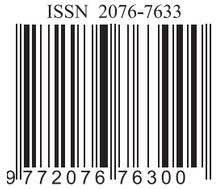
СОДЕРЖАНИЕ 2015 Том 7 № 3

Материалы шестой международной конференции «Распределенные вычисления и грид-технологии в науке и образовании» (Grid'2014)

ISSN 2076-7633

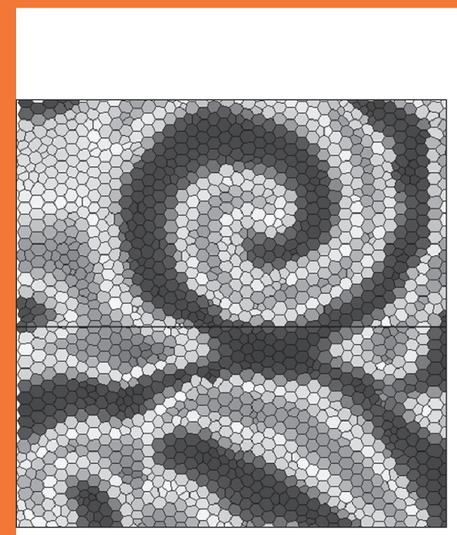


2015 Том 7 № 3

2015
Том
7
№ 3

ISSN 2076-7633

Компьютерные исследования и моделирование



Компьютерные исследования и моделирование

http://crm.ics.org.ru

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ И МОДЕЛИРОВАНИЕ

Журнал публикует результаты оригинальных исследований и работы обзорного характера в области компьютерных исследований и математического моделирования в физике, технике, биологии, экологии, экономике, психологии и других областях знания. Выходит 6 раз в год.

Учредители:
ФГБОУ ВПО «Удмуртский государственный университет», АНО «Ижевский институт компьютерных исследований», ФГБУН «Институт машиноведения им. А. А. Благонравова РАН»

Адрес редакции:
119991, г. Москва, Ленинские горы, д. 1, стр. 12,
кафедра биофизики биологического факультета МГУ имени М. В. Ломоносова,
редакция журнала «Компьютерные исследования и моделирование»
Тел.: +7 (495) 939-02-89; факс: +7 (495) 939-11-15; e-mail: editorial@crm.ics.org.ru

Информация для авторов приведена на внутренней стороне оборота
Подписной индекс 59894 в каталоге «Роспечать-2014»
Официальный сайт: http://crm.ics.org.ru

COMPUTER RESEARCH AND MODELING

Computer Research and Modeling is a quarterly Russian journal publishing original research and review papers on computer-aided studies and mathematical modeling in physics, technology, biology, ecology, economics, psychology and other fields of knowledge.

Address for correspondence:
Department of Biophysics at the Biological Faculty of M. V. Lomonosov Moscow State University
Editorial office of Computer Research and Modeling
119991, Moscow, Leninskie gory, 1, building 12
Tel.: +7 (495) 939-02-89; Fax: +7 (495) 939-11-15; e-mail: editorial@crm.ics.org.ru

Founders of the journal are Udmurt State University, Izhevsk Institute of Computer Science, Institute of Machines Science named after A. A. Blagonravov of RAS

Информация для авторов

Рецензируемый российский журнал «Компьютерные исследования и моделирование» публикует оригинальные и обзорные работы в области компьютерных исследований и математического моделирования в физике, технике, биологии, экологии, экономике, психологии и других областях знания. Журнал освещает работы по компьютерным методам и моделированию систем разной природы в ведущих научных школах России и ближнего зарубежья. Особый интерес для журнала представляют работы, посвященные моделированию в таких бурно развивающихся областях науки, как нанотехнология, биоинформатика, экономика. Приоритетной задачей журнала является освещение развития компьютерных и математических методов изучения процессов в сложных структурированных и развивающихся системах.
Журнал представлен следующими разделами:

– Математические основы и численные методы моделирования;
– Численные методы и основы их реализации;
– Модели в физике и технологии;
– Анализ и моделирование сложных живых систем;
– Модели экономических и социальных систем.

Полнотекстовая электронная версия журнала доступна на официальном сайте журнала **http://crm.ics.org.ru.**

Правила оформления рукописей, представляемых в журнал «Компьютерные исследования и моделирование»

К рассмотрению принимаются рукописи, не опубликованные и не предназначенные к публикации в другом издании. Текст статьи следует предоставлять в электронном варианте в формате LaTeX (желательно) или Microsoft Word. При подготовке статьи обязательно использовать файлы-шаблоны, которые можно загрузить с сайта журнала (раздел «Для авторов»). На первой странице следует указать, для какого раздела предназначена статья, и привести следующие данные на русском и английском языках:

- название статьи;
- информация об авторах: ФИО, полное официальное наименование организации, в которой выполнялась работа, точный почтовый адрес, e-mail;
- краткая аннотация, ключевые слова, код(ы) УДК.

Статья должна быть набрана шрифтом Times New Roman, размер кегля 11 пунктов. Иллюстрации должны быть вставлены в статью и (по возможности) представлены в виде отдельных файлов Corel Draw, EPS или TIFF (300 dpi). При подготовке иллюстраций учитывайте, что они будут отпечатаны в монохромном виде: линии на графиках должны отличаться типом штриховки, а не цветом, нежелательно использовать полутоновую заливку. Пристатейные библиографические ссылки оформляются в соответствии с ГОСТ Р 7.0.5-2008. Ссылки на литературу и другие цитируемые источники в тексте оформляются в квадратных скобках в формате [Фамилия, год]. Список процитированных источников приводится в конце статьи в алфавитном порядке. Подробная инструкция по оформлению статьи находится на сайте журнала.

Куда присылать рукописи
Используйте интерактивную форму на сайте журнала или отправьте рукопись по электронной почте editorial@crm.ics.org.ru (в теме указать: «Статья для журнала»). В теле письма следует указать фамилии авторов с контактной информацией, название статьи и раздела, количество страниц и рисунков.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ И МОДЕЛИРОВАНИЕ

2015 Том 7 № 3

Технический редактор А. В. Бакиев
Компьютерная верстка А. В. Моторина
Корректор Е. В. Огородникова

Подписано в печать 08.06.2015. Вышел в свет 08.06.2015. Формат 60×84 $^{1}/_{8}$.
Печать цифровая. Бумага Xerox 80 г/м^2. Усл. печ. л. 43,94. Уч.-изд. л. 53,67.
Тираж 300 экз. Заказ № 15-47. Цена свободная.
Издатель и типография: АНО «Ижевский институт компьютерных исследований»
426034, г. Ижевск, ул. Кооперативная, д. 5.
E-mail: borisov@ics.org.ru Тел./факс: +7 (3412) 50-02-95
Свидетельство о регистрации СМИ ПИ № ФС 77-43295 от 28.12.2010 г. выдано ФС по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор)

Ки&М

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ И МОДЕЛИРОВАНИЕ

2015 Том 7 № 3

СОДЕРЖАНИЕ

СТЕНДОВЫЕ ДОКЛАДЫ

COMPUTER RESEARCH AND MODELING

2015 Volume 7 Number 3

CONTENTS

POSTER REPORTS

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 6 № 3**

Ки&М

Grid’2014

Уважаемые читатели, перед Вами специальный выпуск журнала «Компьютерные исследования и моделирование», в котором Вы найдете избранные статьи, представленные на шестой международной конференции «Распределенные вычисления и грид-технологии в науке и образовании» (Grid’2014). Конференция проходила с 30 июня по 5 июля 2014 года в Лаборатории информационных технологий Объединенного института ядерных исследований. Данная конференция является традиционной для Лаборатории и проводится каждые два года начиная с 2004 года. Стоит отметить, что за эти десять лет конференция превратилась в уникальный российский форум для обсуждения широкого спектра вопросов, связанных с использованием распределенных и грид-технологий, бурное развитие которых позволило решить широкий класс задач, включая обработку огромного количества данных, поступающих с Большого адронного коллайдера и внесли значительный вклад в открытие бозона Хиггса.

Финансовую поддержку проведению конференции оказала Дирекция Объединенного института ядерных исследований, также спонсорами и партнерами конференции выступили: Supermicro Computer, NIAGARA, Quantum, Jet infosystems, IBM, PARALLEL.RU.

Всего же в работе конференции приняли участие около 200 ученых из научных центров Армении, Беларуси, Болгарии, Венгрии, Монголии, Румынии, Словакии, США, Чехии, Франции, ЮАР и др. Россия была представлена участниками из более, чем 30 университетов и исследовательских центров. В рамках конференции была организована работа восьми секций, на которых обсуждалась текущая и будущая роль грид-технологий, добровольных вычислений, облачных технологий, BigData в моделях компьютинга для мегапроектов в России и мире, таких как NICA и FAIR. Всего же на конференции было заслушано 34 пленарных, свыше 120 секционных и 25 стендовых докладов.

Не все доклады, представленные на конференции Grid’2014, вошли в данный журнал. Однако по представленным работам можно получить представление о современном развитии глобальной грид-инфраструктуры в России и мире, использовании высокопроизводительных вычислений, облачных технологий, а также программного обеспечения для обработки и хранения больших данных в различных областях науки, образования, промышленности и бизнеса.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 395−398

Ки&М

ПЛЕНАРНЫЕ ДОКЛАДЫ

УДК: 004.75

RDMS CMS Computing: Current Status and Plans

V. B. Gavrilov[1], I. A. Golutvin[2], O. L. Kodolova[3], V. V. Korenkov[2,a], L. G. Levchuk[4], S. V. Shmatov[2], E. A. Tikhonenko[2], V. E. Zhiltsov[2]

[1] Institute of Theoretical and Experimental Physics, B. Cheremushkinskaya 25, Moscow, 117218, Russia
[2] Joint institute for nuclear researches, Laboratory of Information Technologies, Joliot-Curie, 6, Moscow reg., Dubna, 141980, Russia
[3] Skobeltsyn Institute of Nuclear Physics, 1(2), Leninskie gory, GSP-1, Moscow 119991, Russia
[4] National Science Center "Kharkov Institute of Physics and Technology", 1 Akademicheskaya St., Kharkov, 61108, Ukraine

E-mail: [a] korenkov@jinr.ru

Получено 3 октября 2014 г.

The Compact Muon Solenoid (CMS) is a high-performance general-purpose detector at the Large Hadron Collider (LHC) at CERN. More than twenty institutes from Russia and Joint Institute for Nuclear Research (JINR) are involved in Russia and Dubna Member States (RDMS) CMS Collaboration. A proper computing grid-infrastructure has been constructed at the RDMS institutes for the participation in the running phase of the CMS experiment. Current status of RDMS CMS computing and plans of its development to the next LHC start in 2015 are presented.

Keywords: grid computing, CMS experiment, RDMS CMS collaboration, CMS Tiers

RDMS CMS компьютинг: текущий статус и планы

В.Б. Гаврилов[1], И. А. Голутвин[2], О. Л. Кодолова[3], В. В. Кореньков[2], Л. Г. Левчук[4], С. В. Шматов[2], Е. А. Тихоненко[2], В. Е. Жильцов[2]

[1] Институт Теоретической и Экспериментальной Физики, Россия, 117218, г. Москва, ул. Большая Черемушкинская, д. 25
[2] Лаборатория информационных технологий, Объединенный институт ядерных исследований Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6
[3] Научно-исследовательский институт ядерной физики имени Д. В. Скобельцына, Россия, 119991, г. Москва, Ленинские горы, ГСП-1, д. 1, стр. 2
[4] Национальный научный центр «Харьковский физико-технический институт», Украина, 61108, г. Харьков, ул. Академическая, д. 1

Компактный мюонный соленоид (CMS) — высокоточный детектор общего назначения на Большом адронном коллайдере (LHC) в ЦЕРН. Более двадцати институтов из России и стран-участниц ОИЯИ вовлечены в коллаборацию RDMS (Россия и страны-участницы) как составной части коллаборации CMS. Для полноценного участия RDMS CMS в действующей фазе эксперимента, в институтах RDMS была создана необходимая компьютерная грид-инфрастуктура. В статье представлены текущий статус компьютинга коллаборации RDMS CMS и планы его развития в контексте следующего старта LHC в 2015 году.

Ключевые слова: грид компьютинг, эксперимент CMS, коллаборация RDMS CMS, центры CMS (Tiers)

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 395−398 (Russian).

Russia and Dubna Member States (RDMS) CMS collaboration was founded in the 1994 year [Matveev, Golutvin, 1996]. The RDMS CMS takes an active part in the Compact Muon Solenoid (CMS) Collaboration [CMS Collaboration …, 1994] at the Large Hadron Collider (LHC) [WhatLHC…] at CERN [CERN Web site]. RDMS CMS Collaboration joins more than twenty institutes from Russia and Joint Institute for Nuclear Research (JINR) member states. RDMS scientists, engineers and technicians were actively participating in design, construction and commissioning of all CMS sub-detectors in forward regions. RDMS CMS physics program has been developed taking into account the essential role of these sub-detectors for the corresponding physical channels. RDMS scientists made large contribution for preparation of study QCD, Electroweak, Exotics, Heavy Ion and other physics at CMS. During LHC Run I RDMS scientists contributed significantly to data taking, data processing and analysis. They played key roles in Higgs hunting, testing the standard model and looking for new physics with CMS. The overview of RDMS CMS physics tasks and RDMS CMS computing activities are presented in [Gavrilov V. et al., 2004; Gavrilov V. et al.. 2006; Gavrilov V. et al., 2008, p. 203–208; Gavrilov V. et al., 2008, p. 156–159; Gavrilov V. et al., 2011]. RDMS CMS computing support should satisfy the LHC data processing and analysis requirements at the running phase of the CMS experiment [CMS Collaboration …, 2005].

During the last decade, a proper grid-infrastructure for CMS tasks has been created at the RDMS CMS institutes, in particular, at Institute for High Energy Physics (IHEP) in Protvino, Joint Institute for Nuclear Research (JINR) in Dubna, Institute for Theoretical and Experimental Physics (ITEP) in Moscow, Institute for Nuclear Research (INR) of the Russian Academy of Sciences (RAS) in Moscow, Skobetsyn Institute for Nuclear Physics (SINP) in Moscow, Petersburg Nuclear Physics Institute (PNPI) of RAS in Gatchina, P.N.Lebedev Physical Institute (LPI) in Moscow and National Scientific Center Kharkov Institute of Physics and Technology (NSC KIPT) in Kharkov. In the CMS global grid-infrastructure these RDMS CMS sites operate as CMS centers of the Tier-2 level with the following names: T2_RU_IHEP, T2_RU_JINR, T2_RU_ITEP, T2_RU_INR, T2_RU_SINP, T2_RU_PNPI, T2_UA_KIPT, T2_RU_RRC_KI.

A stable and successful operation of several RDMS CMS Tier2 centers and years of operating experience lead to creation the CMS Tier-1 center in Russia as an integral part of the central data handling service of the CMS Experiment (CMS Tier-1 in Dubna). Currently, the JINR realizes a large-scale project to create a Tier1 computer center for the CMS experiment in the accordance with a decision (adopted by the WLCG project in 2011) to construct a Tier1 level distributed center for the LHC experiment in Russia on the base of RCC «Kurchatov institute» and JINR. Tier1 center for CMS at JINR is now at a testing phase and a full-scale CMS Tier1 at JINR will be operational in 2015 [Astakhov N. S. et al., 2012].

In line with the CMS computing requirements for the data-taking phase of the experiment, now the RDMS CMS grid-sites provide:

- the computing and data storage resources in full;
- centralized deployment of actual versions of CMS specialized software (CMSSW);
- data transfers between the CMS grid-sites with the usage of the FTS grid-service on basis of VOBOX grid-services for CMS with the Phedex Server;
- SQUID proxy-servers for the CMS conditions DB access;
- certification of network links at the proper data transfer rates between JINR and CMS Tier1 and Tier2 centers;
- daily massive submission of CMS typical jobs by the CMS Hammer Cloud system;
- CMS data replication to the JINR data storage system in the accordance with RDMS CMS physicists' requests;
- participation in the CMS Monte-Carlo physical events mass production in the accordance with the RDMS CMS physicists' scientific program.

As it can be seen from Fig.1, during 2010–2013 years the contribution of RDMS CMS sites intoCMS jobs processing is at level of 5%.

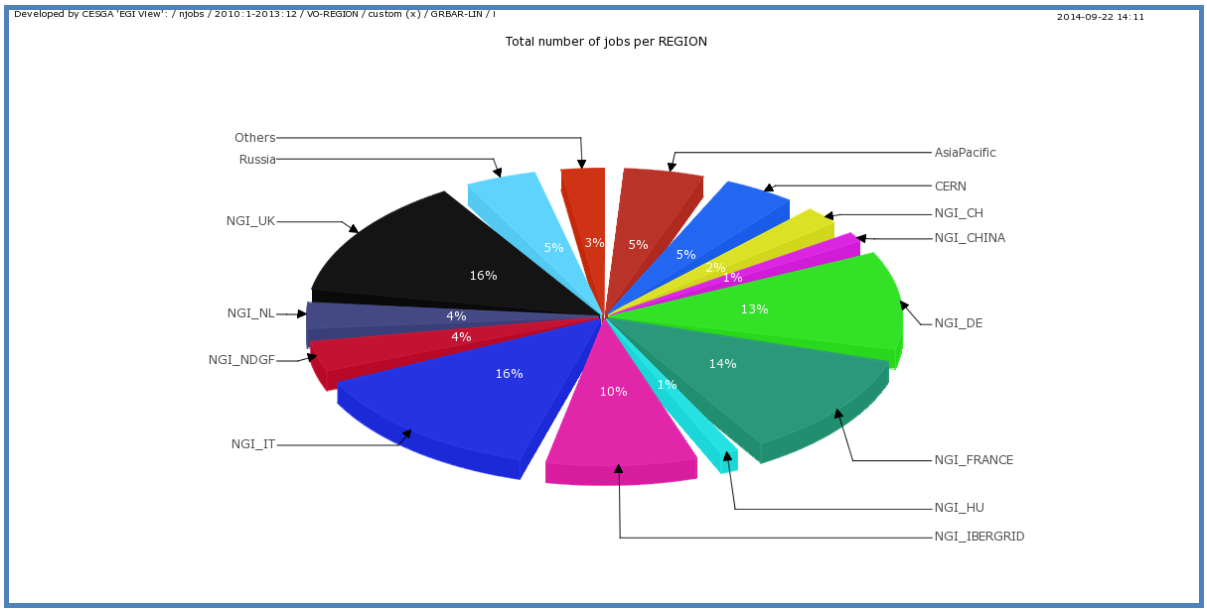


Fig. 1. Number of jobs in CMS Virtual Organization from January, 2010 to December, 2013

From the middle of 2010 to the end of 2013 2 Petabytes data have been transferred to RDMS CMS sites. By the moment about 1.5 Petabytes of CMS data are stored at RDMS CMS sites (0.5 Petabytes are the data of CMS physical groups with which the RDMS sites are associated).

A group of RDMS CMS specialists takes an active part in the CMS Dashboard development (grid monitoring system for the CMS experiments) (/http://dashboard.cern.ch/cms).

The dedicated CMS remote worldwide-distributed centers (ROC) were built in different scientific organizations. The JINR CMS Remote Operation Center (ROC) was founded in the 2009 year to provide participation in CMS operations of a large number of RDMS CMS collaborating scientists and engineers. MSU and IHEP ROCs were started-up two years after. RDMS CMS ROCs provide the following functions:

- monitoring of CMS detector systems;
- data monitoring and express analysis;
- shift operations;
- communications of the JINR shifters with personal at the CMS Control Room (SX5) and CMS Meyrin centre;
- communications between JINR experts and CMS shifters;
- coordination of data processing and data management;
- training and information .

RDMS CMS physicists work in the WLCG environment, and now we are having more than 30 members of CMS Virtual Organization.

Summary

The RDMS CMS computing centers have been integrated into the WLCG global grid-infrastructure providing a proper functionality of grid services for CMS. A significant modernization of the RDMS CMS grid-sites has been accomplished. As result, computing performance and reliability have been increased. In the frames of the WLCG global infrastructure the resources of the both computing centers are successfully used in a practical work of the CMS virtual organization. Regular testing of the RDMS CMS computing centers functionality as grid-sites is provided.

All the necessary conditions for CMS data distributed processing and analysis have been provided at the RDMS CMS computing centers (grid-sites). It makes possible for RDMS CMS physicists to take a full-fledged part in the CMS experiment.

RDMS Tier2 sites contribute significantly to the CMS data processing and analysis tasks. CMS Regional Operation Centers in JINR, MSU and IHEP are operated for remote monitoring of detector systems and data express-analysis.

The nearest plans are to continue support RDMS CMS sites with increasing CPU resources by 14 % and disks resources by 4 % in 2014 and by 25 % and 16 % in 2015 accordingly; in further years — increasing of resources by 10–15% annually.

References

Astakhov N. S. et al. Creation at JINR of the data processing automated system of the TIER-1 level of the experiment CMS LHC // Proc. of GRID'2012 conference, Dubna, 2012. P. 254 (in Russian).

CERN. http://www.cern.ch

CMS Collaboration, Technical Proposal, CERN/LHCC, 94-38, 1994 — http://cmsinfo.cern.ch

CMS Collaboration, The Computing Project, Technical Design Report, CERN/LHCC-2005-023, CMS TDR 7, 2005.

Gavrilov V. at al. Current Status of RDMS CMS Computing // Proc. of the XXI Int. Symposium on Nuclear Electronics and Computing, Dubna, 2008. P. 203–208.

Gavrilov V. at al. RDMS CMS Computing activities before the LHC startup // Proc. of 3rd Int. Conference "Distributed Computing and GRID-technologies in Science and Education", Dubna, 2008. P. 156–159.

Gavrilov V. et al. RDMS CMS Computing // Proc. of the 2nd Int. Conference "Distributed Computing and Grid-Technologies in Science and Education", Dubna, 2006. P. 61.

Gavrilov V. et al. RDMS CMS Computing Model // Proc. of the Int. Conference "Distributed Computing and Grid-Technologies in Science and Education", Dubna, 2004. P. 240.

Gavrilov V. et al. RDMS CMS data processing and analysis workflow // Proc. of XXIII Int. Symp. on Nuclear Electronics & Computing (NEC`2011), Dubna, 2011. P. 148–153.

Matveev V., Golutvin I. "Project: Russia and Dubna Member States CMS Collaboration / Study of Fundamental Properties of the Matter in Super High Energy Proton–Proton and Nucleus–Nucleus Interactions at CERN LHC", 1996-085/CMS Document, 1996 — http://rdms-cms.jinr.ru

WhatLHC. http://public.web.cern.ch/Public/Content/Chapters/AboutCERN/CERNFuture/WhatLHC/WhatLHC-en.html

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 399–406

Ки&М

ПЛЕНАРНЫЕ ДОКЛАДЫ

УДК: 004.021

Grid based high performance computing in satellite imagery. Case study — Perona–Malik filter

B. Belean[1], C. Belean[2], C. Floare[1], C. Varodi[1], A. Bot[1], Gh. Adam[3,4,a]

[1]National Institute for R&D of Isotopic and Molecular Technolgies, 67-103 Donat St., PO 5 Box 700, Cluj-Napoca, 400293, Romania
[2]Babes-Bolyai University, Faculty of Mathematics and Computer Science, Department of Mathematics, 1 M. Kogalniceanu St., Cluj-Napoca, 400084, Romania
[3]Laboratory of Information Technologies, Joint Institute for Nuclear Research, 6 Joliot Curie St., Dubna, 141980, Russia
[4]Horia Hulubei National Institute for Physics and Nuclear Engineering (IFIN-HH), 30 Reactorului St., Magurele - Bucharest, 077125, Romania

E-mail: [a]adamg@jinr.ru

Получено 4 декабря 2014 г.

The present paper discusses an approach to the efficient satellite image processing which involves two steps. The first step assumes the distribution of the steadily increasing volume of satellite collected data through a Grid infrastructure. The second step assumes the acceleration of the solution of the individual tasks related to image processing by implementing execution codes which make heavy use of spatial and temporal parallelism. An instance of such execution code is the image processing by means of the iterative Perona–Malik filter within FPGA application specific hardware architecture.

PACS: 02.60.Lj; 02.70.Bf; 07.05.Pj; 93.55.+z; 93.85.Bc; 95.75.Mn; 95.75.Pq

Supported within the Romania–LIT Hulubei-Mashcheryakov Programme, JINR Orders 94/17.02.2014, p.25, 95/17.02.2014, pp. 76, 77, and 96/17.02.2014, pp. 86–89.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 399–406 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 399–406

Ки&М

ПЛЕНАРНЫЕ ДОКЛАДЫ

Сеточные высокопроизводительные вычисления в получении спутниковых изображний на примере фильтра Перона–Малик

Б. Белеан[1], **К. Белеан**[2], **К. Флоаре**[1], **К. Вароди**[1], **А. Бот**[1], **Г. Адам**[3,4]

[1]*Национальный институт исследования и развития технологии молекулярных изотопов (INCDTIM), Румыния, 400293, Клуж-Напока, Донат ул. 67-103, ПО 5, ПЯ 700*
[2]*Университет им. Бабеша-Бойаи, Факультет математики и информатики, отделение математики, Румыния, 400084, Клуж-Напока, Михаил Когэлничану ул. 1*
[3]*Лаборатория информационных технологий, Объединенный институт ядерных исследований, Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6*
[4]*Национальный научно-исследовательский институт физики и ядерной технологии им. Хории Хулубея (IFIN-HH), Румыния, 077125, г. Мэгуреле-Бухарест, Реакторулуй ул., д. 30*

В данной работе рассматривается подход к эффективной обработке спутниковых изображений, который включает в себя два этапа. Первый этап заключается в распределении быстро взрастающего объема спутниковых данных, полученных через Грид-инфраструктуру. Второй этап включает в себя ускорение решения отдельных задач, относящихся к обработке изображений с помощью внедрения кодов, которые способствуют интенсивному использованию пространственно-временного параллелизма. Примером такого кода является обработка изображений с помощью итерационного фильтра Перона–Малик в рамках специального применения архитектуры аппаратного обеспечения ППВМ (FPGA).

Ключевые слова: фильтр Перона–Малик, обработка спутникового изображение, Грид, высокопроизводительные вычисления, ППВМ (ПЛИС), ЮНИТАР (UNOSAT)

1. Essentials of satellite imagery

The scanning of the Earth surface by satellites is done within a wide wavelength range of the reflected electromagnetic radiation coming from the Sun (microwave, infrared, visible, ultraviolet). The recorded electromagnetic waves are used to create images according to the pattern shown in figure 1. The radiation emitted by an energy source (1) covers a distance and interacts with the atmosphere (2) before reaching the target (3). Radiation is reflected or scattered to the satellite sensor (4), which registers it and then transmits the information on the reflected radiation to a receiving station (5) where the received input is transformed into digital images. The characterization of the target properties or the solution of some specific problem is performed based on the acquired images.

Satellite images are of fundamental interest in meteorology, agriculture, biodiversity conservation, geology, forestry, landscape, regional planning, education, intelligence and warfare. Commercial applications concern: *Insurance companies* — damage estimates based on acquired images before and after disaster; *Mass Media* — satellite imagery based news reports; *Software developers* — image incorporation in flight simulators, games; *Combined with GPS* — localization in geographic information systems, e.g., Google Earth and Google Earth Pro.

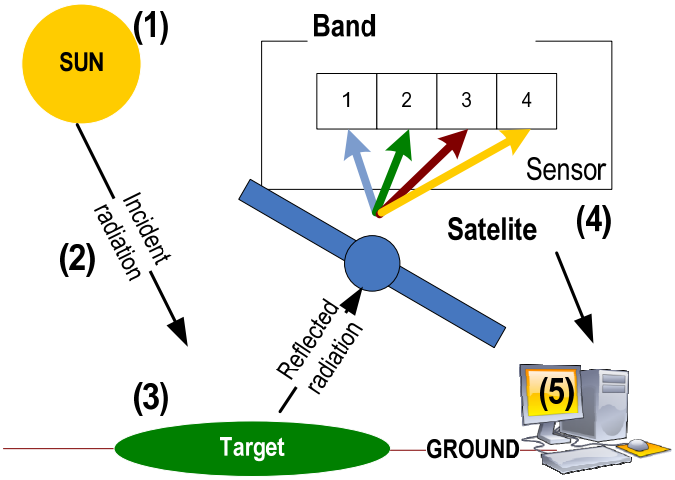


Fig. 1. Acquiring satellite images

Resolution characterization of a satellite image

Concerning the satellite images in remote sensing, there are four types of resolutions: spatial, spectral, temporal, and radiometric. Campbell [Campbell, 2002] defines them as follows. The ***spatial resolution*** represents the pixel size of an image of the surface area measured on the ground, determined by the sensor's instantaneous field of view. The ***spectral resolution*** is defined by the wavelength interval within the electromagnetic spectrum and the number of intervals measured by the sensor. (Examples: *visible images* come from sunlight reflected by the Earth surface; *infrared images* come from infrared sensor measurements of the temperature of the Earth surface; *water vapor images* come from infrared measurement of the temperature in an atmospheric layer about 6–10 km above the Earth surface.) The ***temporal resolution*** is defined by the time interval inbetween two subsequent image acquisitions for a given surface location. The ***radiometric resolution*** is defined by the amount of levels of brightness recorded by the imaging system.

Another resolution may be mentioned, namely, the ***geometric resolution*** which refers to the satellite sensor's ability to effectively image a portion of the Earth's surface in a single pixel and is typically expressed in Ground Sample Distance (GSD) units. GSD is a term containing the overall optical and systemic noise sources and is useful for comparing how well one sensor can "see" an object on the ground within a single pixel [Satellite imagery, 2014].

2. Grid based approach to satellite imagery

The increase of the number of high resolution satellites into orbit and the number of applications which use satellite images lead to "big data" to be processed, which cannot be accommodated to the satellite's local computing infrastructures. The present paper focuses on high throughput computing strategies for processing satellite images which involve grid computing and application specific hardware architectures for high performance computing.

UNOSAT is the United Nations Institute for Training and Research (**UN**ITAR) **O**perational **Sat**ellite Applications Programme. Created in 2000, it provides the worldwide users with high quality satellite imagery and Geographic Information System (GIS) services. These serve for planning sustainable development or monitoring natural disasters.

Excessive loads on the UNOSAT website, putting strong pressure on the computing and storage resources are frequently encountered. Instances:

Use case 1. During ***natural catastrophes*** and ***disasters***.

Use case 2. ***Web visualization from mobile devices*** of the field workers (even though using compression and cropping when interrogating UNOSAT resources).

Use case 3. Need of storage and computing resources ***from UNOSAT*** by ***users having slow internet connection.***

Use case 4. The ***periodically performed updates*** by the UNOSAT administrators (asked by image uploading in the databases; own search and processing tasks on the satellite image databases).

Since 2001, UNOSAT has been based at CERN and is supported by CERN's IT Department in the work it does. **CERN** ***provides Grid approach*** to the operations done on the satellite images: storage, processing, compression. The existing sequential codes for iterative solutions on multicore processors are, however, inefficient [Belean et al., 2013]. The usual approaches to speeding up the sequential solutions [Javier Gallego, 2005; Giuliani et al., 2011; Sarmah, Bhattacharyya, 2011] use ***division*** of the satellite images into ***sub-images*** in order to reduce the size to be processed. Each sub-image can be sent for processing to a different computing element (CE) within the grid. The most frequently encountered technique to divide the satellite images is based on the use of the ***Thiessen polygons*** [Javier Gallego, 2005].

The following limitations characterize the ***sequential iterative algorithms*****: t**he computing power is limited to one CE while processing one sub-image; the CEs are General Purpose Processors (GPPs) (e.g. **Intel® Xeon® Processor E5**); the parallel processing strategies to be applied are limited by the available GPPs.

To secure smooth continuous functioning of the system, a vital condition is to produce ***maximum relevant information*** under ***minimum system load***. The use of FPGA approach to the ***Perona–Malik filter*** offers such a tool.

3. Essentials of Perona–Malik filter for digital image processing

There are ***two basic aims*** of the Perona–Malik filter [Perona, Malik, 1990]: to eliminate disturbances, i.e., filter out noise; to retain and enhance the essential information.

The ***basic idea*** can be summarized as follows. The above aims can be achieved by solving a Neumann boundary problem for a generalized diffusion equation

$$\partial_t u(x,t) = \nabla(g(u,x,t)\nabla u(x,t)). \tag{1}$$

Where the diffusivity function g is to be ***anisotropic***, such that, along some directions $g >> 1$ (strong diffusion) while along other directions $g << 1$ (weak smoothing).

The use of the equation (1) is motivated by the physical fact that, under $g \equiv 1$ Eq. (1) describes the ***heat propagation.*** The solution of the ***heat equation*** is a ***convolution*** of the initial value $u(x, 0) = f(x)$ with a ***Gauss distribution function***. Since the latter is decreasing fast (both in the coordinate and the frequency spaces), the ***fast oscillations are cut out***, hence (1) acts as an effective ***low pass filter***.

The ***digital image processing*** based on the equation (1) is motivated by the following considerations. There is a ***bounded*** domain of the digital image, $\Omega \subset \mathbf{R}^n$ (n = 2, 3) of boundary $\partial\Omega$ of class C^1. The mapping u: $\Omega \to [0, 1]$ then achieves the correspondence from Ω to the ***gray level distribution*** (GLD) of a noisy image. The numerical investigation of the time evolution of the GLD, performed through an ***iterative approach***, results in ***successive instances*** attempting at solving the filtering tasks.

There are two contradictory features of the Gaussian smoothing associated to the heat propagation: ***efficient*** noise filtering, but ***edge smearing*** (image blurring) which results in ***quick loss*** of essential information contained in the original image.

The ***Perona–Malik hypothesis***: [Wielgus, 2010] the diffusivity coefficient in (1) is to be a function of the ***norm of the local gray level distribution gradient.*** This results into the diffusion Perona–Malik problem:

$$\begin{gathered}\partial_t u(x,t) = \nabla(g(|\nabla u|^2)\nabla u(x,t)) \text{ in } \Omega \times (0,+\infty),\\ \partial u / \partial n = 0 \text{ in } \partial\Omega \times (0,+\infty),\\ u(x,0) = f(x) \text{ in } \Omega.\end{gathered} \tag{2}$$

Basic features of the Perona–Malik problem. It is an ***ill-posed*** problem, which ***does not*** admit a weak solution. A solution could, nonetheless, be defined in the sense of distributions. However, under ***spatial discretization by finite differences***, a ***well-conditioned problem*** arises, with an unexpectedly good filtering efficiency. This is usually called the ***Perona–Malik paradox*** in the digital image processing. An important quantity is ***the flux function***,

$$\Phi(s) = s \cdot g(s^2) > 0 \text{ for } s \in (0,+\infty),$$

which is asked to ***vary smoothly with*** s and to have a maximum on $(0,+\infty)$ at some characteristic value $s_0 = \lambda > 0$.

The ***diffusivity function*** $g(s^2)$ enabling such $\Phi(s)$ should be infinitely continuous differentiable and to decrease monotonically from 1 to 0 while s^2 varies from 0 to + ∞.

An expression of $g(s^2)$ inspired by the Gaussian distribution function is

$$g(s^2) = \exp(-s^2 / 2\lambda^2),\ \lambda > 0. \tag{3}$$

Then $\Phi(s)$ has a maximum at $|s| = \lambda$, with $\Phi'(s) > 0$ for $|s| < \lambda$ and $\Phi'(s) < 0$ for $|s| < \lambda$.

An alternative choice of the diffusion function is

$$g(s^2) = 1/(1+s^2\lambda^2),\ \lambda > 0. \tag{4}$$

In the *two-dimensional case*, let ξ and η denote the local coordinates in directions perpendicular and parallel to ∇u respectively. Then the Perona–Malik equation can be rewritten as

$$\partial_t u = \nabla(g(|\nabla u|^2)u_{\xi\xi} + \Phi'(|\nabla u|)u_{\eta\eta}.$$

The coefficient of $u_{\xi\xi}$ is always positive, hence (2) acts as a smearing filter washing details along the contour lines of the function u. The coefficient of $u_{\eta\eta}$ may be both positive and negative, hence, in the perpendicular (gradient) direction slow gradient values are smeared out, while *large gradient values* (like *edges*) are sharpened instead of being blurred.

The experiments using Perona–Malik anisotropic diffusion were visually very impressive: edges remained stable over a very long time. It was demonstrated that edge detection based on this process clearly outperforms the linear Canny edge detector, even without applying non-maxima suppression and hysteresis threshold. This is due to the fact that diffusion and edge detection interact in one single process instead of being treated as two independent processes which are to be applied subsequently. Considering this advantage, this technique would improve edge detection in any image processing based applications.

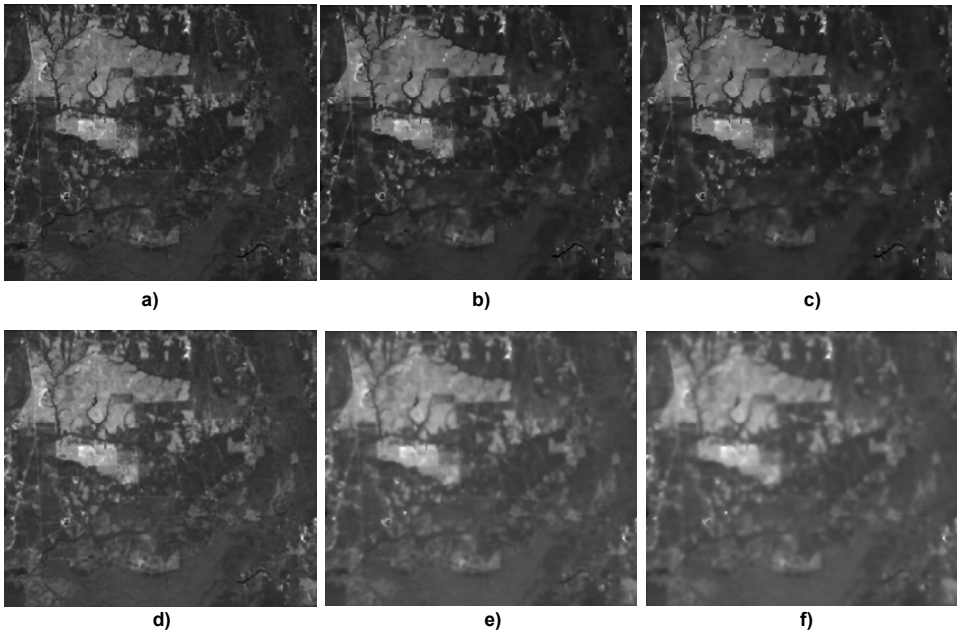


Fig. 2. Anisotropic diffusion showing edge enhancement in case of original image *Florida* [Florida…,2014], with choice (4) for diffusion function: a) *Num_iter* = 5, λ = 10, b) *Num_iter* = 15, λ = 10, c) *Num_iter* = 25, λ = 10, d) *Num_iter* = 5, λ = 30, *e) Num_iter* = 15, λ = 30, f) *Num_iter* = 25, λ = 30

Results of the conventional anisotropic diffusion (Perona & Malik) upon a gray scale image aiming edge enhancement are presented in Fig. 2. The number of iterations is denoted *Num_Iter*. The optimal parameter setup to preserve edge is found in figure 2.c, were all the details are preserved. Thus, it can be seen that for a number of 5 to 25 iterations (Fig. 2.a, 2.b and 2.c), the edge details are preserved. Meanwhile, for large values of λ, details are lost, see Fig. 2.d-f.

4. Fpga based hardware architectures for anisotropic diffusion

In order to propose hardware architecture for Perona–Malik filter, the computational steps of the anisotropic diffusions are independently presented next. Considering the iterative nature of the PDE filtering, the steps described previously are included in one filter iteration.

Given an initial image, to be processed within N iterations (empirically N = 10 to 20):

Steps 1, 2 and 3 shown in Fig. 3 are to be parallelized for efficient computation. Within the Field Programmable Gate Array (FPGA) architecture, both spatial parallelism (enabling the use of multiple computing units) and temporal parallelism (enabling the creation of a pipeline) were implemented. The main estimated advantages of the proposed architecture are as follows: within one iteration, each $p(i,j)$ resulted pixel value its delivered at each $3\times T_{clk}$ cycles, with an initial delay of $T = 12\ T_{clk}$ cycles. In the case of a general purpose processor, within iteration, for one output pixel there are necessary more than 30 T_{clk} cycles. The exponential function can be implemented based on the approach presented in [Belean B. et al., 2012] for the log computation unit.

Computations done for a Virtex 5 FPGA and a general purpose processor are given in Fig. 4. A factor two acceleration of the computations, with good scalability, are obvious.

Conclusions

- In future attempts to implement solutions for specific iterative algorithms, we intend to intensively use Field Programmable Gate Arrays

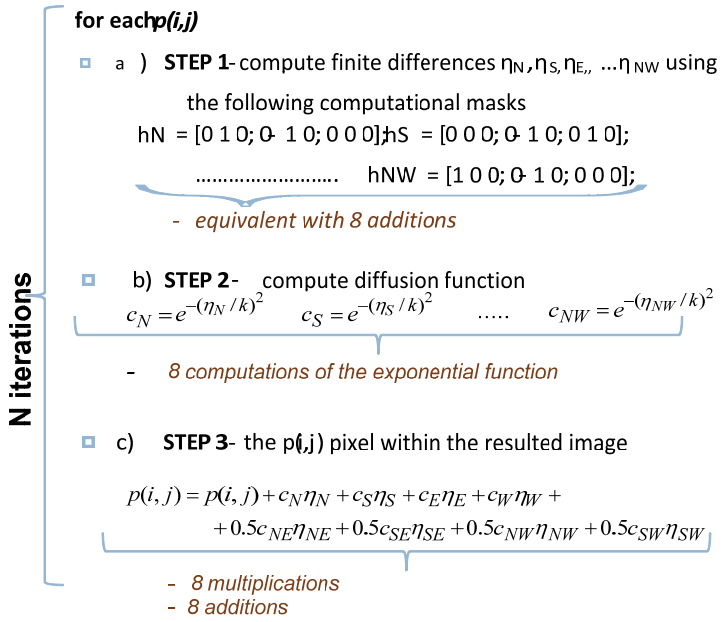


Fig. 3. The three distinct computational steps identified within each of the iterations

- The reported results suggest that the general purpose processors are surpassed by Application Specific Hardware Architectures as it concerns the computing time
- In the next future, we plan to derive a GPU implementation of the Perona–Malik filter and to compare it with the FPGA implementation
- In the processing of grid Big Data sets, extensive use of Application Specific Hardware Architectures might offer efficient solutions.

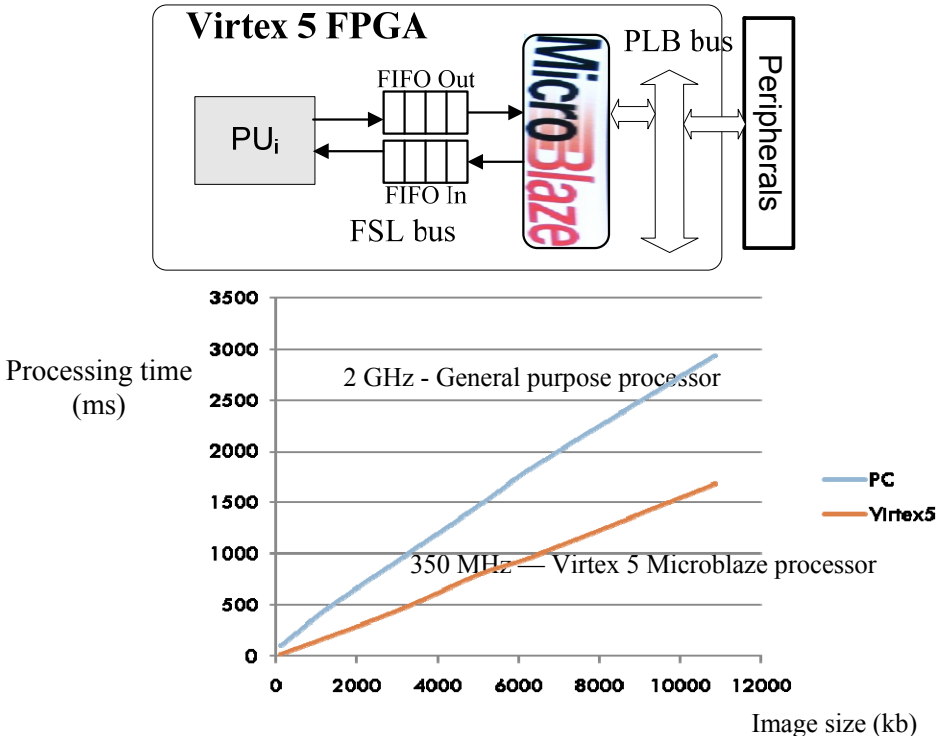


Fig. 4. Schematic presentation of the Virtex 5 FPGA (above) and comparison of the performance dependence of Virtex 5 FPGA with that of a 2 GHz general purpose processor

References

Belean B., Borda M., Bot A. FPGA based hardware architectures for iterative algorithms implementations // 36th International Conference on Telecommunications and Signal Processing. Rome. — 2013.

Belean B., Borda M., Le Gal B., Terebes R. FPGA based system for automatic cDNA microarray image processing. // Computerized Medical Imaging and Graphics. —2012. — Vol. 36. — P. 419–429.

Campbell J. B. Introduction to Remote Sensing. New York London: The Guilford Press. — 2002.

Florida, Google Maps. — 2014. https://www.google.rp/maps/place/Florida

Giuliani G., Ray N., Lehmann A. Grid-enabled Spatial Data Infrastructure for environmental sciences: Challenges and opportunities // Future Generation Computer Syst. — 2011. —Vol. 27. — P. 292–303.

Javier Gallego F. Stratified sampling of satellite images with a systematic grid of points. // ISPRS Journal of Photogrammetry & Remote Sensing. — 2005. — Vol. 59. — P. 369–376.

Perona P., Malik J. Scale space and edge detection using anisotropic diffusion. // Proc. IEEE Computer Soc. Workshop on Computer Vision 16–22 (1987); Perona P., Malik J., Scale-Space and Edge Detection Using Anisotropic Diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1990. — Vol. 12. — P. 629–639.

Sarmah S., Bhattacharyya D.K. A grid-density based technique for finding clusters in satellite image. Pattern Recognition Letters. — 2012. — Vol. 33. — P. 589–604.

Satellite imagery. From Wikipedia, the free encyclopedia. — 2014. http://en.wikipedia.org/wiki/Satellite_imagery

Wielgus M. // Perona–Malik equation and its numerical properties. Praca licencjacka na kierunku MATEMATYKA, Uniwersytet Warszawski, Lipiec. — 2010.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 407–414

Ки&М

ПЛЕНАРНЫЕ ДОКЛАДЫ

УДК: 004.75

ARC-CE: updates and plans

O. Smirnova[1,a], B. Kónya[1], D. Cameron[2], J. K. Nilsen[2] and A. Filipčič[3]

[1] Dept. of Physics, Lund University, 221 00, Lund, Sweden, Professorsgatan 1
[2] Dept. of Physics, University of Oslo, N-0316 Oslo, Norway, 1048 Blindern
[3] Institute Jožef Stefan, 1000 Ljubljana, Slovenia, Jamova 39

E-mail: [a]oxana.smirnova@hep.lu.se

Получено 27 октября 2014 г.

ARC Compute Element is becoming more popular in WLCG and EGI infrastructures, being used not only in the Grid context, but also as an interface to HPC and Cloud resources. It strongly relies on community contributions, which helps keeping up with the changes in the distributed computing landscape. Future ARC plans are closely linked to the needs of the LHC computing, whichever shape it may take. There are also numerous examples of ARC usage for smaller research communities through national computing infrastructure projects in different countries. As such, ARC is a viable solution for building uniform distributed computing infrastructures using a variety of resources.

ARC-CE: новости и перспективы

О. Смирнова[1], Б. Коня[1], Д. Кэмерон[2], Й. К. Нильсен[2] и А. Филипчич[3]

[1] Отд. физики, Университет Лунда, 221 00 Лунд, Швеция
[2] Отд. физики, Университет Осло, N-0316 Осло, Норвегия
[3] Институт Йожефа Стефана, 1000 Любляна, Словения

Вычислительный элемент ARC приобретает всё большую популярность в инфраструктурах WLCG и EGI, и используется не только в контексте систем Грид, но и как интерфейс к суперкомпьютерам и облачным ресурсам. Развитие и поддержка ARC опирается на вклады членов пользовательского сообщества, что помогает идти в ногу со всеми изменениями в сфере распределённых вычислений. Перспективы развития ARC тесно связаны с требованиями обработки данных БАК, в любых их проявлениях. ARC также используется и для нужд небольших научных сообществ, благодаря государственным вычислительным инфраструктурам в различных странах. Таким образом, ARC представляет собой эффективное решение для создания распределённых вычислительных инфраструктур, использующих разнообразные ресурсы.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 404–414 (Russian).

1. Introduction

ARC Compute Element (ARC-CE) is a Grid Compute Element at the core of the ARC middleware developed by NorduGrid [Ellert M et al., 2007]. As of today, it is a key component of ARC middleware, other components being clients for computing and data handling tasks, and information services that advertise ARC-CE status and capacity. ARC-CE is a key enabler of the Nordic Tier-1 operated by NeIC [NeIC Web site], which main characteristic is the distributed nature of both computing and data services. While storage pools across different Nordic countries are federated into a single instance by dCache [dCache Web site], computing services rely on ARC-CE thanks to its capability of caching input data.

Inclusion of ARC into EMI [European Middleware Initiative] middleware stack made it readily available to all sites that support Worldwide LHC Computing Grid (WLCG) [Worldwide LHC Computing Grid] and participate in the European Grid Infrastructure (EGI) [EGI Web site]. It is now used by all large LHC experiments, with ATLAS being the largest user of ARC-CE, followed by an increasing usage by CMS, ALICE and LHCb. In some countries, like the Nordic and Baltic states, as well as Slovenia, ARC-CE is the only Compute Element in use. As can be seen in Figure 1, ARC-CE deployment steadily increases over the past two years.

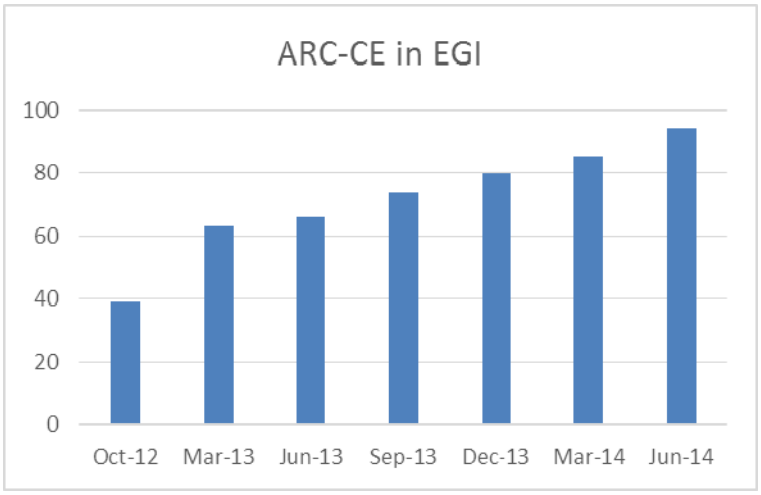


Fig. 1. Number of ARC-CE instances in the EGI database

Figure 2 shows geographical distribution of ARC services, including both ARC-CE and information indices. It serves as a basic service for several national Grid infrastructures, as indicated in the figure.

2. ARC-CE principles

ARC-CE is optimised for data-intensive jobs, taking particular care of staging input data to the local cache for eventual re-use, and staging output data to wherever the job description requests them to be staged.

Data movement is performed by dedicated processes hosted by ARC-CE, and a job will not be submitted to a local batch system until all input data are downloaded by the Compute Element. This may occur as a delay in the job start from the point of view of the user, but in fact this saves time and bandwidth by removing the necessity to move data by the jobs themselves, and moreover, availability of the input data in cache will make the next job that needs same data starting instantly. This approach maximises CPU utilization and minimizes bandwidth. In addition, worker nodes managed by ARC-CE do not require network connectivity if input and output data locations are known in advance, as shown in Figure 3.

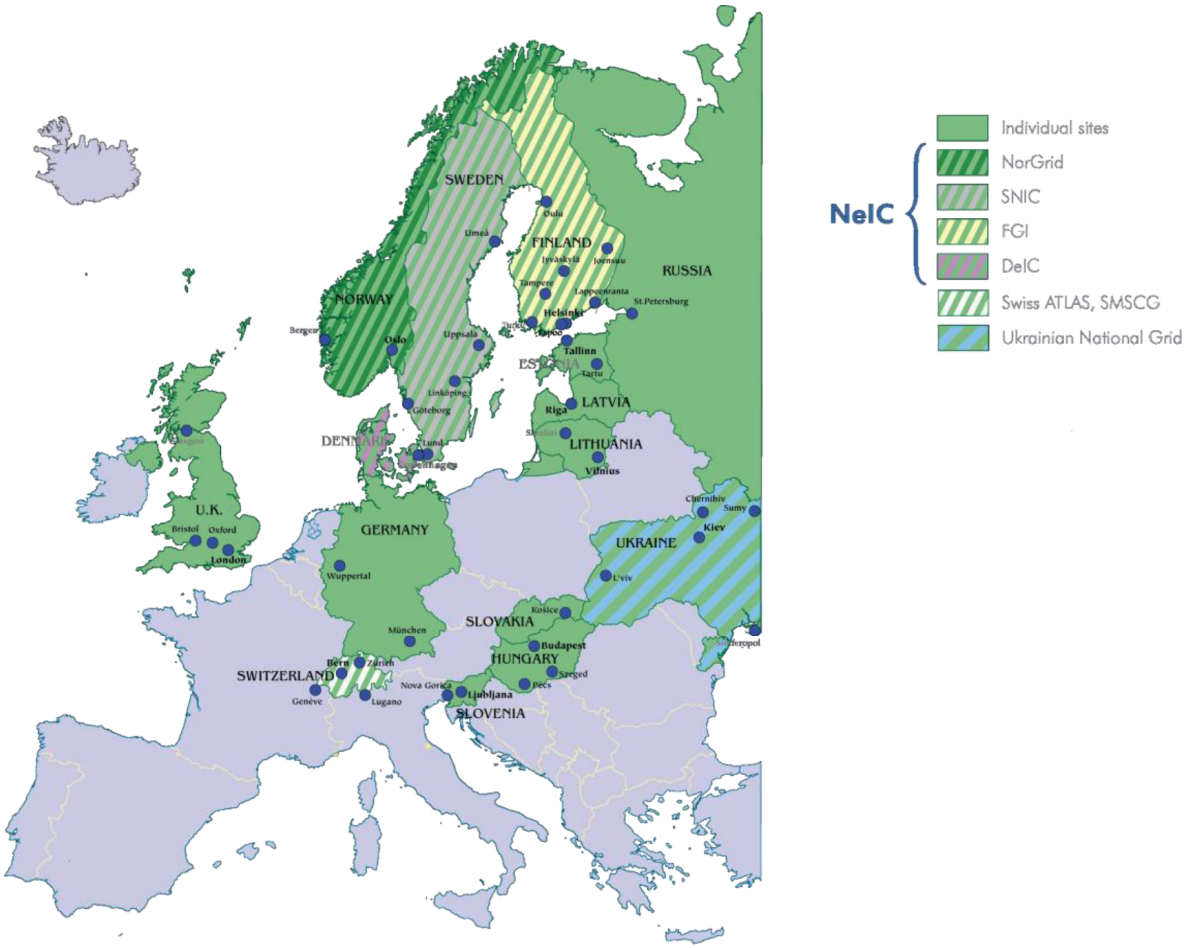


Fig. 2. Geographical distribution of ARC services deployment

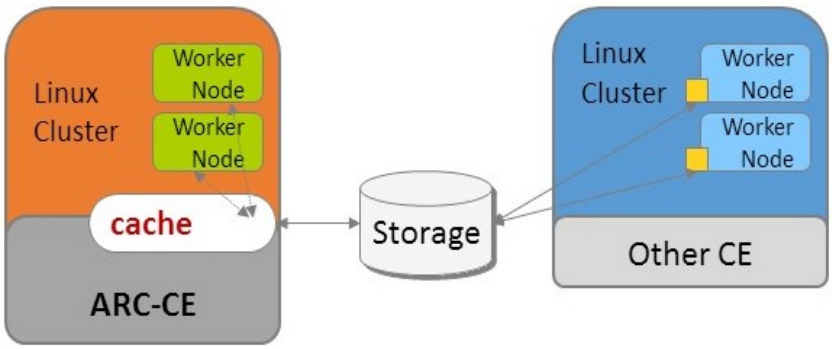


Fig. 3. Comparison of basic ARC-CE principles to those of other CEs: worker nodes managed by ARC-CE need no network connectivity

ARC-CE is a complex service, consisting of many services and utilities. It is quite demanding resource- wise, and needs a fast shared file system as well as a high-end storage server for the cache.

Figure 4 presents an overview of ARC-CE components when installed on a SLURM [SLURM Workload Manager] cluster. In general, ARC-CE supports a large variety of batch systems, including, in addition to SLURM: HTCondor [HTCondor Web site], PBS flavours, Grid Engine flavours, LoadLeveler and LSF. Support levels however differ, relying on community contributions.

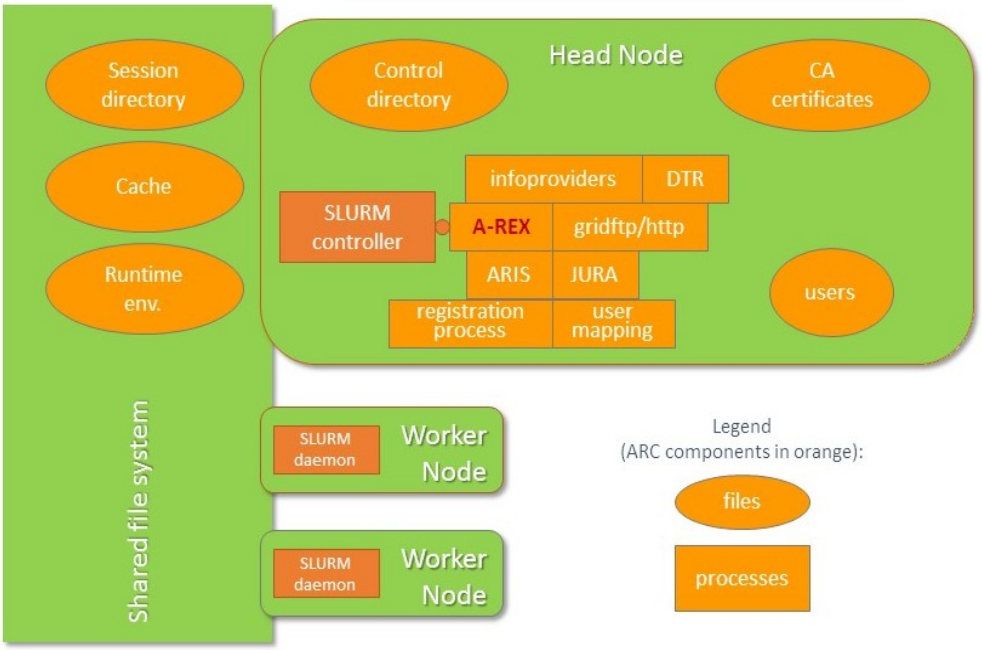


Fig. 4. Overview of ARC-CE components when installed on a SLURM-managed cluster

3. ARC Control Tower

While ARC-CE is designed to move data, it can be used as a generic CE for pilot jobs. This approach, though widely used, does not make use of all the benefits of ARC-CE. In order to optimize resource usage, a special service, ARC Control Tower (aCT), has been developed by NorduGrid and ATLAS [Filipčič A et al, 2011]. aCT is an external service functioning as a workload management system for ARC-CE, specializing on pilot jobs. Its function is to extract job descriptions from pilots, convert them to deterministic ARC-CE jobs, and schedule these "classic" jobs to best suited ARC-CEs. This service is extensively used by ATLAS, but its latest version, aCT2, would allow using it for other workflows as well. Figure 5 shows components of aCT in the current case of ATLAS production deployment. The ATLAS-specific modules are separate from the ARC-specific ones, and only share a common database instance. aCT is developed using ARC API and libraries (ARC SDK), and since it is modular, it is quite straightforward to replace ATLAS-specific components with those supporting other workflows.

In the ATLAS production scenario, aCT presents itself as a computational resource, picking job descriptions from PanDA [Maeno T., 2008], converting them into ARC-specific XRSL job descriptions, and then submitting and managing jobs on ARC-CEs. Upon job completion, aCT fetches output files, handles common failures in case such occurred, and updates PanDA with job status and other required information.

The modules of aCT are known as actors, each responsible for specific actions. ARC actors are: *submitter, status checker*, *fetcher* and *cleaner*, and ATLAS actors are: *autopilot*, *panda2arc*, *atlas status checker* and *validator*. The names of actors speak for themselves; detailed aCT documentation is expected to be released in near future.

4. Gateways to supercomputers

WLCG computing has so far relied mostly on dedicated resources, configured to meet the needs of LHC experiments. However, the current trend of streamlining research computing by investing into

large-scale HPC or Cloud centres motivates LHC experiments to investigate possibilities of using such non-traditional resources. In particular, national research HPC systems are attractive because of guaranteed long-term funding and massive CPU capacities. However, such systems are not designed for high-throughput data processing of the kind used by the LHC experiments. Still, using them for Monte Carlo generation is quite a feasible task.

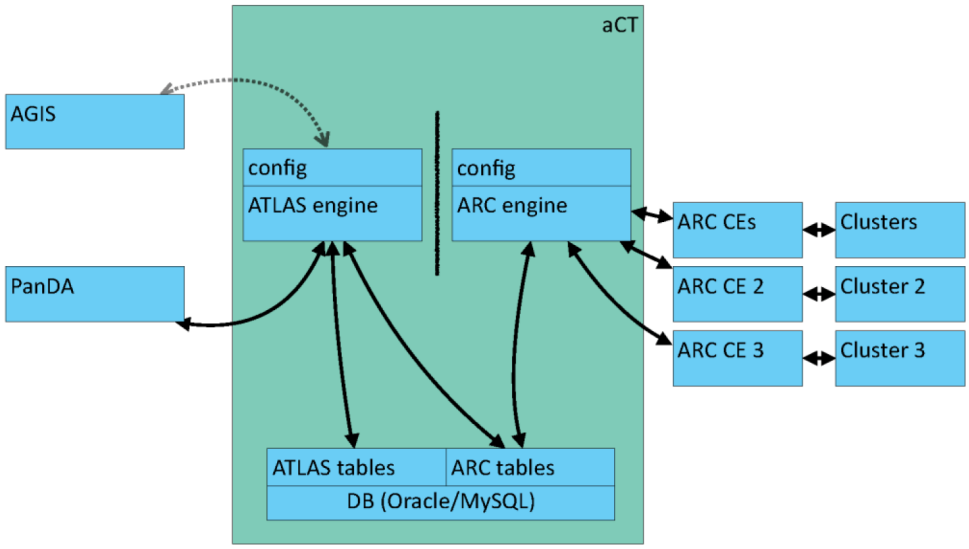


Fig. 5. aCT scheme for deployment with ATLAS' PanDA system; ATLAS-specific modules are clearly separated from ARC-specific ones

ARC does not require installation of any additional software on the worker nodes, neither it requires network connectivity for them. This makes it possible to use ARC-CE as a gateway for HPC systems, establishing interaction to the batch system via *ssh* or via aCT deployed on-site. These approaches are being tested on several HPC systems in Europe, such as *C2PAP/SuperMUC* and *Hydra* in Germany, and *Piz Daint* in Switzerland. Though initial tests are encouraging, and allow for certain types of ATLAS production jobs, still, a lot remains to be done on both HEP and HPC sides to make it useable.

In order to make a reasonable use of HPC resources, the following challenges have to be addressed:

- WAN access on worker nodes of HPC systems is limited (or absent), while it is still needed by many jobs in order to communicate to other services, such as databases
- Job scheduling has to be flexible enough to allow for cases like whole-node, whole-socket or whole-partition scheduling
- Shared file systems used by HPC sites are not necessarily optimal for heavy Input/Output, especially when thousands of processes do simultaneous read and write
- Traditionally, HPC sites offer dedicated login/edge nodes to the users; access to these nodes is strictly controlled and rather limited, in a manner not consistent with multi-user VOs
- In general, HPC policies and procedures are not suitable for WLCG use cases, where multi-user VOs use robot credentials and require installation of hundreds of different versions of proprietary software

Typically, HPC sites are tuned for few classes of massively parallel applications with relatively low I/O, allocate limited (not permanent) time slots to well-identified users, and only allow remote access via a *ssh*-login front nodes. Moreover, HPC systems rarely use Scientific Linux for OS, while it still remains the only supported OS for WLCG. This usage model is clearly not suitable for HEP computing.

There are still ways to meet requirements of both worlds, at least to some extent. Some site policies allow deployment of ARC-CE service machines: they can either be ported to the host operating

system, or binary-compatible packages can be used where possible. If site policies require deploying ARC-CE in a user mode, it can be adapted to run from a non-privileged user account; this however limits the usability of the system, as typically a UID can be mapped to only one batch account. In cases when site nodes have no WAN access, a more complex configuration using aCT as a gateway is possible; this requires manual or semi-automatic synchronization of necessary software and possibly of databases, for off-line usage. Tests showed that this may cause heavy load on shared file systems, thus not every site might be able to deploy such a setup. In cases when even the edge nodes have limited connectivity, ARC-CE needs to be deployed outside of the site, communicating via the *ssh*-back-end, which is currently being developed.

5. ATLAS@Home

Volunteer computing is credited as being one of the Grid pre-cursors, but it is rarely considered as a serious resource for the LHC needs. Still, BOINC-based LHC@home project clocked 2 Tera-FLOPS, which is perhaps not as impressive as hundreds of TeraFLOPS of SETI@home or Einstein@home, but still a valuable contribution. Apart of providing Cloud-like resources for free, it is a very useful mean for public outreach and popularization of LHC physics.

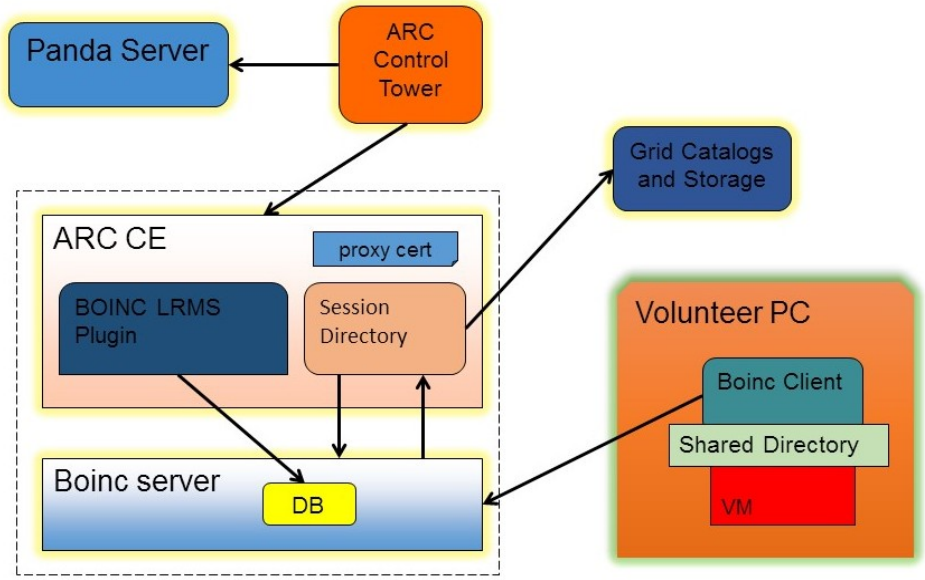


Fig. 6. Scheme of the ATLAS@home volunteer computing project: ATLAS jobs are retrieved from the PanDA server by aCT, and submitted to ARC-CE with a BOINC back-end

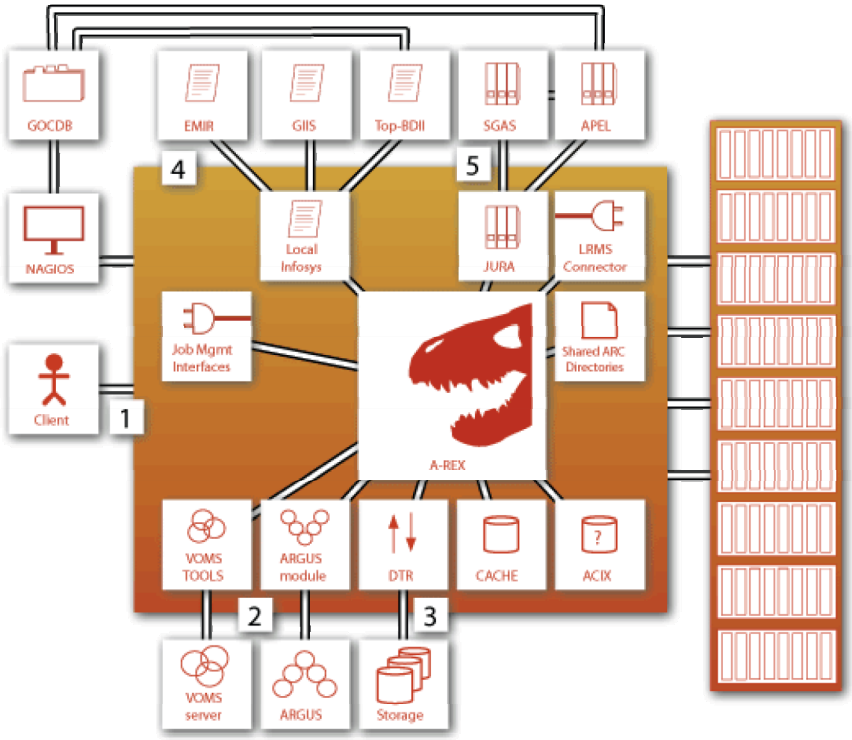
ATLAS recently has launched a BOINC-based volunteer computing project ATLAS@home [ATLAS@home Web site], which makes use of the ARC Control Tower and ARC-CE with a specially crafted BOINC back-end to populate BOINC server with real production jobs. The overall setup is shown in Figure 6.

There are certain considerations to be taken into account, related to the nature of volunteer computing resources. Clearly, one cannot rely on them for top-priority or data-intensive tasks, thus jobs suitable for ATLAS@home are low-priority jobs with high ratio of CPU to I/O, such as non-urgent Monte Carlo simulations. Virtualization is needed for ATLAS software environment, which is achieved through usage of CERNVM images and CVMFS-based software deployment. Since volunteer hosts have no Grid credentials and have access to Grid storages, data need to be staged by ARC middleware components. Ultimately, aCT makes the volunteer resources looking much like a regular queue from the PanDA point of view.

ATLAS@home enjoys a rather unexpected popularity, and at times provides more processing power than some Tier2 centres. Despite the fact that the individual contributors are unknown to the physicists and are not subject to standard WLCG operational procedures, control or monitoring, the service they offer is very valuable and much appreciated.

6. Integration with EGI operations

Thanks to the EMI efforts, ARC-CE is now well-integrated with EGI. Figure 7 shows relation of ARC-CE and its internal services to such EGI components as accounting, VO management, authorisation, monitoring, indexing and cataloguing. This allows for quite a smooth migration from other WLCG CEs (usually, CREAM-CE) to ARC-CE, and an increasing number of WLCG sites are switching to ARC-CE these days. Even if occasional glitches are discovered during deployment of ARC-CEs in previously untested configurations, such issues are quickly solved thanks to the active community of code contributors and openness for new contributions.



1. Job submission (brokering based on info from GIIS, EMIR, Local Infosys and ACIX)
2. Check credentials (VOMS, ARGUS, etc.)
3. Data staging from/to external storage
4. Registration to information indices (EGIIS, EMIR); serving information requests of global aggregators (Top-BDII)
5. JURA parses job logs, prepares and sends job usage records to either SGAS or APEL accounting databases

Fig. 7. Relation of ARC-CE and its internal services to EGI components and other relevant Grid services

7. Summary and outlook

ARC-CE is a well-established Grid computing service, used by WLCG and other Grid sites well beyond its Nordic origin. Being a community-driven effort, ARC benefits from knowledge and expertise brought in by every new site, and the list of ARC code contributors keep growing. Particularly active contribution area is back-ends to various flavors of batch systems, and even to such non-traditional resources as volunteer computing or *ssh*-accessible HPC systems.

ARC Control Tower, serving as a gateway between production systems and ARC-based resources, opens up many new possibilities of adding computing power to WLCG. Most notable examples are aCT usage in conjunction with HPC resources, and an amazing success of the ATLAS@home project, which makes use of aCT, ARC-CE and BOINC. aCT is still work in progress, and more tuning is needed to optimize its performance. Documentation and proper packaging of aCT are other important tasks that welcome contributors.

Future of ARC is inevitably linked to the ever increasing LHC computing requirements; immediate focus is on enhancing support for inclusion of HPC systems into WLCG, and adding support for more batch system options, particularly those related to multi-core processing. aCT2 experience will also help to develop more user-friendly task schedulers, possibly even for other communities outside WLCG.

References

ATLAS@home Web site URL http://atlasathome.cern.ch

dCache Web site URL http://www.dcache.org

EGI Web site URL http://www.egi.eu

Ellert M et al. Future Gener. Comput. Syst. 2007. 23 219-240 ISSN 0167-739X

European Middleware Initiative Web site URL http://www.eu-emi.eu

Filipčič A et al 2011 J. Phys.: Conf. Ser. 331 072013

HTCondor Web site URL http://research.cs.wisc.edu/htcondor/

Maeno T. J. Phys.: Conf. Ser. 119 062036. 2008

NeIC Web site URL http://neic.nordforsk.org

SLURM Workload Manager Web site URL http://slurm.schedmd.com

Worldwide LHC Computing Grid Web site URL http://lcg.cern.ch

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 415–419

Ки&М

ПЛЕНЕРНЫЕ ДОКЛАДЫ

УДК: 004.75

Обновления аппаратно-программной базы ALICE перед вторым запуском Большого адронного коллайдера

А. К. Зароченцев[1,a], Г. Г. Стифоров[2,b]

[1] Санкт-Петербургский государственный университет,
Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35
[2] Лаборатория физики высоких энергий, Объединенный институт ядерных исследований,
Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6

E-mail: [a] andrey.zar@gmail.com, [b] gleb.stiforov@cern.ch

Получено 27 октября 2014 г.

В докладе представлен ряд новостей и обновлений ALICE computing к RUN2 и RUN3.
В их числе:
– ввод в работу новой системы EOS;
– переход к файловой системе CVMFS для хранения ПО;
– план решения проблемы Long Term Data Preservation;
– обзор концепции “O square”, совмещающей офлайн- и онлайн-обработку данных;
– обзор существующих моделей использования виртуальных облаков для обработки данных ALICE.
Ряд нововведений показан на примере российских сайтов.

Ключевые слова: GRID, ALICE, CERN, LHC, WLCG, CVMFS, виртуализация

ALICE computing update before start of RUN2

A. K. Zarochentsev[1], G. G. Stiforov[2]

[1] Saint Petersburg State University, 35 University ave., St. Petersburg, Peterhof, 198504, Russia
[2] Laboratory of High Energy Physics, Joint Institute for Nuclear Research, 6 Joliot Curie St., Dubna, 141980, Russia

Abstract. — The report presents a number of news and updates of the ALICE computing for RUN2 and RUN3.
This includes:
– implementation in production of a new system EOS;
– migration to the file system CVMFS to be used for storage of the software;
– the plan for solving the problem of "Long-Term Data Preservation";
– overview of the concept of "O square", combining offline and online data processing;
– overview of the existing models to use the virtual clouds for ALICE data processing.
Innovations are shown on the example of the Russian sites.

Keywords: GRID, ALICE, CERN, LHC, WLCG, CVMFS, Virtualisation

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 415–419 (Russian).

ALICE: RUN1 и RUN2

В 2011 году завершился первый этап работы LHC, или RUN1. Для сохранения и обработки данных в RUN1 использовалась схема, которая полностью удовлетворяла потребностям эксперимента: вычислительная структура основана на GRID-системах WLCG и ARC, в которых VOBox используется для связи с GRID-структурой ALIEN (ALIce ENvironment) [Shabratova, 2010]. Структура хранения данных основана на xrootd-серверах, привязанных к сайтам через VOBOXEs. Структура доступа и обновления программного обеспечения (далее — ПО) основана на торрентах [Shabratova, 2012]. Авторизация организована через x509-аутентификацию и LDAP. Отдельно написан модуль libXrdAliceTokenAcc для авторизации xrootd. Мониторинг работает на основе MonALISA framework [MONitoring Agents…]. Более подробное описание структуры можно найти в докладах [Shabratova, 2010; Shabratova, 2012].

За первый этап эксперимент сохранил более 16 PB данных и постоянно обрабатывалось в среднем до 50 тысяч задач одновременно. На втором этапе работы LHC энергии столкновений возрастут более чем на 60 %, что даст значительный объем данных. В результате потребуется увеличить объемы хранения и производительность обработки более чем в 2 раза.

Решить задачу получится с помощью наращивания ресурсов и улучшения ПО. Например, улучшения систем обработки данных, качества доступа к хранилищам, системы доступа к обновляемому ПО, а так же виртуализации части ресурсов.

Обновления аппаратно-программной базы ALICE

Команда ALICE произвела ряд серьезных изменений в схеме хранения и обработки данных, которые введет к концу 2014 года, на более чем 90 сайтах по всему миру (рис. 1).

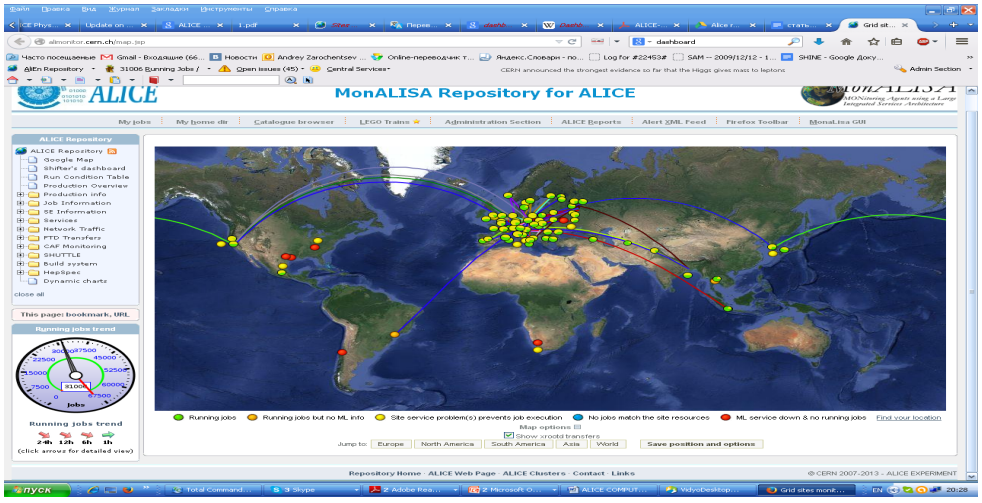


Рис. 1. http://alimonitor.cern.ch/map.jsp xrootd transfers

Например, к концу 2014 года все сайты перейдут на новую версию xrootd. В этой версии исправлены ошибки предыдущих пакетов, добавлена поддержка протокола IPV6 и новых файловых систем, в том числе распределенных, например CEPH [XROOTD project...]. Некоторые сайты планируют перейти на EOS, это система управления xrootd-серверами. EOS позволяет: централизованно администрировать серверы и объединять xrootd-серверы в RAID, что автоматизирует процесс восстановления системы в случае сбоя отдельных компонентов. EOS рекомендуется устанавливать как на сайты уровня Tier-1 (например, сайт RRC-KI-T1) с поддержкой всех компонент, так и на Tier-2 (например, сайт MEPHI) с минимальными требованиями к ресурсам и конфигурации. Подробнее о пакете EOS, для ALICE, можно узнать из докладов Андреаса Петерса [Andreas-Joachim Peters, 2013].

Долгое время для WLCG-сайтов была актуальна проблема с доступом к обновляемому программному обеспечению различных виртуальных организаций (ВО). Стандартным решением проблемы было предоставление менеджеру ВО доступа к некоей директории, доступной на нодах по NFS. На практике данный подход имел недостаток: файловая система NFS замедляла или останавливала работу сайта при 100 и более нодах. Дополнительно к этому обновление ПО зависело от работы VOBox, на котором этот автоматизированный процесс регулярно давал сбои и ошибки.

Альтернативный вариант: засылать весь необходимый обновленный код вместе с данными к задаче, так как самих данных существенно больше в задачах WLCG. В этом случае увеличивался трафик для каждой задачи, что суммарно для всех задач отражалось на производительности. В 2011 и 2012 годах команда ALICE предложила использовать для передачи пакетов ПО p2p-протокол или торренты [Shabratova, 2012].

Данный подход позволял не посылать полный пакет для каждой отдельной задачи. Взамен этого задачи опрашивали ближайшие ноды на наличие пакетов и скачивали необходимые файлы по частям с ближайших источников. В случае отсутствия ближайших источников необходимое ПО скачивалось по http-протоколу с центрального сервера. Использование http-протокола позволяло использовать кэширующий прокси, что экономило трафик, даже если администратор сайта по соображениям безопасности закрывал возможность использования торрентов. Такое решение позволяло экономить трафик по сравнению с вариантом прикрепления пакетов к каждой задаче и решало вопрос с надежностью по сравнению с вариантом использования NFS. В связи с недоверием локальных и сетевых системных администраторов к использованию торрентов зачастую все пакеты всё равно скачивались по http-протоколу. В этом случае трафик экономили только за счет использования прокси.

В 2012 году было найдено новое решение этой проблемы — базируемая на http-протоколе сетевая файловая система CVMFS (CernVM File System (CernVM-FS)) [CernVM File System…, CernVM 3 and…]. CVMFS позволяет использовать кэшируемый прокси для доступа к данным с рабочих нод и централизованно обновлять необходимые данные. В этом случае экономия трафика достигалась, как и в случае с торрентами, за счет использования http- и прокси-сервера. Но CVMFS также позволяет структурировать информацию, создавать отдельные ветви и многое другое**.** До середины 2013 года приняли все виртуальные организации (ВО) WLCG, а к апрелю 2014 года и ALICE планировала перевести все свои сайты на CVMFS, но в итоге все сайты перешли на данную сетевую файловую систему уже к январю 2014. Причем в настоящее время в ALICE computing model ПО с CVMFS используется не только на рабочих нодах, как у остальных ВО, но и на VOBox.

CVMFS изначально была разработана как файловая система для виртуальных машин для минимализации загрузочного образа и гибкости в конфигурации загружаемой системы. На рис. 2 приведена схема mCernVM (micro CERN Virtual machine). mCernVM представляет собой минимальный загрузочный образ, включающий ядро, модуль CVMFS объемом коло 12 МБ и файл contextualization около 64 КБ, включающий настройки подключения к CVMFS-серверу, выбора соответствующей ветки операционной системы и набора программного обеспечения.

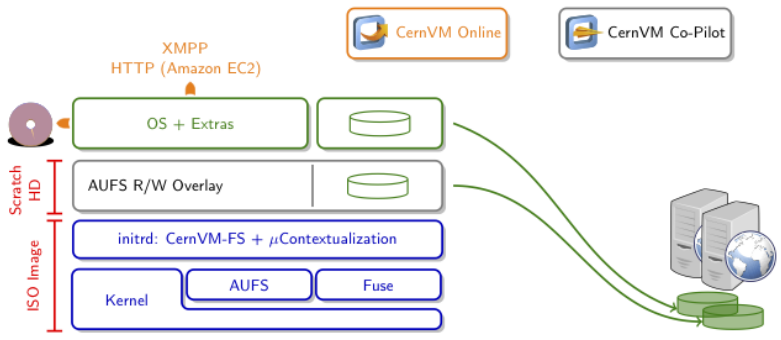


Рис. 2. Структура mCVM

Такой подход позволяет загрузить виртуальную машину с любым доступным набором операционной системы (ОС) и ПО, что дает возможность решить проблему длительного сохранения данных конфигурации для тех или иных вычислений (LTDR — long term data reservation).

В настоящее время довольно много составляющих системы обработки данных CERN переходит в виртуальную среду (HLT-кластер, виртуальные PROOF-кластеры и т. д.) и рассматриваются планы перехода в виртуальную среду и других компонент. В RUN3 планируется перевести вычислительные ресурсы ALICE на облачные системы. Эти системы повысят гибкость использования вычислительных ресурсов, что даст возможность использовать ресурсы поочередно (по требованию) для обработки «сырых» и накопленных данных (онлайн + офлайн). Подробнее о новом подходе к обработке данных “O square” в докладе Предрага Бунчича [Buncic, 2014].

В связи с обновлением структур хранения данных и системы доступа к ПО потребовалось обновление мониторинга. На данный момент у интерфейса http://alimonitor.cern.ch/map.jsp обновили google map до google maps API v3 и добавили возможность наглядно отслеживать xrootd-трафик.

Если ранее взаимодействие сайтов оценивалась только по VOBox, то сейчас отслеживается напрямую обмен файлами по протоколу xrootd (рис.1), как и другая информация по xrootd серверам [Grigoras, 2014]. Отдельно мониторится состояние CVMFS на сайтах [Grigoras, 2014; Grigoras Publishing…].

Страница со статусом отдельных сайтов и описанием проблем сильно облегчила работу администраторов и региональных менеджеров: http://alimonitor.cern.ch/siteinfo/issues.jsp. Значительно расширились возможности личного кабинета, откуда можно запускать собственные расчеты и отслеживать их выполнение в GRID- и AAF- (ALICE Analysis Facility) ресурсах.

Участие Российских сайтов в обновлении аппаратно-программной базы ALICE

Основное достижение российского сектора ALICE GRID — запуск сайта уровня Tier-1 на базе Национального исследовательского центра «Курчатовский институт», RRC-KIAE-T1. В данный момент сайт представляет 150 ТБ дисковых накопителей, более 4000 вычислительных слотов (ядер), планируется внедрять ленточные хранилища. Это сайт — первый из Tier-1, на котором был установлен EOS в качестве системы хранения. Кроме RRC-KIAE-T1 EOS был в 2014 году установлен еще на двух российских сайтах уровня Tier-2, это сайт Санкт-Петербургского государственного университета SPbSU и сайт Национального исследовательского ядерного университета «МИФИ» MEPHI. Про последний сайт стоит сказать отдельно — он был возвращен в активное использование, после двух лет простоя, только в декабре 2014 года.

Российский сегмент в 2013 году пережил спад производительности из-за проблем с сетью GEANT, однако уже к январю 2014 года полностью восстановил старые показатели. В отдельных институтах, таких как СПбГУ и ОИЯИ ведутся работы по адаптации облачных технологий для нужд ALICE computing.

Список литературы

A.-J. Peters EOS CERN Disk Storage, Varna, Bulgaria, 2013. http://nec2013.jinr.ru/files/12/Peters.pdf

CernVM File System. http://cernvm.cern.ch/portal/filesystem

CernVM 3 and μCernVM Beta Release. http://cernvm.cern.ch/portal/ucernvm

Buncic P. ALICE Computing Model RUN2, Tsukuba, Japan, 2014. http://indico.cern.ch/event/274974/contribution/33/material/slides/1.pdf

Grigoras C. News of MonALISA site monitoring, Tsukuba, Japan, 2014. https://indico.cern.ch/event/274974/contribution/87/material/slides/1.pdf

Grigoras C. Publishing ALICE data & CVMFS infrastructure monitoring. 2014. https://indico.cern.ch/event/321470/contribution/4/material/slides/1.pdf

MONitoring Agents using a Large Integrated Services Architecture. http://monalisa.cern.ch/monalisa.html

Shabratova G. The ALICE GRID operation, GRID 2010, Dubna, Russia, 2010. http://grid2010.jinr.ru/files/pdf/grid2010.pdf

Shabratova G. Torrent base of software distribution by ALICE at RDIG, GRID 2012, Dubna, Russia, 2012. http://grid2012.jinr.ru/docs/grid2012.pdf

XROOTD project. http://xrootd.org/

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 421−427

Ки&М

ПЛЕНЕРНЫЕ ДОКЛАДЫ

УДК: 004.7

Preliminary Study of Big Data Transfer over Computer Network

S. E. Khoruzhnikov[1], V. A. Grudinin[1], O. L. Sadov[1], A. Y. Shevel[1, 2], A. B. Kairkanov[1,a]

[1] ITMO University St. Petersburg, 49 Kronverksky Ave., St.Petersburg, 197101, Russia
[2] National Research Centre "Kurchatov Institute" B. P. Konstantinov, Petersburg Nuclear Physics Institute, Orlova Roscha, Gatchina, 188300, Russia

E-mail: [a] arsen.kairkanov@gmail.com

Получено 1 декабря 2014 г.

The transfer of Big Data over computer network is important and unavoidable operation in the past, now and in any feasible future. There are a number of methods to transfer the data over computer global network (Internet) with a range of tools. In this paper the transfer of one piece of Big Data from one point in the Internet to another point in Internet in general over long range distance: many thousands kilometers. Several free of charge systems to transfer the Big Data are analyzed here. The most important architecture features are emphasized and suggested idea to add SDN Openflow protocol technique for fine tuning the data transfer over several parallel data links.

Keywords: data, Linux, transfer, SDN, Openflow, network

Предварительное изучение передачи больших данных по компьютерной сети

С. Э. Хоружников[1], В. А. Грудинин[1], О. Л. Садов[1], А. Е. Шевель[1,2], А. Б. Каирканов[1]

[1] Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Россия, 197101, г. Санкт-Петербург, Кронверкский проспект, д. 49
[2] Национальный исследовательский центр «Курчатовский институт», Петербургский институт ядерной физики имени Б.П. Константинова, Россия, 188300, Ленинградская обл., Гатчина, Орлова роща, ФГБУ ПИЯФ

Передача больших данных по компьютерной сети — это важная и неотъемлемая операция в прошлом, настоящем и в любом обозримом будущем. Существует несколько методов передачи данных по глобальной компьютерной сети (Интернет) с помощью ряда инструментов. В этой статье рассматривается передача данных из одной точки Интернета в другую точку Интернета в основном на большие расстояния: многие тысячи километров. В статье представлен анализ нескольких бесплатных систем передачи больших данных. Подчеркиваются наиболее важные архитектурные особенности и предлагается идея использования технологии ПКС на базе протокола Openflow для улучшения процесса передачи данных по нескольким параллельным каналам связи.

Ключевые слова: данные, Линукс, передача, ПКС, Openflow, сеть

The work is supported by the Saint-Petersburg National Research University of Information Technology, Mechanics & Optics (www.ifmo.ru).

Citation: *Computer Research and Modeling*, 2015, vol. 6, no. 3, pp. 421–427 (Russian).

I. Introduction

The “Big Data” [Big Data, 2014] is known problem for many years. In each period the term “Big Data” does mean different volume and character of the data. Keeping in mind “triple V”: Velocity, Volume, Variety we can pay attention that all those features are relative to current state of the technology. For example in 1980-s the volume of 1 TB was considered as huge volume. There is a range of aspects of the problem: store, analyze, transfer, etc. In this paper we discuss one of important aspects of the Big Data — the transfer over global computer network.

II. The sources of the Big Data

It is known the long list of human activities (scientific and business) which are the generators of large volume of data [Information Revolution ..., 2014; Square Kilometer Array, 2014; Large Synoptic Survey Telescope, 2014; Facility for Antiproton and Ion Research, 2014; International Thermonuclear Experimental Reactor, 2014; CERN, 2014; Lucinda Borovick Richard L. Villars, 2013; The Center for Large-scale Data Systems Research …, 2013; Johnston et al., 2013].

In according [Information Revolution ..., 2014] total volume of business mails in the World in year 2012 is around 3000 PB (3×10^{18}). The consensus estimation for the total volume of stored data is growing 1.5-2.0 times each year starting from 2000. In this paper (and for our tests) we will assume that volume of data around 100 TB (10^{14}) and more could be labeled as Big Data. Quite probably the volume of Big Data will grow with the time.

Another source of Big Data — the preservation of the data for long periods of time: several tens or more years. Many aspects of our personal, society, technical, and business life are now held in digital form. Large volume of those data needs to be stored and preserved. For example, results of medicine tests, data generated by important engines of various kinds (airplane engines, power station generators, etc) and other data have to be archived for long time. The preserved data will be kept in distributed (locally and globally) storage. It is assumed that replicas of preserved data have to be stored in several places (continents) to avoid data loss due to technical, nature or social disasters.

Historically one of the first field where Big Data came into reality was experiments in High Energy Physics (HEP). As the result a number of aspects for data transfer were analyzed and a range of problems was solved. Now more and more scientific and business sectors are dealing (or plan to) with the “Big data” [Information Revolution ..., 2014; Square Kilometer Array, 2014; Large Synoptic Survey Telescope, 2014; Facility for Antiproton and Ion Research, 2014; International Thermonuclear Experimental Reactor, 2014; CERN, 2014; Tierney et al.]. Last time the interest to data transfer of increasing volumes is growing [Nam et al., 2013; Gunter Dan et al., 2012].

III. Freely available utilities/tools for data transfer over the network

The time to transfer over global computer network (Internet) depends on the real data link bandwidth and volume of the data. Taking into account that we talk about volume 100TB and more we can estimate minimum required time for data copy over the network link with 1 Gbit capacity. It will give us about 100MB/sec, hence 100TB/100MB = 1000000 secs = 277.8 hours = 11.6 days. During this time the parameters of the network link might be changed. For example percent of dropped network packages and other data link parameters can be varied significantly. The data link might be suffered of operation interruptions for different period: secs, hours, days. Also important a lot of Linux kernel network parameters. There are several hundreds of kernel network parameters. Not all of them are equally sensitive or influencing. Among most important of them it is good to mention TCP Window size, MTU, congestion control algorithm, etc. Of course quite important the number of independent network links which could be used in parallel. Finally it is seen that in each data transfer of large volume we need to be able to tune (to set) different number of threads, different size of TCP Window, etc.

Now it is time to observe freely available data transfer tools/utilities which might be used to transfer Big Data over the network.

A. Ideas to Compare the data transfer utilities

First of all quick consideration of parameters to compare the data transfer utilities which might help to transfer Big Data.

- Multi-stream data transfer mode — is ability to use several TCP streams in parallel.
- Multi-link data transfer mode — ability to use more than one data link in parallel; important feature especially if it is possible to take into account that available network links are not equal in bandwidth and in conditions (reliability, price, real status, etc).
- Possibility to set parameters low level parameters e.g. TCP Window size, etc.
- Ability in case of failure of the data transfer to continue the data transfer from point of failure.

In reality the data transfer consists of many steps: read the data from the storage, transfer the data over network, write the received data to the storage on remote computer system. In this paper our attention is concentrated more on network transfer process.

B. Low level data transfer utilities/tools

We could mention several utilities for the data transfer over the network (at least part of them are known for around ten years):

- one of low level protocols to transfer the data over the network is UDT [UDT: Breaking …, 2014]. UDT is library which implements data transfer protocol which permit to use *udp*, but not *tcp*. In some cases the library can help to improve data link usage, i.e. to reduce the data transfer time.
- the protocol RDMA over Converged Ethernet (RoCE) [Tierney et all.] has been studied and it was found that in many cases RoCE shows better results than UDP, UDT, conventional TCP.
- MP TCP [MutiPath TCP …, 2014] is interesting protocol which permits to use several data links in parallel for one data transfer. The protocol is implemented as Linux kernel driver.
- openssh family [OpenSSH, 2014] — well known data transfer utilities deliver strong authentication and a number of data encryption algorithms. Data compression before encryption to reduce the data volume to be transferred is possible as well. There are two well known openSSH flavors: patched SSH version [Patched OpenSSH, 2014] which can use increased size of buffers and SSH with Globus GSI authentication. No real restart after failure. No parallel data transfer streams.
- bbcp [BBCP — utility to transfer …, 2014] — utility for bulk data transfer. It is assumed that bbcp is running on both sides, i.e. transmitter, as client, and receiver as server. Utility bbcp has many features including the setting:
    - TCP Window size;
    - number of TCP streams;
    - I/O buffer size;
    - resuming failed copy;
    - authentication with ssh;
    - using pipes, where source or/and destination might be pipe;
    - special option to transfer small files;
    - and many other options dealing with many practical details.
- bbftp [BBFTP — Utility for bulk …, 2014] — utility for bulk data transfer. It implements its own transfer protocol, which is optimized for large files (larger than 2GB) and secure as it does not read the password in a file and encrypts the connection information. bbftp main features are:
    - SSH and Grid Certificate authentication modules;
    - multi-stream transfer;
    - big TCP windows as defined in RFC1323;

- automatic retry;
- customizable time-outs;
- other useful practical features.

- Xdd [Hodson et al., 2013] — utility developed to optimize data transfer and I/O processes for storage systems.
- fdt [Fast Data Transfer, 2014] — Java utility for multi-stream data transfer.
- gridFTP [Grid/Globus data, 2014] is advanced reincarnation of well known utility *ftp* redesigned more than 10 years ago for globus security infrastructure (GSI) environment. The utility has many features and main usage of those are:
  - two security flavors: Globus GSI and SSH;
  - the file with host aliases: each next data transfer stream will use next host aliases (useful for computer cluster);
  - number of parallel data transfer streams;
  - buffer size;
  - restart failed operations and number of restarts.

Many of mentioned utilities are quite effective for data transfer from point of view of link capacity usage. However Big Data transfer assumes significant transmission time (may be many hours, days or more). For long time it is not easy to rely on those quite simple transfer procedures.

C. Middle level File Transfer Service

The FTS3 [File Transfer Service, 2014] is relatively new and advanced tool for data transfer of large volume of the data over the network. It has most features already mentioned above and more. There is advanced data transfer tracking (log) feature, ability to use http, restful, and CLI interfaces to control the process of the data transfer.

Another interesting development is SHIFT [Data Transfer Tools, 2014] which is dedicated to do reliable data transfer in LAN and WAN. There was paid much attention to the reliability, advanced tracking, performance of the data transfer and the usage of parallel data transfer between so called equivalent hosts (between computer clusters).

D. High level data management service: PhEDEx

PhEDEx — Physics Experiment Data Export is used (and developed) in collaboration around Compact Muon Solenoid (CMS) experiment [The CMS Collaboration ..., 2008; Kaselis, 2012; PhEDEx — CMS Data Transfers, 2014; PHEDEX data ..., 2014] at CERN [CERN, 2014]. The experiment does produce a lot of experimental data (in 2013 it was written around 130 PB). Data analysis requires to copy of the data in a range of large computing clusters (about 10 locations in different countries and continents) for analysis and data archiving. Later on the fractions of the data might be copied to smaller computing facilities (more than 60 locations). Total data transfer per day is achieved 350 TB/day [Kaselis, 2012]. It is possible that in nearest future the volume per day will be increased. Because in between several sites there are more than one link in PhEDEx there were developed routing technique which permit to try alternative route when default route is not available.

Finally the system PhEDEx is quite complicated and the management service depends on the physics experiment collaboration environment. It is unlikely that PhEDEx is possible to use without redesign in different environment.

IV. Consideration

Mentioned utilities have several common useful features for data transfer. Among them:

- client-server architecture;

- ability to set the buffer size, TCP Window size, etc;
- ability to perform various operations before real data transfer and after data transfer, use a range of drivers/methods to read/write files to/from secondary storage, etc;
- use more than one of authentication techniques;
- use a number of transfer streams;
- use in some conditions more than one network link for data transfer;
- usage of a number of techniques to make data transfer more reliable.

The utilities are not equal in number of parameters and scope of suggested tasks. Part of them are well suited to be used as independent data transfer utilities in almost any environment. Others, like PhEDEx (in CMS) and comparable systems in collaboration ATLAS [The Rucio project ..., 2014] are dedicated to be used as part of more complicated and specific computing environment.

In other words there is stack of toolkit which might help in many cases to transfer the Big Data over networks. At the same time it is seen that quite a few utilities can use more than one network link.

At the same time no tool suggests fine tuning with parallel data links. Fine tuning is considered as possibility to apply the different policy to each data link. In general parallel data links might be completely different in nature, features, and conditions of use. In particular it is assumed individual QoS for each network link to be used in data transfer and ability to change the policy on the fly. All that give the idea that special application is required which might watch the data links status and change the parameters of data transfer accordingly to real situation in the data links. QoS is planned to be set with protocol Openflow [Open Networking ..., 2013; Nunes et al., 2014]. The special tool PerfSonar [Zurawski et al., 2013] will be used to watch the data links status.

There is special aspect in the procedure of the comparison of the utilities to transfer the Big Data over the computer network. The real networks are different from each other. All above circumstances give the idea that to compare the variety of data transfer utilities (especially for Big Data) demands the customized testbed which is able to simulate at least main network problems, e.g. changing RTT, delays, package drop percent, and so on. Such the testbed development has been started at the network laboratory [Laboratory …, 2014]. The need for testbed is becoming obvious by previously obtained measurement results [Nam et al., 2013]. Here is seen the comparative measurements for one data transfer stream and many streams. The data transfers were performed with special servers so called Data Transfer Nodes (DTN). DTNs have several specific techniques to transfer the data from LAN to WAN destinations. A number of utilities: rsync, scp, bbcp, GridFTP were discussed and measured just for concrete transferred file sizes (11 KB, 3.5 MB, 158 MB, 2.8 GB, and 32 GB) to transfer 6 files in each case. It was discovered that no change in the transfer speed after number of streams more than 8. At the same time no information about the Linux kernel parameters, how authors designated what the speed has been measured: data transfer speed over the data link or transfer speed from disk subsystem to main memory? It is planned to get answer on those questions in developed testbed. Also in the testbed we are taking into account the ideas expressed in [Gunter et al., 2012].

The testbed is intended to be platform to compare different utilities in the same environment. In addition it is planned to use advanced techniques with SDN approach to use parallel data links with use QoS on each data link. As the first step it is planned to perform comparative measurements with the range of data transfer utilities with writing all the measurement conditions details. That permits to compare in future other data transfer methods in exactly same environment in the testbed.

V. The testbed progress

Now the testbed consists of two servers HP DL380p Gen8 E5-2609, Intel(R) Xeon(R) CPU E5-2640 @2.50GHz, 64 GB under Scientific Linux 6.5. Because it is planned to test everything in virtual environment for each mentioned data transfer systems two virtual machines will be used. One VM as transmitter and another VM as receiver. In other words we have around ten VMs. The cloud infrastructure Openstack (version Icehouse) has been deployed to organize above VMs. PerfSonar has been deployed as well.

To study different types of data the special procedure has been developed to generate test directory with files of random length, the total volume of test directory is defined by the parameter of the procedure. During generation of test data it is possible to set mean value for file size and dispersion of the file size. The data inside each file in test directory is intentionally prepared to eliminate possible effect of the data compression (if any) during data transfer.

In initial stage it is planned to compare all the above data transfer systems in local area network to be sure that everything (all scripts) is functioning properly. The distinct problem is to write all logs, parameters, etc during the measurement. As it was mentioned earlier in the paper many parameter values in the directory /proc might affect the speed of the data transfer. That means the requirement to write automatically whole directory /proc into some place, let say "log directory". In addition there is need to write all the parameters used when data transfer starts. Also it is required to write all messages from data transfer engine/utility. Finally the data link status is intended to be written as well. All mentioned information has to be saved in "log directory". All those features have been implemented in the scripts dedicated to do measurements.

Developed scripts with short descriptions are available in https://github.com/itmo-infocom/BigData.

References

BBCP — utility to trasnfer the data over network — http://www.slac.stanford.edu/~abh/bbcp/.

BBFTP — Utility for bulk data transfe — http://doc.in2p3.fr/bbftp/.

Big Data — http://en.wikipedia.org/wiki/Big_data.

CERN — http://www.cern.ch/.

Data Transfer Tools — http://fasterdata.es.net/data-transfer-tools/

Facility for Antiproton and Ion Research — http://www.fair-center.eu/.

Fast Data Transfer — http://monalisa.cern.ch/FDT/.

File Transfer Service — FTS3 — http://www.eu-emi.eu/products/-/asset_publisher/1gkD/content/fts3; https://svnweb.cern.ch/trac/fts3

Grid/Globus data transfer tool. Client part is known as globus-url-copy — http://toolkit.globus.org/toolkit/data/gridftp/

Gunter Dan et al. Exploiting Network Parallelism for Improving Data Transfer Performance, High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:, DOI: 10.1109/SC.Companion.2012.337 — http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6496123

Hodson Stephen W., Poole Stephen W., Ruwart Thomas M., Settlemyer Bradley W. // Moving Large Data Sets Over High-Performance Long Distance Networks // Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008 Oak Ridge, 37831-6164 // http://info.ornl.gov/sites/publications/files/Pub28508.pdf [1.12.2013]

index.php?halsid=ig58511e1q1ekqq75uud43dn66&view_this_doc=hal-00825087&version=5

Information Revolution: Big Data Has Arrived at an Almost Unimaginable Scale — http://www.wired.com/magazine/2013/04/bigdata/.

International Thermonuclear Experimental Reactor — http://www.iter.org/.

Johnston William E., Dart Eli, Ernst Michael, Tierney Brian // Enabling high throughput in widely distributed data management and analysis systems: Lessons from the LHC — https://tnc2013.terena.org/getfile/402 (text) and https://tnc2013.terena.org/getfile/716 (presentation)

Kaselis R., Piperov S., Magini N., Flix J., Gutsche O., Kreuzer P., Yang M., Liu S., Ratnikova N., Sartirana A., Bonacorsi D., Letts J. CMS Data Transfer operations after the first years of LHC

collisions // International Conference on Computing in High Energy and Nuclear Physics 2012 (CHEP2012) IOP Publishing Journal of Physics: Conference Series 396 (2012) 042033. 8 p.

Laboratory of the Network Technology — http://sdn.ifmo.ru/

Large Synoptic Survey Telescope — http://www.lsst.org/lsst/.

Lucinda Borovick Richard L. Villars // White paper. The Critical Role of the Network in Big Data Applications — http://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns224/ns944/critical_big_data_applications.pdf [last read 1.12.2013]

MutiPath TCP — Linux Kernel Implementation — http://mptcp.info.ucl.ac.be/, http://multipath-tcp.org/

Nam Hai Ah et al. The Practical Obstacles of Data Transfer: Why researchers still love scp // November 2013 NDM'13: Proceedings of the Third International Workshop on Network-Aware Data Management — http://dl.acm.org/citation.cfm?id=2534695.2534703&coll=DL&dl=ACM&CFID=563485433&CFTOKEN=25267057

Nunes Bruno Astuto A., Mendonca Marc, Nguyen Xuan-Nam, Obraczka Katia, and Turletti Thierry // A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks — http://hal.inria.fr/

Open Networking Foundation White Paper Software-Defined Networking: The New Norm for Networks // https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf (last read: 1.11.2013)

OpenSSH — http://openssh.org/

Patched OpenSSH — http://sourceforge.net/projects/hpnssh/

PhEDEx — CMS Data Transfers — https://cmsweb.cern.ch/phedex

PHEDEX data transfer system — https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhedexAdminDocsInstallation) and http://hep-t3.physics.umd.edu/HowToForAdmins/phedex.html

Square Kilometer Array — http://skatelescope.org/.

The Center for Large-scale Data Systems Research at the San Diego Supercomputer Center — http://clds.sdsc.edu/ [last read 1.12.2013]

The CMS Collaboration 2008 The CMS experiment at the CERN LHC JINST 3 S08004

The Rucio project is the new version of ATLAS Distributed Data Management (DDM) system services — http://rucio.cern.ch/

Tierney Brian, Kissel Ezra, Swany Martin, Pouyoul Eric // Efficient Data Transfer Protocol for BigData — www.es.net/assets/pubs_presos/eScience-networks.pdf // Lawrence Berkeley National Laboratory, Berkeley, CA 94270 // School of Informatics and Computing, Indiana University, Bloomington, IN 47405

UDT: Breaking the Data Transfer Bottleneck — http://udt.sourceforge.net/.

Zurawski J., Balasubramanian S., Brown A., Kissel E., Lake A., Swany M., Tierney B., Zekauskas M. // perfSONAR: On-board Diagnostics for Big Data — http://www.es.net/assets/pubs_presos/20130910-IEEE-BigData-perfSONAR2.pdf [last reading date: 1.11.2013]. 6 p.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 429−437

Ки&М

ПЛЕНЕРНЫЕ ДОКЛАДЫ

УДК: 004.023

Высокопроизводительные вычисления на гибридных системах: будут ли решены «задачи большого вызова»?

А. В. Богданов[а], А. Б. Дегтярев[б], В. Н. Храмушин[в]

Санкт-Петербургский государственный университет (СПбГУ),
Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., 35

E-mail: [а]bogdanov@csa.ru, [б]deg@csa.ru, [в]v.khram@gmail.com

Получено 2 февраля 2015 г.

На примере расчета течений проводится анализ возможностей современных гибридных распределенных вычислительных систем для расчета «задач большого вызова». Приводятся соображения, что только многоуровневый комплексный подход к такой проблеме позволит эффективно масштабировать подобные задачи. Подход подразумевает использование новых математических моделей процессов переноса, разделение на динамическом уровне явлений переноса и внутренних процессов и использование новых парадигм программирования, учитывающих особенности современных гибридных систем.

Ключевые слова: гибридная система, «задачи большого вызова», тензорная математика, аэрогидродинамика, вычислительный эксперимент

High performance computations on hybrid systems: will "grand challenges" be solved?

A. V. Bogdanov, A. B. Degtyarev, V. N. Khramushin

St. Petersburg State University, 35 Universitetskii prospekt, Petergof, Saint-Petersburg, 198504, Russia

Abstract. — Based on CFD computations we provide the analysis of the possibilities for using modern hybrid distributed computational environments for large complex system simulation. We argue that only multilevel approach supported by new mathematical models of transport properties, dynamical representation of the problem with transport and internal processes separated, and modern paradigm of programming, taking into account specific properties of heterogeneous system, will make it possible to scale the problem effectively.

Keywords: hybrid system, “grand challenges”, tensor mathematics, aerohydrodynamics, computational experiment

Исследования выполняются при поддержке грантов РФФИ (№ 13-07-00747), СПбГУ (№ 9.38.674.2013, № 0.37.155.2014) и Комплексной программы ДВО РАН «Дальний Восток» (№ 15.3312-III-СО-08-023), вычислительные эксперименты осуществлялись на базе оборудования Ресурсного центра «Вычислительный центр СПбГУ».

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 429−437 (Russian).

Введение

Так называемые «задачи большого вызова» — «Grand challenges» — термин политики США, использованный в конце 1980-х годов в целях обозначения необходимости финансирования исследований в области высокопроизводительных вычислений и коммуникаций. Само понятие «Grand challenges» в научной литературе было введено нобелевским лауреатом Kenneth G. Wilson [Wilson, 1987; Wilson, 1989]. В это время был выделен круг фундаментальных и прикладных проблем, жизненно важных для развития человечества, эффективное решение которых возможно только с использованием сверхмощных вычислительных ресурсов.

С тех пор конкретный список задач много раз видоизменялся и конкретизировался, а сам термин «Grand challenges» стал нарицательным. В список входят комплексные задачи физики, нанотехнологий, аэронавтики, биологии, национальной безопасности, науки о Земле, энергетики, окружающей среды и т. д.

Если постараться понять, какие вычислительные средства необходимы для решения «задач большого вызова», то очень обобщенно можно заключить, что имеем дело с тремя принципиальными конфигурациями, которые направлены на решение:

1. задач обработки больших и сверхбольших объемов данных;
2. большого количества слабосвязных задач;
3. одной сильносвязной задачи, требующей большого объема памяти и производительности вычислительных мощностей.

Понимание того, что одним локальным аппаратным решением невозможно обеспечить решение большинства задач «Grand challenges», привело к разработке идей распределенных вычислений, появлению кластерных технологий и GRID. Появление многоядерных процессоров в какой-то мере также попытка эффективного объединения и создания локального высокопроизводительного вычислительного ресурса. Развитие сетевых технологий, улучшение канальной инфраструктуры, появление новых решений интеграции разнородных вычислительных ресурсов — все это сыграло положительную роль в решении задач «Grand challenges», однако не обеспечило желаемого уровня, особенно при решении задач третьего типа. Основная проблема в ресурсном обеспечении решения задач третьего типа заключается в неэффективности объединения большого количества вычислительных элементов при наличии временны́х затрат на подготовку данных (синхронизация, обмен данными и пр.). В то же время повышение сложности задачи ведет к вовлечению в расчетный процесс все большего количества вычислительных ядер, что становится с определенного момента фактором, снижающим ускорение. К настоящему моменту практически любые решения распределенных или кластерных вычислений для задач третьего типа оказываются неэффективными. Единственное возможное решение в сложившихся условиях компьютерного рынка, с нашей точки зрения, должно быть основано на гибридной архитектуре. Объединение традиционных многоядерных процессоров с GP GPU (General-Purpose Graphics Processing Units) дает возможность локального объединения на порядки большего количества вычислительных ядер, чем в кластерной и распределенной архитектуре. Однако в этом случае при отображении комплексной задачи на такую архитектуру возникает большое количество не решенных до сих пор задач (аппаратного, программного и даже математического плана). Так могут ли высокопроизводительные вычисления для решения задач «Grand challenges» быть решены на гибридной архитектуре? Рассмотрению этой проблемы посвящена данная статья.

Методологические основы решения задачи

Решение любой комплексной проблемы с использованием вычислительной техники можно с философской точки зрения представить в трехмерном логическом пространстве (рис. 1), в котором координатные оси образуют главные направления исследований.

«Проект» — согласование исходной задачи и ожидаемых результатов в избранной языковой среде. «Явление» — детальное описание физических законов. «Развитие» — этапы трансформации моделируемых процессов при построении вычислительного эксперимента.

На базисных направлениях формируются ортогональные предметные плоскости в качестве независимых областей научных знаний, определяющих искомую цель.

Плоскость ③ — «Язык компьютерных вычислений» — формирует стадии вычислительного эксперимента. (1) подготовка исходных данных («начало»), (2) алгоритм решения задачи («процесс»), (3) всесторонний анализ трансформации физических полей («результат»).

Плоскость ② — «Математические принципы» — образуется осями «Развитие» и «Явление». Таким образом, она представляет ход реализации и задействованные процессы в едином комплексе этапов вычислительного эксперимента. При этом алгоритмы и функциональная среда вычислительных объектов и операций относятся строго к моделированию конкретного физического «Явления». Элементы этой плоскости, подобно подлежащему и сказуемому в естественном языке, объединяют элементы программирования в математические и геометрические правила, методы вовлечения числовых объектов — в подготовку, исполнение и последующее представление результатов численного моделирования.

В плоскости ① — «Законы и правила механики» — обособляются аналитические зависимости и законы. Их формализация может осуществляться методами функционального программирования либо оформляться в виде привычных алгоритмических построений и статичных пакетов программ.

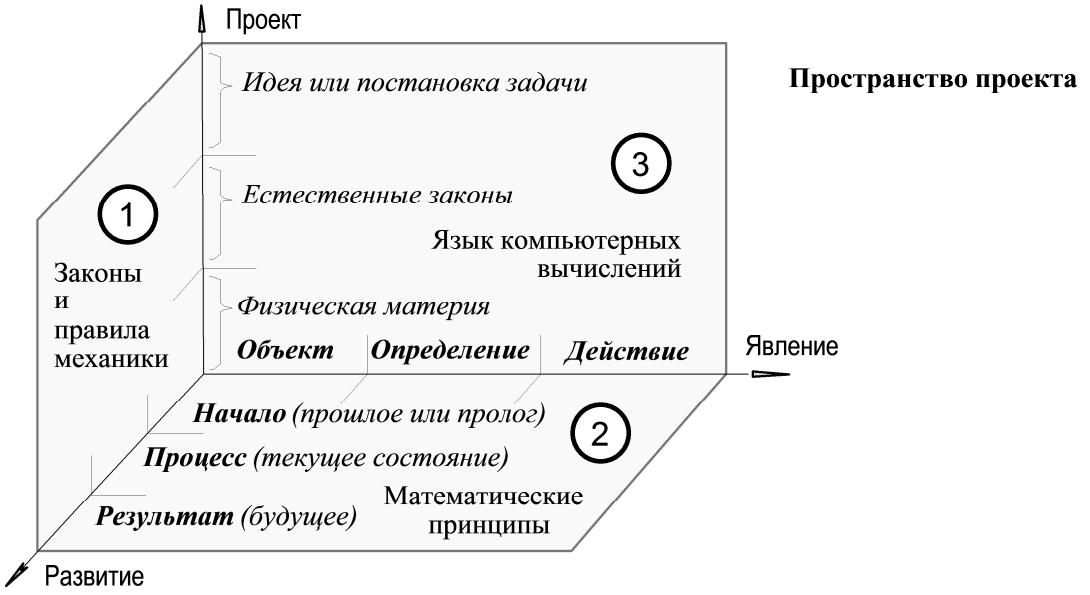


Рис. 1. Логическое пространство проектной задачи о построении вычислительного эксперимента

Предметные плоскости, по сути, являются особыми областями для научных изысканий, в которых формируются триады из формально независимых объектов программирования, физических законов и конкретных действий по реализации прямых вычислительных экспериментов.

Каждая комплексная задача, и в первую очередь те задачи, которые входят в список «Grand challenges», должна решаться в этих трех плоскостях. Рассмотрим один из наиважнейших аспектов в этом направлении — моделирование и расчет различных течений. В этой области находится большое количество задач: от гидрометеорологии до расчета горения в современных двигателях. Традиционно расчет течений связывают с решением уравнения Навье–Стокса. Проблемы, возникающие в этом направлении, мы можем, согласно высказанной концепции, разделить на три большие группы.

1. Некорректность уравнения Навье–Стокса, лежащего в основе моделирования, поскольку является некоторой идеализацией реальных процессов.

2. Идеализацией также является геометрия рассматриваемых течений. Это касается возникновения в расчетах острых кромок, углов и т. д., которые были обусловлены методами ап-

проксимации с помощью крупных сеток. Это вносит очень большие проблемы в решение задач, поскольку в реальности чаще всего эти особенности отсутствуют. В настоящий момент размер сеток может быть существенно уменьшен, что говорит о том, что и методы борьбы с возникающими проблемами должны быть другими.

3. Каждый раз при переносе задач течения на новые архитектуры возникает проблема портинга этих приложений. Поскольку довольно давно стало понятно, что появление новых архитектур связано с появлением новых библиотек программирования, то для течений также необходимо создание новых библиотек. Таким образом, мы должны по новому подходить к проблеме программирования задач гидроаэродинамики.

Рассмотрим подробнее эти три группы вопросов.

Математические принципы организации вычислений задач течения

Для того чтобы обратиться к первой группе вопросов, необходимо сначала выяснить, почему в ряде случаев уравнение Навье–Стокса — это хорошо, а в ряде случаев не считается.

Вопрос этот становится понятен на основе таких простых рассуждений. Представим наше уравнение в виде

$$u_t + uu_x - \mu u_{xx} = H\left(u,x\right), \tag{1}$$

где H является некоторым функционалом параметров потока. Блестящий анализ О. А. Ладыженской [Ладыженская, 1972] показывает, что основные проблемы этого уравнения связаны с нелинейным вторым членом в левой части. Если H может быть представлен в виде градиента некоторого функционала, проблема решается путем представления u в виде градиента некоторой функции V и интегрирования уравнения один раз. В действительности именно таким образом вводится представление Коула–Хопфа.

$$u = -2\mu\frac{1}{v}\frac{dv}{dx}, \tag{2}$$

которое линеаризует наше уравнение для случая, когда H является градиентом некоторого функционала.

Тогда эффективность численной реализации уравнения Навье–Стокса будет заключаться в удачной аппроксимации потенциальной части уравнения. В случае отсутствия подходящей аппроксимации потенциальной части использование этого уравнения просто нежелательно и необходимо переходить к другим методам решения, включая прямое имитационное моделирование явлений переноса.

Более сложные подходы связаны с использованием формального представления

$$K \equiv \frac{d}{dx}\left(\frac{d}{dx}\right)^{-1} K \tag{3}$$

и попыткой придания определенного смысла обратному оператору к d/dx. Такой подход весьма успешен в квантовой теории поля, но для его эффективности процедуре должен быть придан определенный физический смысл [Боголюбов, Ширков, 1984]. В нашем случае имеет смысл вернуться к выводу уравнений Навье–Стокса и пересмотреть соответствующие процедуры. Выясняется, что основные проблемы появляются уже в исходном уравнении Больцмана, которое столь же математически неудовлетворительно, как и уравнение Навье–Стокса. Для выхода из этой проблемы удобно рассмотреть обобщения уравнения Больцмана. Имея в виду намеченную выше процедуру, удобно исходить из нелокального варианта уравнения [Богданов, 1976]. Применяя к такому обобщению уравнения Больцмана стандартную процедуру, мы получим уравнение Навье–Стокса, но с нелокальными операторами переноса [Власов, 1978]. При раз-

ложении сингулярных ядер соответствующих операторов получаются поправки, которые и регуляризуют обратный оператор в нашей процедуре. Интересно, что при этом получается теория переноса, совершенно аналогичная теории линейной реакции на внешнее возмущение [Кадanов, Бейм, 1964]. Нам представляется, что именно такая теория и должна быть положена в основу построения эффективных алгоритмов расчета течений.

Геометрическая идеализация

Проблема геометрической идеализации течений в первую очередь связана с постановкой граничных и начальных условий задачи. Традиционный подход формализации границ течений проистекает из теоретической постановки проблемы и вводит в рассмотрение острые кромки, углы и иные элементы, легко учитываемые в аналитическом подходе к рассмотрению задачи, однако крайне негативно влияющие на численное решение. Сходимость численных методов в таких областях оказывается крайне низкой, тем более, что в реальности мы всегда имеем дело с гладкими поверхностями.

Необходимо отметить, что эксперименты показывают: в гидродинамике зависимость от определения точной границы достаточно слабая [Богданов и др., 1983]. Отсюда в классическом численном подходе для разрешения упомянутой проблемы зачастую заменяют реальную границу некоторой «эффективной» границей, модифицируют на ней граничные условия для удовлетворения законов сохранения и получают более эффективную вычислительную процедуру. Этот подход успешно реализуется при переходе к методу крупных частиц, а взаимодействие с поверхностью заменяется взаимодействием с единичной крупной частицей, которая может быть деформирована исходя из характера граничных условий. В том или ином подходе эффективным решение оказывается в случае возможности изолированного решения двух возникающих задач — ранее описанной проблемы математических принципов организации вычислений и геометрического представления течений. Поэтому принципиальным положением можно считать отделение динамики течений от геометрии. При этом, конечно, все численные решения должны быть строжайше связаны с конкретными геометрическими базисами, масштабами и физическими размерностями.

Самый эффективный способ решения подобных проблем достигается разделением и дальнейшим описанием каждой из них в сопряженных пространствах. На одном из них развивается динамика, а в другом описывается базовый объект. В этом случае задача становится самосогласованной.

Пример такого разделения представлен в [Дегтярев, Храмушин, 2014]. Здесь показаны техника разделения численного решения по независимым физическим процессам, и в то же время инструментальные средства тензорной математики для непосредственного рассмотрения физических процессов в конкретной ячейке-частице вычислительного эксперимента. Это позволяют обобщить задачу до уровня полного моделирования нестационарных процессов в аэрогидромеханике. В простейшем случае, для такого обобщения скалярная плотность крупных частиц жидкости заменяется тензорным вычислительным оператором, обеспечивающим линейную аппроксимацию параметров состояния жидкости внутри сеточной ячейки, с одновременной формализацией предыстории зависимого движения и деформации несвободных, но энергонезависимых частиц жидкости с помощью тензорной (*инерционной*) массы. Тензорные формализации в гидромеханике элементарных частиц сплошной среды привносят необходимые и достаточные инструменты для прямого моделирования конвективных составляющих течения в неподвижных сеточных ячейках, так же как и для прогноза зависимого смещения и деформации потока смежных частиц жидкости, в том числе вблизи граничных поверхностей или свободных разрывов в тяжелой жидкости.

Таким образом, доопределяются фундаментальные свойства физической теории поля в канонической записи исходных уравнений движения в ближайшем окружении смежных число-

вых ячеек-частиц, а тензорные формализации лишь подчеркивают достоинства явных численных схем для достижения сквозного контроля состояния крупных частиц жидкости в условиях всеобъемлющего распараллеливания вычислительных процессов.

Трехмерная тензорная математика не противоречит традиционным математическим моделям аэрогидромеханики и служит основой для объединенной формализации алгоритмических и функциональных подходов в программировании, включающих операции тензорной арифметики [Программа построения числовых объектов...] для моделирования реологических свойств и обобщенной механики частиц жидкости в больших массивах числовых структур, связываемых в сеточные аппроксимации с помощью координатных пространств тензорной геометрии[1].

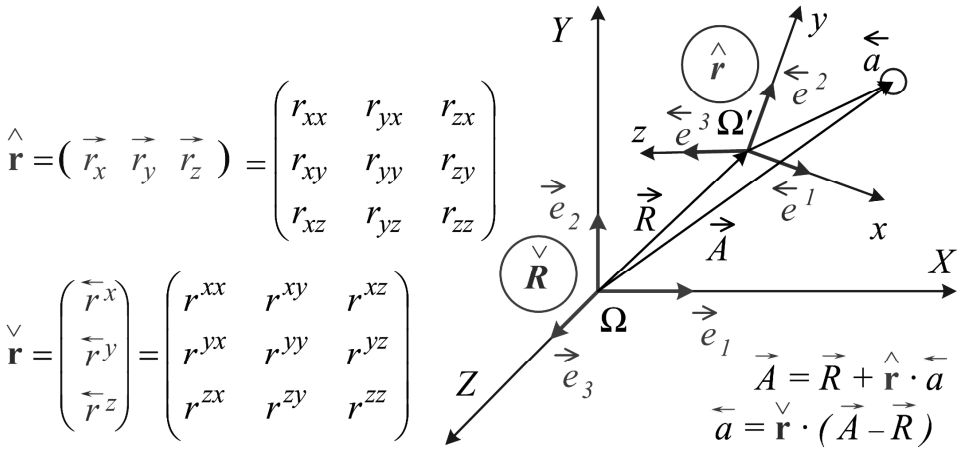


Рис. 2. Разметка ортов, векторное и матричное представление локального базиса крупной частицы жидкости. Ω отмечает центр глобальной системы отсчета, Ω' — пространственная привязка локального базиса. Прописные символы относятся к глобальным отсчетам в единой абсолютной системе координат; строчные — к аппроксимациям в связанных базисах крупных частиц жидкости, что в точности соответствует определениям конечных разностей или дифференциалов, исчисляемых по методу хорд Ньютона

Запись тензорных величин: $\overset{T}{\underset{\Omega}{\vec{\boldsymbol{R}}}}$ — координаты исходной узловой точки, Ω — индексы местоположения узла в сеточной области; верхний индекс: T — время от начала проведения вычислительного эксперимента. $\overset{t}{\underset{+}{\vec{\boldsymbol{R}}}}$ — смежный узел со сдвигом в сторону «+» от исходного центра масс и со смещением во времени на величину t; $\hat{r} = \vec{r}_k = r_{ik}$ [м^3] — тензор формы крупной частицы жидкости; $\overset{<}{M} = M^i{}_j = \check{\rho} \cdot \hat{r}$ [кг] — смешанный тензор, соотносящий внутреннее состояние частицы $\check{\rho}$ на абсолютную систему отсчета в локальном базисе $\hat{r}$.

В принятых обозначениях кинематика отдельно взятой частицы жидкости и связанных с ней внутренних потоков (живых сил) представляется простым многочленом с линейными тензорными элементами и вторым порядком по приращениям скалярного времени:

$$^{+}\vec{A} = \vec{R} + \vec{V} \cdot t + \vec{F} \cdot \overset{<}{m} \cdot t^2/2 + (\hat{r} + \hat{v} \cdot t + \hat{f} \cdot \overset{<}{m} \cdot t^2/2) \cdot \overleftarrow{a},$$

где — $\vec{F}$ [Н] — массовые; $\hat{f}$ [Н·м^2] — поверхностные силы.

[1] Операции тензорной геометрии исполняются на аппаратном уровне, что поддерживается в функциональной среде однородных координат в машинной графике OpenGL.

Программная реализация

Теперь, когда мы достигли разделения сопряженных пространств, и программирование необходимо проводить в этих же терминах. Иными словами, саму компьютерную архитектуру необходимо разбить таким же образом и выбрать для каждого из пространств оптимальное описание. Это оказывается выгодным как с точки зрения отображения задачи, так и с точки зрения оптимального выбора архитектуры.

Учитывая, что разделяемые сопряженные пространства в рассматриваемой задаче и описываемые в них процессы являются разными, для эффективного решения мы должны построить адекватную вычислительную инфраструктуру. На сегодняшний день такой структурой может быть только виртуальный гетерогенный вычислительный комплекс, построенный на базе традиционных многоядерных процессоров с многопоточными графическими ускорителями (CPU + GPGPU).

В этом случае действительно, когда описывается динамика течений, отделенная от геометрии, можно всегда уменьшить размерность и просчитать ее на узле мощных многоядерных процессоров. Но когда работа производится со сложной геометрией, то размерность пространства уменьшить не удается. Единственный выход повышения эффективности кода в этом случае — программирование при помощи векторной арифметики. Однако работа в гибридной архитектуре при попытке программировать в векторной парадигме на GPGPU не привела до сих пор к успеху, сравнимому с работой на векторных процессорах. Это связано в первую очередь с невозможностью передачи данных в векторном формате и планирование векторных операций на несколько шагов вперед, что является важной характеристикой любого векторного компьютера.

В современной вычислительной практике сложно представить себе возможность разработки специализированных алгоритмов гидромеханики для встраивания в микропроцессорные схемы новейших суперкомпьютеров. И в то же время ориентация на повсеместное согласование алгоритмов и собственно архитектурных особенностей вычислительных систем является необходимым условием реализации наиболее эффективных информационно-вычислительных систем. Показательным примером эффективного согласования аппаратных и программных возможностей является контекстная графическая среда OpenGL, в составе которой имеется полный набор операций для трехмерной визуализации с использованием математического аппарата «однородных координат» — своеобразной тензорной геометрии с векторами местоположения графических объектов внутри обрамленных матриц [4x4].

В трехмерной тензорной математике со всеми вычислительными объектами должны связываться логические предикаты для установки типа числовых структур и их принадлежности к абсолютному или локальному базису. Они используются для автоматических преобразований в соответствии с исходными установками в уравнениях аэрогидромеханики. Как минимум для такого предиката необходимо три двоичных бита:

“000” — T — *скалярная величина в абсолютном базисе или безразмерный инвариант;*

“001” — $\vec{A}$ — *вектор в глобальной системе координат (СК);*

“010” — $\overleftarrow{a}$ — *вектор в локальном базисе крупной частицы;*

“011” — t — *размерная скалярная величина в локальной системе отсчета;*

“100” — $\hat{r}$ — *тензор формы крупной частицы жидкости в глобальной СК;*

“101” — $v^{>}$ — *смешанный базис проекций векторов на локальную СК;*

“110” — $m^{<}$ – *базис локальных векторов в проекциях глобальной СК;*

“111” — $\rho^{\vee}$ – *определение тензорной величины в локальном базисе.*

Непрерывный контроль битовых масок-признаков особенно важен в случае распараллеливания вычислительных операций на сложных гибридных системах. В них моделирование раз-

деляется на большие пространственные блоки, между которыми должна осуществляться автоматическая конвертация числовых структур для согласования решений на смежных границах или в областях с перехлестом нерегулярных сеточных аппроксимаций, где возможны непредсказуемые перестроения структуры физических полей аэрогидромеханики.

Программная среда инженерного вычислительного эксперимента

Изложенный подход разделения и дальнейшего описания проблемы в сопряженных пространствах при организации вычислительного эксперимента требует адаптивного его управления со сквозным контролем текущего состояния всех числовых объектов. При этом устанавливаются функциональные или контекстные требования к языковой среде программирования для проектирования, построения и реализации прямых вычислительных экспериментов. Эти требования заключаются в следующем:

1) элементарные пространственно-временные объекты и базовые физические явления должны представляться в виде вычислительных структур и числовых величин в размерном виде, что в первую очередь требуется для визуального контроля и автоматического применения гибридных схем;

2) свойства вычислительных операций и элементарных числовых объектов инвариантно определяются в проекциях глобальной системы координат и однозначно соответствуют расчетным аппроксимациям в локальных базисах.

Инженерная реализация вычислительных экспериментов в аэрогидромеханике, безусловно, потребует адекватного математического сопровождения. В нем должны сочетаться алгоритмические и функциональные средства прикладного программирования с предельно эффективным отображением моделируемых физических явлений на архитектуру высокопроизводительных вычислительных систем. Трехмерная тензорная математика может стать элементом пространства (рис. 1) для унифицированного согласования базовых законов механики (1) и компьютерных языков программирования (3) с образованием необходимых и достаточных математических принципов (2) для изысканий в проектировании и реализации прикладных вычислительных экспериментов в гидромеханике.

В заключение можно сформулировать особенности специализированной среды программирования, в которой с числовыми объектами связываются алгоритмические операции.

1. Операции логические или эмпирические связаны с конструированием законов гидромеханики, которые задают способы формирования и методы анализа крупных частиц жидкости для выбора вычислительных моделей или трансформации числовых объектов.

2. Операции сложения применяются к любым числовым объектам после приведения к единому геометрическому базису и физическим размерностям в соответствии с предопределенными законами гидромеханики (пп. 1).

3. Операция «произведение» может исполняться только с сопряженными векторами и тензорами для интерполяционного проецирования физических полей в дуальных координатных базисах. Запрещается любое изменение ранга тензора в операциях произведения, если это не покрывается логическим синтезом (пп. 1) или анализом числовых объектов (пп. 2).

Алгоритмические последовательности будут управляться числовыми объектами:

1) скалярные или инвариантные величины, например время t, участвуют в операциях произведения или представляются производными для любых числовых объектов;

2) векторные величины участвуют в операциях сложения с сопоставимыми векторами и в операциях произведения с тензорами для изменения координатных базисов;

3) тензорные величины синтезируют или характеризуют свойства крупных частиц жидкости, геометрические деформации, физические явления и процессы гидромеханики в функции скалярного времени. С использованием тензорных объектов формулируются основные законы гидромеханики, а их конструирование и анализ образуют этапы численного моделирования.

Заключение

Вычислительная гидромеханика обладает наибольшим историческим авторитетом в построении сложных программно-технических средств для прикладных инженерных задач. Новейшие компьютерные технологии формально не связаны с решением прикладных задач гидромеханики и не противоречат развитию фундаментальных идей для совершенствования вычислительной математики и методов проектирования сложных вычислительных экспериментов. Настоящая работа посвящена синтезу функциональных и алгоритмических методов в прямых вычислительных экспериментах в аэрогидромеханике с явными численными схемами, органично объединяющихся в теоретических построениях тензорной математики, наилучшим образом отображающей законы гидромеханики и вычислительной геометрии на архитектуру современных гибридных информационно-вычислительных комплексов.

Список литературы

Богданов А. В. К выводу обобщенного кинетического уравнения больцмановскоо типа // Вестник ЛГУ, Сер. мат., мех., астр. — 1976. — № 13. — С. 66.

Богданов А. В., Горбачев Ю. Е., Дубровский Г. В., Фурсенко А. А., Жмакин А. И. и др. Теоретические модели релаксационной газодинамики и методы расчета неравновесных течений структурного газа. III. Численное исследование влияния релаксационных процессов на газодинамику течений. Изд. ЛИЯФ, препринт ФТИ, № 861, 36 с. 1983.

Боголюбов Н. Н., Ширков Д. В. Введение в теорию квантованных полей (изд. 4-е). — М.: Наука, 1984. — 600 с.

Власов А. А. Нелокальная статистическая механика. — М.: Наука, 1978. — 264 с.

Дегтярев А. Б., Храмушин В. Н. Проектирование и построение вычислительных экспериментов в гидромеханике с использованием явных численных схем и алгоритмов тензорной математики // Математическое моделирование. — 2014. — 26 (11). — С. 4–17.

Каданов Л., Бейм Г. Квантовая статистическая механика. Методы функций Грина в теории равновесных и неравновесных процессов. — М.: Мир, 1964. — 255 с.

Ладыженская О. А. О динамической системе, порождаемой уравнениями Навье–Стокса // Краевые задачи математической физики и смежные вопросы теории функций. 6, Зап. научн. сем. ЛОМИ, 27, Ленинград. отд. — Л.: Наука, 1972. — С. 91–115.

Программа построения числовых объектов и функций трехмерной тензорной математики для вычислительных экспериментов в гидромеханике (Tensor). — СПбГУ, Роспатент № 2013619727.

Federal Plan for High-end Computing: Report of the High-end Computing Revitalization Task Force (HECRTF), Executive Office of the President, Office of Science and Technology Policy, 2004

High Performance Computing Act of 1991.

Task Force on Grand Challenges. Final Report of the NSF Advisory Committee for Cyberinfrastructure [электронный ресурс]. — 2011. — URL: http://www.nsf.gov/od/oci/taskforces (дата обращения: 17.01.2015).

The Fourth Paradigm: Data-Intensive Scientific Discovery. Hey T., Tansley S., and Tolle K. (Eds.) Microsoft Research, Redmond, Washington, October 2009.

Wilson K. G. Grand challenges to computational science, in Modern physics in America: a Michelson-Morley centennial symposium. Cleveland, Ohio, 30–31 October 1987, AIP conference proceedings 169, ed. William Fickinger and Kenneth L. Kowalski.

Wilson K. G. Grand Challenges to Computational Science (Cornell Center for Theory and Simulation in Science and Engineering: 1987). — P. 2.

Wilson K. G. Grand challenges to computational science. Future Generation Computer Systems 5, i.2-3, 1989. Pp. 171–189. DOI: 10.1016/0167-739X(89)90038-1.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 439−444

Ки&М

ПЛЕНЕРНЫЕ ДОКЛАДЫ

УДК: 004.43, 004.94

GIS INTEGRO for petroleum and gas investigations

E. N. Cheremisina, A. E. Senner[a]

All-Russian Research Institute of geological, geophysical and geochemical systems,
Russia, 117105, Moscow, Varshavskoe Rd., 8
E-mail: [a]senner_a_e@mail.ru

Получено 20 февраля 2015 г.

GIS INTEGRO is the geo-information software system forming the basis for the integrated interpretation of geophysical data in researching a deep structure of Earth. GIS INTEGRO combines a variety of computational and analytical applications for the solution of geological and geophysical problems. It includes various interfaces that allow you to change the form of representation of data (raster, vector, regular and irregular network of observations), the conversion unit of map projections, application blocks, including block integrated data analysis and decision prognostic and diagnostic tasks.

The methodological approach is based on integration and integrated analysis of geophysical data on regional profiles, geophysical potential fields and additional geological information on the study area. Analytical support includes packages transformations, filtering, statistical processing, calculation, finding of lineaments, solving direct and inverse tasks, integration of geographic information.

Technology and software and analytical support was tested in solving problems tectonic zoning in scale 1:200000, 1:1000000 in Yakutia, Kazakhstan, Rostov region, studying the deep structure of regional profiles 1:S, 1-SC, 2-SAT, 3-SAT and 2-DV, oil and gas forecast in the regions of Eastern Siberia, Brazil.

The article describes two possible approaches of parallel calculations for data processing 2D or 3D nets in the field of geophysical research. As an example presented realization in the environment of GRID of the application software ZondGeoStat (statistical sensing), which create 3D net model on the basis of data 2d net. The experience has demonstrated the high efficiency of the use of environment of GRID during realization of calculations in field of geophysical researches.

Keywords: geo-information, integrated interpretation of geophysical data, GIS INTEGRO, parallel calculations, GRID

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 439–444 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 439–444

Ки&М

ГИС ИНТЕГРО при решении задач на нефть и газ

Е. Н. Черемисина, **А. Е. Сеннер**

Всероссийский научно-исследовательский институт геологических, геофизических и геохимических систем, Россия, 117105, г. Москва, Варшавское ш., д. 8

В основу системы интегрированной интерпретации геофизических данных при изучении глубинного строения Земли положена система ГИС ИНТЕГРО, являющаяся геоинформационной системой функционирования разнообразных вычислительных и аналитических приложений при решении различных геологических задач. ГИС ИНТЕГРО включает в себя многообразные интерфейсы, позволяющие изменять форму представления данных (растр, вектор, регулярная и нерегулярная сеть наблюдений), блок преобразования картографических проекций, а также прикладные блоки, включающие блок интегрированного анализа данных и решения прогнозно-диагностических задач.

Методический подход базируется на интеграции и комплексном анализе геофизических данных по региональным профилям, геофизических потенциальных полей и дополнительной геологической информации на изучаемую территорию.

Аналитическое обеспечение включает пакеты трансформаций, фильтрации, статистической обработки полей, расчета характеристик, выделения линеаментов, решения прямых и обратных задач, интегрирования геоинформации.

Технология и программно-аналитическое обеспечение апробировались при решении задач тектонического районирования в масштабах 1:200000, 1:1000000 в Якутии, Казахстане, Ростовской области, изучения глубинного строения по региональным профилям 1:ЕВ, 1-СБ, 2-СБ, 3-СБ и 2-ДВ, прогноза нефтегазоносности в районах Восточной Сибири, Бразилии.

Ключевые слова: геоинформация, интерпретация геофизических данных, ГИС ИНТЕГРО, параллельные вычисления, ГРИД

The building of the models of bowels of the earth is an actual problem, both with standpoint of the scientific studies, and with standpoint of practical activity mankind.

The scientific aspect of the decision this is attempts of understanding the processes of the formation and evolutions of our planet. Practical — in forecasting and searching of extractable natural resources for our vital activity. The rash technical progress during last two centuries has generated the huge amount absolutely new devices, technologies, materials. The amounts industrial production grows in groups of ten and a hundreds once. This has brought about need cutting the increase the mining as traditional resource (coal, oil, ore etc.) so and extractable natural resources practically not claimed earlier (the uranium, nickel, aluminum etc.). On expert estimations in surface layer of the earth (the depth not more 1 kilometer), available to mining useful resources by traditional technology, is extracted more than 70 percents useful extractable natural resources already.

Enumerated above and some other factors illustrate urgency of the building of the models of bedding of recourses for the reason forecasting, detailed elaboration and estimations of the volumes and extractable natural resources useful fossilized. GIS INTEGRO provides:

The building of the models is based on data processing the geophysical measurements, got by different methods of the geological exploring. Date in most cases present itself regular two-dimensional or three-dimensional network. A network node contains values of measured geophysical parameters (for instance, gravitation gradient, magnetic field value, etc.).

In our institute designed and developed software complex GIS INTEGRO, designed for processing of geophysical data. It is Russian innovation technology for solution of nature-use problems.

GIS INTEGRO provides:

Techniques and tools for integrated data analysis.

Regional forecast of oil and gas perspectives.

Environmental monitoring of oil-and- gas bearing regions.

Information and analytical support of nature-use management.

With standpoint of the processing for building of the models of earth bowels need mention the following typical features the data:

Large and very large volume. So, for instance, network of the size 100x100x20 contains $2*10^5$ nodes. Every node contains several values of the physical values. Image such small network contains the order of a million of values. In practice meets need of the processing the networks, containing too much values. The volume given herewith exceeds 2 Gbytes — greatly possible volume for keeping in operative memory the most wide-spread IBM PC computer architectures.

Possibility to ambiguous interpretation to observed anomaly. Revealed as a result of processing spottiness can be caused either as significant on its size anomaly, located from surface deeply, so and small anomaly, lying to surfaces close.

Enumerated above factors (either as variety of other) determine at building of the models of earth bowels use complex algorithms data processing, multipass processing the same material at variation of importance's parameter processing, use statistical and iterative methods of the processing. One pass calculation time maybe forms several hours, and in some case can reach the tens of hours.

The appearance of the environment GRID create the possibility of the considerable reduction execution time for some classes of geophysical tasks, were multisequencing of the process data processing possible.

This article discuss two the most perspective approaches of the multisequencing, not requiring significant modification existing software and giving significant advantage on time performing the calculation.

The first approach.

The typical particularity of the three-dimensional nets in the field of geophysics is vastly greater amount of the nodes on axes ***X*** and ***Y*** (the plane ***X0Y*** is parallel earth surface) in contrast with amount of the nodes on ***Z*** axis (***Z*** is directed vertically downwards) (Fig. 1.).The reason - is quick growth of relative error of the measurements with depth and brings about small validity of the deep measurements.

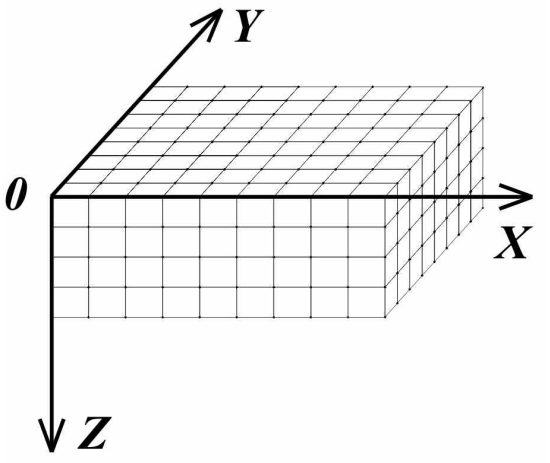


Fig. 1. Typical geophysics data network

Exists the extensive class of the tasks, which algorithms are based on calculation in each network node of the certain values as a surrounded nodes function, removed from base node not far then any value R. In this case possible following scheme of the multisequencing.

The source net divides on separate fragments by vertical planes paralleled ***ZOX*** or ***ZOY***. The processing algorithm placed in nodes GRID parametric tunes in on processing single specified fragment. Processing of each fragment is produced by separate node GRID, which is sent whole net. After termination of the processing all fragment in nodes GRID got results are integrated in resulting net and are delivered researcher (Fig. 2).

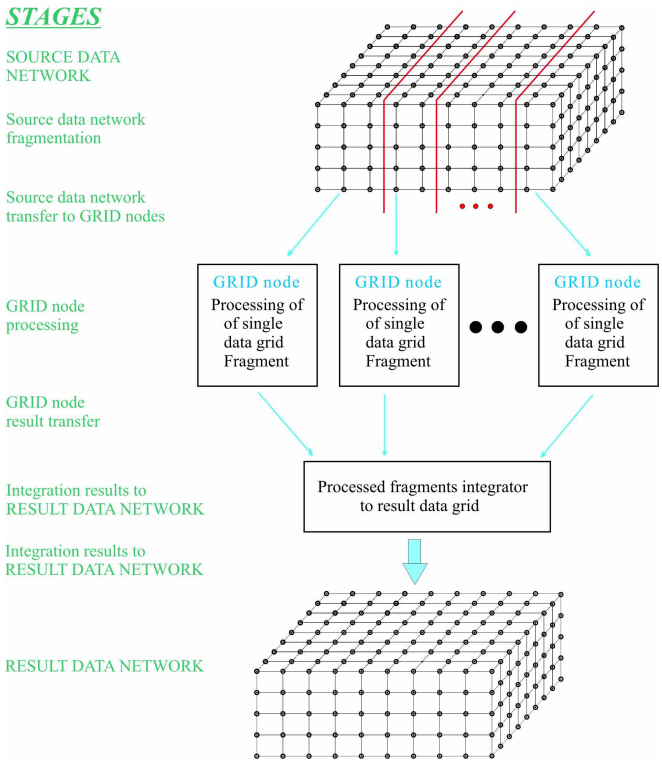


Fig. 2. Scheme of processing using GRID

The algorithms of integrations result usually it is enough simple and require not big running time. Therefore it possible considers that reduction of calculation time proportional to account of the amount fragment partition.

If source network great, that possible optimization of volume transferring data. To GRID node transfer only extended fragment, consisting directly of fragment plus nodes removed no more than on ***R*** from border nodes of the fragment of the net (Fig. 3). The amount real transferred data decrease in this case proportional to amount of source net fragments.

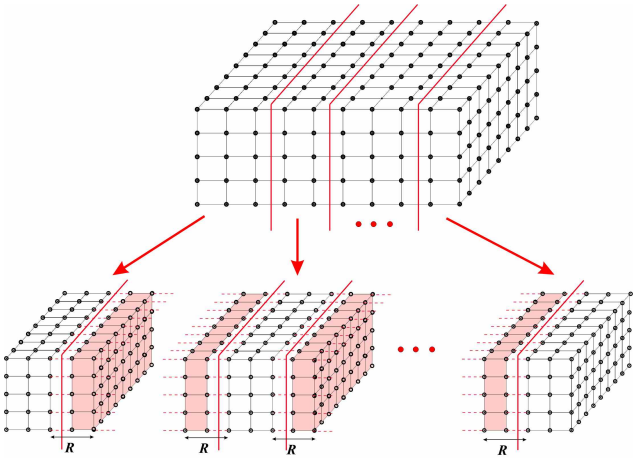


Fig. 3. Transfer optimization

The volume of the issues possible to optimize, sending to GRID node only extended fragment, consisting directly of fragment plus nodes removed no more than on ***R*** from border nodes of the fragment of the net.

The second approach.

Realization of software package ***ZondGeoStat*** (statistical sounding) for finding linear anomaly by vector flap in 2D task of the geophysical studies demonstrates second approach of the possibility of the use the multisequencing. Statistical sounding allows provide tracing the statistical parameters in the moving changing size window. It allows to go to ***3D*** object, carrying information on deep construction of the earth of the under investigation territory.

The algorithm of package is based on calculation of the necessary features inwardly certain square area of the fixed size, named by window. The centre of window complies with one of the nodes of the net. The results of the calculation single window fit are situated in central node window. The single pass of the calculation consists in processing all possible fixed size windows (Fig. 4), where ***Nx***, ***Ny*** — are number of nodes ***X*** and ***Y*** axis, ***R*** — size of current window:

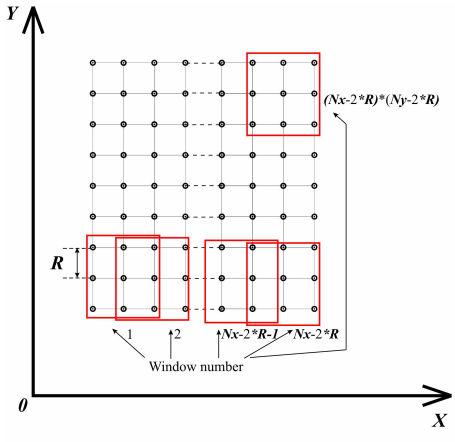


Fig. 4. Single pass. R=1

The Following single pass differs the size used window (Fig. 5).

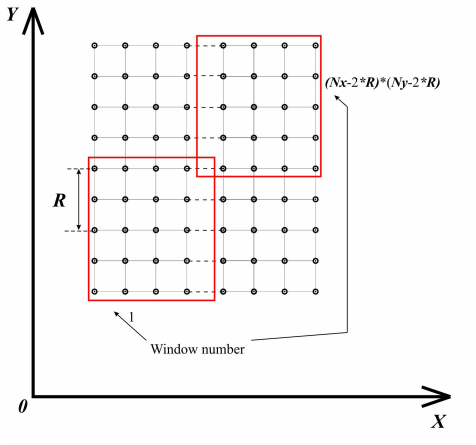


Fig. 5. Single pass. R=2

Accordingly on following stage of the calculation changes the size a window and procedure of the calculation is repeated.

The finally calculation is collection of the calculation of the set of the different size windows. The typical amount of the different sizes window under given study forms from 10 up 30.

Created in GRID environment realization of ***ZondGeoStat*** use individual GRID node for calculating of single pass, in other words for processing all source data grid by fixed size window. So — logical scheme of using GRID environment in this case very similar to scheme of Figure 1.

Conclusion.

The derived experience has demonstrated high efficiency of the of use GRID environment in the field of geophysical and geological investigations.

It seems perspective further adaptation other software package data processing in GRID environment.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 445–454

Ки&М

ПЛЕНАРНЫЕ ДОКЛАДЫ

УДК: 004.023

Deriving semantics from WS-BPEL specifications of parallel business processes on an example

V. Dimitrov

University of Sofia, Faculty of Mathematics and Informatics, Bulgaria, 1164 Sofia, 5 James Bourchier Blvd.

E-mail: cht@fmi.uni-sofia.bg

Получено 27 октября 2014 г.

WS-BPEL is a widely accepted standard for specification of business distributed and parallel processes. This standard is a mismatch of algebraic and Petri net paradigms. Following that, it is easy to specify WS-BPEL business process with unwanted features. That is why the verification of WS-BPEL business processes is very important. The intent of this paper is to show some possibilities for conversion of a WS-BPEL processes into more formal specifications that can be verified. CSP and Z-notation are used as formal models. Z-notation is useful for specification of abstract data types. Web services can be viewed as a kind of abstract data types.

Извлечение семантики из спецификаций WS-BPEL обработки параллельных процессов в бизнесе на примере

В. Димитров

Университет Софии, Факультет Математики и Информатики, Болгария, 1164, г. София, б-р Джеймс Баучера, д. 5

Аннотация. — WS-BPEL — это широко распространённый стандарт для спецификации распределенных и параллельных бизнес-процессов. Этот стандарт не подходит для алгебраических парадигм и парадигм направленных графов Петри. Исходя из этого, легко определить бизнес-процесс WS-BPEL с нежелательными особенностями. Именно поэтому проверка бизнес-процессов WS-BPEL очень важна. Цель этой статьи состоит в том, чтобы показать некоторые возможности для преобразования процессов WS-BPEL в более формальные спецификации, которые могут быть проверены. CSP и система обозначений Z используются как формальные модели. Система обозначений Z полезна для спецификации абстрактных типов данных. Web-сервисы могут рассматриваться как своего рода абстрактные типы данных.

Research in this paper are funded by Bulgarian Science Fund under contract ДФНИ-И01/12 “Modern programming languages, environments and technologies, and their application in education of software professionals”.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 445–454 (Russian).

Motivation

There are two kinds of business processes in WS-BPEL [OASIS…, 2007]: executable and abstract ones.

The behavioral semantics of executable business processes is well defined in WS-BPEL standard. There is only one problem with WS-BPEL extensions, because they go outside the notation framework; they are open and unpredictable, but without them the framework semantics is consistent.

The other category are the abstract business processes. Their intention is to describe Web services interactions without details on Web services internal implementations. The standard defines concretization procedure that can create an executable business process from an abstract one. But it is possible to generate with this procedure an executable business process with behavior different from that of the abstract one. It is possible, the executable business process to contain interactions that change the original ones, i.e. the executable process is not simply specialization of the abstract one. The standard requires for every abstract process to exist at least one executable business process, that is concretized by the procedure defined in the standard, and that is compatible with the abstract one. In such a way, the standard guarantees that above mentioned deviations are not available for at least one executable business process.

Abstract business process represents a class of executable business processes compatible with it. It is more productive to verify abstract business processes because:

1. Verification of an abstract business process means a verification of the whole class of executable business processes that it represents.
2. Abstract business process does not contain implementation details that have no impact on Web services interactions.

WS-BPEL business process is specified in two parts. First one is the WSDL [W3C…, 2001] specification of the Web services involved in the interaction. This specification includes the business process specification as a Web service. The second part is the WS-BPEL business process specification as Web services interactions. These both specifications are complementary because the business process, usually, is a Web service specified in WSDL. On the other hand, the business process as Web service is implemented in WS-BPEL. WSDL standard is extended to capture Web Services participating in the WS-BPEL business process and this is an essential part of WS-BPEL.

WSDL specifies only the interfaces and hides implementation details. WSDL specifications could be formalized. Why such a formalization is needed? WSDL is a XML based notation. Authors of XML argue that XML is readable for humans and computers. But XML specifications are verbose and not readable for humans. That is why, it is better, if Web services can be specified in some more compact notation, that is well better accepted by the humans. Such a tool is the Z notation [ISO/IEC 13568:2002]. Specifications in it tend to be very compact.

Z-notation is mainly used for specification of abstract data types, but it can be used for specification of algebras.

Web services can be viewed as abstract data types. In the example below, the WSDL and WS-BPEL specifications are a specification of abstract business process taken from the standard.

Formalization of the WSDL specification

First, messages exchanged among Web services are defined (shippingPT.wsdl):

```
<wsdl:definitions
  targetNamespace="http://example.com/shipping/interfaces/"
  xmlns:ship="http://example.com/shipping/ship.xsd"
  xmlns:tns="http://example.com/shipping/interfaces/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
 <wsdl:types>
  <xsd:schema>
   <!-- import ship schema -->
  </xsd:schema>
 </wsdl:types>
 <wsdl:message name="shippingRequestMsg">
  <wsdl:part name="shipOrder" type="ship:shipOrder" />
 </wsdl:message>
 <wsdl:message name="shippingNoticeMsg">
  <wsdl:part name="shipNotice" type="ship:shipNotice" />
 </wsdl:message>
 <wsdl:portType name="shippingServicePT">
  <wsdl:operation name="shippingRequest">
   <wsdl:input message="tns:shippingRequestMsg" />
  </wsdl:operation>
 </wsdl:portType>
 <wsdl:portType name="shippingServiceCustomerPT">
  <wsdl:operation name="shippingNotice">
   <wsdl:input message="tns:shippingNoticeMsg" />
  </wsdl:operation>
 </wsdl:portType>
</wsdl:definitions>
```

The application data schemas are imported in this part of the WSDL specification. Such a types here are shipOrder and shipNotice. They are the application data containers. Only the properties, defined on these messages, have impact on the message exchange among the business process Web services. These data types are modeled as basic types in Z notation:

$$[shipOrder, shipNotice]$$

Messages are modelled with Z schemas:

$$\begin{array}{|l} \textit{shippingRequestMsg} \\ \hline shipOrder : shipOrder \\ \hline \end{array}$$

$$\begin{array}{|l} \textit{shippingNoticeMsg} \\ \hline shipNotice : shipNotice \\ \hline \end{array}$$

Every message part in the Z-schemas is specified as a field with the same part name and the same type name.

Port types define Web services. They could be represented as abstract data types. In Z-notation, abstract data types are defined with schema type (Z schema) and operations (Z-schemas) applied on it. There is no way in the Z-notation, operations to be defined on the basic types. That is why initially, the basic type WebService is introduced and then it is used in the port types Z-schemas. The field ws is not very elegant approach for introducing the Web services, but it works.

All port types are only with one operation. They are represented with the corresponding Z-schema. Operations are one way. Each of them has only one input parameter.

$[WebService]$

$shippingServicePT \triangleq [ws: WebService]$

$shippingCustomerPT \triangleq [ws: WebService]$

shippingRequest

$\Delta shippingServicePT$

$input?: shippingRequestMsg$

shippingNotice

$\Delta shippingCustomerPT$

$input?: shippingNoticeMsg$

These specifications of the Web services do not contain any information about the Web services structure or behavior.

Properties definition in the WSDL specification is:

```
<wsdl:definitions
 targetNamespace="http://example.com/shipping/properties/"
 xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
 xmlns:vprop="http://docs.oasis-open.org/wsbpel/2.0/varprop"
 xmlns:ship="http://example.com/shipping/ship.xsd"
 xmlns:sif="http://example.com/shipping/interfaces/"
 xmlns:tns="http://example.com/shipping/properties/"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <wsdl:import location="shippingPT.wsdl"
  namespace="http://example.com/shipping/interfaces/" />
 <!-- types used in Abstract Processes are required to be finite
 domains. The itemCountType is restricted by range -->
 <wsdl:types>
  <xsd:schema
   targetNamespace="http://example.com/shipping/ship.xsd">
    <xsd:simpleType name="itemCountType">
     <xsd:restriction base="xsd:int">
      <xsd:minInclusive value="1" />
      <xsd:maxInclusive value="50" />
    </xsd:restriction>
   </xsd:simpleType>
  </xsd:schema>
 </wsdl:types>
 <vprop:property name="shipOrderID" type="xsd:int" />
 <vprop:property name="shipComplete" type="xsd:boolean" />
 <vprop:property name="itemsTotal" type="ship:itemCountType" />
 <vprop:property name="itemsCount" type="ship:itemCountType" />
 <vprop:propertyAlias propertyName="tns:shipOrderID"
  messageType="sif:shippingRequestMsg" part="shipOrder">
```

```
    <vprop:query>
      ship:ShipOrderRequestHeader/ship:shipOrderID
    </vprop:query>
  </vprop:propertyAlias>
  <vprop:propertyAlias propertyName="tns:shipOrderID"
    messageType="sif:shippingNoticeMsg" part="shipNotice">
    <vprop:query>ship:ShipNoticeHeader/ship:shipOrderID</vprop:query>
  </vprop:propertyAlias>
  <vprop:propertyAlias propertyName="tns:shipComplete"
    messageType="sif:shippingRequestMsg" part="shipOrder">
    <vprop:query>
      ship:ShipOrderRequestHeader/ship:shipComplete
    </vprop:query>
  </vprop:propertyAlias>
  <vprop:propertyAlias propertyName="tns:itemsTotal"
    messageType="sif:shippingRequestMsg" part="shipOrder">
    <vprop:query>
      ship:ShipOrderRequestHeader/ship:itemsTotal
    </vprop:query>
  </vprop:propertyAlias>
  <vprop:propertyAlias propertyName="tns:itemsCount"
    messageType="sif:shippingRequestMsg" part="shipOrder">
    <vprop:query>
      ship:ShipOrderRequestHeader/ship:itemsCount
    </vprop:query>
  </vprop:propertyAlias>
  <vprop:propertyAlias propertyName="tns:itemsCount"
    messageType="sif:shippingNoticeMsg" part="shipNotice">
    <vprop:query>ship:ShipNoticeHeader/ship:itemsCount</vprop:query>
  </vprop:propertyAlias>
</wsdl:definitions>
```

A new type for the properties is introduced and it Z-schema is:

$$itemCountType == 1..50$$

The properties are then represented as types:

$$shipOrderID == \mathbb{N}_1$$
$$shipComplete ::= False \mid True$$
$$itemsTotal == itemCountType$$
$$itemsCount == itemCountType$$

There are two deviations in the Z-notation schemas from the WSDL specification. The order numbers are positive numbers – not simply integers as is defined in the WSDL specification. In Z-notation, there is no Boolean type and it is modelled with two values False and True.

The aliases are properties placed on the messages. Here, they are modelled as functions from thr message type to the property type. There is no need to model XPath queries, because they are extensions to WS-BPEL. Queries written in other languages can be modelled in the same way. The aliases specification is more abstract:

$shipOrderID\_shipOrder: shippingRequestMsg \rightarrow shipOrderID$
$shipOrderID\_shippingNotice: shippingNoticeMsg \rightarrow shipOrderID$
$shipComplete\_shipOrder: shippingNoticeMsg \rightarrow shipComplete$
$itemsTotal\_shipOrder: shippingRequestMsg \rightarrow itemsTotal$
$itemsCount\_shipOrder: shippingRequestMsg \rightarrow itemsCount$
$itemsCount\_shippingNotice: shippingNoticeMsg \rightarrow itemsCount$

One property could have many aliases with the same name. In the Z-notation, above defined functions are global ones and their names must be unique. So, alias name is formed by the property name and the part name, in which it is defined. It is mapping from message type to property type.

Formalization of partner link type has no sensible interpretation here and they are modeled in the context of the WS-BPEL business process.

Finally, as result of the modelling effort, the specification is very compact and very simple. It does not include the business process as a Web service. This specification could be used for Web Services development, but it is very simple without invariants.

Formalization of the WS-BPEL specification

The specification of the example abstract business process in WS-BPEL is:

```
<process name="shippingService"
 targetNamespace="http://example.com/shipping/"
 xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/abstract"
 xmlns:plt="http://example.com/shipping/partnerLinkTypes/"
 xmlns:props="http://example.com/shipping/properties/"
 xmlns:ship="http://example.com/shipping/ship.xsd"
 xmlns:sif="http://example.com/shipping/interfaces/"
 abstractProcessProfile=
  "http://docs.oasis-open.org/wsbpel/2.0/process/abstract/ap11/2006/08">
 <import importType="http://schemas.xmlsoap.org/wsdl/"
  location="shippingLT.wsdl"
  namespace="http://example.com/shipping/partnerLinkTypes/" />
 <import importType="http://schemas.xmlsoap.org/wsdl/"
  location="shippingPT.wsdl"
  namespace="http://example.com/shipping/interfaces/" />
 <import importType="http://schemas.xmlsoap.org/wsdl/"
  location="shippingProperties.wsdl"
  namespace="http://example.com/shipping/properties/" />
 <partnerLinks>
  <partnerLink name="customer" partnerLinkType="plt:shippingLT"
   partnerRole="shippingServiceCustomer"
   myRole="shippingService" />
 </partnerLinks>
 <variables>
  <variable name="shipRequest" messageType="sif:shippingRequestMsg" />
  <variable name="shipNotice" messageType="sif:shippingNoticeMsg" />
  <variable name="itemsShipped" type="ship:itemCountType" />
 </variables>
```

```
<correlationSets>
  <correlationSet name="shipOrder" properties="props:shipOrderID" />
</correlationSets>
  <sequence>
    <receive partnerLink="customer" operation="shippingRequest" variable="shipRequest">
      <correlations>
        <correlation set="shipOrder" initiate="yes" />
      </correlations>
    </receive>
    <if>
      <condition>
        bpel:getVariableProperty('shipRequest', 'props:shipComplete')
      </condition>
      <sequence>
        <assign>
          <copy>
            <from variable="shipRequest" property="props:shipOrderID" />
            <to variable="shipNotice" property="props:shipOrderID" />
          </copy>
          <copy>
            <from variable="shipRequest" property="props:itemsCount" />
            <to variable="shipNotice" property="props:itemsCount" />
          </copy>
        </assign>
        <invoke partnerLink="customer" operation="shippingNotice" inputVariable="shipNotice">
          <correlations>
            <correlation set="shipOrder" pattern="request" />
          </correlations>
        </invoke>
      </sequence>
    <else>
      <sequence>
        <assign>
          <copy>
            <from>0</from>
            <to>$itemsShipped</to>
          </copy>
        </assign>
        <while>
          <condition>
            $itemsShipped &lt; bpel:getVariableProperty('shipRequest', 'props:itemsTotal')
          </condition>
          <sequence>
            <assign>
              <copy>
                <opaqueFrom/>
                <to variable="shipNotice" property="props:shipOrderID" />
              </copy>
              <copy>
                <opaqueFrom/>
                <to variable="shipNotice" property="props:itemsCount" />
```

```
          </copy>
        </assign>
        <invoke partnerLink="customer" operation="shippingNotice" inputVariable="shipNotice">
          <correlations>
            <correlation set="shipOrder" pattern="request" />
          </correlations>
        </invoke>
          <assign>
            <copy>
              <from>
                $itemsShipped + bpel:getVariableProperty('shipNotice', 'props:itemsCount')
              </from>
              <to>$itemsShipped</to>
            </copy>
          </assign>
        </sequence>
      </while>
    </sequence>
  </else>
 </if>
</sequence>
</process>
```

At the beginning, the partner link, in which the business process participates, is defined. The role of the process in this link is fixed. The partner link is defined with the partner link type taken from the WSDL specification (shippingLT.wsdl):

```
<wsdl:definitions
 targetNamespace="http://example.com/shipping/partnerLinkTypes/"
 xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
 xmlns:sif="http://example.com/shipping/interfaces/"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
 <wsdl:import location="shippingPT.wsdl"
  namespace="http://example.com/shipping/interfaces/" />
 <plnk:partnerLinkType name="shippingLT">
  <plnk:role name="shippingService" portType="sif:shippingServicePT" />
  <plnk:role name="shippingServiceCustomer" portType="sif:shippingServiceCustomerPT" />
 </plnk:partnerLinkType>
</wsdl:definitions>
```

The partner link type shippingLT connects a service (shippingService) with its consumer (shippingServiceConsumer). In the WS-BPEL specification, the business process role is a fixed service provider. The roles of port types are defined in shippingLT.wsdl and are represented as operations in shippingPT.wsdl.

The business process logic written is pseudo code is:

```
receive shipOrder
if condition shipComplete
 send shipNotice
else
 itemsShipped := 0
 while itemsShipped < itemsTotal
```

```
itemsCount := opaque          // non-deterministic assignment corresponding e.g. to
                              // internal interaction with back-end system
send shipNotice
itemsShipped = itemsShipped + itemsCount
```

The process is instantiated when a shipOrder is received. If the received order has been executed then a shipNotice is replied. This situation is checked in the message header property shipComplete. Otherwise, a cycle is started for the order execution. At every step, part of the items are delivered and a notification is send. The counter is incremented with the number of the sent items. The cycle exits when all items are delivered and only then the process is terminated. In the abstract process, the number of delivered items at every step is non-deterministic. This information is retrieved from the backend system that actually register how many items have been send. From interactions point of view, this process is very simple.

Initially, the process waits to receive an order message from a consumer and then replies with one or more messages. There are no error handlers, no compensators, no return values. There is no need for correlation sets coordination: when a new instance of the process is created, the process can be restarted in parallel to wait for new order, and the current instance is executing the received yet order.

In CSP, the process is very simple as is shown below:

```
channel customer 0;

shippingService() = customer?shipOrder -> (checkOrder(shipOrder) |||
        shippingService());
checkOrder(shipOrder) = (shipComplete -> customer!shipOrder -> Skip) []
        (shipNotComplete -> executeOrder(shipOrder));
executeOrder(shipOrder) =
        (itemsShipped -> Skip) []
        (itemsNotShipped -> change_itemsCount -> customer!shipOrder ->
        executeOrder(shipOrder));

var count = 10;

shippingServiceCustomer() =
        if (count > 0) {customer!count -> {count--} -> receive()} else {Skip};
receive() = customer?shipNotice -> receive();

System() = shippingServiceCustomer() ||| shippingService();
#assert System() deadlockfree;
```

PAT, product for specification and verification of CSP models, is used here. In this specification, there is only one channel between service provider and service consumer. This channel is modelling the partner link from the WS-BPEL specification. The channel could have some capacity, but here only a message can be exchanged through it, like in the classic CSP.

The WS-BPEL process is modelled as the CSP process shippingService. This process, initially, is waiting to receive a shipping order through the channel. When the process receives an order, it starts its execution, but in parallel restarts a new copy to wait for a new order.

The subprocess checkOrder checks the order. There are two possible events from the check: the order is executed yet or not. With these two events is modelled the check in the WS-BPEL process. The order is received as a parameter by the subprocess checkOrder.

If the order has been executed yet, the process sends through the channel a notification, which may be is the order message in some format. Otherwise, it starts the subprocess executeOrder. In this

process, all manipulations with variables, messages and properties are abstracted to result events. If all order items have been sent then the process terminates. Otherwise, the backend system is initiated. The last one sends information when some delivery is done. This is marked by an occurrence of change_itemsCount. Through the channel, the consumer is informed about that delivery. Then follows a recursive execution of the subprocess with the left items.

In this specification, instead of shippingNotice is returned shippingOrder. The idea is that a document-message, like shippingOrder, carries the business process state and no more other messages are needed. It possible, the CSP process to use different message in that case, but this would not change the interaction flow.

In the CSP specification, there are consumer and system subprocesses. They are added for verification purposes of the whole system.

The abstraction of data manipulations into events is the main approach in this conversion from WS-BPEL to CSP. The representation of a cycle as a recursive subprocess (subprogram) is the other used approach. The conditional statements are modelled as choices among events. The correlation sets are ignored in the model, because they are used only in the consumer part. The process simply returns through the channel data (the order) that contain the dialog identifier.

Conclusion

Attractive results in WSDL formalization with Z notation have not been achieved, because there are no behavior for the modelled Web Services. The business process WSDL specification is simply an interface to several Web services.

On the other hand, the business process WS-BPEL specification specifies a behavior. Its formalization in CSP is maximally abstracted from implementation details saving the original interaction flow. The CSP specification can then be formally verified. The CSP model is very compact and readable.

This example of formalization demonstrates an approach to formal verification of business processes.

References

ISO/IEC 13568:2002, Information technology — Z formal specification notation — Syntax, type system and semantics,

http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=21573

OASIS, Web Services Business Process Execution Language Version 2.0, OASIS Standard, 11 April 2007, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html

W3C, Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, http://www.w3.org/TR/wsdl

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 455−462**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

JINR TIER-1-level computing system for the CMS experiment at LHC: status and perspectives [1]

N. S. Astakhov, A. S. Baginyan, S. D. Belov, A. G. Dolbilov, A. O. Golunov, I. N. Gorbunov, N. I. Gromova, I. A. Kashunin, V. V. Korenkov [a], V. V. Mitsyn, S. V. Shmatov, T. A. Strizh, E. A. Tikhonenko, V. V. Trofimov, N. N. Voitishin, V. E. Zhiltsov

Joint institute for nuclear researches, Laboratory of Information Technologies,
Joliot-Curie, 6, Moscow reg., Dubna, 141980, Russia

E-mail: [a] korenkov@jinr.ru

Получено 28 октября 2014 г.

The status and perspectives of the Tier1 center for the CMS (Compact Muon Solenoid) experiment at Joint Institute for Nuclear Research (JINR) are presented.

Keywords: grid computing, CMS experiment, CMS Tiers

Статус и перспективы вычислительного центра оияи 1-го уровня (TIER-1) для эксперимента CMS на большом адронном коллайдере

Н. С. Астахов, А. С. Багинян, С. Д. Белов, А. Г. Долбилов, А. О. Голунов, И. Н. Горбунов, Н. И. Громова, И. А. Кашунин, В. В. Кореньков, В. В. Мицын, С. В. Шматов, Т. А. Стриж, Е. А. Тихоненко, В. В. Трофимов, Н. Н. Войтишин, В. Е. Жильцов

*Лаборатория информационных технологий, Объединенный институт ядерных исследований
Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6*

Представлены статус и перспективы развития вычислительного центра 1-го уровня (Tier1), создаваемого в ОИЯИ, для эксперимента CMS (Компактный мюонный соленоид).

Ключевые слова: грид компьютинг, эксперимент CMS, центры CMS (Tiers)

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 455–462 (Russian).

[1] The work is carried out within the federal program “Research and development on the priority directions of the development of the scientific — technological complex in Russia for 2007-2013” (contract №07.524.12.4008).

Introduction

The 6 million billion proton-proton collisions were produced by the Large Hadron Collider (LHC) [The Large Hadron Collider] at CERN in its first physics Run (2010-2012). Around 5 billion of these collisions were recorded in real time by the ATLAS and CMS experiments each for further processing, reconstruction of physics objects and physics analysis. Including simulation events, all in all, the LHC experiments have generated during the LHC Run 1 about 200 PB of data.

Data storage, processing and analysis of such a huge amount of data have been completed in the framework of the distributed computing infrastructure within the Worldwide LHC Computing Grid (WLCG) Project [LHC Computing…, 2005]. The WLCG computing model joints the three-level recourse centers (tiers) and originally assumed hierarchical structure according to their functionality. Then data distribution and replication was optimized by allowing transfers between any two centres. Now WLCG is formed by more than 170 centres spread around the world among them the Tier-0 center in CERN and thirteen Tier-1 centers. 2 million jobs run every day in this global infrastructure [The Worldwide LHC Computing Grid].

CMS Tier1 center at JINR (Dubna)

For CMS Tier1 centers are in Germany, United Kingdom, USA (FNAL), Italy, France, Spain, Taipei and JINR (Dubna). Starting 2011 the WLCG Tier-1 site is under development in the Russian Federation for all four LHC Experiments [CMS Dashboard; Korenkov 2013]. The special Federal Target Programme Project is aimed to construction of a Tier-1 computer-based system in National Research Center "Kurchatov Institute" and JINR for processing experimental data received from LHC and provision of grid services for a subsequent analysis of the data at the distributed centers of the LHC computing grid. It is shared that the National Research Center "Kurchatov Institute" is responsible for support of ALICE, ATLAS, and LHCb experiments, while the JINR provides Tier-1 services for the CMS experiment. In 2012 the WLCG Overview Board approved the plan of creating a Tier1-level center in Russia.

In agreement with the CMS Computing model [Grandi, Stickland, Taylor, 2005], the JINR Tier-1 site will provide acceptance of an agreed share of raw data and Monte Carlo data and provision of access to the stored data by other CMS Tier-2/Tier-3 sites of the WLCG, will serve FTS-channels for Russian and Dubna Member States Tier-2 storage elements including monitoring of data transfers.

The Tier1 CMS infrastructure at JINR consists of the following elements (services) (Figure 1): Data storage subsystem, Computing system (Computing Elements), Data transfer subsystem (FTS), Management of data transfer and data storage (CMS VOBOX), Load distribution system, and CMS Tier-1 network infrastructure.

Since October 2013 the JINR Tier-1 supports CMS as the tape-less Tier-1 with 1200 cores (17K HS06) and 450 TB disk-only dCache storage system. The prototype of mass-storage system constitutes 130 TB dCache pools and 72 TB tapes. All required grid services were installed and successfully validated and tested for high memory (6GB) jobs, in particular, File Transfer Service FTS 2.2.8, CMS Data Transfers service PhEDEx 4.1.2 for disk-only dCache and MSS, authorization and authentication Argus service, 2x Frontier Squids (access to the calibration data via the local cache), site BDII (Berkeley DB Information Index) and top-level BDII, User Interface service UI, Credential Management Service MyProxy, Workload Management System WMS, Logging and Bookkeeping service LB, LCG File Catalog - LFC for internal testing.

Since the LHC Run-2 start-up in line with the WLCG and LHC Experiments requirements, the JINR has to provide a support of a number of the main Tier-1 services for the CMS experiment: user-visible services (Data Archiving Service, Disk Storage Services, Data Access Services, Reconstruction Services, Analysis Services, User Services) and specialized system-level services (Mass storage system, Site security, Prioritization and accounting, Database Services).

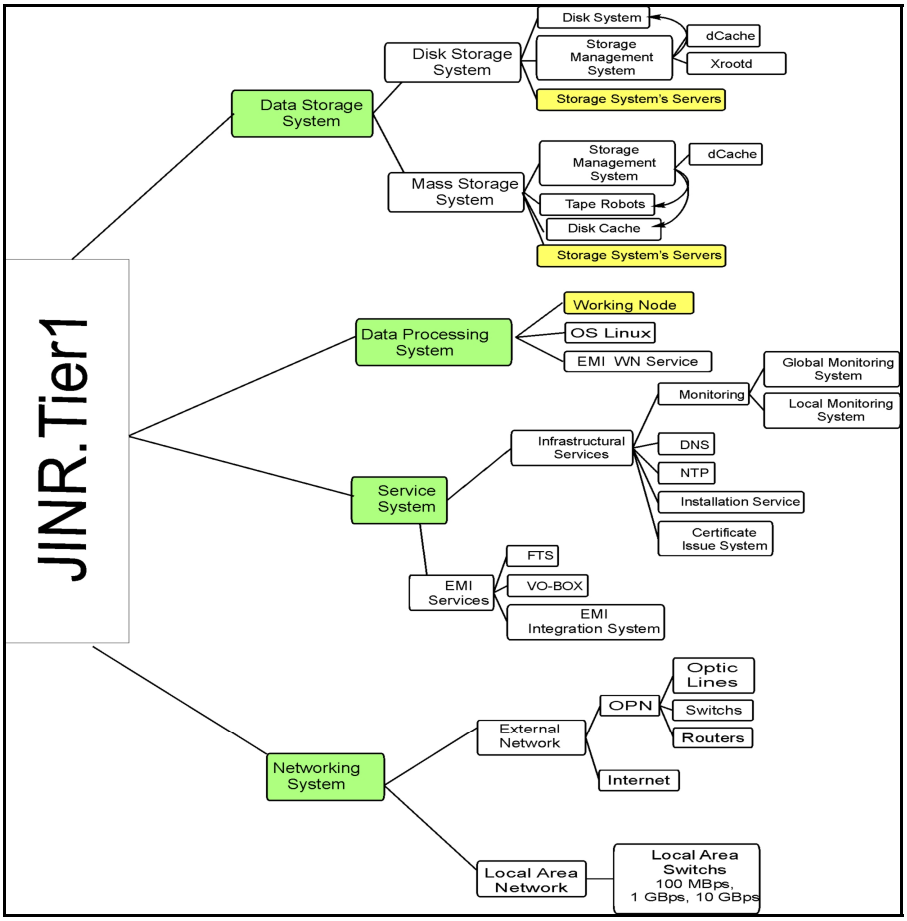


Fig. 1. JINR Tier1 infrastructure scheme

The network bandwidth as part of LHCOPN for Tier-0-Tier-1 and Tier-1-Tier-1 connections was about 2 Gbps for 2012 and now is 10 Gbps. The JINR link to public network with a bandwidth of 20 Gbps is used to connect the Tier-1 with all other Tier-2/Tier-3 sites.

In 2015 the CMS Tier-1 site in JINR will provide computing facilities about 10% of the total existing CMS Tier-1 resources (excluding CERN).

Structure of JINR CMS Tier1 local network

Figure 2 presents the Tier1 network topology at JINR. To realize it, we have to commutate 160 disk servers, 25 blade servers and 60 infrastructure servers.

For the networks with a star topology, each network host is connected to a central node with a point-to-point connection. All traffic passes through the central node. An advantage of the star topology is simplicity of supplementing additional nodes, while its primary disadvantage is that the hub represents a single point of failure. The type of the network topology in which some of the nodes of the network are connected to more than one other node in the network makes it possible to take advantage of some of the redundancy that is provided by a physical connected mesh topology. A fully connected network is a communication network in which each of the nodes is connected with one another. In graph theory it is known as a complete graph [Education-portal].

Network designers implement mesh topology and Spanning Tree Protocol (STP) on switches in order to prevent loops in the network, i.e. to use STP in situations where you want redundant links, but not loops. A failure of primary links activates the backup links so that users can continue using the network. Without STP on switches, such a failure can result in a loop. STP defines a tree that spans all

the switches in an extended network. STP forces certain redundant data paths into a blocked state and leaves other paths in a forwarding state. If a link in the forwarding state becomes unavailable, STP reroutes data paths through the activation of the standby path.

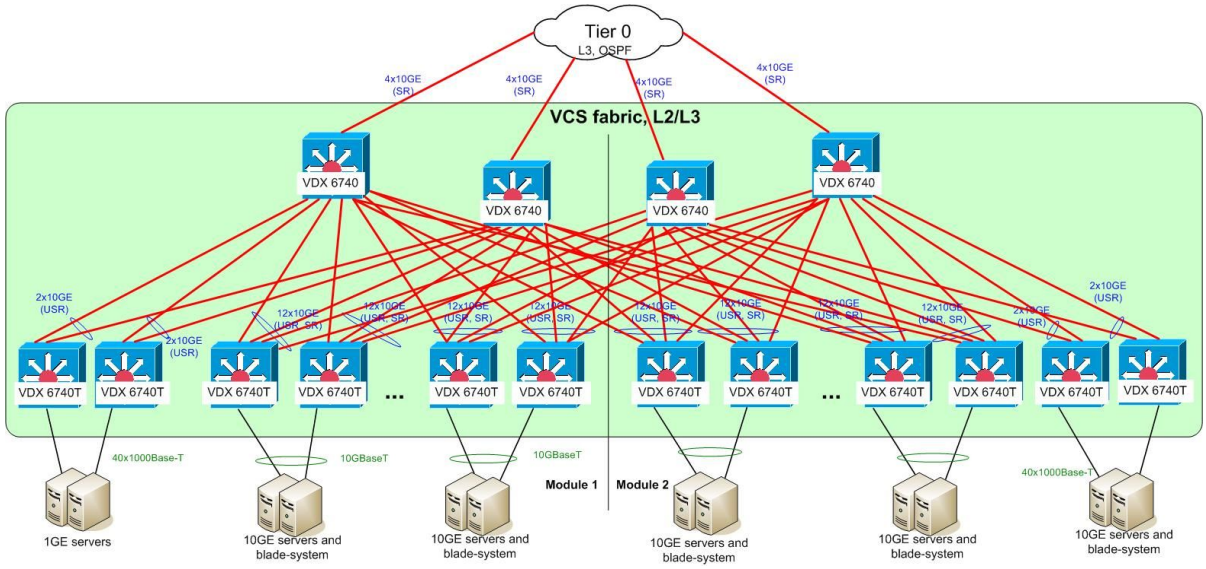


Fig. 2. Network topology of JINR CMS Tier 1 center

Recently the IS-IS protocol for network calculating has been applied. For this protocol Dijkstra's algorithm from graph theory is used. It compares and calculates the shortest path through all nodes in the network. It is constructing a shortest-path tree from the first vertex to every other vertex in the graph. On its basis a modern protocol Transparent Interconnection of Lots of Links (TRILL) was developed [Transparent Interconnection of…].

A new Layer 2 routing protocol, Transparent Interconnection of Lots of Links (TRILL), offers many advantages over STP. While STP maintains a single path and blocks all redundant paths, TRILL provides Layer 2 multipath and multi-hop routing. The proposed TRILL protocol enhances Layer 2 routing by introducing multipath and multihop routing capabilities. The new capabilities represent significant improvements over Spanning Tree Protocol (STP). TRILL provides Layer 2 multiple paths by splitting the traffic load among several paths. TRILL also is faster at self-healing during a network failure. While one link gets unavailable others continue transfer traffic. The maximum reduction in a network bandwidth does not exceed 50 % and taking into account our design in the network Tier 1 at JINR it is not more than 25 %. Accordingly, all the nodes continue to operation, only their bandwidth decreases.

In conclusion, it is worth noting, that the topology Tier 1 at JINR will be based on protocol TRILL. This protocol will help network designers to create a coherent Virtual Cluster Switching (VCS) fabric with distributed switches and allow creating highly reliable, mobile and multi-port systems.

Monitoring and statistics

The various metrics (example for JINR Tier-1 is given in Fig. 3) are based on the result of common WLCG tests and CMS specific tests, in particular NAGIOUS tests are applied (Fig. 4). These tests are used to establish site availability and readiness for the CMS collaboration usage. Test results are summarized in the status summary table at Site Status Board (SSB) monitoring at the CMS Dashboard [CMS Dashboard, URL].

The JINR CMS Tier1 site shows good results in the monitoring rank of CMS Tier1 sites on availability and readiness (see Fig.5).

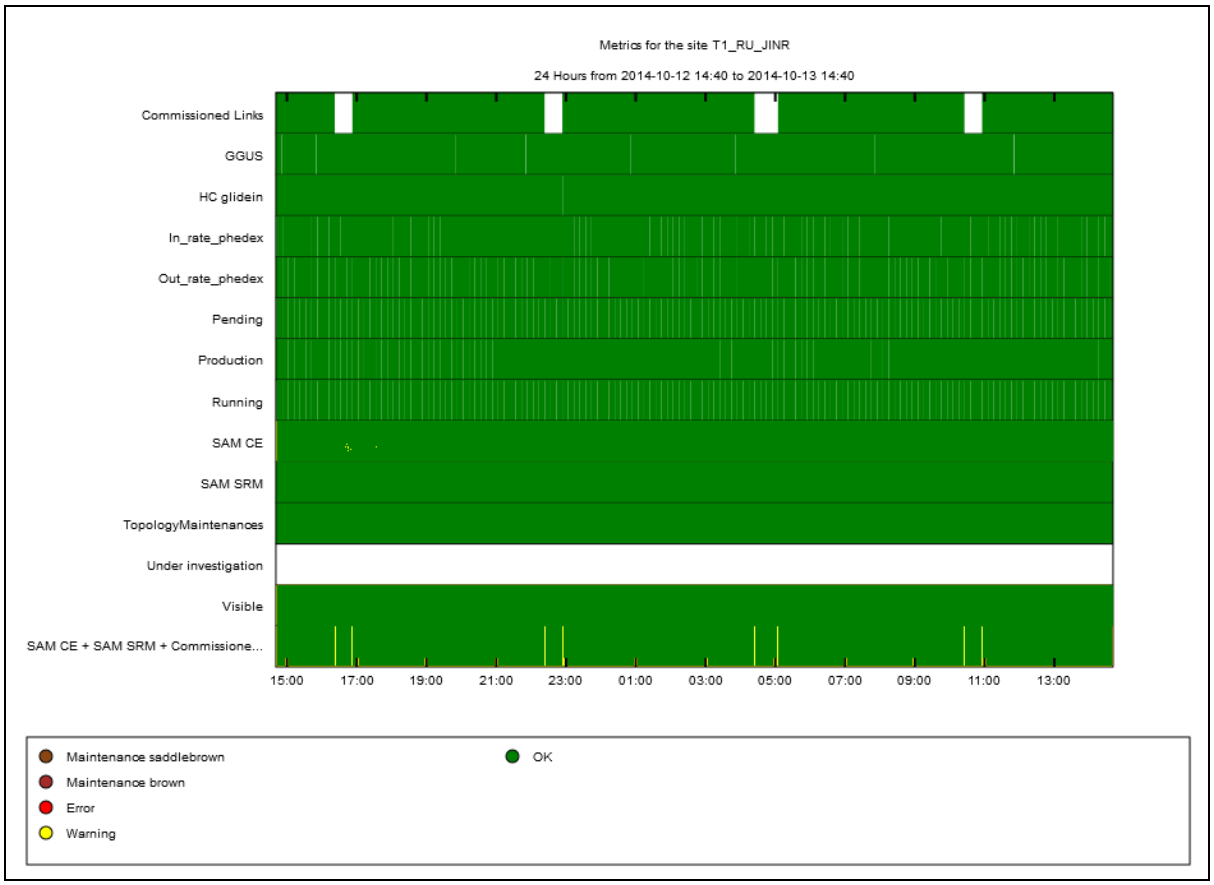


Fig. 3. Metrics for the JINR Tier-1 site

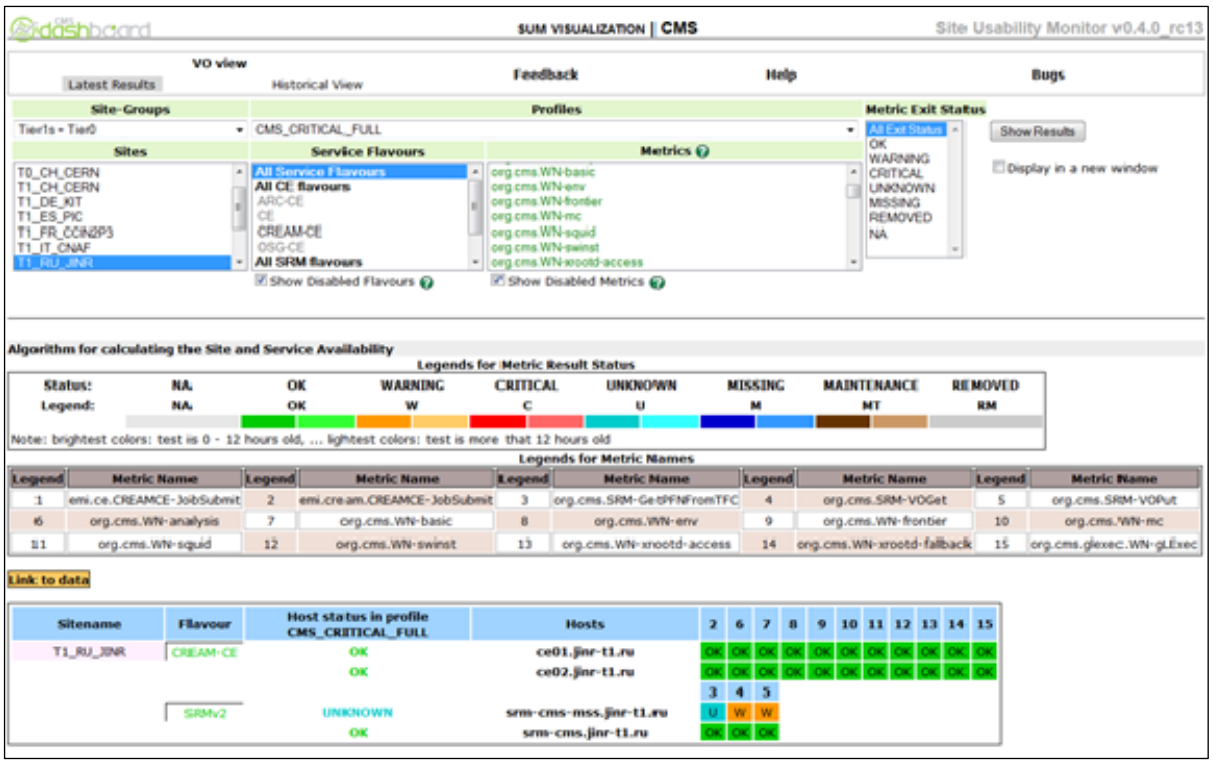


Fig. 4. The JINR Site usability based on the NAGIOS test

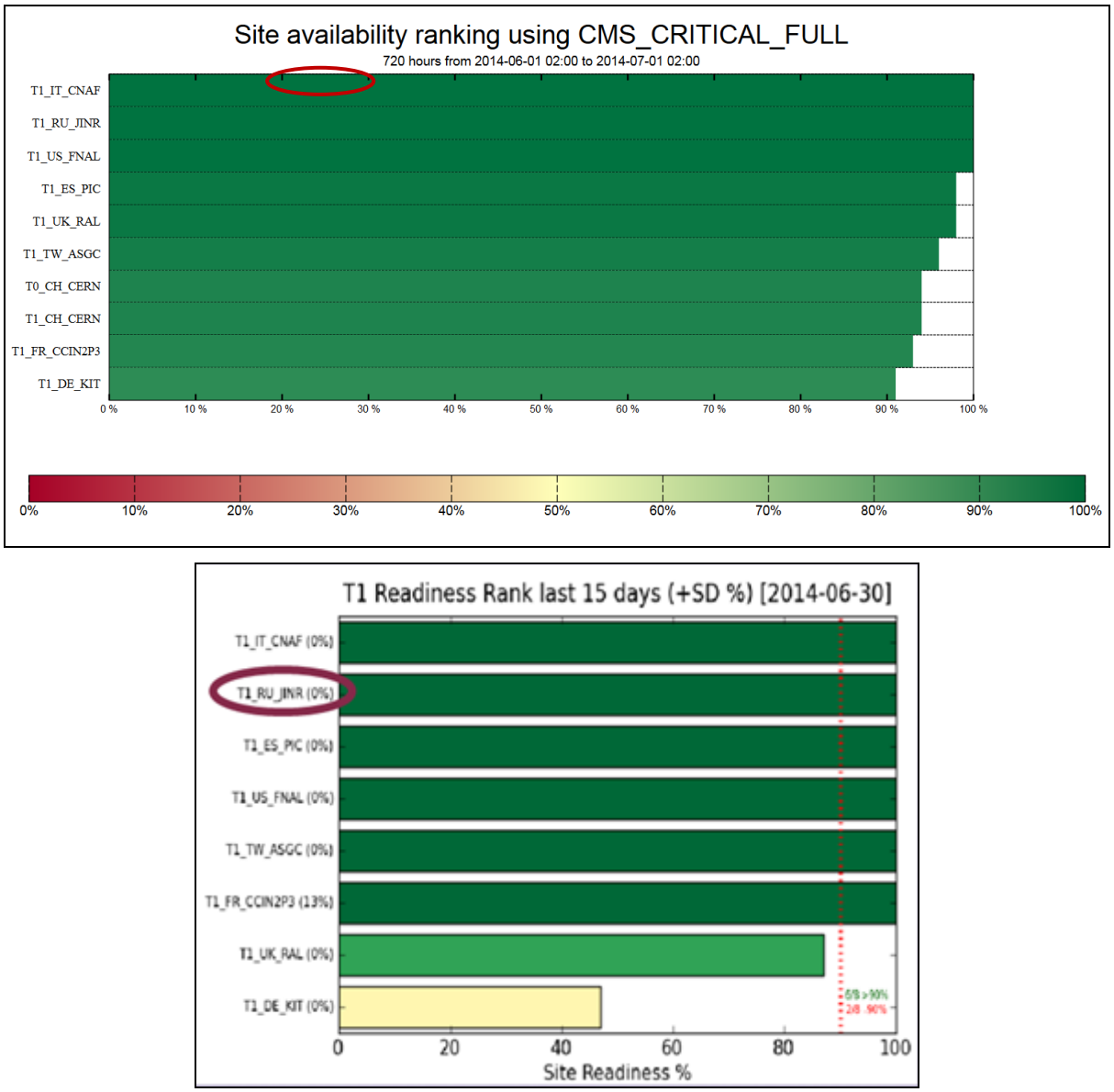


Fig. 5. Availability (top) and readiness(bottom) of CMS Tier-1 Centers

The JINR Tier sites is enabled to process more than 230 000 jobs per months (Fig. 6, left), i. e. about 6 % of all CMS jobs, with very high efficiency (~ 90%) (Fig. 6, right). The utilization metrics shown the efficiency of usage of job slots for all CMS Tier-1 sites are given in Figure 7.

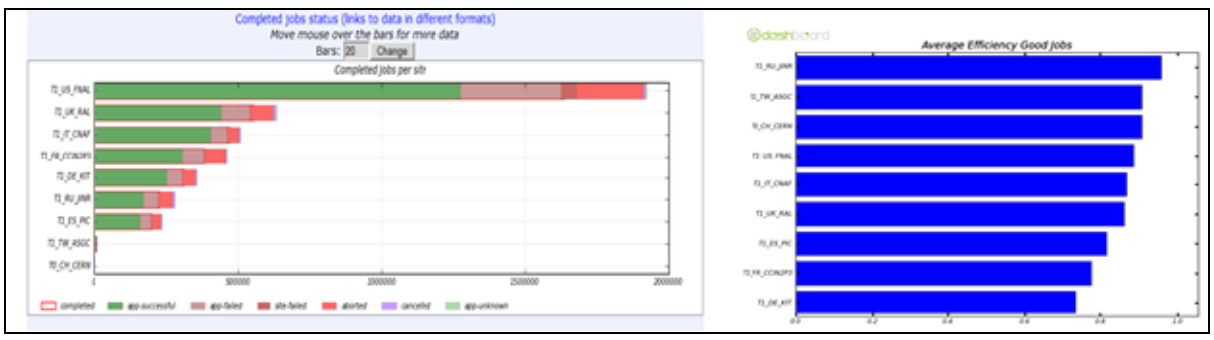


Fig. 6. Complited jobs per site for one month (left) and Average CPU efficiency for Good Jobs (right)

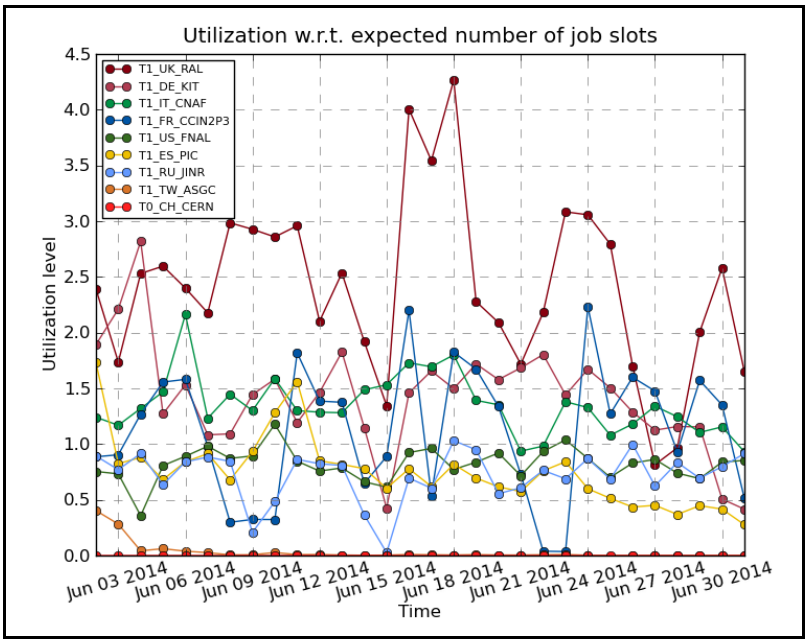


Fig. 7. CMS T1 Site Activity Summary (utilization level)

Summary

Current status and activities on creation of CMS Tier1 center at JINR were reported at NEC'2013 [Korenkov, 2013] and GRID'2012 conferences and are published in [Korenkov et al., 2012; Astakhov et al., 2013; Astakhov et al., 2013/04]. In February 2015 the JINR CMS Tier1 resources will be increased to the level that was outlined in JINR's rollout plan: CPU 2400 cores (28800 HEP-Spec06), 2.4 PB disks, and 5.0 PB tapes. It is planned the JINR CMS Tier-1 site will be included in the WLCG infrastructure as a Tier-1 production-level member with resources indicated above from February 2015 for WLCG use by the CMS Collaboration.

References

Astakhov N. S., Belov S. D., Dmitrienko P. V., Dolbilov A. G., Gorbunov I. N., Korenkov V. V., Mitsyn V. V., Shmatov S. V., Strizh T. A., Tikhonenko E. A., Trofimov V. V., Zhiltsov V. E. CMS Tier-1 at JINR, in NEC'2013 Proceedings, Dubna, 2013, pp.19–23.

Astakhov N. S., Belov S. D., Gorbunov I. N., Dmitrienko P. V., Dolbilov A. G., Zhiltsov V. E., Korenkov V. V., Mitsyn V. V., Strizh T. A., Tikhonenko E. A., Trofimov V. V., Shmatov S. V. The Tier-1-level computing system of data processing for the CMS experiment at the Large Hardon Collider. 15 p., "Information Technologies and Computation Systems", 2013/04, pp. 27–36 (in Russian).

CMS Dashboard. http://dashb-ssb.cern.ch/dashboard/request.py/siteviewhome

Education-portal. http://education-portal.com/academy/lesson/how-star-topology-connects-computer-networks-in-organizations.html#lesson

Grandi C., Stickland D., Taylor L. CMS NOTE 2004-031 (2004), CERN LHCC 2004-035/G-083; CMS Computing Technical Design Report, CERN-LHCC-2005-023 and CMS TDR 7, 20 June 2005.

Korenkov V.V. CMS Tier 1 at JINR. // XXIV International Symposium on Nuclear Electronics & Computing, NEC2013. 2013. http://nec2013.jinr.ru/files/13/Korenkov.ppt

Korenkov V. V., Astakhov N. S., Belov S. D., Dolbilov A. G., Zhiltsov V. E., Mitsyn V. V., Strizh T. A., Tikhonenko E. A., Trofimov V. V., Shmatov S. B. Creation at JINR of the data processing automated system of the TIER-1 level of the experiment CMS LHC. // Proceedings of the 5th Inter. Conf. "Distributed Computing and Grid-technologies in Science and Education", ISBN-5-9530-0345-2, Dubna, 2012, pp. 254-265 (in Russian).

LHC Computing Grid Technical Design Report. CERN-LHCC-2005-024, 2005; Worldwide LHC Computing Grid (WLCG), http://lcg.web.cern.ch/LCG/public/default.htm

The Large Hadron Collider. http://home.web.cern.ch/topics/large-hadron-collider

The Worldwide LHC Computing Grid. http://wlcg-public.web.cern.ch/

Transparent Interconnection of Lots of Links. http://www.ipinfusion.com/products/zebos/protocols/data-center-ethernet/TRILL

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 463−467

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Cloud Infrastructure at JINR

A. V. Baranov[1,a], N. A. Balashov[1,b], N. A. Kutovskiy[1,2,c], R. N. Semenov[1,d]

[1] Laboratory of Information Technologies, Joint Institute for Nuclear Research, Joliot Curie 6, Dubna, 141980, Russia
[2] National Scientific and Educational Centre of Particle and High Energy Physics of the Belarusian State University, Pervomayskaya Str.18, Minsk, 220040, Belarus

E-mail: [a] baranov@jinr.ru , [b] balashov @jinr.ru , [c] kut @jinr.ru , [d] roman @jinr.ru ,

Получено 30 сентября 2014 г.

Cloud technologies are already wide spread among IT industry and start to gain popularity in academic field. There are several fundamental cloud models: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). The article describes the cloud infrastructure deployed at the Laboratory of Information Technologies of the Joint Institute for Nuclear Research (LIT JINR). It explains the goals of the cloud infrastructure creation, specifics of the implementation, its utilization, current work and plans for development.

Keywords: cloud technologies, virtualization

Облачная инфраструктура ОИЯИ

А. В. Баранов[1], Н. А. Балашов[1], Н. А. Кутовский[1,2], Р. Н, Семенов[1]

[1]Лаборатория информационных технологий, Объединенный институт ядерных исследований Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6
[2]Национальный научно-учебный центр физики частиц и высоких энергий Белорусского государственного университета, Беларусь, 220088, г. Минск, ул. Первомайская, д. 18

Облачные технологии широко распространены в ИТ и начинают набирать популярность в научной среде. Существует несколько базовых моделей облачных сред: инфраструктура как услуга (*IaaS*, англ. *Infrastructure-as-a-Service*), платформа как услуга (*PaaS*, англ. *Platform-as-a-Service*), программное обеспечение как услуга (*SaaS*, англ. *Software-as-a-Service*). В данной статье рассматривается облачная инфраструктура, созданная в Лаборатории информационных технологий Объединённого Института Ядерных Исследований (ЛИТ ОИЯИ). Описаны цели создания облачной инфраструктуры, особенности ее реализации, использование, текущие работы и планы по развитию.

Ключевые слова: облачные технологии, виртуализация

Citation: *Computer Research and Modeling*, 2012, vol. 4, no. 3, pp. 463–467 (Russian).

1. Introduction

The JINR cloud service was deployed in order to increase an efficiency of the overall IT infrastructure of Laboratory of information technologies functioning: more efficient servers and services management, better hardware utilization, higher services and storage systems reliability. It is built upon an Infrastructure as a Service (IaaS) model. Such model provides network access to computational, software and information resources (networks, servers, storage devices, services and application software), allowing to allocate those resources on-demand according to dynamically changing requirements: cloud users can obtain, configure and deploy VMs themselves with the minimal assistance of the IT specialists. The cloud service is expected to reduce the total cost of ownership of the computing infrastructure as well as to reduce a complexity of its support

2. Service Implementation

The JINR cloud service is based on an open-source IaaS system OpenNebula [OpenNebula project, http://opennebula.org]. The two main components of the system can be marked out:

- front-end node (FN): contains the system core and user interfaces to interact with the service;
- cluster nodes (CNs): the physical servers which host the users' virtual machines (VMs).

While CNs are the physical machines, the FN is a virtual one hosted on one of the CNs itself.

Two user interfaces are available to access the service:

- command line interface (CLI);
- web-based graphical user interface “Sunstone”.

Figure 1 shows the interactions between the cloud service components.

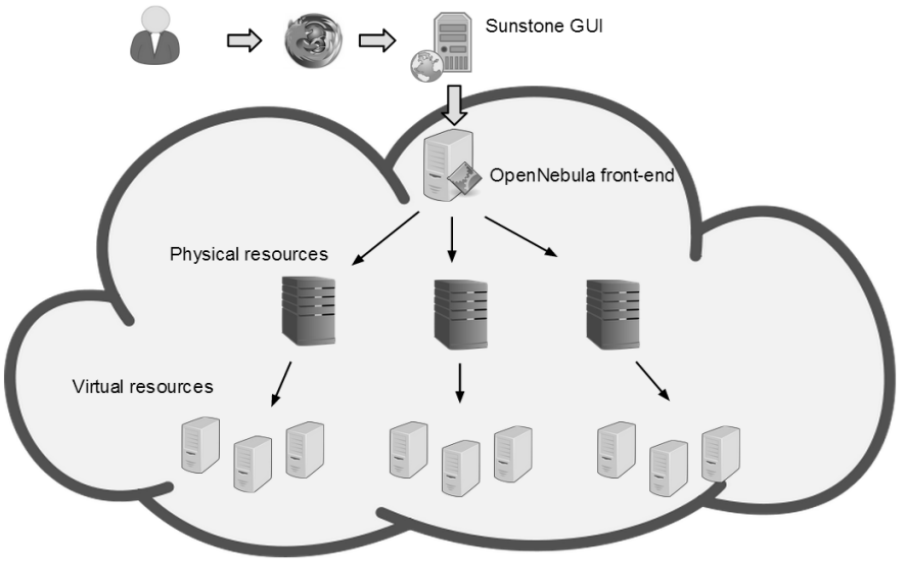


Fig. 1. JINR Cloud service structure scheme showing interactions between its components

Currently the service uses two virtualization technologies to provide VMs:

- OpenVZ [OpenVZ project, http://openvz.org] (an operating system-level virtualization);
- KVM [KVM project, http://www.linux-kvm.org] (provides full hardware virtualization).

The reason why two different virtualization technologies are used is to better utilize hardware resources and at the same time fit the variety of the emerging tasks: OpenVZ containers are lightweight and fast but they are bound to use the hosts operating system kernel, while KVM virtual machines support any type of operating systems inside the VMs but have higher overhead.

Originally OpenNebula had no OpenVZ containers support but its extensible and modular architecture allowed us to add such support by implementing the custom driver

JINR cloud service has two types of CNs:

- servers with two mirrored disk drives (RAID1) used to host highly reliable VMs;
- servers with one disk used for educational, research or test VMs.

The virtual machines are monitored currently by Opennebula, some their parameters are tracked by built-in monitoring system and its information is available on the Sunstone dashboard.

To make a request on resources or own quotas extension easily for end-users the custom plugin for Sunstone was developed. It's a simple web-form integrated into Sunstone menu. All that web-form's fields need to be filled by user. Pressing “Send” button automatically generates an email to cloud service manager for request approval.

The VMs can be accessed either with use of rsa/dsa-key or password. A plugin implementing Kerberos authentication was developed for user authentication in Sunstone. To increase the security of information exchange between the service web-interface and user’s browsers SSL encryption is used.

3. Service Usage

Currently the service usage is developed in three directions:

- test, educational and research tasks as part of JINR participation in various projects using cloud and grid technologies;
- systems and services deployment with high reliability and availability requirements;
- increasing computing capacities of the grid-infrastructures during peak loads.

The following services and testbeds are currently deployed in JINR cloud:

- EMI-based [EMI project, www.eu-emi.eu] testbed (used for trainings, performing JINR obligations in international projects such as WLCG [WLCG project, http://wlcg.web.cern.ch], etc);
- ATLAS T3MON [Andreeva et al., 2011] + PanDA [PanDA project, https://twiki.cern.ch/twiki/bin/view/PanDA/PanDA] testbed [Belov et al., 2012] (monitoring tools development for ATLAS Tier-3 sites, PanDA software development for distributed analysis);
- DIRAC-based [DIRAC project, http://diracgrid.org] testbed for BES-III [Web-portal of BES-III experiment, http://bes3.ihep.ac.cn.] experiment (monitoring tools development for BES-III distributed computing infrastructure);
- DesktopGrid testbed (to estimate the volunteer' computing technology for possible use in solving JINR users' tasks);
- web-service HEPWEB (provides a possibility to use different tools for Monte-Carlo simulation in high-energy physics);
- test instances of the JINR document server (JDS) and JINR Project Management Service (JPMS).

Moreover a set of OpenNebula testbeds are deployed in the JINR cloud service for development and debugging OpenVZ driver for current and new OpenNebula software releases. Each of such testbeds consists of 2-3 KVM VMs:

- one FN of test cloud instance,
- 1-2 CNs with OpenVZ hypervisor installed.

Services and testbeds currently deployed in the JINR cloud are shown in figure 2.

4. Current work and plans

Current work and features to do are listed below:

- implement authentication in VMs through Kerberos;

- create a support mailing list to interact with the end-users (to inform them about news, maintenance, new features, etc);
- estimate the possibility to implement Software as a Service (SaaS) model and/or the ability to provide access to virtual machines with pre-installed applications;
- improve quotas request form;
- deploy web-portal containing HOWTOs, FAQs and other information to improve end-users' experience with JINR cloud service.

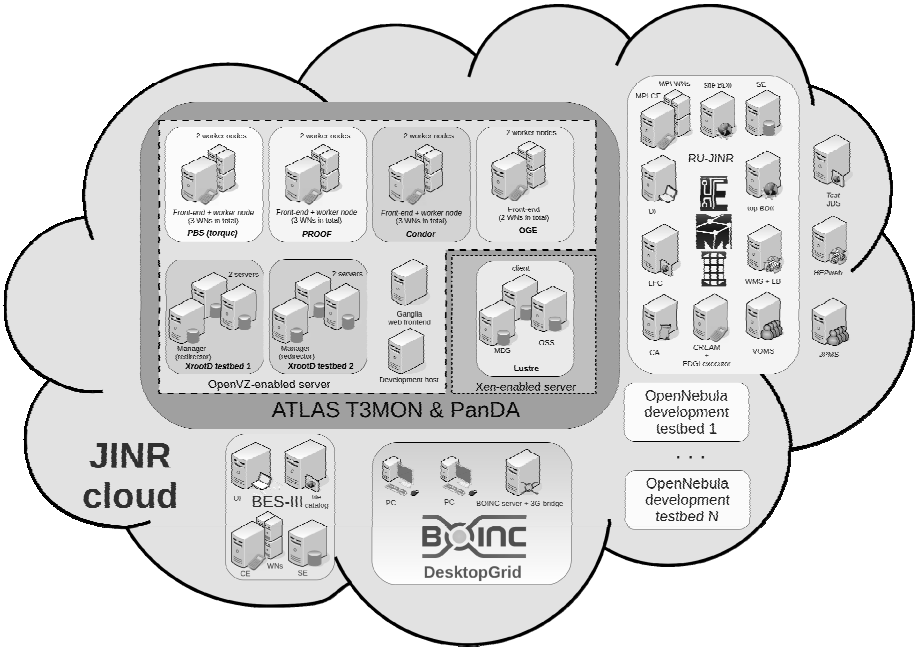


Fig. 2. Services and testbeds currently deployed in the JINR cloud

Conclusions

The JINR cloud service made possible to better utilize hardware resources. It also significantly simplified the job of system administrators by automating many virtual machines management tasks and by giving the users the ability to create and manage VMs by themselves within the limit of the granted quotas.

The service is actively used to cover users' demands as well as to carry out JINR commitments in Russian and international projects.

OpenNebula platform showed its stability and the ease of use. The source codes and platform architecture occurred to be well designed and easy to understand that makes it flexible and really easy to extend its functionality with custom plugins and drivers.

References

Andreeva J. et al. Tier-3 Monitoring Software Suite (T3MON) proposal. ATLAS note. — 2011.

Belov S. et al. VM-based infrastructure for simulating different cluster and storage solutions used on ATLAS Tier-3 sites // Journal of Physics: Conference Series. — 2012. — Vol. 396. Part 4. — P. 5. doi:10.1088/1742-6596/396/4/042036.

DIRAC project, [Электронный ресурс]. URL: http://diracgrid.org.

EMI project, [Электронный ресурс]. URL: www.eu-emi.eu.

KVM project, [Электронный ресурс]. URL: http://www.linux-kvm.org.
OpenNebula project, [Электронный ресурс]. URL: http://opennebula.org.
OpenVZ project, [Электронный ресурс]. URL: http://openvz.org.
PanDA project, [Электронный ресурс]. URL: https://twiki.cern.ch/twiki/bin/view/PanDA/PanDA.
Web-portal of BES-III experiment, [Электронный ресурс]. URL: http://bes3.ihep.ac.cn.
WLCG project, [Электронный ресурс]. URL: http://wlcg.web.cern.ch.

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 469−473**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75, 004.052.2, 004.052.32

BES-III Distributed Computing Status

S. Belov[1], Z. Deng[2], W. Li[2], T. Lin[2], I. Pelevanyuk[1], V. Trofimov[1], A. Uzhinskiy[1], T. Yan[2], X. Yan[2], G. Zhang[2], X. Zhao[2], X. Zhang[2], A. Zhemchugov[1]

[1] Joint institute for nuclear researches, Laboratory of Information Technologies, Joliot-Curie, 6, Moscow reg., Dubna, 141980, Russia
[2] Institute of High Energy Physics, Chinese Academy of Sciences, 19B YuquanLu, Shijingshan District, Beijing, 100049, China

Получено 30 сентября 2014 г.

The BES-III experiment at the IHEP CAS, Beijing, is running at the high-luminosity e+e- collider BEPC-II to study physics of charm quarks and tau leptons. The world largest samples of J/psi and psi' events are already collected, a number of unique data samples in the energy range 2.5–4.6 GeV have been taken. The data volume is expected to increase by an order of magnitude in the coming years. This requires to move from a centralized computing system to a distributed computing environment, thus allowing the use of computing resources from remote sites — members of the BES-III Collaboration. In this report the general information, latest results and development plans of the BES-III distributed computing system are presented.

Keywords: BES-III, distributed computing, grid systems, DIRAC Interware, data processing

Распределенные вычисления для эксперимента BES-III

С. Белов[1], Ц. Ден[2], В. Ли[2], Т. Линь[2], И. Пелеванюк[1], В. Трофимов[1], А. Ужинский[1], Т. Янь[2], С. Янь[2], Г. Чжан[2], С. Чжао[2], С. Чжан[2], А. Жемчугов[1]

[1] Лаборатория информационных технологий, Объединенный институт ядерных исследований Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6
[2]Институт физики высоких энергий, Китайской академии наук, Китай, 100049, г. Пекин, ЮкуаньЛу 19Б

В 2009 году в Пекине заработал детектор BES-III (Beijing Spectrometer) [1] ускорителя BEPC-II (Beijing Electron–Positron Collider). Запущенный еще в 1989 году BEPC за время своей работы предоставил данные для целого ряда открытий в области физики очарованных частиц. В свою очередь на BES-III удалось получить крупнейшие наборы данных для J/ ψ, ψ ' и ψ частиц при энергии ускорителя 2.5–4.6 ГэВ. Объемы данных с эксперимента (более 1 ПБ) достаточно велики, чтобы задуматься об их распределенной обработке. В данной статье представлена общая информация, результаты и планы развития проекта распределенной обработки данных эксперимента BES-III

Ключевые слова: BES-III, распределённый компьютинг, грид системы, DIRAC Interware, обработка данных

This work is supported in part by the joint RFBR-NSFC project No.14-07-91152 and NSFC projects 11179020 and 11375221.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 469–473 (Russian).

Introduction

The BES-III experiment at the BEPC-II collider located in the Institute of High Energy Physics (Beijing, China) started data taking in 2009 after a major upgrade of already existing accelerator BEPC and detector BES-II [Ablikim M. et al., 2010]. The experiment is run by an international collaboration of more than 400 members from 52 institutes in 12 countries from around the world. The main physics goals of the experiment are precision measurements in the tau-charm domain. The BES-III experiment has already taken the world's largest data samples of J/ψ ($1.2 \cdot 10^9$ events) and ψ' decays ($0.3 \cdot 10^9$ events), as well as a large amount of ψ (3770) data and a number of unique samples of data in the energy range 2.5–4.6 GeV. The total volume of experimental data is already about 0.9 PB, of which about 300 TB is event summary data for physics analysis (DSTs). This amount of data is rather large to be processed in a single computing center. Use of distributed computing looks like an attractive option to increase the computing resources of the experiment and to speed up the data analysis.

The BES-III computing model

Raw experimental data are taken from the BES-III detector and stored to the tape storage managed by CASTOR. The maximum data rate is about 40 MB/s. After reconstruction DSTs are produced and used in further physics analysis. DSTs are stored in a disk pool managed by Lustre and can be accessed only from internal IHEP network. The total amount of DSTs currently is about 300 TB. Both inclusive and exclusive Monte-Carlo simulation (MC) is made for each data sample as well. Experimental data taken with a random trigger are used in the simulation to reproduce noise and machine background individually for each run. The total amount of MC DSTs is more than 50 TB now. The BES-III offline software is based on the Gaudi framework [CERN Web site] and runs on Scientific Linux OS.

The BES-III distributed computing system

Grid computing became a routine tool for data processing in high energy physics after successful deployment in the LHC experiments. However, the main difficulty for a widespread use of the grid tools developed in the WLCG project is their large scale and complexity. It is not easy to adapt the distributed computing software that was designed for LHC experiments for use in a medium scale experiment. Limited manpower makes it even more difficult to maintain. For BES-III, the situation is even worse, because very few participating sites are members of WLCG. As a result there are few experienced grid users and developers and there is lack of grid computing infrastructure. Another problem is that network connectivity between institutes participating in the BES-III experiment is typically low. All these considerations motivate the following approach to the BES-III distributed computing model.

It is assumed that remote sites participate only in MC production and physics analysis, while all reconstruction of real experimental data is done at IHEP. Three operation models are considered, depending on the capabilities and priorities of each site:

a) MC simulation runs at remote sites. The resulting data are copied back to IHEP and then MC reconstruction runs there. (This model is convenient for sites with no SE or with only a small one);

b) MC simulation and reconstruction runs at remote sites. The resulting data are copied back to IHEP;

c) DSTs are copied from IHEP and other sites and analyzed using local resources.

Distributed analysis is postponed for later stage of the project.

BES-III grid solution

The DIRAC (Distributed Infrastructure with Remote Agent Control) software [DIRAC Web site] is chosen to be the main BES-III grid solution. DIRAC was designed originally for the LHCb experi-

ment, but with time it evolved into a generic product which could be used to access distributed computing resources in various communities of users. The main reasons why DIRAC is suitable for BES-III needs are the following:

- DIRAC provides all the necessary components to build ad-hoc distributed computing infrastructures interconnecting resources of different types, allowing interoperability and simplifying interfaces.
- DIRAC provides job management, data management, information system, monitoring, security system.
- DIRAC is rather easy to install, configure and maintain.
- DIRAC supports grids based on different middleware (gLite, EGI, VDT, ARC, etc).
- DIRAC requires no grid middleware installation on site. Remote hosts can be accessed through an SSH tunnel and the application runs via local resource management system.

DIRAC is adopted as a core part of the BES-III grid system. A production installation of DIRAC has been set up for BES-III, with nine remote sites and the DIRAC server running at the IHEP central site in Beijing. DIRAC job submission was tuned to fit to BES-III needs. Computing elements like gLite-CREAM and SSH-CE are used on the BES-III sites.

Interesting feature of DIRAC job management system is a capability to use cloud resources. Cloud computing becomes very popular technology nowadays, thanks to its flexibility and universality. For BES-III community cloud computing looks attractive because it allows to compensate peak overrun of resources and to use existing resources more effectively. VMDIRAC is an extension for DIRAC which allows to submit jobs to the clouds. Two servers for OpenStack and for OpenNebula have been set up at IHEP. Virtual resources from University of Turin, JINR and Soochow University are used for BES-III job processing.

Data management system

Data management in BES-III includes data storage, data transfer and catalogs. Main data storage in IHEP is managed by Lustre. and available from internal network only. To access data from outside a bridge connecting Lustre storage and the external network is needed. The problem was solved by introducing an extension to the dCache system, which allows to connect Lustre storage to the dCache server at IHEP as a disk pool and to synchronize the namespace. Using this bridge the data can be reached from internal IHEP network using all dCache-supported protocols. The BES-III transfer system based on FTS, provides reliable data transfer between IHEP and remote sites via both SRM and GridFTP protocols. SRM-capable storage elements dCache, Bestman and Storm are used at the BES-III sites. Catalogs are based on the DIRAC FileCatalog (DFC) with the MySQL backend. Variety of physics tasks of the experiment requires high granularity of data. Dynamic datasets based on metadata queries are used as containers of files. DIRAC provides a mechanism of the dataset management but more functionality needs to be developed to meet the experiment needs.

The BES-III Monitoring

A monitoring system is necessary to ensure the reliable data production using the BES-III distributed computing and to simplify maintenance and troubleshooting of the system. Regretfully, there is no low-level or high-level monitoring system provided by DIRAC, except the monitoring of DIRAC services themselves. Information about failure of the jobs or unavailability of the resources appears after several days after the event. Development of the monitoring system is required to provide enough input to decrease the number of failed jobs, to understand the failure reasons, to show system malfunction before failure occurs, to control overall status of the grid, to optimize data transfers, to check storage availability etc. By the end of 2013 the first prototype of BES-III grid monitoring system has been

developed and deployed. Simple jobs are submitted by a monitoring agent hourly via DIRAC job management system, both running the standard BES-III applications and providing system tests followed by sending the information back to the system. This information is collected, analyzed and available via the web page integrated into the BES-III DIRAC web portal. The number of tests is implemented to provide an information about the most important metrics of the BES-III grid: network ping test, WMS test (sending simple job), simple BOSS job (full simulation of 50 events), combined test of CVMFS, environment and resources availability, CPUlimit test, network, and SE latency test. Analysis includes site reliability estimation and identification of problematic hosts. An example of the monitoring page is shown in Fig. 1.

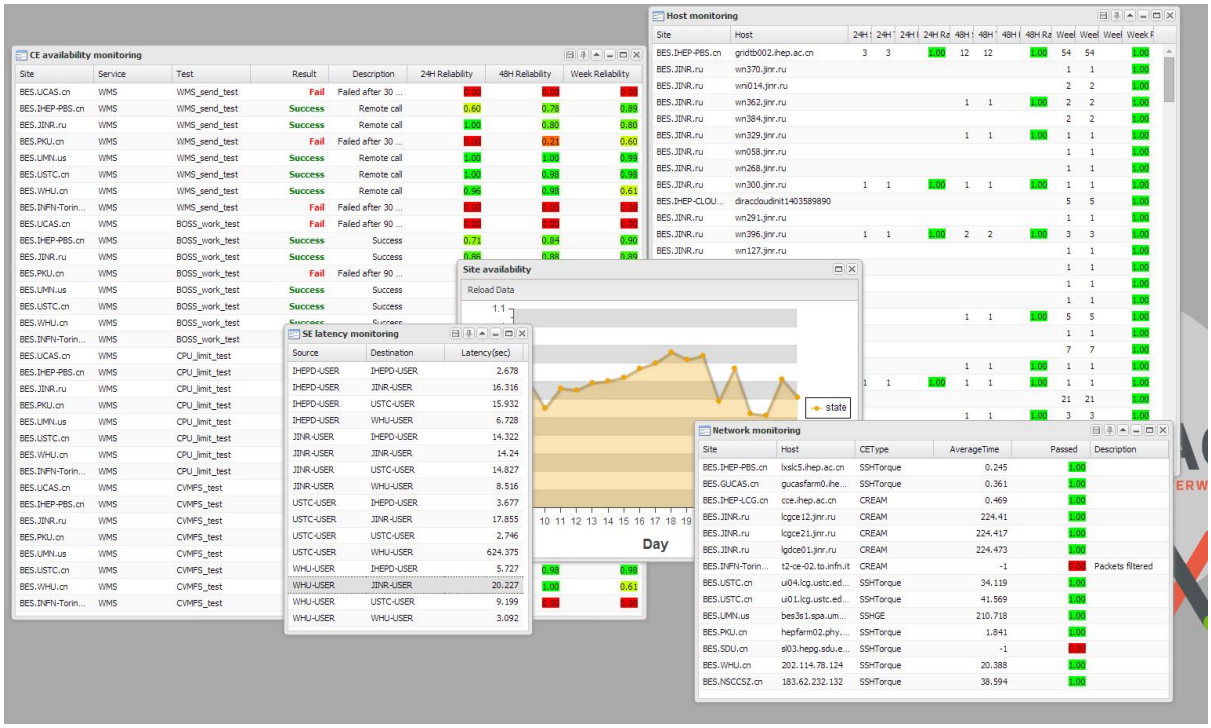


Fig. 1. Examples of the reports at BES-III monitoring system

Currently the existing monitoring system is being moved to be part of the DIRAC Resource Status System (RSS) service is in progress. RSS service is a new part of DIRAC that provides scheduling mechanism to collect and keep information about computing resources and to take decisions on use of these resources based on their availability. This mechanism suits very well to carry out functional tests and to perform the monitoring Being implemented in scope of the RSS framework the monitoring system can be used not only for BES-III experiment but as a generic solution for all DIRAC projects.

Summary

The BES-III distributed computing is operational since 2013. Since then more than 350000 jobs were executed and about 250 TB of disk space are managed by the system. While the basic infrastructure is built and the system is already put in production, more development is necessary to improve the dataset management, to integrate job management and data management systems, to implement fully functional monitoring & accounting system and to use clouds resources effectively. Experience and approaches to the organization of distributed computing gained by the BES-III collaboration may be interesting and useful for other medium scale experiments willing to use grid for their data processing.

References

Ablikim M. et al. "Design and Construction of the BESIII Detector'' Nucl. Instrum. Meth. A614 (2010) 345–399.
CERN Web site. http://www.cern.ch/gaudi
DIRAC Web site. http://diracgrid.org/

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 475−483

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.02

Running applications on a hybrid cluster

A. V. Bogdanov[1,a], I. G. Gankevich[1], V. Yu. Gayduchok[2], N. V. Yuzhanin[1]

[1]Saint Petersburg State University, University ave. 35, St. Petersburg, Peterhof, 198504, Russia
[2]Saint Petersburg Electrotechnical University "LETI", St. Professora Popova 5, St. Petersburg, 197376, Russia
E-mail: [a] bogdanov@csa.ru

Получено 17 декабря 2014 г.

A hybrid cluster implies the use of computational devices with radically different architectures. Usually, these are conventional CPU architecture (e.g. x86_64) and GPU architecture (e. g. NVIDIA CUDA). Creating and exploiting such a cluster requires some experience: in order to harness all computational power of the described system and get substantial speedup for computational tasks many factors should be taken into account. These factors consist of hardware characteristics (e.g. network infrastructure, a type of data storage, GPU architecture) as well as software stack (e.g. MPI implementation, GPGPU libraries). So, in order to run scientific applications GPU capabilities, software features, task size and other factors should be considered.

This report discusses opportunities and problems of hybrid computations. Some statistics from tests programs and applications runs will be demonstrated. The main focus of interest is open source applications (e. g. OpenFOAM) that support GPGPU (with some parts rewritten to use GPGPU directly or by replacing libraries).

There are several approaches to organize heterogeneous computations for different GPU architectures out of which CUDA library and OpenCL framework are compared. CUDA library is becoming quite typical for hybrid systems with NVIDIA cards, but OpenCL offers portability opportunities which can be a determinant factor when choosing framework for development. We also put emphasis on multi-GPU systems that are often used to build hybrid clusters. Calculations were performed on a hybrid cluster of SPbU computing center.

Keywords: GPGPU, HPC, computational clusters, OpenFOAM, LINPACK, ViennaCL, CUDA, OpenCL

The research was carried out using computational resources of Resource Center Computational Center of Saint Petersburg State University (T-EDGE96 HPC-0011828-001) and partially supported by Russian Foundation for Basic Research (project No. 13-07-00747) and Saint Petersburg State University (project No. 9.38.674.2013)

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 475–483 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 475−483

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

Запуск приложений на гибридном кластере

А. В. Богданов[1], **И. Г. Ганкевич**[1], **В. Ю. Гайдучок**[2], **Н. В. Южанин**[1]

[1] *Санкт-Петербургский государственный университет, Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35*
[2] *Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», Россия, 197376, г. Санкт-Петербург, ул. Профессора Попова, д. 5*

Гибридный кластер подразумевает использование вычислительных ресурсов с различными архитектурами. Как правило, в таких системах используется CPU распространенной архитектуры (например, x86_64) и GPU (например, NVIDIA CUDA). Создание и эксплуатация подобного кластера требует определенного опыта: для того чтобы задействовать все вычислительные мощности такой системы и получить существенное ускорение на задачах, требуется учесть множество факторов. К таким факторам относятся как характеристики оборудования (например, особенности сетевой инфраструктуры, хранилища, архитектуры GPU), так и характеристики программного обеспечения (например, реализация MPI, библиотеки для работы с GPU). Таким образом для эффективных научных расчетов на подобных системах требуется помнить о характеристиках ускорителя (GPU), особенностях программного обеспечения, характеристиках задачи и о многих других факторах.

В этой статье анализируются достоинства и недостатки гибридных вычислений. Будут приведены результаты запуска некоторых тестов и научных приложений, использующих GPGPU. Основное внимание уделено программных продуктах с открытым исходным кодом, которые поддерживают работу с GPGPU.

Существует несколько подходов для организации гетерогенных вычислений. В данной статье мы рассмотрим приложения, использующие CUDA и OpenCL. CUDA довольно часто используется в подобных гибридных системах, в то время как переносимость OpenCL-приложений может сыграть решающую роль при выборе средства для разработки. Мы также уделим внимание системам с несколькими GPU, которые все чаще используются в рамках подобных кластеров. Вычисления проводились на гибридном кластере ресурсного центра «Вычислительный центр СПбГУ».

Ключевые слова: GPGPU, высокопроизводительные вычисления, вычислительные кластеры, OpenFOAM, LINPACK, ViennaCL, CUDA, OpenCL

Introduction

Recent years have shown growing interest to hybrid computations. It became clear that conventional architectures have limited performance, which is in many cases inferior to performance of hybrid ones, let alone energy consumption and heat generation [Huang, Xiao, Feng, 2009]. Today one can face many different hybrid architectures, the vast majority of them are usually employ SIMD-accelerator (GPU, Cell, MIC, etc.) and a conventional CPU. This report concerns GPGPU as one of the earliest accelerator implementation.

One should clearly understand that such systems are not a panacea: while there are many tasks that can be smoothly mapped on GPGPU there are some classes of algorithms that can not benefit from implementing them for GPGPU. This report concerns aspects of running applications on a hybrid cluster that contains several GPGPUs on each node.

One can look at TOP-500 list and find out that the most powerful supercomputers are hybrid clusters: today (June 2014 TOP-500 list) about 35 % of overall performance of TOP-500 list systems is given by some accelerator extension cards [TOP 500 list statistics…] which is almost four times more than in 2010 (hybrid systems provided only 9 % of total performance at that time).

It can be explained with a simple fact: hybrid systems are constantly evolving that leads to performance growth while preserving and improving GFLOPS/Watt ratio. Software for hybrid systems is improving too. Software companies develop new libraries for GPGPU and applications that use such libraries, there are already several standards for GPGPU (e. g. OpenCL, OpenACC).

Such evolution can be seen on the example of GPGPU and other accelerators. Manufacturers try to ease programming of such systems. Some compilers can automatically split tasks between CPU and GPU. In case of MIC there are two basic approaches: native compilation and offload when MKL automatically offloads parts of a program to accelerator [Intel Xeon Phi Coprocessors…].

But one should clearly understand that not all tasks benefit from GPGPU usage (some tasks can show speedup while others show even slowdown when running on GPGPU) and remember that this area is still developing, so it will probably take time for such systems to become quite common and quite simple for programming.

1. GPU use cases

There are several ways one can harness GPU.

Conventional usage. The obvious case. GPU is used for graphics computations, relatively simple. One or multiple GPUs per one node. Probably there is no need for building a cluster in that case, since compute-intensive calculations are a part of the third case.

GPUs for virtual machines. The next approach is to use some GPUs within one or several virtual machines. Virtualization technologies are wide-spread due to advantages they offer to organizations and end users. Virtual CPU and network devices become ubiquitous while virtualization of powerful GPUs is an actively developed area which becomes a point of interest for major manufacturers. Such techniques are implemented, for example, in VMWare products and XenServer. There are two basic approaches in this area.

- Dedicated GPU (GPU pass-through). This variant is similar to conventional GPU usage: hypervisor just gives a virtual machine unrestricted access to the whole GPU. So, such virtual machine can use the whole GPU, while other VMs have no access to this GPU.
- Virtual GPU. Fully virtualized GPU is used in this case. Such approach can be exemplified by NVIDIA GRID K2 GPU managing by XenServer. Such GPU can be divided into several virtual GPUs with different characteristics. Each virtual GPU can be assigned to a separate virtual machine.

A scheme in Figure 1 depicts this two approaches.

GPUs within computational clusters. Finally, one can create cluster which nodes will contain one or more GPGPUs. Such clusters are usually referred as hybrid clusters. While the second case

(especially virtual GPUs) is still not common this case is the most frequently used approach for scientific computations. The main issues that arise in this case are listed below.

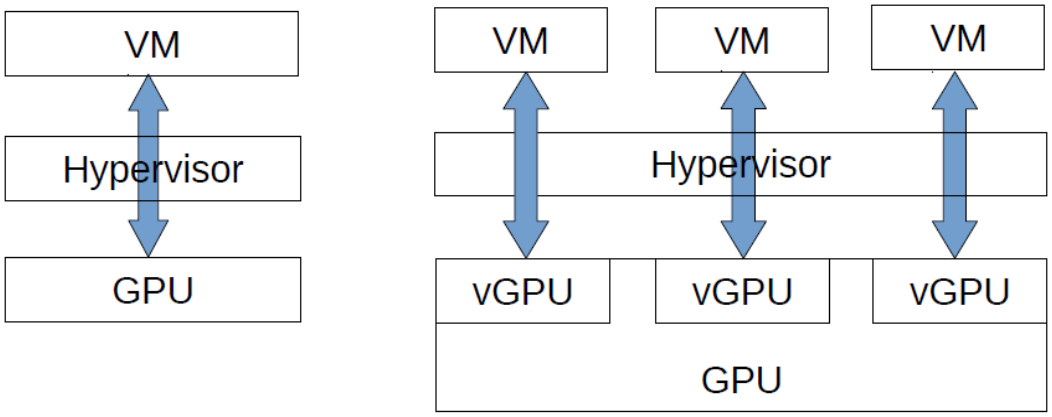


Fig. 1. Dedicated GPU and virtual GPUs

- How many GPUs should be installed into one node? The answer to this question is determined by several factors: node interconnect (that can be a “bottleneck” when the number GPUs is growing), the number of CPU cores (CPU will assign tasks to GPUs), etc.
- How to share cluster resources between users? One can solve this problem (it can be a real challenge in situation with limited resources that should be accessible for many users with radically different tasks) via Portable Batch System (PBS) [Torque resource manager]. There are several implementations of PBS that varies in some parameters, but the main idea is still the same: administrator creates several queues with different limitations. So, he can decide which users should have access to a particular queue.
- How to restrict user from using GPUs (or some other cluster resources) that assigned to another one? This question is solved by choosing appropriate managing system (for example, some reliable PBS implementations that can manage GPU access and so forth) and proper configuration as well.

2. Platform specifications

All tests for this report were performed using hardware of resource center of Saint Petersburg State University. This center has hybrid cluster with the following characteristics:

- 24 nodes;
- 2 CPU Intel Xeon X5650 (6 cores per CPU, total 12 cores);
- 96 RAM;
- 3 (16 nodes) or 8 (8 nodes) GPUs NVIDIA M2050 per node;
- Ethernet 10G network;
- Infiniband 4x QDR (40 Gbit/sec) network.

Peak performance of this complex is 59.6 TFLOPS. GPU peak performance is 0.5 TFLOPS (double precision), while CPU performance is 0.075 TFLOPS. Each node is running CentOS 6.4. Cluster management system is PBS.

3. Basic tests

For assessing the GPU performance and scalability of different tasks on GPU it's quite convenient to start from LINPACK. It is de facto standard benchmark for HPC systems (TOP-500 uses the best run of LINPACK test for creating the list). This test solves a dense system of linear

equations using LU factorization. Figure 2 depicts the LINPACK test results (performance in GFLOPS for one node with 3 GPUs depending on the matrix size).

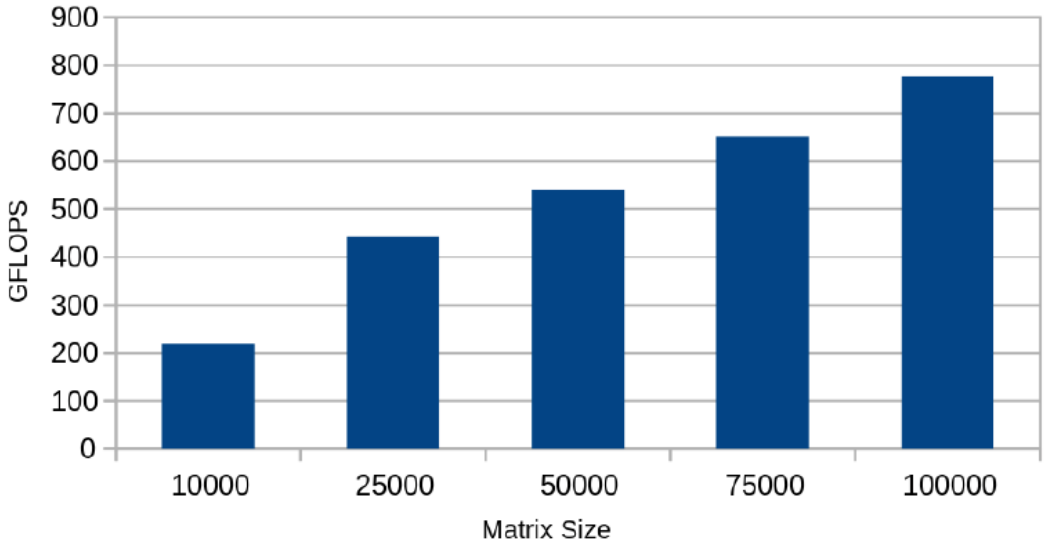


Fig. 2. LINPACK test results for nodes with 3 GPUs.

While LINPACK test uses double precision (TOP-500 takes into account only double precision LINPACK) such precision is not necessary for many tasks. We performed several tests that can be smoothly mapped on GPU (single and double precision). These tests show similar behavior: performance gain increases with task size. Coarse grained algorithms implementations show good scalability when run on multi-GPU systems. GPUs shows about twice as much performance for single precision in comparison to double precision for some tests. It is suits quite good to performance declared by manufacturer (0.5 TFLOPS for double and 1 TFLOPS for single precision). But these tests are synthetic, they can be efficiently mapped to GPGPU. Users don't need such tests (except for initial hardware testing) because they want to run real applications that solves their scientific problems. Some real world applications that uses GPGPU are discussed in the next section.

4. Applications

There are many applications that support GPGPU, open source and free, as well as commercial: Abinit, ANSYS, GROMACS, Matlab, OpenFOAM (with Ofgpu library or some other library for GPGPU), QuantumEspresso, etc. The list of such applications is constantly growing. We are interested in open source applications, but we will start from two well-known commercial applications because different scientists work with different applications (tools, libraries) and get used to different instruments, commercial and non-commercial, while our aim is to provide users with information about general rules for running GPGPU applications, point to a possible “bottlenecks”, assess performance which can be gained on hybrid systems, we look at clusters from administrator perspective.

Figure 3 depicts the MATLAB 2011b BLAS level 3 test for single and double precision that was run on CPU and GPU.

This diagrams once again shows how task size influences the performance. Tasks that can be vectorized (matrix operations in this example) can be mapped to GPGPU and benefit from GPGPU usage.

The next example is ANSYS Fluent case. We decide to run small case on different number CPU cores and GPUs. ANSYS Fluent version 15 allows user to use both CPU and GPU within a run. The only rule is CPU cores number should be divisible by GPUs count (that's why for some configurations there is no time value). The results are depicted in Figure 4.

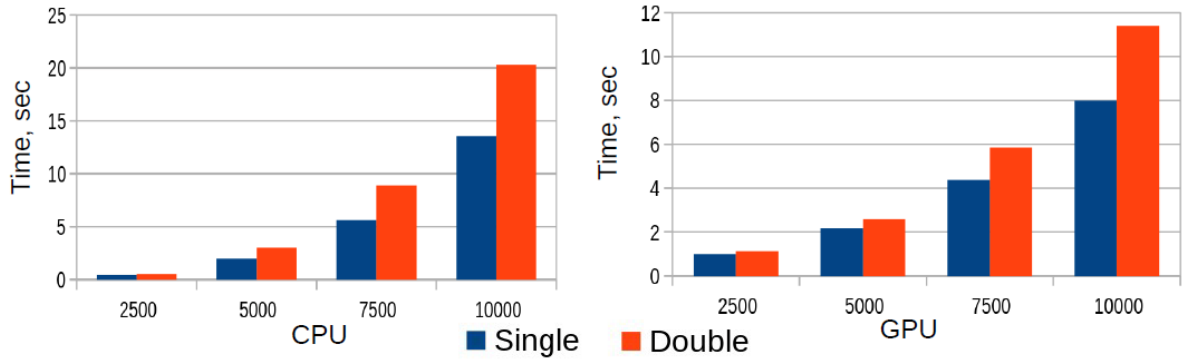


Fig. 3. MATLAB R2011b BLAS 3 level test

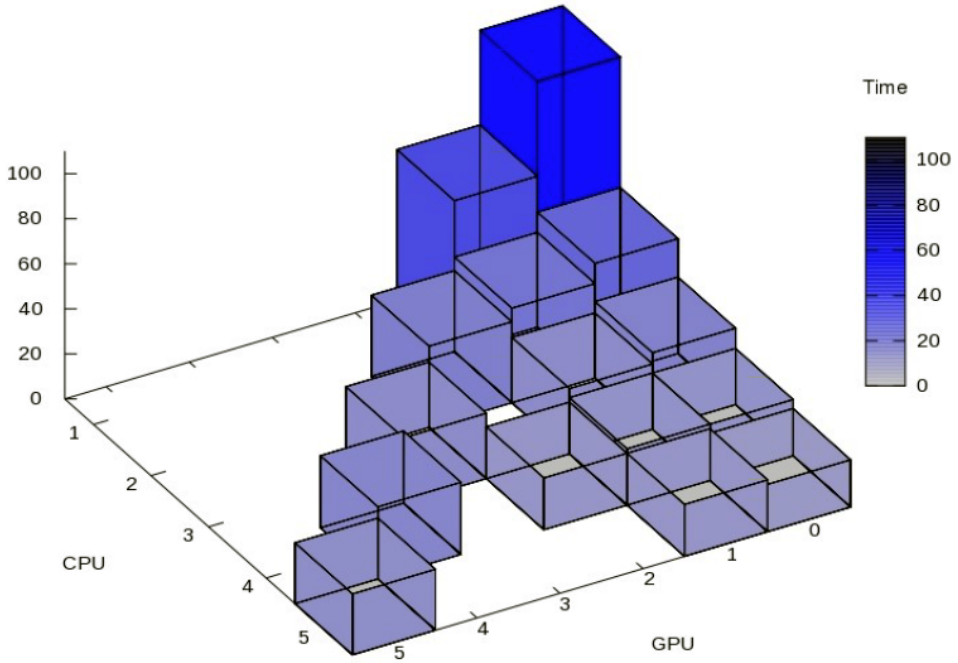


Fig. 4. ANSYS Fluent test results (small case)

As one can see small case has not so good scalability even if user runs certified large commercial product.

As an open source alternative to MATLAB we used ViennaCL in our tests. It has CUDA library and OpenCL versions and also version for MIC [ViennaCL…]. The results of LU decomposition test is depicted in Figure 5 (performance means here performance ratio, as compared with run time of the test with 500 matrix size).

And finally OpenFOAM. OpenFOAM is an open source platform for solving CFD problems [Jasak, Jemcov, Tukovic, 2007.]. There are several libraries for running OpenFOAM on GPU. Ofgpu is an open source library for GPU computations. It provides users with GPU linear system solvers and uses CUSP [GPU v1.1 Linear Solver Library...] to work with matrices.

There are several libraries for running OpenFOAM on GPGPU. Such libraries can be commercial or free. We decided to go for open source of gpu library.

OpenFOAM CFD calculations involve solving systems of linear equations. This task is usually offloaded onto GPUs.

Ofgpu provides users with 2 linear system solvers: PCGgpu (preconditioned conjugate gradient solver for symmetric matrices for GPGPU) and (PBiCGgpu - preconditioned biconjugate gradient

solver for asymmetric matrices for GPGPU). These solvers should be specified in the file: <case>/sysem/fvSolution.

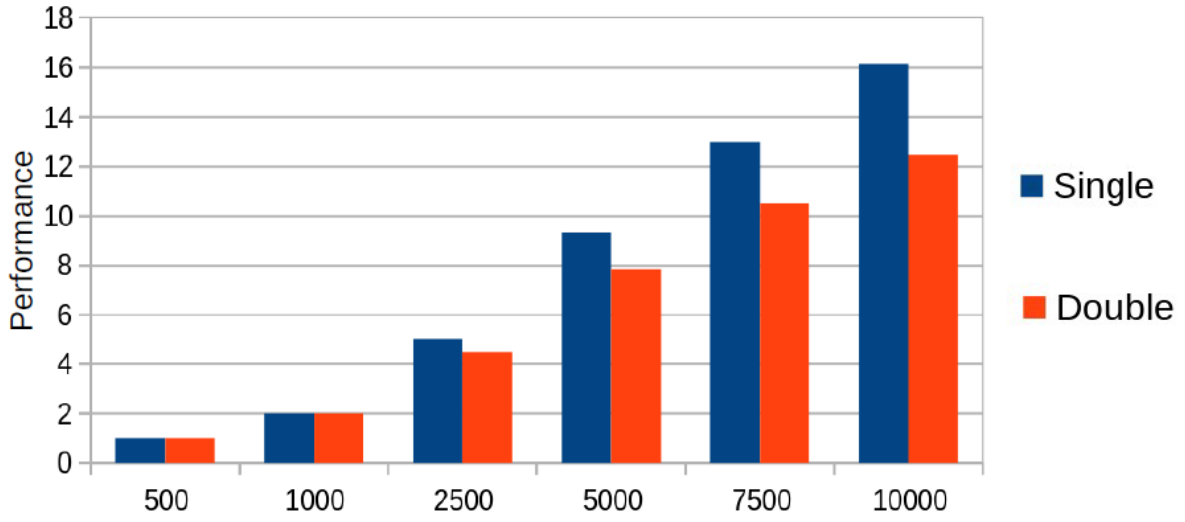


Fig. 5. ViennaCL LU decomposition test results

Ofgpu is built as a separate library and it should be loaded when OF solver use "*gpu" linear system solvers. User can specify GPU device to use in the file: <case>/sysem/controlDict ("cudaDevice" parameter).

Figure 6 and 7 depicts test results for CPU and GPGPU runs for different mesh sizes. Test case corresponds to a steady turbulent flow in a tube (OpenFOAM 2.2.2 single precision with OFGPU 1.1, compiled using Intel compilers (Intel Cluster Studio 2013; ICC 13.0.1), for CPU runs IMPI 4.1.0.024 from Intel Cluster Studio 2013 was used, CUDA Toolkit 5.5 was used for GPGPU version).

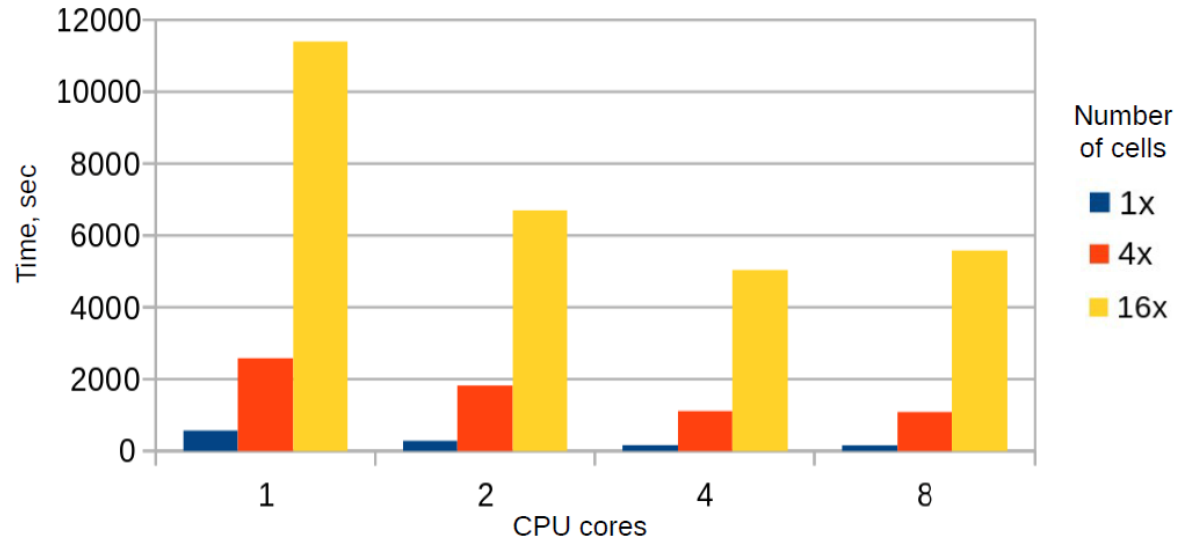


Fig. 6. OpenFOAM CPU test results

As one can see the finer mesh is used the more speedup is achieved.

But Ofgpu allows users to run OpenFOAM computations only on one GPU. Multi-GPU systems are not supported. One can run different tasks using different GPUs (by specifying GPU number), but one can not use several GPU within one task. We are currently working on multi-GPU version of ofgpu that will be able to harness multiple GPUs on different nodes on a hybrid cluster. The results will be published in future works.

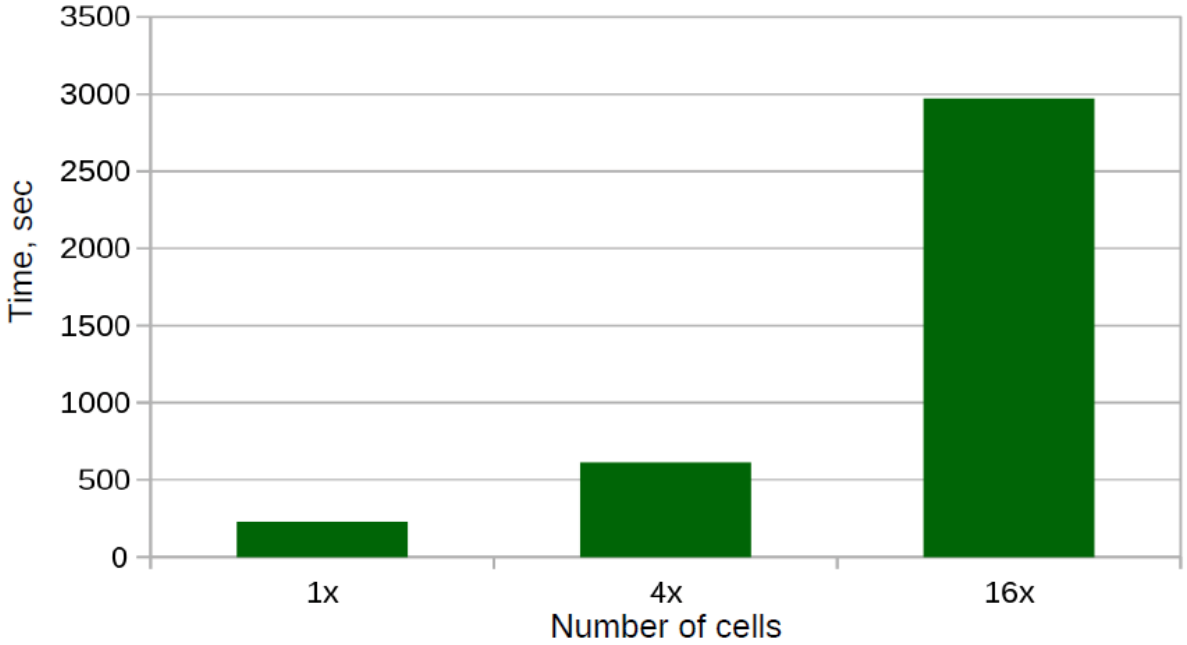


Fig. 7. OpenFOAM GPGPU test results

Conclusions

Hybrid clusters offer great peak performance, however, real performance depends on task. It is advisable to run coarse grained algorithms on such systems in order to decrease amount of data transferred between CPU and GPU because data transfer speed can be a bottleneck. In addition to this cluster architecture can introduce network speed as a new possible bottleneck. Software implementations can increase or decrease performance (for example, one should think about MPI implementation in case of MPI program). Performance gained is task size dependent. For example, one can get substantial speedup on GPU for CFD task when using finer mesh (but anyway there is a question whether or not one needs such fine mesh). Another possible problem is limited memory on GPU. When writing an application memory size should be taken into account. Programmer should choose the way he will use accelerators. He could write code for the accelerator using special API or delegate this work to compiler or special library that will offload the accelerator. The first method is advisable since compilers (or special libraries) are still can not split code for CPU and accelerators in a manner that will lead to great speedup. But programming such systems requires more accuracy from a programmer. Multi-GPU systems introduce new questions. And finally, one can get slightly different results on CPU and GPU working with floating point numbers. It concerns rounding and accordance to a floating point standard (e.g. IEEE 754) and it should be taken into account too.

Cluster management system can ease maintenance of such complexes. Sharing resources between users (especially GPUs) can be a challenge, but good management system can ease this task.

So, hybrid clusters are an actively developing area, promising systems that will probably be widely discussed subject in the future. Despite the fact that such systems introduce several issues that are not specific to traditional architectures they offer advantages that have never been specific to traditional architectures too.

References

GPU v1.1 Linear Solver Library for OpenFOAM, http://www.symscape.com/gpu-1-1- openfoam.

Huang S., Xiao S., Feng W. On the Energy Efficiency of Graphics Processing Units for Scientific Computing. 2009

Intel Xeon Phi Coprocessors web pages, http://www.intel.com/

Jasak H., Jemcov A., Tukovic Z. Openfoam: A c++ library for complex physics simulations. International Workshop on Coupled Methods in Numerical Dynamics, IUC, Dubrovnik, Croatia, 2007.

Resource Center Computational Center Website, http://cc.spbu.ru

TOP 500 list statistics (category "Accelerator/Co-processor Performance Share"), http://www.top500.org/statistics/list/

Torque resource manager web pages, http://www.adaptivecomputing.com/products/open-source/torque/

ViennaCL web pages, http://viennacl.sourceforge.net/

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 485−491

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.43

Performance of the OpenMP and MPI implementations on ultrasparc system

A. Bogdanov[1,a], P. Sone K. Ko[2,b], K. Zaya[2,c]

[1] Institute for High-performance computing and the integrated systems, St. Petersburg, 199397, Russia
2 St.Petersburg State Marine Technical University, 3 Lotsmanskaya Str., St. Petersburg, 190008 Russia

E-mail: [a] bogdanov@csa.ru, [b] pyaesonekoko@gmail.com, [c] kyawzaya4436@gmail.com

Получено 4 декабря 2014 г.

This paper targets programmers and developers interested in utilizing parallel programming techniques to enhance application performance. The Oracle Solaris Studio software provides state-of-the-art optimizing and parallelizing compilers for C, C++ and Fortran, an advanced debugger, and optimized mathematical and performance libraries. Also included are an extremely powerful performance analysis tool for profiling serial and parallel applications, a thread analysis tool to detect data races and deadlock in memory parallel programs, and an Integrated Development Environment (IDE). The Oracle Message Passing Toolkit software provides the high-performance MPI libraries and associated run-time environment needed for message passing applications that can run on a single system or across multiple compute systems connected with high performance networking, including Gigabit Ethernet, 10 Gigabit Ethernet, InfiniBand and Myrinet. Examples of OpenMP and MPI are provided throughout the paper, including their usage via the Oracle Solaris Studio and Oracle Message Passing Toolkit products for development and deployment of both serial and parallel applications on SPARC and x86/x64 based systems. Throughout this paper it is demonstrated how to develop and deploy an application parallelized with OpenMP and/or MPI.

Keywords: OpenMP, Parallel Programming, MPI (Message Passing Interface), SPARC System

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 485–491 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 485−491

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

Производительность OpenMP и реализация MPI на системе ultrasparc

А. В. Богданов, Пуае Сон Ко Ко, Кьяв Зайя

[1] *Института высокопроизводительных вычислений и Информационных Систем, Россия, 199397, г. Санкт-Петербург*
[2] *Санкт-Петербургский государственный морской технический университет, Россия, 190008, г. Санкт-Петербург, ул. Лоцманская, д. 3*

Данная работа нацелена на программистов и разработчики, заинтересованных в использовании технологии параллельного программирования для увеличения производительности приложений. Программное обеспечение Oracle Solaris Studio обеспечивает современную оптимизацию и распараллеливание компиляторов для языков C, C ++ и ФОРТРАН, продвинутый отладчик, и оптимизированные математические и быстродействующие библиотеки. Также включены чрезвычайно мощный инструмент анализа производительности для профилирования последовательных и параллельных приложений, инструмент анализа для обнаружения состязания при передаче данных и блокировки в памяти параллельных программ и IDE. Программное обеспечение Oracle Message Passing Toolkit обеспечивает высокопроизводительные MPI библиотеки и сопряжённую среду во время работы программы, необходимую для приложений передачи сообщений, которые могут работать на одной системе или по всему множеству вычислительных систем с высокопроизводительным сетевым оснащением, включая Gigabit Ethernet, 10 Gigabit Ethernet, InfiniBand и Myrinet. Примеры OpenMP и MPI представлены по всему тексту работы, включая их использование через программные продукты Oracle Solaris Studio и Oracle Message Passing Toolkit для развития и развертывания последовательных и параллельных приложений на основе систем SPARC и x86/x64. В работе продемонстрировано, как развивать и развертывать приложение, распараллеленное с OpenMP и/или MPI.

Ключевые слова: OpenMP, параллельное программирование, MPI (Message Passing Interface), система SPARC

Multicore Processor Technology

In a multicore processor architecture there are multiple independent processing units available to execute an instruction stream. Such a unit is generally referred to as a *core*. A processor might consist of multiple cores, with each core capable of executing an instruction stream. Since each core can operate independently, different instruction streams can be executed simultaneously. Nowadays all major chip vendors offer various types of multicore processors. A block diagram of a generic multicore architecture is shown in Figure 1.

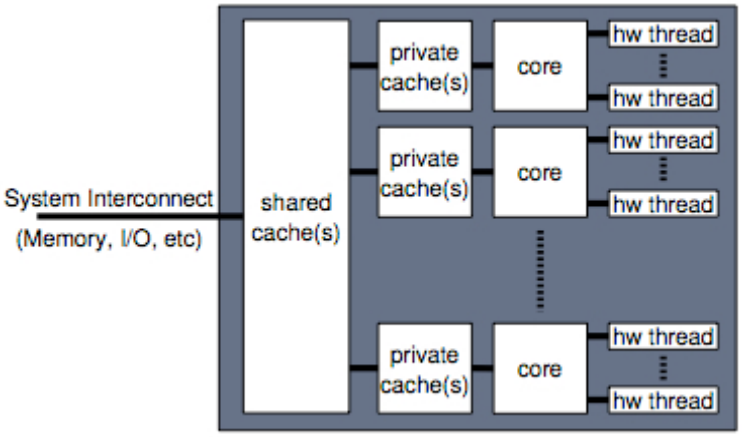


Fig. 1. Block diagram of a generic multicore architecture

In some architectures, each core has additional hardware support to efficiently execute multiple independent instruction streams in an interleaved way. For example, while one instruction stream waits for data to come from memory, another stream may be able to continue execution. This is transparent to the application and reduces, or even entirely avoids, processor cycles being wasted while waiting. It also adds a second level of parallelism to the architecture. Although a very important feature to improve both the throughput and single application parallel performance, we will not make this distinction in the remainder.

On the memory side, multiple levels of fast buffer memory can be found. These are generally referred to as *cache memory* or cache(s) for short. Today first level caches are typically local to the core. Higher-level caches can be local, but may also be shared across the cores. Typically at least the highest level of cache often is shared.

The instruction streams can be completely unrelated. For example, one might watch a video on a laptop, while having an email client open at the same time. This gives rise to (at least) two instruction streams. We say “at least” because each of these applications could be internally parallelized. If so, they might each execute more than one instruction stream.

On a dual-core processor, one core can handle the application showing the video, while the other core executes the email client. This type of parallel execution is often referred to as *throughput computing*. A multicore architecture greatly improves throughput capacity.

What is a Thread?

A thread consists of a sequence of instructions. A thread is the software vehicle to implement parallelism in an application. A thread has its own state information and can execute independently of the other threads in an application. The creation, execution and scheduling of threads onto the cores is the responsibility of the operating system. This is illustrated in Figure 2.

In general it is best for performance to make sure the hardware resources used are not overloaded and do not exceed their capacity. In case a resource is overloaded, the common phrase is to say that

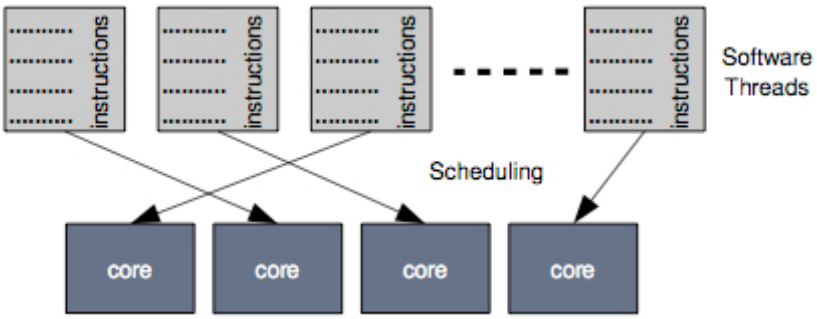


Fig. 2. Software threads scheduled onto the cores

this resource is oversubscribed. For example, when executing more than one application on a single core, the operating system has to switch between these programs. This not only takes time, but information in the various caches might be flushed back to main memory as well. In that respect, one should see the operating system itself as an application too. Its various daemons have to run in conjunction with the user level programs. This is why it is often most efficient to not use more software threads than cores available in the system, or perhaps even leave some room for these daemons to execute as well.

The exception is if a core has hardware support for multiple threads. In this case, some level of oversubscription of a core could be beneficial for performance. The number of software threads to use depends on the workload and the hardware implementation details.

On current operating systems, the user can have explicit control over the placement of threads onto the cores. Optimally assigning work to cores requires an understanding of the processor and core topology of the system. This is fairly low-level information, but it can be very beneficial to exploit this feature and improve the performance by carefully placing the threads.

To improve cache affinity, one can also pin the threads down onto the cores. This is called binding and essentially bypasses the operating system scheduler. It could work well in a very controlled environment without oversubscription, but in a time-shared environment it is often best to leave the scheduling decisions up to the operating system.

Why Parallelization?

Parallelization is another optimization technique to further enhance the performance. The goal is to reduce the total execution time proportionally to the number of cores used. If the serial execution time is 20 seconds for example, executing the parallel version on a quad core system ideally reduces this to 20/4 = 5 seconds. This is illustrated in Figure 3.

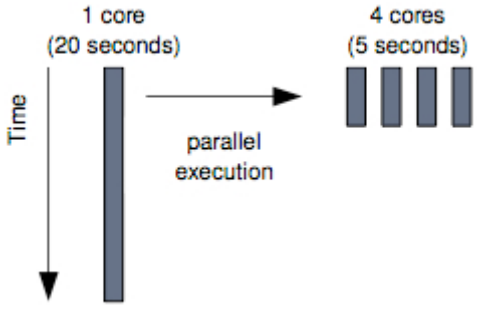


Fig. 3. Parellelization reduces the execution time

Parallelization attempts to identify those portions of work in a sequential program that can be executed independently. At run time this work is then distributed over the cores available. These units of work are encapsulated in threads.

The programmer relies on a programming model that will express parallelism inherent in an application. Such a parallel programming model specifies how the parallelism is implemented, and the parallel execution managed.

An Application Programming Interface (API) consists of a library of functions available to the developer. POSIX Threads (or *Pthreads*), Java Threads, Windows Threads and the Message Passing Interface (MPI) are all examples of programming models that rely on explicit calls to library functions to implement parallelism.

Another approach might utilize compiler directives such as #pragma constructs in C/C++ to identify and manage the parallel portions of an application's source code. OpenMP is probably the most well known example of such a model.

Parallel Architectures

In this section an overview of various types of parallel systems is given. These are generic descriptions without any specific information on systems available today.

The Symmetric Multiprocessor (SMP) Architecture
The Non-Uniform Memory Access (NUMA) Architecture
The Hybrid Architecture
The Cache Coherent Non-Uniform Memory Access (cc-NUMA) Architecture

Parallel Programming Models

There are many choices when it comes to selecting a programming model for a parallel system.

Automatic Parallelization
The OpenMP Parallel Programming Model
The Message Passing Interface (MPI) Parallel Programming Model
The Hybrid Parallel Programming Model

Performance Results

The results were obtained on a Sun SPARC Enterprise T5120 server from Oracle. The system had a single UltraSPARC T2 processor with 8 cores and 8 hardware threads per core. In Figure the elapsed times in seconds for the Automatically Parallelized and OpenMP implementations are plotted as a function of the number of threads used. Note that a log scale is used on the vertical axis.

For up to 8 threads, both versions perform equal. For 16 threads the Automatically Parallelized version performs about 9% faster than the OpenMP version.

Both versions scale very well for up to 8 threads. When using 32 threads, the performance deviation compared to the Automatically Parallelized version is about 30% for the OpenMP version. For 64 threads, the elapsed time is about twice as high. This difference is caused by the parallel overheads increasing as more threads are used. If more computational work was performed, this overhead would not be as dominant.

In Figure5 the performance of the MPI implementation is plotted as a function of the number of processes. Both the computational time (solid line) as well as the total time spent in the MPI functions (bar chart) are shown.

The time spent in the MPI functions is obviously zero if only one process is used. When using two processes it is just below one second, going up to 2.4 seconds on 64 threads. The computational work is very small. As a result, the cost of message passing is relatively dominant and no overall performance gain is achieved when running in parallel. If more work were performed, this would be different.

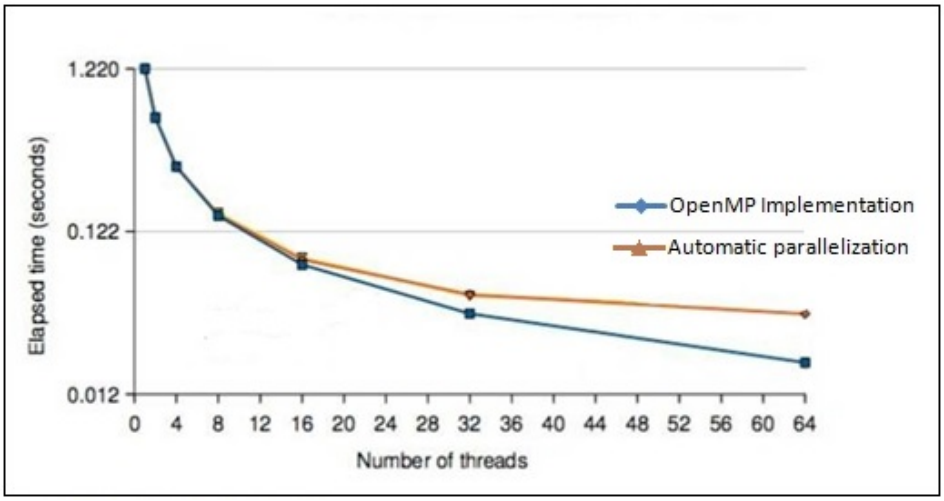


Fig. 4. Performance of the Automatically Parallelized and OpenMP implementations

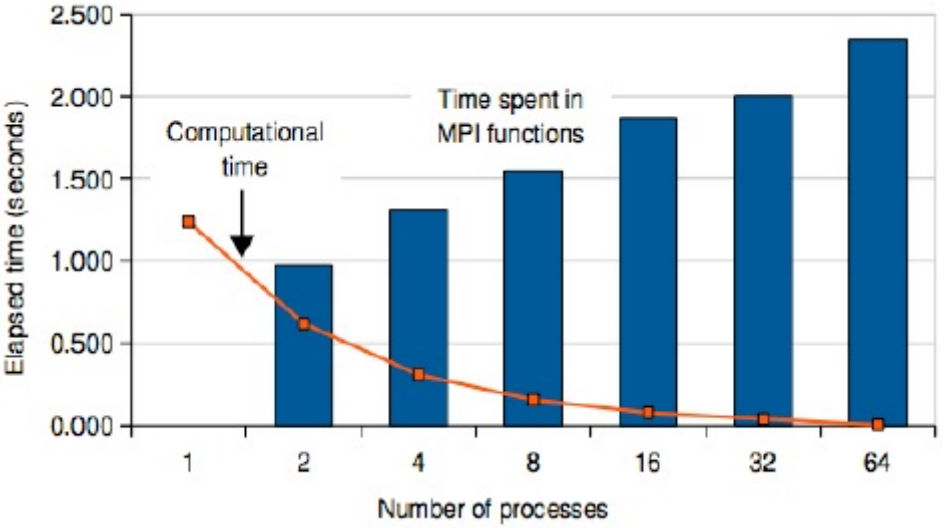


Fig. 5. Performance of the MPI implementation

Conclusion

The goal of parallel computing is to reduce the elapsed time of an application. To this end, multiple processors, or cores, are used to execute the application. The expected speed up depends on the number of threads used, but also on the fraction of the execution time that can be parallelized. Only if this fraction is very high, scalable performance to a very high number of cores can be expected. MPI is used to distribute the work over the nodes, as well as handle the communication between the nodes. More fine-grained portions of work are then further parallelized using Automatic Parallelization and/or OpenMP. Together with the Oracle Message Passing Toolkit, the Oracle Solaris Studio compilers can be used to develop and deploy these kinds of applications. OpenMP is a de-facto standard to explicitly implement parallelism. Like Automatic Parallelization, it is suitable for multicore and bigger types of shared memory systems. It is a directive based model, augmented with run time functions and environment variables. The Oracle Solaris Studio compilers fully support OpenMP, as well as additional features to assist with the development of applications using this programming model. The choice of the programming model has substantial consequences regarding

the implementation, execution and maintenance of the application. We strongly recommend to carefully consider these before making a choice.

References

Barbara Chapman, Gabriele Jost, Ruud van der Pas, "Using OpenMP", The MIT Press, 2008.

Darryl Gove, "Solaris Application Programming", Prentice Hall, 2008.

High Performance Computing and Communications Glossary, http://wotug.kent.ac.uk/parallel/acronyms/hpccgloss

MPI Forum, http://www.mpi-forum.org

Open MPI Home Page, http://www.open-mpi.org

OpenMP Specifications, http://openmp.org/wp/openmp-specifications

Oracle Message Passing Toolkit Home Page, http://www.sun.com/software/products/clustertools

Oracle Solaris Studio Documentation, http://developers.sun.com/sunstudio/documentation

Oracle Solaris Studio Numerical Computation Guide, http://docs.sun.com/app/docs/doc/819-3693

Oracle Solaris Studio Performance Analyzer Reference Manual, http://docs.sun.com/app/docs/doc/821- 0304.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 493–498

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.65

Storage database in cloud processing

A. V. Bogdanov[a], Thurein Kyaw Lwin[b]

Saint Petersburg State University, University ave. 35, Peterhof, St. Petersburg, 198504, Russia

E-mail: [a] bogdanov@csa.ru, [b] trkl.mm@mail.ru

Получено 27 октября 2014 г.

Storage is the essential and expensive part of cloud computation both from the point of view of network requirements and data access organization. So the choice of storage architecture can be crucial for any application. In this article we can look at the types of cloud architectures for data processing and data storage based on the proven technology of enterprise storage. The advantage of cloud computing is the ability to virtualize and share resources among different applications for better server utilization. We are discussing and evaluating distributed data processing, database architectures for cloud computing and database query in the local network and for real time conditions.

Keywords: Storage database, cloud processing, storage architecture, cloud architecture, data processing

Хранилища баз данных в обработке в облаке

А. В. Богданов, Тхурейн Киав Лвин

[1] *Санкт-Петербургский государственный университет,*
Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

Хранение — это существенная и дорогая часть облачных вычислений как с точки зрения требований сети, так и организации доступа к данным, поэтому выбор архитектуры хранения может быть критическим для любого приложения. В этой работе мы сможем посмотреть на типы облачных архитектур для обработки и хранения данных, основанных на доказанной технологии хранения в сети масштаба предприятия. Преимущество облачных вычислений — это способность визуализировать и разделять ресурсы среди различных приложений для наилучшего использования сервера. Мы обсуждаем и оцениваем распределенную обработку данных, архитектуры баз данных для облачных вычислений и очередь баз данных в локальной сети и для условий реального времени.

Ключевые слова: Хранилища баз данных, обработка в облаке, архитектуры хранения, облачная архитектура, обработка и хранение данных

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 493–498 (Russian).

Introduction

The cloud computing and cloud data stores have been a precursor and facilitator to the emergence of big data. Cloud computing is the commoditization of computing time and data storage by means of standardized technologies. Some cloud services provide consumers with space for storage and use of data for free, others charge a particular payment for services provided to subscribers. There are also private clouds, which are owned and operated by organization. In fact, it is protected network for storing and sharing critical data and programs. To create a private cloud requires hardware, software and other tools from different vendors and managing of the physical servers with the external and internal layers. Hybrid cloud, as is clear from the title, combined resources of different public and private clouds in a single service or a decision [Technical Details…]. The basis of all cloud services, products and solutions are software tools that functionality can be divided into three types, means for processing data and running applications (computing servers) to move data (network) and for storage (SAN).

The structure of Cloud and capabilities

Cloud computing processing and storage quickly gained popularity because they are not only provide a solution to the most complex and pervasive problems in the IT sector, but also open up a number of new features. In some environments, these technologies help to reduce costs and often necessary to expand the range of goals and objectives or services ensure compliance, reaching the necessary indicators of availability, performance, security and data protection [Armbrust, 2010].

Cloud Solutions

Cloud Solutions means to create and store content, as well as strategies that determine where and how the content is used. These solutions are used to create virtual infrastructures, organizations of all sizes which can accommodate the necessary applications and tools, as well as development environments and test new features [Technical Details…].

Some of common terms and phrases that characterize cloud solutions are.

• Optimized and cost-effective: the expansion of the range of services and ensure an appropriate level of service with the resources available

• Ability to create a variety of service options: resource allocation levels for different budgets and requirements for quality of service

• Flexible, scalable continuously: the possibility of expanding without complications

• Reliable, flexible and dynamic: adaptation to the ever-changing needs and availability

• Fast or automatic allocation of resources: quick access to resources and services

• Secure and supporting multi-client architecture: a safe separation of user data integrity

• Measure and manage: to provide metrics for reporting, analysis and management of services

• Scaling by increasing density: taking advantage of multi-client architecture to reduce costs

Distributed data processing

Distributed data processing is an opportunity to integrate fragmented data resources. One approach to centralizing data is to simply decommission existing database systems and to build a new integrated database. An alternative approach is to build an integration layer on top of pre-existing systems [Bobrovsky, 2013]. Building an integration layer on top of existing database systems is a challenge in complexity and performance, but this option sometimes makes the most business and engineering sense. In a data-sharing environment, there is no single best architecture that will solve all problems. Large installations of database systems may be accessed by hundreds of thousands of times a minute. The irreducible latency present even in a fully optical network is not capable of supporting

such a performance requirement. Indeed, local disks are also too slow, and most of this sort of information is cached off disk and into memory. In some organizations, if critical data is unavailable for even a matter of minutes, it could affect millions of dollars of revenue. This is why remote data access is not used in such large-scale situations where high availability is critical [Nikulchev, 2008]. There are many small and medium weight applications with modest performance requirements for data. Often, such applications are designed to work with a copy of data because getting a copy and loading it on a local database seems like the easiest solution. Such design does not factor in the cost of maintaining a separate copy of the data. When the applications are put into production and begin having problems keeping their data in sync, these costs become all too apparent. Such applications would probably do better to remotely reference their data. In such cases, it is a good architecture to remotely reference application databases for shared data. Such “distributed” databases need to incorporate some high availability design, depending on the weight of the applications served and their availability requirements. Each application should be analyzed to determine its performance and reliability requirements.

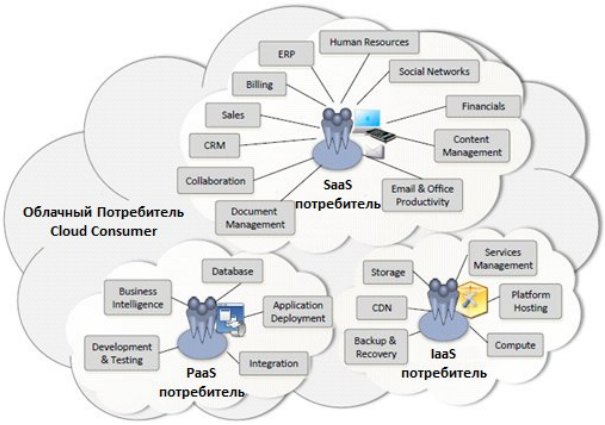


Fig. 1. Structure of cloud services

To choose a proper solution one must carefully look at existing tools and make a proper choice both of instruments and distributed system architecture.

DBMS application as a service

DBMS application is a layer between models DbaaS, SaaS, PaaS and cloud database. As for the DbaaS, it is a managed service in the mode of payment for use and providing access to a database of external programs [Babu, 2009]. Actually cloud DBMS scales almost to the size of PaaS and DbaaS volume and manages the database and the number of client connections to the database. Unlike the classical DbaaS, PaaS model is fairly obvious and clear. PaaS provides a ready hosting, where there is pre tuned and working database, but the interaction with the developed software it is necessary to organize manually, through the local interfaces, management and updating the system needs to be done separately, and payment is taken monthly, for hosting as a whole (this is one of the fundamental differences). When this is not excluded, than the servicing of DbaaS PaaS is implemented via predevelopment tools. The large uncertainties remain only in the ratio of the cloud model with DbaaS. First, the difference is mainly on the technical details of the implementation by the service provider (and the market is divided in the same way), but the end user in most cases can not apply to a single universal concept of "cloud DBMS."

What is the difference of Cloud DBMS and DbaaS

Cloud SMS — is a fully automated multi-user and unlimited scalable service that provides database functionality, but operated and administered "unnoticed" by the service provider [Pluzhnik,

Nikulchev]. It should not be confused cloud DBMS and database running on a virtual machine. Cloud model provides high flexibility and scalability of service, quick access to programming interfaces and settings. The user can connect to the system at any moment — for example, one hour — set the desired settings to load data to generate queries and get results without worrying about versions DBMS, its administration and configuration. Database as a Service — providing a simple but functional profile of saturated solutions "database in the cloud" for the needs of medium and small businesses and IT departments of large corporations. It usually does not occur directly in the provider's own data center, and functions as an add-on classical cloud services [Nikulchev, 2008]. Almost always specific DbaaS is one particular database provided in the cloud directly to the developer. Typical examples: Caspio, EnterpriseDB, Heroku Postgres, Xeround [Rittinghouse, 2010]. It is quite evident and displays the difference in business models: cloud DBMS suitable is for large-scale standard tasks and DbaaS is such for specialized application, using a particular brand of database engine, with the possibility of direct communication with its developers [Haak, 2011]. Furthermore, DbaaS allows much more accurate system to pick a right load, in particular, by controlling the amount of client connections.

Analysis of the Database and Hybrid Local Time Condition

We analyze the local database, build over a relational database MS SQL Server and occupied almost 26683.48 MB of memory on the database server. In the hybrid database information about the project and clients are moved from the local database to MS SQL Server in the cloud storage. In this process the memory on the database server of about 46.12 MB, and in the cloud server about 25 GB were used, thus producing the testing of queries. The results of two experiments are shown in Figure 2.

Table 1. The experiment result table

Record of Seconds		The number of records in a query	Time Extraction		Average query time		The number of all Project
Local	Hybrid		Local	Hybrid	Local	Hybrid	
0,534101	0,523230	1	177,533	183,334	1,7975	1,833	100
0,232501	0,220650	2	406,403	411,739	4,0862	4,117	200
0,140102	0,157200	3	621,604	627,270	6,238	6,272	300
0,101307	0,110017	4	880,467	883,8712	8,8168	8,838	400
0,090013	0,090732	5	1075,044	1099,451	10,960	10,994	500
0,073105	0,073070	6	1305,506	1329,938	13,055	13,077	600
0,060137	0,060032	7	1600,401	1604,110	16,014	16,041	700
0,050722	0,050643	8	1940,710	1943,777	19,407	19,437	800
0,04155	0,041502	9	2262,453	2265,402	22,624	22,653	900
0,03565	0,037819	10	2620,356	2622,060	26,203	26,220	1000

In Fig. 2. it is clearly seen, that for large databases with complex inquiries for the semi structured data there is no substantial loss of the time for data transmission in this experiment [Shokin, 2010]. Cloud storage is effectively a boundless data tank. It is important for effective performance in computations that while many solutions scale horizontally, when data is copied in parallel by cluster or parallel computing processes the throughput scales linear with the number of nodes for reading or writing. This includes products and solutions, that are used to deploy public, private and hybrid clouds.

Conclusion

We summarized above the database requirements for cloud databases and compared the suitability of database architectures for cloud computing. We also tested the queries in the local server and in the cloud, used for the construction of information systems and we can make conclusions about its

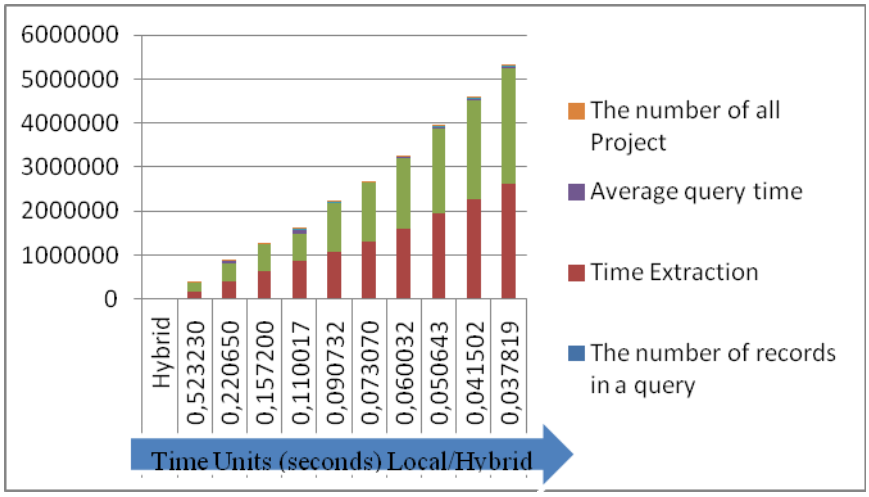


Fig. 2. The experiment results Graph

effectiveness with technology, developed for semi structured data. The steps for organization of such a system are:

1. Estimation of total system parameters (maximum number of users for simultaneous operation, the ability to scale services, the availability of personalized access).

2. Validation of the project (having own server capacity, cost comparison with the cost of launching of rental services).

3. Evaluation the time for data access, query performance evaluation for cloud infrastructures.

4. Constructing of automatic allocation system and sending requests in a distributed database.

To solve the first stage uses multi-criteria and decision-making methods, the second stage is realized on the basis of economic- mathematical methods of evaluation; third and fourth - based on optimization techniques and effective evaluation of search queries. Cloud storage systems are the easiest and most affordable way to solving the problem of highly loaded database. Easily run into work, they will not require re-deploy the application or any modification in the configuration files. The end goal of Database-as-a-Service is to automate database administration tasks and resolve its issues and challenges so that the end user doesn't have to think about anything on the database level. And just like the other "as-a-Service" models, it should also be on demand and easily deployable model. The database is requirements for the cloud databases and compares the suitability of different database architectures to cloud computing. Based on our performance results, we believe that the Cloud Database vision can be made a reality, and we look forward to demonstrating an integrated prototype of next Big Data Solution. Whether we need to assembling, managing or developing on a cloud computing platform and need a cloud-compatible database.

References

Armbrust M. A view of cloud computing / M. Armbrust, A. Fox, R. Griffith etc. // Communications of the ACM. — 2010. — Vol. 53, No. 4. — P. 50–58.

Babu S. Automated control in cloud computing: challenges and opportunities / S. Babu, J. Chase, S. Parekh // 1st workshop on Automated control for datacenters and clouds. — 2009. — P. 13–18 (DOI: 10.1145/1555271.1555275).

Beloglazov A. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints / A. Beloglazov, R. Buyya // IEEE Transactions on Parallel and Distributed Systems. — 2013. — Vol. 24, No. 7. — P. 1366–1379.

Bobrovsky S. MS. Week / Re № 30-31 (850-851) 19 2013 Noryabrya.

Haak S. Autonomic benchmarking for cloud infrastructures: an economic optimization model / S. Haak, M. Menzel // 1st ACM / IEEE Workshop on Autonomic Computing for Economics. — 2011. — P. 27–32 (DOI: 10.1145/1998561.1998569).

Nikulchev E. V. Construction of the model download channels in data networks based on geometric approach / E.V. Nikulchev, S.V. Payain // News of higher educational institutions. Problems printing and publishing. — 2008. — № 6. — S. 91–95.

Pluzhnik E. V., Nikulchev E. V. // Semistructured DATABASE hybrid cloud infrastructure.

Rittinghouse J. W. Cloud computing-implementation, management, and security / J.W. Rittinghouse, Ransome J.F. — NY: Taylor and Francis Group, 2010.

Shokin Y. I. Technology development of software systems information support of scientific activities, dealing with semistructured documents / Y. Shokin, A. M. Fedotov, V. B. Barakhnin // Computational technologies. — 2010. — T. 15, No. 6. — S. 111–125.

Technical Details Architecture cloud data processing and storage. http://www.seagate.com/ru/ru/tech-insights/cloud-compute-and-cloud-storage-architecture-master-ti/

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 499–504

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Improvement of computational abilities in computing environments with virtualization technologies

A. V. Bogdanov[1,a] , Kyaw Zaya[2,b], Pyae Sone Ko Ko[2,c]

[1] Institute for High-performance computing and the integrated systems, St. Petersburg, 199397, Russia
2 St.Petersburg State Marine Technical University, 3 Lotsmanskaya Str., St. Petersburg, 190008 Russia

E-mail: [a] bogdanov@csa.ru, [b] kyawzaya4436@gmail.com, [c] pyaesonekoko@gmail.com

Получено 4 декабря 2014 г.

In this paper, we illustrates the ways to improve abilities of the computing environments by using virtualization, single system image (SSI) and hypervisor technologies' collaboration for goal to improve computational abilities. Recently cloud computing as a new service concept has become popular to provide various services to user such as multi-media sharing, online office software, game and online storage. The cloud computing is bringing together multiple computers and servers in a single environment designed to address certain types of tasks, such as scientific problems or complex calculations. By using virtualization technologies, cloud computing environment is able to virtualize and share resources among different applications with the objective for better server utilization, better load balancing and effectiveness.

Keywords: Virtualization, Hypervisor, Shared Memory System, Single System Image

Усовершенствование вычислительных возможностей в вычислительной среде с помощью технологий виртуализации

А. В. Богданов, Кьяв Зайя, Пуае Сон Ко Ко

[1] Института высокопроизводительных вычислений и Информационных Систем, Россия, 199397, г. Санкт-Петербург
[2] Санкт-Петербургский государственный морской технический университет, Россия, 190008, г. Санкт-Петербург, ул. Лоцманская, д. 3

В этой работе мы показываем способы увеличения возможностей вычислительных сред с помощью виртуализации, единого системного образа (SSI) и коллаборации технологий гипервизора с целью улучшения вычислительных возможностей. В последнее время облачные вычисления как новое сервисное понятие, стали популярными для обеспечения различных сервисов для пользователя типа разделения мультимедиа, офисное on-line программное обеспечение, игры и on-line хранилища. Облачные вычисления объединяют множество компьютеров и серверов в единую среду, разработанную для обращения к определённым типам задач, таких как научные задачи или сложные вычисления. С помощью технологий виртуализации среда облачных вычислений способна виртуализировать и разделять ресурсы между различными приложениями с целью лучшего использования сервера, лучшего балансирования загрузки и эффективности.

Ключевые слова: Виртуализация, гипервизор, система разделяемой памяти, единый системный образ

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 499–504 (Russian).

Introduction

Now a day new technologies allow us to create relatively cheap multi-machine systems with shared computing resource, simplicity to manage heterogeneous resources and control large amount data transfer among them. Such systems provide relatively low computational cost, highly scalable, have high levels of reliability are proven tools for designing, debugging and performance analysis of parallel programs. The new approach is to combine virtualization technology with single system image. Since few years, virtual machines were offered enhanced resource management, using features such as live migration. SSI supports easier programming and administration for cluster environment. After evaluating different configurations, we show that combining the approaches allows us to better decisions challenges such as flexibility for the use of available resources and simplicity of use. In other terms, virtual machines add a level of management flexibility between the hardware and applications, while SSI provides an abstraction of distributed resources. Simultaneous use of both technologies could improve the overall platform resources, productivity and efficiency of the cluster running applications. Advantage of cloud computing is the ability to virtualize and share resources among different applications in order to ensure better utilization of the server [Al-Kiswany et al., 2011] [Nagarajan et al., 2007]. Cloud computing, a large distributed system that uses distributed resources to provide services to end-users through the implementation of several technologies. Virtualization technologies which are heavily relied on by the Cloud Computing environments provide the ability to transfer virtual machines (VM) between the physical systems using the technique of live migration mainly for improving the efficiency [Sabahi, 2012]. Dynamic server consolidation through live migration is an efficient way in cloud computing environments.

Hypervisor

A hypervisor, also known as a virtual machine monitor (VMM), is computer hardware platform virtualization software that allows several operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory and resources to itself. The hypervisor is controlling the host processor and resources, distributing what is needed to each operating system in turn and ensuring that the guest operating systems or virtual machines are unable to disrupt each other [Yong, Network…]. In a virtual environment, the virtual machine monitor (VMM) is the master control program with the maximum privilege level, and the VMM manages one or more operating systems. Hypervisors are classified into two types:

Native hypervisors: Software systems that run directly on the host's software as a hardware control and guest operating system monitor. A guest operating system thus runs on another level above the hypervisor. This is the classic implementation of virtual machine architectures.

Host hypervisors: Software applications that run within a conventional operating system environment. Considering the hypervisor layer being a distinct software layer, guest operating systems thus run at the third level above the hardware.

Virtualization

Virtualization enables the consolidation and pooling of resources so that they can be distributed across a variety of applications to compensate for the limited resources and reduce growing business needs. Virtualization provides a logical abstraction of physical computing resources and creates computing environments that are not limited to the physical configuration or implementation.

Virtualization Methods

Operating System-Based Virtualization

Virtualization is enabled by a host operating system that supports multiple isolated and virtualized guest OS's on a single physical server with the characteristic that all are on the same operating

system kernel with exclusive control over the hardware infrastructure. The host operating system can view and has control over the Virtual Machines [Sabahi, 2012] [Litty, 2005].

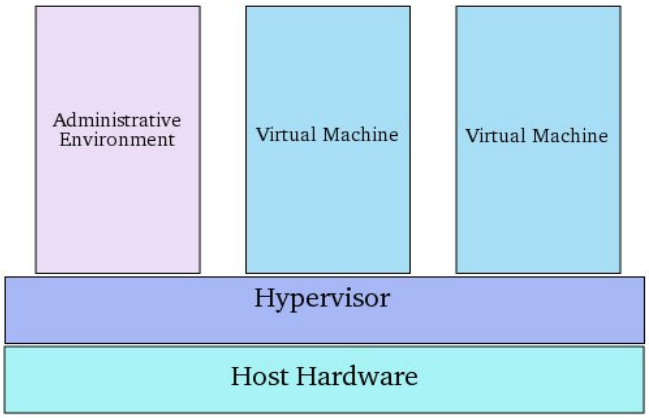


Fig. 1. Hypervisor (Virtual Machine Monitor)

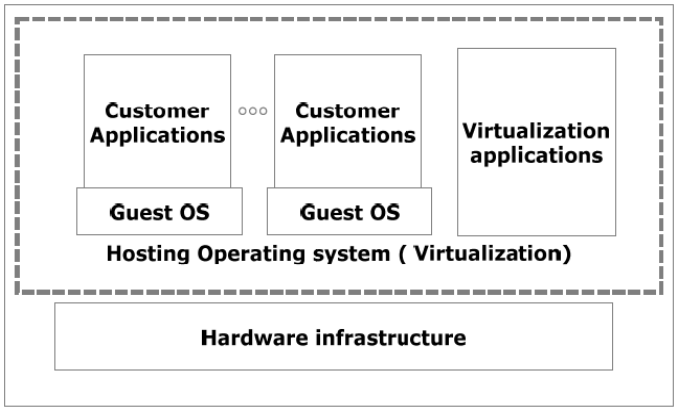


Fig. 2. Operation System-Based Virtualization

Application-based virtualization

An application-based virtualization is hosted on top of the hosting operating system. This virtualization application then emulates each VM containing its own guest operating system and related applications [Sabahi, 2012].

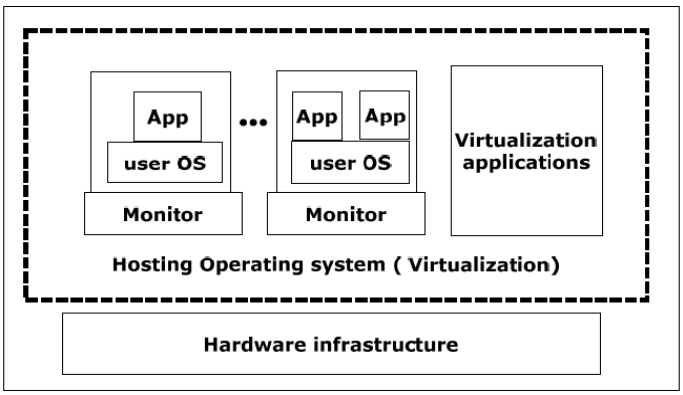


Fig. 3. Application-Based Virtualization

Hypervisor-based virtualization

The hypervisor is available at the boot time of machine in order to control the sharing of system resources across multiple VMs. Some of these VMs are privileged partitions which manage the virtu-

alization platform and hosted Virtual Machines. In this architecture, the privileged partitions view and control the Virtual Machines. This approach establishes the most controllable environment and can utilize additional security tools such as intrusion detection systems [Sabahi, 2012; Litty, 2005].

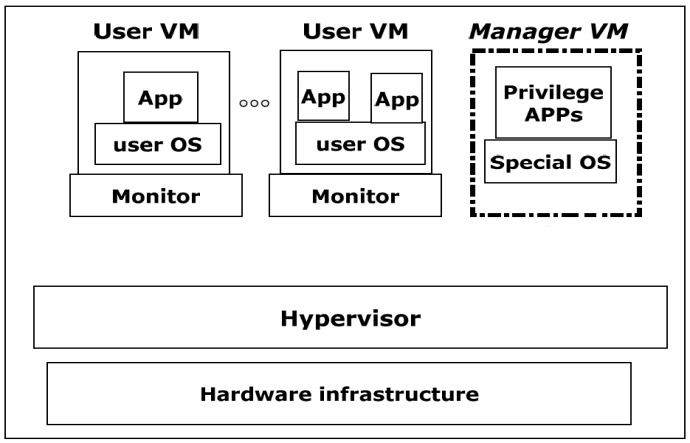


Fig.4. Hypervisor-Based Virtualization

Single System Image

SSI means that all the distributed resources are organized to a uniform unit for users, users can not be aware of the existence of every node that makes up of the computer system. SSI includes some attributes such as single memory space, single process space, single I/O space, single log on point, single file system, single loads management, and so on. The key attributes of SSI are single memory space and single process space [Rajkumar Buyya, 2001; Yong, Network…]. The SSI of a cluster can be implemented on the hardware level, the under-ware level, the middleware level and the application level.

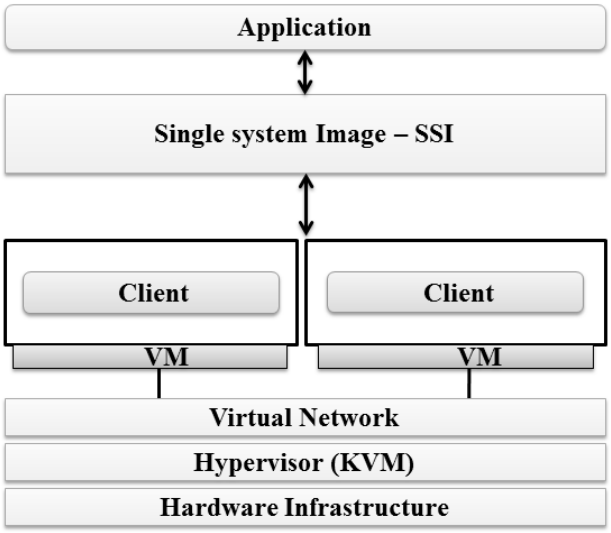


Fig.5. Integration of Single System Image and Virtualization

Live Migration

Transfers applications on a guest OS to any other healthy node and also provide the pre-migration mechanism where all the states are transferred before the application starts executing on the

new healthy node. The sequence in which the live migration occurs is explained below [Desai, 2012; Nagarajan et al., 2007]:

(a) Pre Migration: Whenever the migration is initiated, host inquires for available resources and if found, it will reserve them for the process.

(b) Pre Copy: All the pages are sent to the new node while the guest OS still continues the execution of the application. The changes made during this time are sent as dirty bits so that they can be identified on the new node.

Analyzing performance

After all configurations, we analyzed the performance of our virtual distributed computing environment. We manage heterogeneous resources, create a virtual computing clusters under the control of a hypervisor and combine computing resources into a single computing system, which shown in figure 5. Without virtualization technologies computing environment inability to control computer processes and dynamic load balancing between nodes to improve performance during execution of parallel and multi-threaded applications.

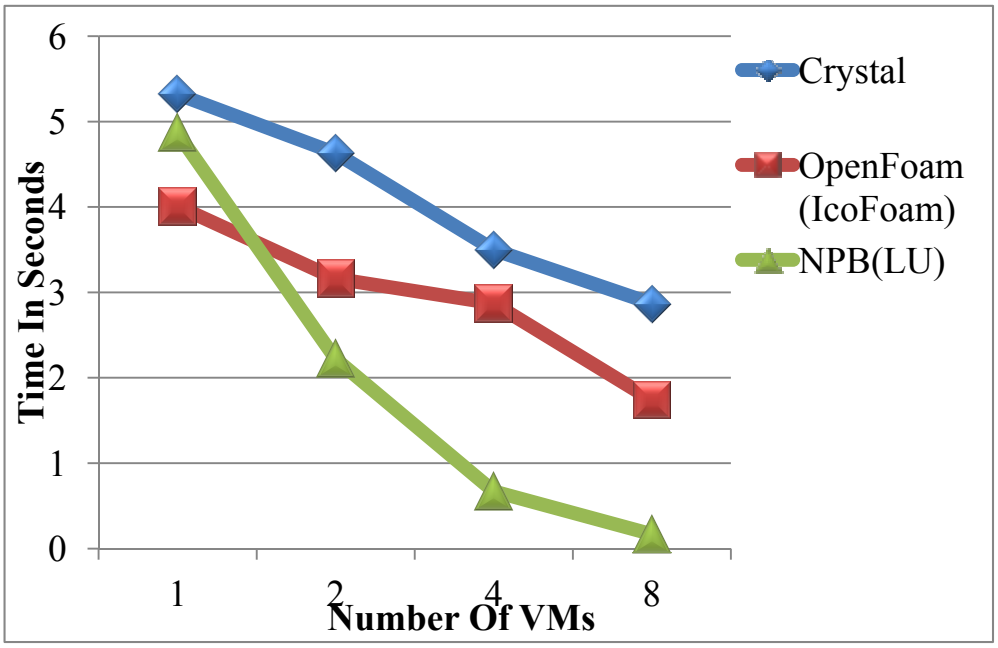


Fig. 6. Performance Testing On Virtual Computing Environment

Conclusion

Virtualization, that is increasingly attractive for commercial systems, as implemented by a small hypervisor that runs below the usual OS layer, has the same potential to benefit HPC applications in the dimensions of flexible OS variety, productivity, performance, reliability, availability, security and simplicity. Combination of virtualization and single system image allows processes to more easily migrate between nodes in the cluster for better load balancing and effectiveness of the computing environment [Mergen et al., 2003].

References

Al-Kiswany S., Subhraveti D., Sarkar P. and Ripeanu M. VMFlock: Virtual machine co-migration for the cloud. In HPDC'11 (San Jose, USA, June 2011).

Desai M. Department of Computer Science, University of Southern California Los Angeles, CA. High Performance Computing and Virtualization, 2012.

Litty L. "Hypervisor-based Intrusion Detection," M.S. thesis, Dept. Computer Science, University of Toronto, 2005.

Mergen M. F., Uhlig V., Krieger O., Xenidis J. Virtualization for high performance. 2003.

Nagarajan A. B., Mueller F., Engelmann Ch., Scott S. L. Proactive fault tolerance for HPC with Xen virtualization. ICS 2007: 23-32.

Rajkumar Buyya T. C. "Single system image (ssi)," the International Journal of High Performance Computing Applications, vol. 15, pp. 124–135, 2001.

Sabahi F. Secure Virtualization for Cloud Environment Using Hypervisor-based Technology, 2012.

Yong L. Network Center of Nanchang University, Single System Image with Virtualization Technology for Cluster Computing Environment, Nanchang 330031, China.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 505–509

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.27

The development of an ARM System on Chip based Processing Unit for Data Stream Computing

Mitchell A. Cox[a], Robert Reed, Bruce Mellado

School of Physics, University of the Witwatersrand,
1 Jan Smuts Avenue, Braamfontein, Johannesburg, 2000, South Africa

E-mail: [a]mitchell.cox@cern.ch

Получено 2 октября 2014 г.

Modern big science projects are becoming highly data intensive to the point where offline processing of stored data is infeasible. High data throughput computing, or Data Stream Computing, for future projects is required to deal with terabytes of data per second which cannot be stored in long-term storage elements. Conventional data-centres based on typical server-grade hardware are expensive and are biased towards processing power. The overall I/O bandwidth can be increased with massive parallelism, usually at the expense of excessive processing power and high energy consumption. An ARM System on Chip (SoC) based processing unit may address the issue of system I/O and CPU balance, affordability and energy efficiency since ARM SoCs are mass produced and designed to be energy efficient for use in mobile devices. Such a processing unit is currently in development, with a design goal of 20 Gb/s I/O throughput and significant processing power. The I/O capabilities of consumer ARM System on Chips are discussed along with to-date performance and I/O throughput tests.

Keywords: high data throughput computing, big data, arm system on chips

Разработка системы ARM на базе блока обработки данных для вычислений потока данных, реализованного на основе ИС

М. А. Кокс, Р. Рид, Б. Мелладо

Университет Витватерсранда, Южная Африка, 2000, Йоханнесбург, 1 Ян Смут Авеню

Современные масштабные научные проекты становятся все более информационно ёмкими, и обработка хранимых данных в режиме offline является невозможной. Требуется высокая пропускная способность при вычислениях или Вычисления Потока Данных, чтобы иметь возможность обрабатывать терабайты данных в секунду; такие данные не могут быть элементами длительного хранения. Общепринятые датацентры, основанные на стандартном аппаратном обеспечении, являются дорогими и настроены на вычислительную мощность. Общая пропускная способность может быть увеличена с помощью массивного параллелизма, чаще всего за счет повышенной вычислительной мощности и потребления энергии. Система ARM на основе ИС (SoC) может решить проблему системы ввода/вывода и соотношение CPU, доступность и эффективность использования энергии, так как ARM SoC являются элементами массового производства и разработаны на основе эффективного использования энергии в мобильных устройствах. На данный момент такой элемент обработки находится в разработке и нацелен на пропускную способность ввода/вывода в 20 Гб/с и значительную вычислительную мощность. Рассмотрены возможности ввода/вывода потребления системы ARM на основе ИС вместе с вычислением производительности и тестами на пропускную способность ввода/вывода.

Ключевые слова: высокая вычислительная пропускная способность, большие данные, система на ARM чипе

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. We would also like to acknowledge the School of Physics, the Faculty of Science and the Research Office at the University of the Witwatersrand, Johannesburg.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 505–509 (Russian).

1. Introduction

Projects such as the Large Hadron Collider (LHC) at CERN and the Square Kilometer Array (SKA) in South Africa generate enormous amounts of raw data which presents a serious computing challenge.

A simple plot, shown in Fig. 1, of the increase in CPU processing power in MIPS (Million Instructions Per Second) and hard drive read-write speed in MB/s (MegaByte/s) over many years clearly demonstrates the fact that hard drive I/O rates are insufficient, and will not become sufficient in the near future, to store the entirety of raw data from modern scientific experiments such as the SKA and LHC [Dursi, 2012].

The increase in Ethernet throughput, however, is at a similar rate to the increase in CPU processing power. Based on Amdahl's Laws it has been recommended that approximately one compute instruction per bit of data is required for a balanced system and this relationship is clear when comparing CPU and Ethernet in Fig. 1 [Szalay et al., 2010]. It appears upon inspection that CPU performance and Ethernet throughput are well balanced but in reality high-end Ethernet in not commonly available except on very high-end systems. For example, 1 Gb/s Ethernet from 2002 is suitably balanced with a 2002 performance CPU. It is imbalanced when it is coupled with a modern CPU with an order of magnitude higher performance, however this is a very common situation since high-end CPUs are more prevalent than cutting-edge Ethernet.

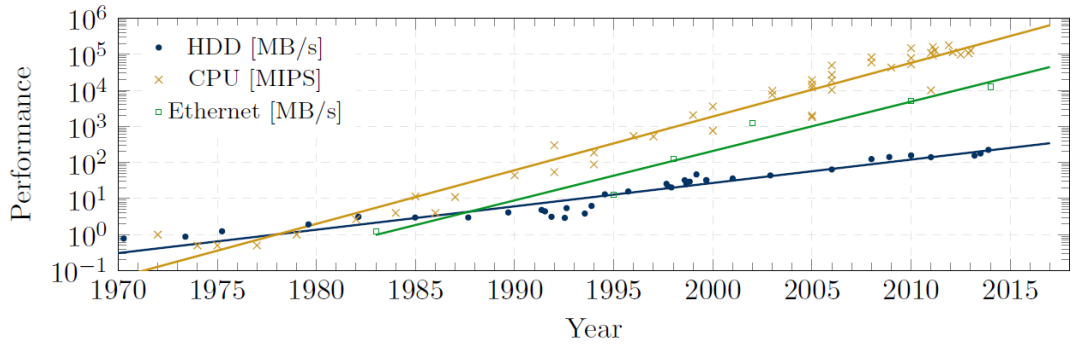


Fig. 1. Hard drive (HDD) and Ethernet 802.3 data throughput and CPU performance on a log scale since 1970 [Dursi, 2012; Wikipedia IEEE 802.3, 2014]

A specialised triggering and data acquisition system is currently employed by the LHC to reduce the amount of data produced to a manageable quantity for offline storage. This solution is not always suitable and so a paradigm shift is necessary to deal with future workloads and new projects. The cost, energy efficiency, processing performance and I/O throughput of the computing system to achieve this task is vitally important to the success of future big science projects. Current x86-based microprocessors such as those commonly found in personal computers and servers are typically biased towards processing performance and not I/O throughput and are therefore less-suitable for cost-effective high data throughput applications due to the necessity for massive parallelism.

High Volume throughput Computing (HVC) provides a suitable paradigm for data stream computing applications `Zhan J et al., 2012]. HVC is a datacenter based computing paradigm where the focus is on loosely-coupled throughput-oriented workloads in terms of either requests (service type applications), processed data (big data applications) or the maximum number of simultaneous subscribers (interactive real-time applications). The definition does not include data-intensive MPI workloads since these are suitably covered by High Performance Computing (HPC).

One of the first steps to the development of an effective HVC system is a high data throughput Processing Unit (PU). This PU should be well balanced in terms of CPU performance and I/O throughput and latency to maximise energy efficiency and cost.

ARM System on Chips (SoCs) are found in almost all mobile devices due to their low energy consumption, high performance and low cost and are the basis for the PU under develop-

ment [Rajovic et al., 2013]. Section 2 provides a brief overview of the specifications and performance for the SoC that was used for the PCIe testing. Two of these SoCs were connected via their PCI-Express interface and tested. This test setup is described and preliminary results are given in Section 3. Section 4 concludes.

2. ARM System on Chips

ARM System on Chips (SoCs) are low cost, energy efficient and high performance which has led to their extensive use in mobile devices. Several ARM platforms have been tested by the group at the University of the Witwatersrand, Johannesburg but only the specifications and test results for the Freescale i.MX6 quad-core ARM SoC is presented in Tab. 1 [Reed et al., 2014].

3. PCI-Express Pair Testing

PCI-Express throughput tests have been performed on a pair of Freescale i.MX6 quad-core ARM Cortex-A9 SoCs clocked at 1 GHz, located on Wandboard development boards [Wandboardorg 2012 Wandboard]. The results are presented in Tab. 2 and a photo of the custom test setup designed by the author is in Fig. 2. Three tests were run to ascertain the maximum data throughput that can be obtained from the i.MX6 SoC: a simple CPU based memcpy command and two Direct Memory Access (DMA) transfers, initiated by the Endpoint (EP) or slave and the Root Complex (RC) which is the host.

Table 1: CPU benchmark results and specifications
of the Freescale i.MX6Q Cortex-A9 SoC [Reed et al., 2014]

Core Revision	r2p2
Clock (MHz)	996
Cores	4
Feature Size (nm)	40
SP GFLOPS	5.12
DP GFLOPS	2.40
Load Power (W)	5.03
Idle Power (W)	2.02
Calc. Power (W)	3.01
DP GFLOPS/W	0.80
Ethernet (Mb/s)	470
PCIe (Gb/s)	5

Unfortunately the i.MX6Q SoC does not have a DMA unit on the PCIe controller and so the Image Processing Unit DMA unit was used instead. This is a workaround provided by the manufacturer.

Table 2: PCI-Express throughput results of a i.MX6 (Wandboard) pair

	CPU memcpy	**DMA (EP)**	**DMA (RC)**
Read (MB/s)	94.8 ± 1.1%	174.1 ± 0.3%	236.4 ± 0.2%
Write (MB/s)	283.3 ± 0.3%	352.2 ± 0.3%	357.9 ± 0.4%

The theoretical maximum throughput for the PCI-Express Gen 2 x1 link that was used is 500 MB/s. The best result is using DMA initiated by the RC but it is only 72% of the theoretical maximum. The RC-mode drivers are more optimized than the EP-mode drivers due to limited manufacturer support for EP-mode. The read results are lower than write because of overheads to

initiate the read. The PU architecture will take these differences into account and use a data push rather than a pull based approach.

4. Discussion, Conclusions and Future Work

Data stream computing, or more formally High Volume throughput Computing (HVC), is required for projects such as the LHC and SKA which produce enormous amounts of raw data. A general purpose ARM System on Chip based processing unit is being developed at the University of the Witwatersrand, Johannesburg which hopes to enable affordable and energy efficient HVC.

PCI-Express is superior to Ethernet in energy efficiency, I/O throughput and latency. Typical commodity ARM SoCs do not support Ethernet faster than 1 Gb/s however PCI-Express may be used for higher data throughput communications. Unfortunately, PCI-Express is not suitable for longer distance communications but the solution to this may be found in a PCI-Express to Ethernet bridge.

Initial throughput measurements presented for a pair of Freescale i.MX6 quad-core Cortex-A9 SoCs are 72 % of the theoretical maximum 500 MB/s for the available x1 link. Six of these SoCs would therefore be connected in parallel to provide 20 Gb/s throughput at a power consumption of less than 50 W. As a proof of concept the final Cortex-A9 prototype aims to provide 20 Gb/s aggregated throughput.

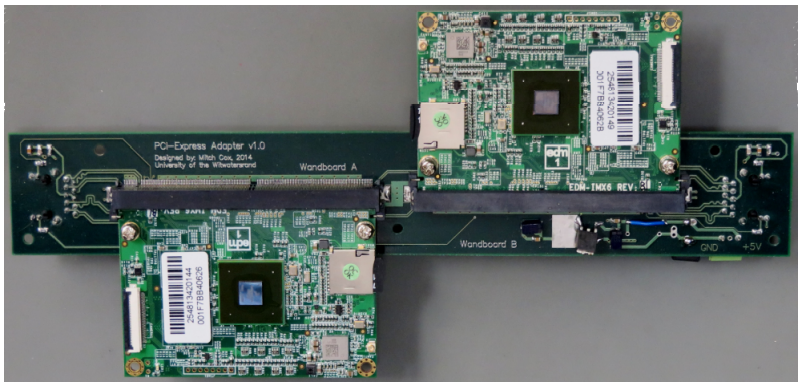


Fig. 2. PCI-Express test setup for a pair of i.MX6 SoCs (Wandboards)

The next stage of research by the author will be to test a small PCIe cluster of Cortex- A9 SoCs. The use of multiple energy efficient commodity ARM SoCs interconnected via PCI- Express and a single higher-end SoC for external communications via multiple 10 Gb/s Ethernet connections is theoretically well suited as a HVC Processing Unit.

Future big science experiments may be jeopardised by prohibitive data processing costs but the research presented in this paper, as well as future research and development of a HVC processing unit, may lead to a possible solution to this problem with its high data throughput, energy efficient and affordable computing capabilities.

References

Dursi J. Parallel I/O doesn't have to be so hard:The ADIOS Library Tech. rep. SciNet. 2012. URL: http://wiki.scinethpc.ca/wiki/images/8/8c/Adios-techtalk-may2012.pdf

Rajovic N. et al. Journal of Computational Science 4 439-443 ISSN 18777503. 2013. URL: http://www.sciencedirect.com/science/article/pii/S1877750313000148

Reed R. et al. A CPU Benchmarking Characterization of ARM Based Processors // The 6th International Conference: Distributed Computing and Grid-technologies in Science and Education (Dubna, Russia). 2014.

Szalay A. S. et al. ACM SIGOPS Operating Systems Review 44 71 ISSN 01635980. 2010. URL: http://dl.acm.org/citation.cfm?id=1740390.1740407

Wandboardorg 2012 Wandboard - Freescale i.MX6 ARM Cortex-A9 Opensource Community Development Board accessed: 18 February 2014 URL: http://www.wandboard.org/

Wikipedia. IEEE 802.3 — Wikipedia, The Free Encyclopedia accessed:16 September 2014 URL: http://en.wikipedia.org/wiki/IEEE_802.3

Zhan J. et al. High Volume Throughput Computing:Identifying and Characterizing Throughput. Oriented Workloads in Data Centers 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IEEE). 2012. PP 1712-1721 ISBN 978-1-4673-0974-5 URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6270846

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 511−516**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.272

Applications of on-demand virtual clusters to high performance computing

I. G. Gankevich[a], S. G. Balyan[b], S. A. Abrahamyan[c], V. V. Korkhov[d]

Saint Petersburg State University, University ave., 35, Peterhof, St. Petersburg, 198504, Russia

E-mail: [a]igankevich@cc.spbu.ru, [b]serob.balyan@gmail.com, [c]suro7@live.com, [d]vladimir@csa.ru

Получено 1 октября 2014 г.

Virtual machines are usually associated with an ability to create them on demand by calling web services, then these machines are used to deliver resident services to their clients; however, providing clients with an ability to run an arbitrary programme on the newly created machines is beyond their power. Such kind of usage is useful in a high performance computing environment where most of the resources are consumed by batch programmes and not by daemons or services. In this case a cluster of virtual machines is created on demand to run a distributed or parallel programme and to save its output to a network attached storage. Upon completion this cluster is destroyed and resources are released. With certain modifications this approach can be extended to interactively deliver computational resources to the user thus providing virtual desktop as a service. Experiments show that the process of creating virtual clusters on demand can be made efficient in both cases.

Применение создаваемых по требованию виртуальных кластеров в высокопроизводительных вычислениях

И. Г. Ганкевич, С. Г. Балян, С. А. Абраамян, В В. Корхов

*Санкт-Петербургский государственный университет,
Россия, 198504 г. Санкт-Петербург, Петергоф, Университетский просп., д. 35*

Виртуальные машины обычно ассоциируются с возможностью создавать их по требованию для предоставления клиентам разнородных веб-сервисов, однако, автоматическое создание виртуальных машин для запуска на них вычислений общего назначения на практике широко не используется. Такой сценарий использования виртуализации полезен в среде высокопроизводительных вычислений, где большинство ресурсов не потребляется разнородными сервисами, а используется для пакетной обработки данных. В этом случае для запуска каждого приложения создается отдельный кластер виртуальных машин, а запись выходных данных производится на сетевое хранилище. После того как приложение завершает свое выполнение, кластер уничтожается, высвобождая занятые вычислительные ресурсы. После определенных изменений данный подход может быть использован для предоставления виртуального рабочего стола в интерактивном режиме. Эксперименты показывают, что процесс создания виртуальных кластеров по требованию может быть эффективно реализован в обоих случаях.

The research was carried out using computational resources of Resource Center Computational Center of Saint Petersburg State University (T-EDGE96 HPC-0011828-001) and partially supported by Russian Foundation for Basic Research (project No. 13-07-00747) and Saint Petersburg State University (project No. 9.38.674.2013).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 511–516 (Russian).

Introduction

Virtualisation is employed for many components of computing system such as processor, memory, storage subsystem and network interface, however, virtualising all of them at once is not required in high performance computing where efficiency of the resulting computing system is of utmost importance. First, most of the virtualisation comes at a cost of slight performance decrease, which grows with the size of the application. Second, virtualisation of some components (i.e. network interfaces, processors) complexifies system's architecture without simplifying problem solution. This led to scarce use of virtualisation technologies in high performance computing; contrary to this we feel that there is at least one way of using it to simplify architecture of the system and decrease its maintenance effort.

This way consists of storing applications which run on cluster and their dependencies inside lightweight containers. Upon submitting a task containers are mounted on each host given by a resource manager and parallel application is executed inside each of them. The benefit of this approach is that a separate operating system and optimised libraries can be installed for each application without the need to alter host computer configuration. In addition to this, it is easy to maintain different versions of applications (due to licensing or compatibility problems) in different containers and to update them in contrast to common shared-folder-for-all-applications configuration where there is a lot of manual work. Finally, since container virtualisation does not imply processor virtualisation overheads this approach can be made efficient in terms of application performance.

The rest of the paper discusses related work (Section 1), describes system's architecture with containers (Section 2) and evaluates its efficiency on basis of real-world application and some synthetic tests (Section 3).

1 Related work

Research works on the subject of virtual clusters can be divided into two broad groups: works dealing with provisioning and deploying virtual clusters in high performance environment or GRID and works dealing with overheads of virtualisation. Works from the first group typically assume that virtualisation overheads are low and acceptable in high performance computing and works from the second group in general assume that virtualisation has some benefits for high performance computing, however, the authors are not aware of the work that touches both subjects in aggregate.

In [Chen, Wo, Li, 2009] authors evaluate overheads of the system for on-line virtual cluster provisioning (based on QEMU/KVM) and different resource mapping strategies used in this system and show that the main source of deploying overhead is network transfer of virtual machine images. To reduce it they use different caching techniques to reuse already transferred images as well as multicast file transfer to increase network throughput. Simultaneous use of caching and multicasting is concluded to be an efficient way to reduce overhead of virtual machine provisioning.

In [Ye K. et al, 2010] authors evaluate general overheads of Xen para-virtualisation compared to fully virtualised and physical machines using HPCC benchmarking suite. They conclude that an acceptable level of overheads can be achieved only with para-virtualisation due to its efficient inter domain communication (bypassing dom0 kernel) and absence of high L2 cache miss rate when running MPI programs which is common to fully virtualised guest machines.

In contrast to these two works the main principles of our approach can be summarised as follows. Do not use full or para-virtualisation of the whole machine but use virtualisation of selected components so that overheads occur only when they are unavoidable (i.e. do not virtualise processor). Do not transfer opaque file system images but mount standard file systems over the network so that only minimal transfer overhead can occur. Finally, amend standard task schedulers to work with virtual clusters so that no programming is needed to distribute the load efficiently. These principles are paramount to make virtualisation lightweight and fast.

2 System's configuration

The system comprises many standard components which are common in high performance computing, these are distributed parallel file system which stores home directories with experiment's input and output data, cluster resource scheduler which allocates resources for jobs and client programmes to pre- and post-process data; the non-standard component is network-attached storage exporting container's root files systems as directories. Linux Container technology (LXC) is used to provide containerisation, GlusterFS is used to provide parallel file system and TORQUE to provide task scheduling. The most recent CentOS Linux 7 is chosen to provide stable version of LXC (>1.0) and version of kernel which supports all containers' features. Due to limited number of nodes each of them is chosen to be both compute and storage node and every file in parallel file system is stored on exactly two nodes. Detailed hardware characteristics and software version numbers are listed in Table 1.

Table 1. Hardware and software components of the system

Component	Details	Component	Details
CPU model	Intel Xeon E5440	Operating system	CentOS Linux 7 (Core)
CPU clock rate (GHz)	2.83	Kernel version	3.10
No. of cores per CPU	4	LXC version	1.0.5
No. of CPUs per node	2	GlusterFS version	3.5.1
RAM size (GB)	4	TORQUE version	5.0.0
Disk model	ST3250310NS	OpenMPI version	1.6.4
Disk speed (rpm)	7200	IMB version	4.0
No. of nodes	12	OpenFOAM version	2.3.0
Interconnect speed (Gbps)	1		

Creating virtual cluster in such environment requires the following steps. First, a client submits a task requesting particular number of cores. Then according to distribution of these cores among compute nodes a container is started on each node from the list with SSH daemon as the only program running inside it. Here there are two options: either start containers with network virtualisation (using “macvlan” or “br” LXC network type) and generate sufficient number of IP addresses for the cluster or use host network name space (“none” LXC network type) and generate only the port number to run ssh daemon on. The next step is to copy (possibly amended) node file from host into the first container and launch submitted script inside it. When the script finishes its work SSH daemon in every container is killed and all containers are destroyed.

For this algorithm to work as intended client's home directory should be bind-mounted inside the container before launching the script. Additionally since some MPI programmes require “scratch” directories on each node to work properly, container's root file system should be mounted in copy-on-write mode, so that all changes in files and all the new files are written to host's temporary directory and all unchanged data is read from read-only network-mounted file system; this can be accomplished via Union or similar file system and that way application containers are left untouched by tasks running on the cluster.

To summarise, only standard Linux tools are used to build the system: there are no opaque virtual machines images, no sophisticated full virtualisation appliances and no heavy-weight cloud computing stacks in this configuration.

3 Evaluation

To test the resulting configuration OpenMPI and Intel MPI Benchmarks (IMB) were used to measure network throughput and OpenFOAM was used to measure overall performance on a real-world application.

The first experiment was to create virtual cluster, launch an empty (/bin/true) MPI programme in it and compare execution time to ordinary physical cluster. To set this experiment up in the container the same operating system and version of OpenMPI as in the host machine was installed. No network virtualisation was used, each run was repeated several times and the average was displayed on the graph (Fig. 1). The results show that a constant overhead of 1.5 second is added to every LXC run after the 8th core: one second is attributed to the absence of cache inside container with SSH configuration files, key files and libraries in it and other half of the second is attributed to the creation of containers as shown in Figure 2. The jump after the 8th core marks bounds of a single machine which means using network for communication rather than shared memory. The creation of containers is fully parallel task and takes approximately the same time to complete for different number of nodes. Overhead of destroying containers was found to be negligible and was combined with "mpirun" time. So, usage of Linux containers adds some constant overhead to the launching of parallel task depending on system's configuration which is split between creation of containers and filling the file cache.

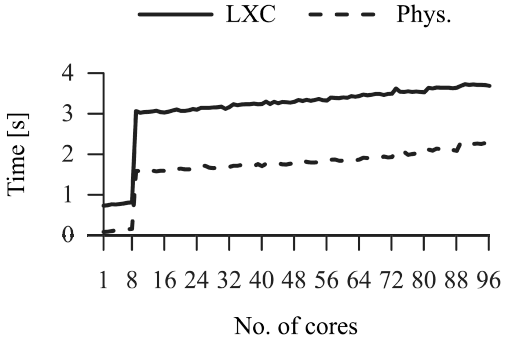


Fig. 1. Comparison of LXC and physical cluster performance running empty MPI programme

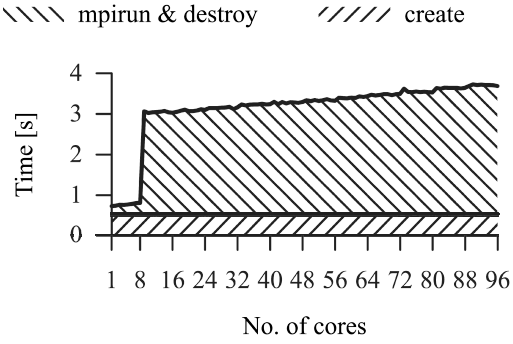


Fig. 2. Breakdown of LXC empty MPI programme run

The second experiment was to measure performance of different LXC network types using IMB suite and it was found that the choice of network virtualisation greatly affects performance. As in the previous test container was set up with the same operating system and the same IMB executables as the host machine. Network throughput was measure with "exchange" benchmark and displayed on the graph (Fig. 3). From the graph it is evident that until 2^{14} bytes message size the performance is approximately the same for all network types, however, after this mark there is a dip in performance of virtual ethernet. It is difficult to judge where this overhead comes from: some studies report that under high load performance of bridged networking (veth is always connected to the bridge) is decreased [James, 2004], but IMB does not have high load on the system. Additionally, the experiment showed that as expected throughput decreases with the number of cores due to synchronisation overheads (Fig. 4).

The third and the last experiment dealt with real-world application performance and for this role the OpenFOAM was chosen as the complex parallel task involving large amount of network communication, disk I/O and high CPU load. The dam break RAS case was run with different number of cores (total number of cores is the square of number of cores per node) and different LXC network types and the average of multiple runs was displayed on the graph (Fig. 5). Measurements for 4 and 9 cores were discarded because there is a considerable variation of execution time for these numbers on physical machines. From the graph it can be seen that low performance of virtual ethernet decreased final performance of OpenFOAM by approximately 5-10% whereas "macvlan" and "none" performance is close to the performance of physical cluster (Fig. 6). So, the choice of network type is the main factor affecting performance of parallel applications running on virtual clusters and its overhead can be eliminated by using "macvlan" network type or by not using network virtualisation at all.

To summarise, there are two main types of overheads when using virtual cluster: creation overhead which is constant and small compared to average time of a typical parallel job and network overhead which can be eliminated by not using network virtualisation at all.

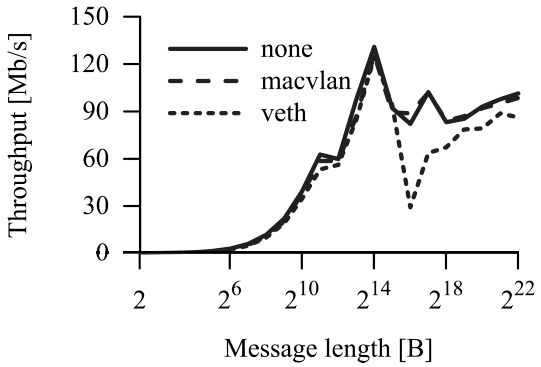


Fig. 3. Average throughput of "exchange" MPI benchmark

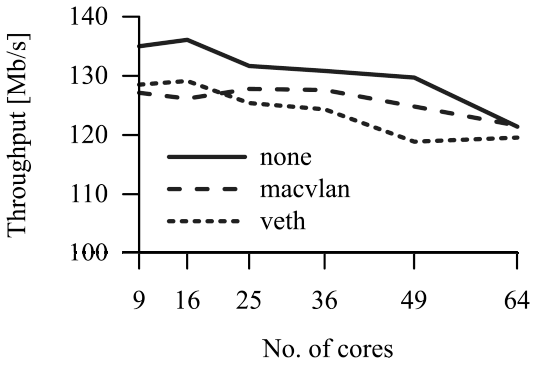


Fig. 4. Throughput for 16Kb messages

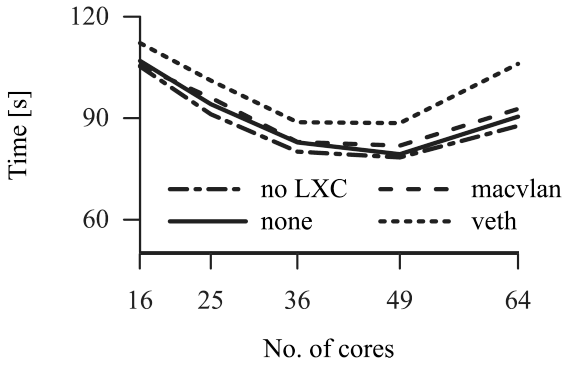


Figure 5. Average performance of OpenFOAM with different LXC network types

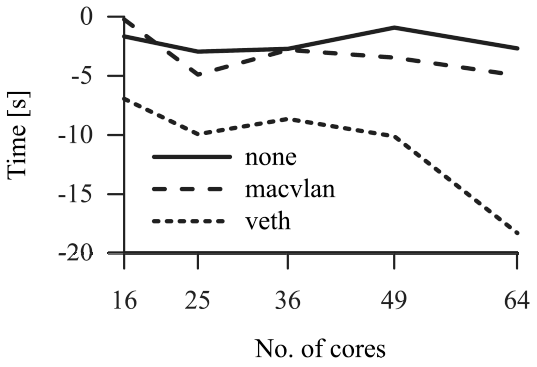


Figure 6. Difference of OpenFOAM performance on physical and virtual clusters. Negative numbers show slowdown of virtual cluster

Conclusions and future work

Presented approach for creating virtual clusters from Linux containers was found to be efficient and its performance comparable to ordinary physical cluster: not only usage of containers does not incur processor virtualisation overheads but also network virtualisation overheads can be totally removed if host's network name space is used and network bandwidth saved by automatically transferring only those files that are needed through network-mounted file system rather than the whole images. From the point of view of system's administrator storing each HPC application in its own container makes version and dependencies control easy manageable and their configuration does not interfere with the configuration of host machines and other containers.

Acknowledgements

Authors would like to thank Andrey Zarochentsev for his advice on using UnionFS to make root file system writeable which was the turning point to eliminate existing over-sophisticated solution.

References

Chen Y., Wo T., Li J. An efficient resource management system for on-line virtual cluster provision // Cloud Computing, 2009. CLOUD'09. IEEE International Conference on. — IEEE, 2009. — С. 72–79.

James T. Y. Performance evaluation of Linux Bridge // Telecommunications System Management Conference. — 2004.

Ye K. et al. Analyzing and modeling the performance in Xen-based virtual cluster environment // High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on. — IEEE, 2010. — С. 273–280.

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 517–520**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.04

Efficient processing and classification of wave energy spectrum data with a distributed pipeline

I. G. Gankevich [a], A. B. Degtyarev [b]

Saint Petersburg State University, University ave. 35, Peterhof, St. Petersburg, 198504, Russia

E-mail: [a]igankevich@cc.spbu.ru, [b]deg@csa.ru

Получено 1 октября 2014 г.

Processing of large amounts of data often consists of several steps, e.g. pre- and post-processing stages, which are executed sequentially with data written to disk after each step, however, when pre-processing stage for each task is different the more efficient way of processing data is to construct a pipeline which streams data from one stage to another. In a more general case some processing stages can be factored into several parallel subordinate stages thus forming a distributed pipeline where each stage can have multiple inputs and multiple outputs. Such processing pattern emerges in a problem of classification of wave energy spectra based on analytic approximations which can extract different wave systems and their parameters (e.g. wave system type, mean wave direction) from spectrum. Distributed pipeline approach achieves good performance compared to conventional "sequential-stage" processing.

Keywords: distributed system, big data, data processing, parallel computing

Эффективная обработка и классификация энергетических спектров морского волнения на основе распределенного вычислительного конвейера

И. Г. Ганкевич, А. Б Дегтярев

Санкт-Петербургский государственный университет, Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

Обработка больших массивов данных обычно происходит в несколько последовательно выполняемых этапов, таких как пред- и постобработка, после каждого из которых промежуточные данные записываются на диск; однако, для каждой задачи этап предварительной обработки может отличаться, и в таком случае непосредственная передача данных по вычислительному конвейеру от одного этапа (звена) к другому будет более эффективным с точки зрения производительности решением. В более общем случае некоторые этапы можно разделить на параллельные части, сформировав таким образом распределенный вычислительный конвейер, каждое звено которого может иметь несколько входов и выходов. Такой принцип обработки данных применяется в задаче о классификации энергетических спектров морского волнения, которая основана на аппроксимациях, позволяющих извлекать параметры отдельных систем волн (тип волн, генеральное направление волн и т. п.). Система, построенная на этом принципе показывает более высокую производительность по сравнению с часто применяемой поэтапной обработкой данных.

Ключевые слова: распределенные системы, большие данные, обработка данных, параллельные вычисления

The research was carried out using computational resources of Resource Center Computational Center of Saint Petersburg State University (T-EDGE96 HPC-0011828-001) and partially supported by Russian Foundation for Basic Research (project No. 13-07-00747) and Saint Petersburg State University (project No. 9.38.674.2013 and 0.37.155.2014).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 517–520 (Russian).

Introduction

The problem of classification of wave energy spectra is both data- and compute-intensive which makes it on one hand amenable to data-centric programming approaches like Hadoop and on the other hand to parallel programming techniques. In the "mapping" phase spectra should be pre-processed and converted to some convenient format and in the "reduction" phase resulting spectra are classified using genetic optimisation algorithm. These steps represent general algorithm for data processing with Hadoop, however, classification algorithm is itself parallel which makes it difficult to program in Java (the language in which Hadoop programmes are usually written). Therefore, we feel that Hadoop is not the most efficient way to solve the problem and a distributed programme which mimics useful Hadoop behaviour should be used instead.

This work is a short preview of a alternative technological framework being developed and it is compared to Hadoop implementation only.

1 Implementation

The NDBC dataset[1] consists of spectra which are sorted by year and station where measurements were made. Data for each spectrum is stored in five variables which are used to reconstruct original frequency-directional spectrum with the following formula:

$$S(\omega,\theta)=\frac{1}{\pi}\left[\frac{1}{2}+r_1\cos(\theta-\alpha_1)+r_2\sin\left(2(\theta-\alpha_2)\right)\right]S_0(\omega).$$

Here ω denotes frequency, ϑ — wave direction, $r_{1,2}$ and $\alpha_{1,2}$ are parameters of spectrum decomposition and S_0 is the non-directional spectrum [Marshall D. Earle, 1996]; values of $r_{1,2}$, $\alpha_{1,2}$, S_0 are acquired through measurements. Detailed properties of the dataset used in evaluation are listed in Table 1.

Table 1. Dataset properties

Dataset size	144MB
Dataset size (uncompressed)	770MB
Number of wave stations	24
Time span	3 years (2010–2012)
Total number of spectra	445422

The algorithm of processing spectra is as follows. First, current directory is recursively scanned for input files. All directories are recursively distributed to processing queues of each machine in the cluster. Processing begins with joining corresponding measurements for each spectrum variables into a tuple which is subsequently classified by a genetic algorithm (this algorithm is not discussed in the paper and in fact can be replaced by any other suitable classification algorithm). While processing results are gradually copied back to the machine where application was executed and when the processing is complete the programme terminates. The resulting implementation is shown in Figure 1.

Directory structure can be arbitrary and the only thing it serves is to distribute the data, however, files containing corresponding measurements should be placed in a single directory so that no joining of variables residing in different machines can happen. In this test spectra were naturally sorted into directories by year and station.

The feature which makes this implementation different from other similar approaches is that both processors and disks work in parallel throughout the programme execution. Such behaviour is achieved with assigning a separate thread (or thread pool) for each device and placing tasks in the queue for the corresponding device in this pool. As tasks that read from the disk complete they pro-

[1] http://www.ndbc.noaa.gov/dwa.shtml

duce tasks for CPUs to process this data and place them into the processor task queue. In similar way when data processing tasks complete they place tasks to write the data into the disk task queue. In similar vein via a separate task queue network devices transmit the data to a remote node. So, each device has its own thread (or thread pool) and all of them work in parallel by placing tasks in each other's task queues. Since tasks "flow" from one queue to another and queues can reside on different machines this approach is called distributed pipeline.

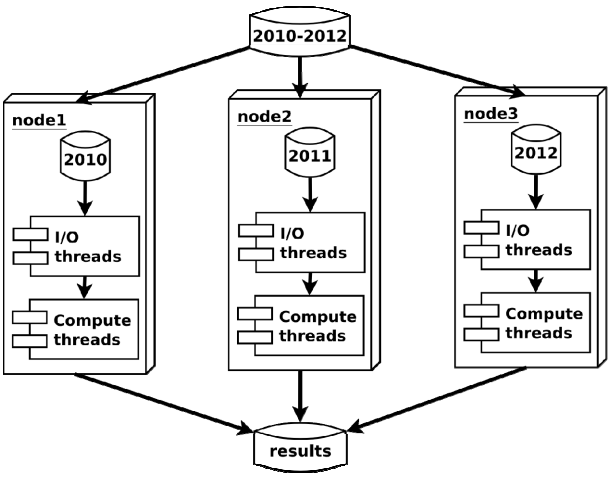


Fig. 1. Implementation diagram for distributed pipeline

2 Evaluation

The system setup which was used to test the implementation consisted of commodity hardware and open-source software (Table 2) and evaluation was divided into two stages. In the first stage Hadoop was installed on each node of the cluster and was configured to use host file system as a source of data so that performance of parallel file system which is used by default in Hadoop can be factored out from the comparison. To make this possible the whole dataset was replicated on each node and placed in the directory with the same name. In the second stage Hadoop was shut down and replaced by newly developed application and dataset directories were statically distributed to different nodes to nullify the impact of parallel file system on the performance.

Table 2. Hardware and software components of the system

Component	Details
CPU model	Intel Q9650
CPU clock rate (GHz)	3.0
No. of cores per CPU	4
No. of CPUs per node	1
RAM size (GB)	4
Disk model	ST3250318AS
Disk speed (rpm)	7200
No. of nodes	3
Interconnect speed (Mbps)	100

Component	Details
Operating system	Debian Linux 7.5
Hadoop version	2.3.0
GCC version	4.7
Compile flags	-std=c++11

In the test it was found that Hadoop implementation has low scalability and maximum performance of approx. 1000 spectra per second and alternative implementation has higher scalability and maximum performance of approx. 7000 spectra per second (Fig. 2). The source of Hadoop inefficiency was found to be temporary data files which are written to disk on each node. These files represent sorted chunks of the key-value array and are part of implementation of merge sort algorithm used to

distribute the keys to different nodes. For NDBC dataset the total size of these files exceeds the size of the whole dataset which appears to be the consequence of Hadoop not compressing intermediate data (the initial dataset has compression ratio of 1:5, see Table 1). So, the sorting algorithm and careless handling of compressed data led to performance degradation and inefficiency of Hadoop for NDBC dataset.

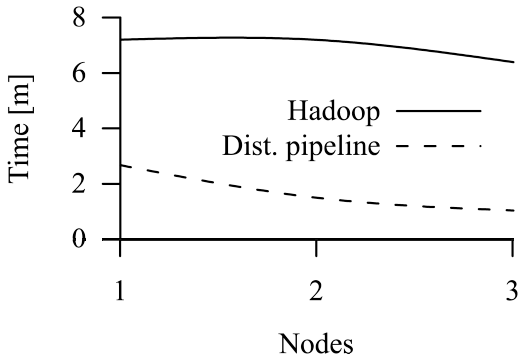


Fig. 2. Performance of Hadoop and distributed pipeline implementations

The sorting is not needed to distribute the keys and in the alternative implementation directory hierarchy is used to determine machine for reduction. For each directory a separate task is created which subsequently creates tasks for each sub-directory and each file. Since each task can interact with its parent when the reduction phase is reached reduction tasks are created on the machines where parents were executed previously.

Conclusions and future work

No redundant sorting nor any kind of temporary files are used in the alternative implementation which allows it to scale well and show better performance compared to Hadoop approach. The future work is to incorporate dynamic distribution of files to hosts and fault tolerance into the implementation.

References

Marshall D. Earle. Nondirectional and Directional Wave Data Analysis Procedures (NDBC Technical Document 96-01), 1996, URL: http://www.ndbc.noaa.gov/wavemeas.pdf.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 521–527

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.023

An interactive tool for developing distributed telemedicine systems

V. P. Guskov[1,a], D. E Gushchanskiy[2], N. V. Kulabukhova[2], S. Abrahamyan[3], S. Balyan[3], A. B. Degtyarev[2], A. V. Bogdanov[2]

[1]Herzen State Pedagogical University of Russia, Moika river embankment 48, St. Petersburg, 191186, Russia
[2] Saint Petersburg State University, University ave., 35, Peterhof, St. Petersburg, 198504, Russia
[3] State Engineering University of Armenia (Polytechnic), Teryan Str. 105, Yerevan, 0009 Republic of Armenia

E-mail: [a]vadim.p.guskov@gmail.com

Получено 1 октября 2014 г.

Getting a qualified medical examination can be difficult for people in remote areas because medical staff available can either be inaccessible or it might lack expert knowledge at proper level. Telemedicine technologies can help in such situations. On one hand, such technologies allow highly qualified doctors to consult remotely, thereby increasing the quality of diagnosis and plan treatment. On the other hand, computer-aided analysis of the research results, anamnesis and information on similar cases assist medical staff in their routine activities and decision-making.

Creating telemedicine system for a particular domain is a laborious process. It's not sufficient to pick proper medical experts and to fill the knowledge base of the analytical module. It's also necessary to organize the entire infrastructure of the system to meet the requirements in terms of reliability, fault tolerance, protection of personal data and so on. Tools with reusable infrastructure elements, which are common to such systems, are able to decrease the amount of work needed for the development of telemedicine systems.

An interactive tool for creating distributed telemedicine systems is described in the article. A list of requirements for the systems is presented; structural solutions for meeting the requirements are suggested. A composition of such elements applicable for distributed systems is described in the article. A cardiac telemedicine system is described as a foundation of the tool

Keywords: artificial intelligence, decision support systems, distributed telemedicine systems, interactive tool, remote consultations

The work is partly supported by Resource Center "Computer Center SPbSU" and project 9.38.674.2013 of Saint-Petersburg State University.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 521–527 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 521–527

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

Интерактивный инструментарий для распределенных телемедицинских систем

В. П. Гуськов[1], Д. Е. Гущанский[2], Н. В. Кулабухова[2], С. А. Абраамян[3], С. Г. Балян[3], А. Б. Дегтярев[2], А. В. Богданов[2]

[1]Российский государственный педагогический университет им. А. И. Герцена,
Россия, 191186, г. Санкт-Петербург, набережная реки Мойки, д. 48
[2]Санкт-Петербургский государственный университет,
Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35
[3]Государственный инженерный университет Армении (Политехник),
Республика Армения, 375009, г. Ереван, ул. Терьяна, д. 105

Для жителей удалённых районов часто может составлять проблему прохождение квалифицированного медицинского обследования. Доступный медицинский персонал может отсутствовать или не обладать экспертными знаниями достаточного уровня. Помочь в такой ситуации могут телемедицинские технологии. С одной стороны, такие технологии позволяют врачам высокой квалификации оказывать удалённые консультации, повышая тем самым качество постановки диагноза и составления плана лечения. С другой стороны, средства автоматизированного анализа результатов проведённых исследований, анамнеза и информации об аналогичных случаях помогают облегчить выполнение рутинных действий и оказать медицинскому персоналу поддержу в принятии решений.

Создание телемедицинской системы для конкретной предметной области — это трудоёмкий процесс. Не достаточно подобать подходящих специалистов и заполнить базу знаний аналитического модуля. Необходимо также организовать всю инфраструктуру системы, удовлетворяя предъявляемые требования по надёжности, отказоустойчивости, защите персональных данных и так далее. Снизить трудоёмкость разработки телемедицинских комплексов может инструментарий, содержащий многократно используемые инфраструктурные элементы, общие для систем такого рода.

В данной работе описан интерактивный инструментарий для создания распределённых телемедицинских систем. Приводится список требований, предъявляемый к получаемым системам, и архитектурные решения, позволяющие удовлетворить эти требования. В качестве примера применения созданного инструментария описывается кардиологическая телемедицинская система.

Ключевые слова: интерактивный инструментарий, искусственный интеллект, распределённые телемедицинские системы, системы поддержки принятия решений, удалённые консультации

Introduction

In a narrow sense, telemedicine is a procedural complex for transporting medical data at a distance via computer technologies and high-trunk connection. However, the modern state of information technologies allows considering telemedicine as an interdisciplinary field focused on creation of the unified information space that contains ordered resources and functions of the different subjects of healthcare.

The necessity of telemedicine development is acknowledged by the leading countries of the world. For instance, telemedicine technologies are widely used by USA, Norway, Greece, Great Britain and many others [Телемедицинские центры и ресурсы..., 2014]. Projects of telemedicine networks formations are considered among the most important medical projects funded by European Community. They are particularly important for the countries with vast remote areas, where traditional medicine availability is lacking [К вопросу развития телемедицины..., 2013]. Considering the large geographic extent of Russian Federation, the significance of such problems is growing continuously.

While a huge variety of telemedicine systems is being used for different purposes: remote surgery, videoconferences and consultations, medical e-learning and others, the most important branch of telemedicine in Russia is providing highly-qualified medical treatment for people regardless of their location. Appropriate medical services should be available in rural hospitals, towns with a lack of specialists, remote settlements without any medical institutions, during expeditions and so on.

With the growth of IT-infrastructure and network speed, the need for efficient distributed medical data processing increases. It creates a problem of the medical data protection, which should be handled with personal rights and state security in mind [Рыжов, 2011].

Commonly, remote consulting is considered as a basic function of telemedicine systems. The function includes deferred and real-time consulting via text messages or video, medical data transmission and analysis.

Remote consulting can be extended to videoconferences. Vast experience in holding videoconferences, even international ones, proves the efficiency of the approach. Russia is systematically involved in the number of videoconferences such as Moscow — San Francisco, Moscow — Geneva, Arkhangelsk — Oslo etc [Юсупов и др., 2002].

Another steadily growing part of telemedicine is the development and use of electronic health records (EHR). EHR are the key part of implementation of distributed health technologies. They grant easy access to medical history and thereby increase the efficiency of medical consulting and treatment. Consolidation of databases which belong to different medical facilities provides wide range of possibilities for statistical analysis and data mining.

In addition to consulting and medical data handling, medical e-learning is a significant component of the telemedicine. According to the analysis of different telemedicine projects from various countries, 48% of the projects are related to medical educational purposes [К вопросу развития телемедицины..., 2013]. The remote consulting, information support of general practitioners and medical scientists, lifelong distance learning determine the complete solution for problems of telemedicine application by improving the training of specialists. Periodical issues of the journal of medical education ‘Academic Physician & Scientist’ about the topic could be considered as a proof of large interest in educational aspects of telemedicine.

Synthetic technologies of medical data processing are rapidly being developed as a part of telemedicine. They consist of different kinds of information, expert and intelligent systems. Their aggregating factor is an AI-based knowledge extraction from medical data and providing this knowledge in the form of decision support systems [Нечаев, Дегтярев, 2011]. Automated consultations using various diagnostic systems are considered to be the main focus of such systems application. Their purpose is to support decision making about appropriate treatment on real cases, utilizing achievements in theoretical and practical techniques of AI. A lot of factors and the complexity of interactions during medical decision making are the reason why medicine is one of the most difficult areas to apply automatic logic inference [Богданов и др., 2003]. There may be a number of reasons:

The complexity of medical knowledge impedes problem solving because of lack of standardization in terminology, formats and measurement scales. Diagnostic system encodings are becoming more and more universal lately, however the extensive range of symptoms, data acquisition formats and organization of records storage still remain a matter of personal preference;

Flexible and easy to use computer methods of medical knowledge representation have not been created yet. The same could be told about flexible algorithms of inference;

Large amount and complexity of medical knowledge result in high requirements on computer systems for its storing and processing. So far, such requirements could have only been met using mainframes with different 'intelligent' terminals, which have various processing and visualization speed;

There is no any 'driving force' capable to encourage the spread of such computer systems and their inclusion into medical practice.

Medical theory and practice are quite complex. The lack of clear definitions of deceases and conditions of the organism, parallel classifications, various and sometimes doubtful measurement scales complicate the work of physicians. The effective telemedicine usage requires development of strict encoding and diagnostic standards. The unified terminology is important for knowledge formalization for intelligent systems. The standardization will help to solve the problem of introducing new technologies and defining knowledge bases for new generation of medical information systems more precisely.

The main requirements for the development of decision support systems in medical applications are associated with flexibility and ability to evolve and provide practical advices in the environment with information heterogeneity. Hierarchical system pattern, the possibility of connecting networks, operational reliability and survivability as well as continuity and the ability to upgrade continuously make such systems extremely complex both in development and maintenance.

Creating a telemedicine system for a particular domain is a laborious process. It is not enough to find qualified medical experts and to fill the knowledge base of the analytical module. It is also necessary to organize all infrastructure of the system to meet the requirements in terms of reliability, fault tolerance, protection of personal data and so on. The motivation of current work is to create a toolkit for a broad range of telemedicine systems to decrease the amount of work needed for the development of telemedicine systems.

The research foundation

The idea of creating tools for development of distributed telemedicine systems is based on experience with ADEPT-C software package. ADEPT-C (Advanced Data Evaluation Parallel Tools - in Cardiology) is designed to ensure the correct functionality of a RTCES (Real-Time Cardiology Expert System) [Bogdanov et al., Volume 1, Issue 2, 2006]. Its purpose is to arrange data collection of remote patients' statuses, formation of EHRs and decision making support for diagnosing physicians in the form of automated consultations. It implements a software shell for a cardiology real-time expert system designed for operational control and medical consulting of patients with cardiovascular diseases who are at a considerable distance from the leading medical centers. ADEPT-C operates on client-server architecture and uses multiprocessor supercomputers as its core. ADEPT-C toolkit is the basis for creating a new generation of RTCES functioning on high performance supercomputers [Bogdanov et al., Volume 1, Issue 3, 2006]. The system includes components which can be implemented as reusable components and used as a basis in creating modern distributed telemedicine systems in specific problem areas, thus it served as a foundation of the present work. Below is the short description of the system functionality [Bogdanov et al., Volume 1, Issue 1, 2006].

The RTCES is intended for use only by persons with special medical training and system authorization. These persons are divided into the following categories:

- Physician generalists. The category includes primary care physicians, family physicians, emergency room doctors and medical staff providing medical stations in organizations with high requirements on workers' health (e.g. airports, railroad stations, etc).

- Specialists. This category contains the problem area specialists from clinics and hospitals. The main difference from the first group is their expertise in identification of specific symptoms, which are the base of the differential diagnosis.
- Experts. This group consists of leading specialists in the medicine problem area who have mastered their diagnosis skills and are capable of making additional changes to the knowledge base of the system.

According to the three groups of users, the system allows performing the following tasks:

- Automated anamnesis collection and conducting EHR of a personalized patient, including his or her passport and insurance information, results of the current survey and inspection, medical devices data, the diagnosis;
- Automated remote consulting — the decision making support for the attending physician (generalist) in order to create a further diagnosis;
- Automated differential diagnosis — the decision making support for the attending specialist in order to establish a final diagnosis. This task is performed based on the results of an automated consultation;
- Real-time remote expert consulting for an attending physician (performed on the results of automated consultations in ambiguous cases);
- Education and training of physicians.

ADEPT-C software package is designed to accomplish the following tasks:

- Allowing users (based on their permissions) to access to a single database containing information about patients.
- Organization of rule-based inference for the differential diagnosis of cardiac diseases.
- Stochastic modeling of human heart electrical activity in order to detect and diagnose its pathological dynamics.
- Modification and adoption of new information into the database and the knowledge base of RTCES based on the results of the current system usage.
- Protection of personal data during their transfer between components of the complex.

The fundamentally essential part of the telemedicine complex organization is knowledge and experience transfer from leading research institutions, therapeutic and diagnostic centers to physicians and other specialists who are far away. Interconnection of computer systems into a single telecommunication network under the multilevel structure based on a telemedicine resource center allows implementation of knowledge integration and rapid transfer of structured data between separate contributors for the further usage in diagnosis and treatment.

Architecture of the system

There are four main components in the designed interactive tool (see figure 1). The first is a facade — component that is responsible for interaction with external environment. It provides API for creating client applications for specific problem areas, e.g. services of data access and consultation requests. In addition, the facade contains modules of encrypted data transfer and user authorization and authentication.

The second component is the remote consultation organization module. It maintains the list of available specialists and a queue of patients, for which diagnosis is required. When doctor is ready for the next case, the module chooses patient from the top of the priority queue and helps to establish connection between them.

The medical data storage is the third toolkit component. The main part of it is the distributed database for history of cases. The tool user can deploy the database nodes on his hardware and configure the structure of stored EHRs. Furthermore, the component contains API for integration of external data sources. Records from extraneous data warehouses should be converted for the further usage in the system.

The last component — automatic consultation module — can be seen on the figure 2. The central part of that module is the inference engine. Its function is to provide recommendations by means of

logical reasoning. The engine uses information from different sources. First, it has access to knowledge base filled in by problem area experts. Using rules from knowledge base and symptoms from working memory, the reasoner provides the most possible diagnosis. Additional information can be extracted by statistical analysis of anonymous records from data storage. Furthermore, the module contains procedure components registry. External procedures can be integrated into the system using this registry. Procedure can add extra facts to working memory and precise existed symptoms.

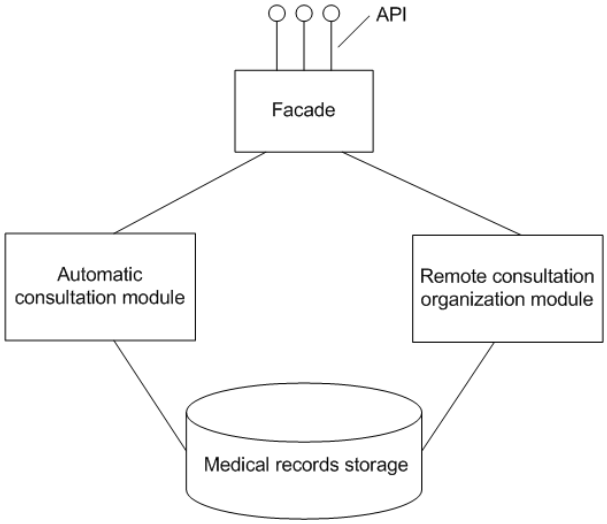


Fig. 1. Common architecture

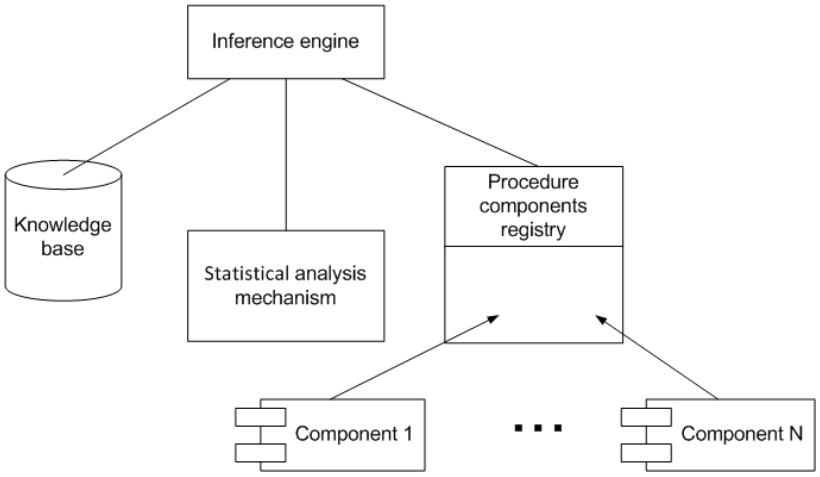


Fig. 2. Automatic consultation module

Summarizing the previous topics, the list of the interactive tool functions is presented below:

- Medical data storage;
- Encrypted data transfer;
- User authorization and authentication;
- Client API;
- Inference engine;
- Distributed knowledge base;
- Statistical analysis tool;
- Procedure component registry.

These activities are useful for creation of telemedicine system infrastructure. If the tool takes responsibility to provide these functions, it will be easier to construct the telemedicine system for specific problem area.

Conclusion

The present work is an initial stage for future development. Creation of such tools allows faster implementation of functionalities of information systems for telemedicine purposes. A fast and easy way to create telemedicine systems can boost their use in healthcare and greatly improve the quality of life.

References

К вопросу развития телемедицины в Республике Армения [Текст] / М. Н. Авакян [и др.] // Медицинское образование и профессиональное развитие. — 2013. — № 4. — С. 42–54.

Нечаев Ю. И., Дегтярев А. Б. Интеллектуальные системы: концепция и приложения. — СПб.: Изд-во С.-Петерб. ун-та, 2011. — 269 с.

Богданов А. В., Бухановский А. В., Вальденберг А. В., Дегтярев А. Б., Нечаев Ю. И. Информационно-аналитическая система в области телемедицины. Патент на изобретение RUS 2251965 29.01.2003.

Рыжов Р. С. Актуальные проблемы правового обеспечения накопления конфиденциальной информации о гражданах в телемедицине // Теория и практика общественного развития. — 2011. — №7. — С. 247–249.

Юсупов Р. М., Полонников Р. И., Дюк В. А., Блажис А. К., Кувакин В. И., Иванов А. Ю., Воробьев О. В., Сотников А. Д. Развитие телемедицины на Северо-Западе России // Труды СПИИРАН. Вып. 1, т. 1 — СПб: СПИИРАН, 2002.

Телемедицинские центры и ресурсы Internet по телемедицине. Функции телемедицинских центров. 2014. fzoz.ru/articles/informatsionnye-tekhnologii-dlya-vracha-glava-26-telemeditsinskie-tsentry-i-resursy-interne

Bogdanov A., Degtyarev A., Nechaev Y., Valdenberg A. Designing a High-Performance Telemedicine System. // Healthcare IT Management, Volume 1, Issue 2, Euromedical Communications NV, Brussels, Belgium, 2006 — P. 30–32.

Bogdanov A., Degtyarev A., Nechaev Y., Valdenberg A. Designing a High-Performance Telemedicine System, Healthcare IT Management, Volume 1, Issue 3, Euromedical Communications NV, Brussels, Belgium, 2006. — P. 31–34.

Bogdanov A., Degtyarev A., Nechaev Y., Valdenberg A. Designing a High-Performance Telemedicine System, Healthcare IT Management, Volume 1, Issue 1, Euromedical Communications NV, Brussels, Belgium, 2006. — P. 29–32.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 529–532

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.02

Visualization of work of a distributed application based on the mqcloud library

O. O. Iakushkin, V. M Grishkin

Saint Petersburg State University, University ave. 35, Peterhof, St. Petersburg, 198504, Russia

E-mail: [a] Oleg.jakushkin@gmail.com

Получено 10 октября 2014 г.

Abstract. — Independent components communicating with each other due to complex control make the work of complex distributed computer systems poorly scalable within the framework of the existing communication middleware. Two major problems of such systems' scaling can be defined: overloading of unequal nodes due to proportional redistribution of workload and difficulties in the realization of continuous communication between several nodes of the system. This paper is focused on the developed solution enabling visualization of the work of such a dynamical system.

Keywords: distributed systems, cloud computing, services, monitoring, networking, peer-to-peer

Визуализация работы распределенного приложения на базе библиотеки mqcloud

О. О. Якушкин, В. М. Гришкин

Санкт-Петербургский государственный университет, Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

Независимые компоненты, взаимодействующие между собой при помощи комплексного управления, делают работу сложных распределенных вычислительных систем плохо масштабируемой в рамках имеющегося промежуточного коммуникационного программного обеспечения. Можно выделить две основные проблемы масштабирования таких систем: перегрузка неравноценных узлов из-за равномерного перераспределения нагрузки и сложности в реализации продолжительного взаимодействия нескольких узлов системы. В данной работе мы рассмотрели созданное решение позволяющее обеспечивать визуальное отображение работы такой динамической системы.

Ключевые слова: сетевые распределенные вычисления, облачные вычисления, сервисные системы, сети передачи данных, мониторинг, одноранговые сети

The work has been done with partial support of SPbU grant 9.38.674.2013, SPbU grant 9.37.157.2014 and RFBR grant No. 13-07-747. The research has been done using the computing resources of the resource center "SPbU Computing Facility" (http://cc.spbu.ru).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 529–532 (Russian).

Introduction

Modeling the architecture of a distributed system plays a fundamental role in terms of bottleneck analysis, development from the perspective of horizontal scaling as well as protection of the presented system from external factors [Degtyarev et al., 2010; Bogdanov et al., 2006; Bogdanov et al., 2010]. This paper is focused on the creation of a means of interactive monitoring of the distributed network architecture in a service system. Means of interactive monitoring of various network topologies are studied here. The existing solutions are evaluated and the necessity of research required to develop our own ones is substantiated as well. The developed solution and its qualitative characteristics are described.

In addition to the existing internode communication library [Iakushkin, Grishkin, 2014.], a statistics presentation system for the nodes involved in communication has been developed. The existing means of data presentation in similar systems have been studied and a way of presenting the aggregated data of network statistics has been developed. Then an application with an option of displaying the system's current operation and tracing the changes dynamically has been created. The system has been launched using the computer powers of the resource center "SPbU Computing Facility".

The assigned development and testing tasks have been accomplished; the developed system successfully copes with the assigned tasks within the framework of the current stage of its development.

In future this system will be applicable during extension of the service base in the course of debugging and monitoring of the services of the systems of SPbU research projects No. 01201453381 and No. 01201453375. The system has an open interface and can be used by other projects for the sake of debugging network communication between distributed system components.

While working on a prototype of the communication system for monitoring and diagnostics of the state of cultural heritage sites as well as for usage in particular simulation experiments in modeling hydrophysical and hydrometeorological processes in the waters of the Baltic Sea within the framework of SPbU research projects No. 01201453381 and No. 01201453375, we have developed a non-broker transportation system enabling dynamical workload redistribution during nodes communication, with the support of various communication means.

Task description

Peer-to-peer (P2P) communication libraries for message exchange, such as ZeroMQ and NanoMsg, contain a pre-installed set of program and component interaction patterns. Other P2P communication libraries, such as libjingle, aimed at information streaming provide a narrow set of communicative primitives that do much succeed in expanding the abstraction pool already introduced by the Berkeley Software Distribution (BSD) sockets. The absence of well-defined rules for the approach to expanding the set of system interaction patterns that could be safe for the system and elaborately scaled entails uncertainty during product development and requires deep knowledge of all the architecture of the transportation solution in order to make any changes.

Thus, the users of our system had to add to the standard message transmission patterns, such as Push-Pool, Publish-Subscribe, Request-Reply and Pair, certain asynchronous Request-Reply capabilities that make it possible to meet clients' requests at the server's convenience and some long chains of Request-Reply pairs with an option of switching from the server to the client. Within the framework of the traditional non-broker architecture the users would have to create extensions to the existing libraries that currently do not offer formalized interfaces for expansion. In other words, to exercise total control over the behavior of the pattern introduced by them, having information about only one of the system's nodes, that could, in the process of horizontal scaling, result in the loss of stability of the entire system's operation and the application's responsiveness for the end users.

While working on the tasks assigned, we have developed a library offering interfaces that make it possible to systemically expand the set of offered means of communication. Let us note that the addition of communication patterns was connected with a range of problems: the uniform addressing scheme required for horizontal scaling; the unification of interfaces for the development of new communication components and further interaction with them; the opportunity to change the transportation level protocol without any need of rewriting the already created means of communication.

The systems that scholars have to work with within the framework of their research projects are very difficult for debugging. That is caused by many things, the problems of end service applications and the space of network communications uniting them being the major factors. Moreover, after the system has been launched, it is quite a difficult task to monitor its operability and bottlenecks.

That made us develop a means of interactive monitoring of the distributed network architecture, in addition to the already created library. This work has been focused on the creation of such a means and on the study of methods and practices aimed at solving such tasks.

The assigned task solution

Development stages

Our research consisted of three main parts.

- The study of the existing solutions aimed at the visualization of distributed systems binding elements with each other and displaying auxiliary information.
- The selection of the existing components that can become a base for the development of the suggested solution.
- The implementation, debugging and testing of the solution.

The existing solutions

During the first stage we have studied many existing solutions for visualization of specialized distributed systems. Yet, none of them is able to solve the task assigned. The most interesting solutions are the following:

- The system of download data presentation of the Windows Azure Service Bus message broker software;
- The modeling system of the RabbitMQ message broker software;
- The Boston subway traffic system visualization.

At this stage it has been decided to present the system nodes and their communications as a directed acyclic graph with floating tips above the elements containing information about the current state and work history.

Technological base

At the second stage the existing technological solutions with an open source code have been defined within the framework of the technologies suggested in the application. They have formed the basis of the developed solution. Let us consider the selected libraries.

- C# (.NET/Mono) statistics collection service and testing system services.
  - o ASP.NET 4.0 (mono) for web interface presentation;
  - o ASP.NET SignalR for the server's communication with the web interface;
  - o EntityFramework and SQLite for structured data storage;
  - o protobuf-net for reading the input data.
- JavaScript for application business logic processing.
  - o JQuery and SignalR for application operation logic control;
  - o D3 and dagre-d3 for displaying node communication, presentation of statistical data and work with the SVG format.

Protocol Buffers have been chosen for the formatting of the input data due to their high performance in terms of message reading and small-size message packages. The existence of such a solution made it possible to make the system an open one in terms of accepting statistics not only from the nodes dependent on the developed communication library but also from third-party solutions. The open interface makes it possible to load data created by systems written in more than ten popular programming languages such as Java, C++, Python, etc.

Debugging and testing

The selected technological base enabled us to create, during a short period of time, a new system integrated into the general solution. This system makes it possible to display on a near real time basis any changes in the system state and their communication characteristics.

The test system where the solution has been launched consists of three nodes bound by the Fast Ethernet network. Service applications communicating with each other have been launched on these nodes. The system features the following message transmission patterns:

- Publish-Subscribe;
- Request-Response;
- Pipe (in the form of a closed ring and directed graph).

The created subsystem-observer is located on one of the nodes.

The following groups of load modes concerning message transmission between the nodes per one second (simultaneously and subsequently) have been tested in it:

- 10, 100, 1000 messages per second;
- 1000, 5000, 10000 messages per second;
- 10000, 15000, 20000 messages per second;
- 10000, 50000, 100000, 200000 messages per second.

The system slightly influences the network workload of the data-sending nodes (in all the cases less than one per cent of the network load: 0.5 Mbit/s). The system does not consume more than five per cent of the CPU resource, that being true both for the workload of the 3 studied nodes and 5 listening clients and the workload of 100 connected nodes and 500 listening clients.

Conclusion

Our study has resulted in the development of an interactive system for creating a model of a dynamical distributed system of directly communicating nodes. This system can be helpful in the process of debugging, development and protection of the service systems used within the framework of SPbU research projects.

References

Bogdanov A. V., Degtyarev A. B., Nechaev Y., Valdenberg A. Design of Telemedicine System Architecture // Healthcare IT management. — 2006. — Vol. 1, No. 2. — P. 31–33.

Bogdanov A. V., Degtyarev A. B., Nechaev Y., Valdenberg A. Design of High-Performance Telemedicine System // Healthcare IT management. — 2006. — Vol. 1, No. 1. — P. 29–31.

Bogdanov A. V., Degtyarev A. B., Lwin S. M., Lwin T. K. Problems of Development of Complex Multilayered Applications in Distributed Environment // Proceedings of the 4th Intern. Conf. “Distributed Computing and Grid-Technologies in Science and Education”. Dubna: JINR, 2010. — P. 51–56.

Degtyarev A. B., Logvinenko Y. Agent system service for supporting river boats navigation // Procedia Computer Science. — 2010. — Vol. 1, No. 1. — P. 2717–2722.

Iakushkin O., Grishkin V. Messaging middleware for cloud applications: Extending brokerless approach // Emission Electronics (ICEE). — 2014. — P. 5–9.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 533–537

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.02

Decomposition of the modeling task of some objects of archeological research for processing in a distributed computer system

O. O. Iakushkin[a], A. B. Degtyarev, S. V. Shvemberger

Saint Petersburg State University, University ave. 35, Peterhof, St. Petersburg, 198504, Russia

E-mail: [a] Oleg.jakushkin@gmail.com

Получено 10 октября 2014 г.

Abstract. — Although each task of recreating artifacts is truly unique, the modeling process for façades, foundations and building elements can be parametrized. This paper is focused on a complex of the existing programming libraries and solutions that need to be united into a single computer system to solve such a task. An algorithm of generating 3D filling of objects under reconstruction is presented. The solution architecture necessary for the system's adaptation for a cloud environment is studied.

Keywords: distributed systems, cloud computing, modeling, reconstructions, service application architecture

Декомпозиция задачи моделирования некоторых объектов археологических исследований для работы в распределенной вычислительной среде

О. О. Якушкин, А. Б. Дегтярев, С. В. Швембергер

Санкт-Петербургский государственный университет, Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

В то время как каждая задача воссоздания артефактов уникальна, моделирование фасадов, фундаментов и конструктивных элементов строений может быть параметризовано. В работе рассмотрен комплекс существующих программных библиотек и решений, которые необходимо объединить в единую вычислительную систему для решения такой задачи. Представлен алгоритм генерации трехмерного заполнения реконструируемых объектов. Рассмотрена архитектура решения, необходимая для переноса системы в облачную среду.

Ключевые слова: сетевые распределенные вычисления, облачные вычисления, моделирование, реконструкция, сервисная архитектура

The work has been done with partial support of SPbU grant 9.38.674.2013, SPbU grant 9.37.157.2014 and RFBR grant No. 13-07-747. The research has been done using the computing resources of the resource center "SPbU Computing Facility" (http://cc.spbu.ru).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 533–537 (Russian).

Introduction

Visualization of architectural monuments' reconstruction and modeling of historical events of the past have always been an overarching challenge of historical studies. Our study is devoted to highlighting a pile of tasks aimed at reconstruction of certain classes of architectural objects requiring computer-aided modeling as an optimal solution in terms of authenticity and effort.

Although each task of recreating artifacts is truly unique, the modeling process for façades, foundations and building elements can be parametrized. Practically it means that the details of each single stone/block/slab constituting the reconstructed object may at the first stage be elaborated in a distributed computer environment without an artist's aid.

We have studied a complex of the existing programming libraries and solutions that need to be united into a single computer system to solve such a task. Its essential character is defined by a current practice, when, for the sake of the authenticity of the image and structures' collapse, artists have to model whole buildings using a single stone, that being incomparable to a computer-aided solution in terms of effort.

Task decomposition

The algorithm of object recreation

At present, various solutions are used for 3D object modeling. These are polygonal modeling packages; solutions with an option of giving volume and additional properties to objects during rendering; architectural modeling programs. At SPbU, the most frequently used modeling packages for interactive installations are Maya 2014 and 3ds Max 2014. They have no option of filling the given volume with similar objects with unique characteristics "out of the box". In other words, the software that artists use for modeling has no tools that make it possible to create stone blocks directly. Moreover, they require substantial computing resources for processing of large arrays of created objects which is crucial when the recreated model should be destroyable by means of animation. As a result, an approach has been developed, enabling the creation of single stones and whole walls as individual objects by introduction of possible random disturbances.

An algorithm of creating structures made of unique stones that can be composed (and decomposed) in a 3D space has been developed. The algorithm is based on the work with point clouds and their clusterization. This algorithm is based on working with parallelepiped-shaped bound enclosed volumes (*). Local grids acting as an analogue of the surface of a torus with a dimension greater by one than that of the studied volume are defined in *. For instance, if we have a ball with one (*)-type volume in diameter in the form of a cube with the side one, we shall see, upon moving the ball's center to the zero point, that the ball is split into eighths, each of them being located in the cube's angles see fig. 1 for 2D example.

Let us assume that there is a primitive, a point cloud, which is a mapping of a real world object into the world of points (i.e. that of a sphere).

Given an enclosed space of the * C type partitioned into n equal subspaces of the * type, each of them also partitionable. The number of inclusions must be limited. Each subspace in partitioning must have its own unique index.

Let us assume that the primitive has been inscribed into the smallest subspace, so that its position and size functionally depend on the index of the subspace it is being inscribed into. Let us suppose that similar dependence exists between the inscribed subspace and the space it is inscribed into.

Here the inclusion of the subspaces into the C space as well as their aspect ratio will define such masonry parameters as width, height, length and the number of layers.

By choosing the initial primitive and its distribution functions, as well as by noise masking and distortion operations performed above the points of the C space, it is possible to obtain the desired characteristics of the material's surface appearance.

Then the masonry array is decomposed into particular fragments by means of clusterization.

Fig 1: A circle in a 2d box *

The algorithm implementation

Today, there are several main libraries studied for the sake of the implementation of this algorithm. These are Point Cloud Library (PCL) and OpenVDB. Despite the fact that PCL has broad image capturing options to work with point clouds as well as data segmentation and filtration options, we have developed a small OpenVDB-based prototype. This library was chosen due to its convenience when working with nested hierarchical data structures.

The polygonal grid optimality is an important aspect of creating geometry. To optimize the grid developed as a result of using the algorithm, the OpenSubdiv library was used. This library makes it possible to decrease the number of polygons in the end model with no loss in quality. The algorithm's operability has been tested within the framework of the prototype.

Solution adaptation for cloud architecture

We have performed further task decomposition into the necessary components and means in order to prepare the project for implementation to solve field-specific tasks.

Modeling of a large number of objects made of such structures, i.e. the streets of an ancient town, requires high computational power. It is noteworthy that many libraries working with 3D graphics support GPU coprocessors "out of the box", OpenSubdiv being one of them.

Strict requirements to the modeling speed as well as the volume of needed calculations require the development of a service system existing in a distributed computer system. As a result, the solution is divided into a client application working in the modeling software used by the artist and a server application performing analyses and calculations automatically.

Porting out the main mechanism of model development to the service layer makes it possible to make the client thinner and the system in general, in the long view, applicable to deployment into various means of visual modeling.

To take into account any changes in the service system, the extension installed into the programming package should be updatable. This paper provides a description of how such an update should take place (Fig. 2).

The given service architecture makes it possible to extend the functionality at the data preprocessing and post-processing levels beyond the 3D modeling system. Thus, we can, at the preprocessing stage, request visual information about the object the user is working with and offer modeling service settings for such an object. At the post-processing stage, we can offer the user to choose the point cloud region he/she wants to visualize. The given architecture provides for soft updates of

the system. In other words, it enables creation, testing and deployment of new components of the system without interrupting the users' current sessions for updates.

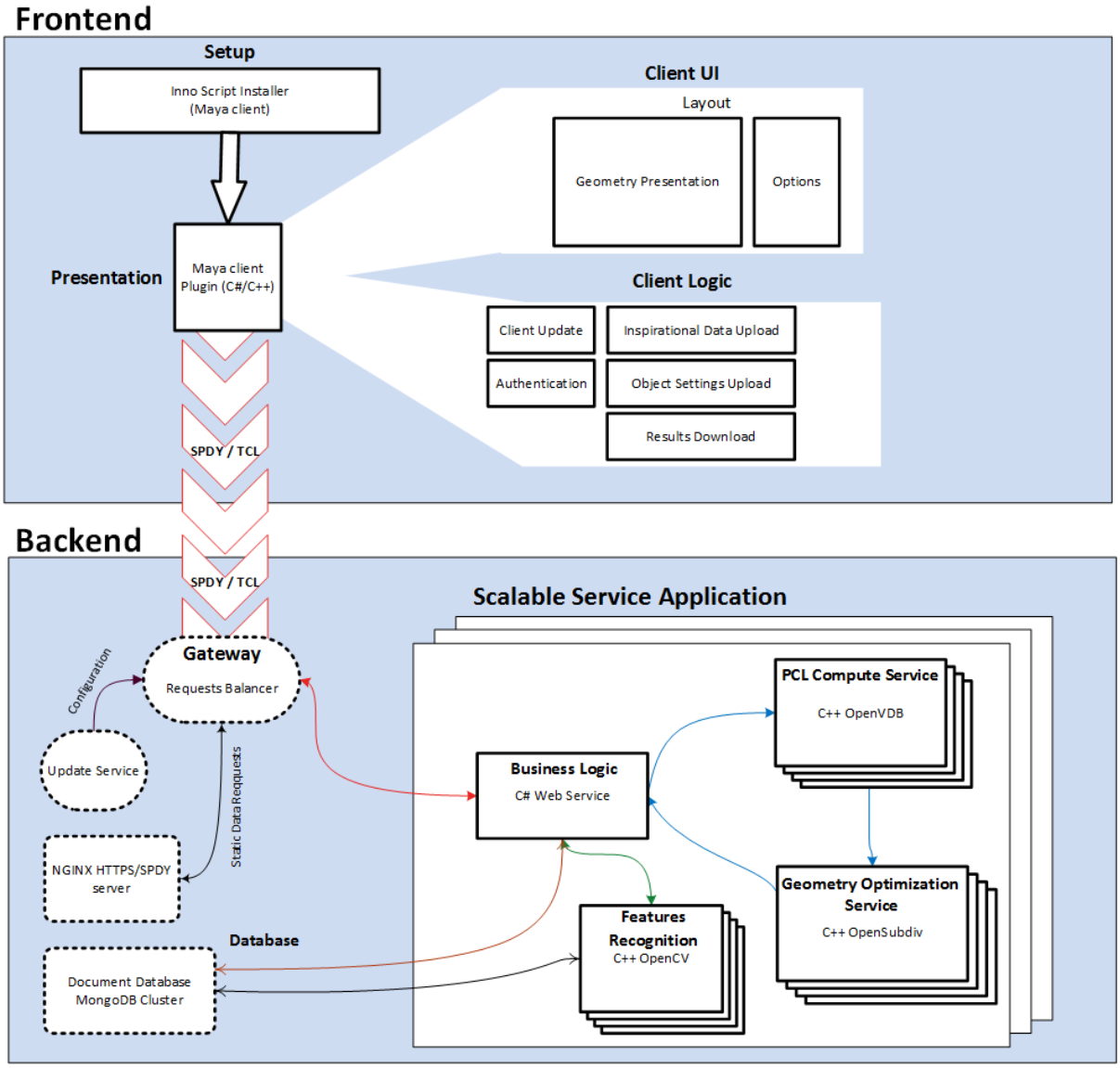


Fig. 2: The architecture of the offered service solution

Conclusion

Solutions of the modeling task of particular objects of archeological research have been elaborated. The architecture enabling the given system's work in a distributed computer system using service architecture has been described. Its scaling and further development points have also been described.

References

Museth, K., Lait, J., Johanson, J., Budsberg, J., Henderson, R., Alden, M., & Pearce, A. OpenVDB: an open-source data structure and toolkit for high-resolution volumes // In ACM SIGGRAPH, Courses (p. 19). ACM, 2013;

Schäfer, H., Niessner, M., Keinert, B., Stamminger, M., & Loop, C. State of the Art Report on Real-time Rendering with Hardware Tessellation // InEurographics 2014-State of the Art Reports (pp. 93–117). The Eurographics Association, 2014.

Watt, M., Coumans, E., ElKoura, G., Henderson, R., Kraemer, M., Lait, J., & Reinders, J. Multi-threading for Visual Effects // CRC Press. — 2014.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 539–542

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Exact calculation of a posteriori probability distribution with distributed computing systems

K. I. Kholodkov[a], I. M. Aleshin[b]

Schmidt Institute of Physics of the Earth RAS, Russia, 123995, Moscow, B. Grouzinskaya str. 10, b. 1

E-mail: [a] keir@ifz.ru, [b] ima@ifz.ru

Получено 10 октября 2014 г.

Abstract. — We'd like to present a specific grid infrastructure and web application development and deployment. The purpose of infrastructure and web application is to solve particular geophysical problems that require heavy computational resources. Here we cover technology overview and connector framework internals. The connector framework links problem-specific routines with middleware in a manner that developer of application doesn't have to be aware of any particular grid software. That is, the web application built with this framework acts as an interface between the user 's web browser and Grid's (often very) own middleware.

Our distributed computing system is built around Gridway metascheduler. The metascheduler is connected to TORQUE resource managers of virtual compute nodes that are being run atop of compute cluster utilizing the virtualization technology. Such approach offers several notable features that are unavailable to bare-metal compute clusters.

The first application we've integrated with our framework is seismic anisotropic parameters determination by inversion of SKS and converted phases. We've used probabilistic approach to inverse problem solution based on a posteriory probability distribution function (APDF) formalism. To get the exact solution of the problem we have to compute the values of multidimensional function. Within our implementation we used brute-force APDF calculation on rectangular grid across parameter space.

The result of computation is stored in relational DBMS and then represented in familiar human-readable form. Application provides several instruments to allow analysis of function's shape by computational results: maximum value distribution, 2D cross-sections of APDF, 2D marginals and a few other tools. During the tests we've run the application against both synthetic and observed data.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 539–542 (Russian).

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 539–542**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

Точное вычисление апостериорной функции распределения вероятности при помощи вычислительных систем

К. И. Холодков, И. М. Алёшин

Институт физики Земли им. О. Ю. Шмидта РАН, Россия, 123995, г. Москва, Б. Грузинская ул., 10, стр. 1

Представленная работа описывает опыт создания и развёртывания веб-приложения и грид-инфраструктуры для решения задач геофизики, требующих большого количества вычислительных ресурсов. В работе представлен обзор технологии и механизма платформы интеграции геофизических приложений с распределёнными вычислительными системами. Разработанная платформа предоставляет собой промежуточное программное обеспечение, предоставляющая удобный доступ к развёрнутым на ее основе геофизическим приложениям. Доступ к приложению осуществляется через веб-браузер. Интеграция новых приложений облегчается за счёт предоставляемого стандартного универсального интерфейса взаимодействия платформы и новым приложением.

Для организации распределённой вычислительной системы применено ПО Gridway, экземпляр которого взаимодействует с виртуализированными вычислительными кластерами. Виртуализация вычислительных кластеров предоставляет новые возможности при утилизации вычислительных ресурсов по сравнению с традиционными схемами организации кластерного ПО.

В качестве пилотной задачи использована обратная задача определение параметров анизотропии коры и верхней мантии по данным телесейсмических наблюдений. Для решения использован вероятностный подход к решению обратных задач, основанный на формализме апостериорной функции распределения (АПФР). При этом вычислительная задача сводится к табулированию многомерной функции. Результат вычислений представлен в удобном для анализа высокоуровневом виде, доступ и управление осуществляется при помощи СУБД. Приложение предоставляет инструменты анализу АПФР: расчет первых моментов, двумерные маргинальные распределения, двумерные сечения АПФР в точках ее максимума. При тестировании веб-приложения были выполнены вычислены как синтетических, так и для реальных данных.

Introduction

One of our previous implementation of determination of anisotropic parameters by seismic data inversion problem utilized the EGEE grid infrastructure. The infrastructure used the gLite middleware, despite the use of gLite middleware the computation procedure startup was carried out by hand mostly.

Large scale computational resources are difficult to utilize and maintain and middleware provides a sort of unified interface to it. Obviously researches that lack sufficient large scale computing skills require another level of usability improvement. Often this improvement is archived by building a so-called wrapper above complex middleware interfaces. This is what we've done in [Aleshin, 2013; Kholodkov, 2013]. That research involved the creation of own grid infrastructure with Globus Toolkit across three Academic institutions. Major drawbacks include complexity of application changes and overall system akwardness. The later was caused by security and transfer services bypass. We've written our own implementation for user authentication and job data delivery but we couldn't drop GSI and GridFTP out of Globus infrastructure. This led us to several conceptual improvement — the new wrapper has to be two-level one. The first level — an integration platform that makes all interaction with grid middleware and provides simplified API for application development, which is the second level of a two-layer concept. Additionaly we've switched to Gridway.

The implementation

New implementation utilizes the same hardware as previous experiment. Particularly, we've made use of virtualization to provide flexible resources to computation project and to allow easy maintenance. The virtualization uses hardware acceleration so compute power isn't wasted.

Switching to Gridway allowed us to build a simplified grid. Gridway metascheduler runs on separate node and talks to TORQUE queue manages directly thanks to DRM4G[DRM4G, 2014] patches to Gridway.

The integration platform connects application with grid middleware. The platform is built using common modern coding techniques for easy modification and extension. Most parts of platform are coded in php with CodeIgniter, which is an Model-View-Controller programming framework. Additionaly the platform utilizes a V8 Javascript Engine (node.js) for utility jobs. Application-specific modules are separated from common shared code allowing the researcher to easy port similar (data independent class) applications to run on this platform. As pilot project we've implemented determination of anisotropic parameters of crust and upper mantle by teleseismic data problem.

Pilot problem

The pilot provides a simple web-interface for determination of anisotropic parameters of crust and upper mantle by teleseismic data problem. The major difference from our previous approach [Алёшин, 2009] is use of APDF formalism to find best fit. The idea of APDF takes data and model error estimation into accout in the way that the outcome of the solver is best-fit-probability to parameter set relation.

Renewed web interface is now inverse-problem oriented and the only question regarding the grid itself user has to answer is about application split factor — how small should slices of entire application be — and this will be addressed in future.

In addition to monitoring the new interface features automatic result processing by generating a set of parameter-set-to-APDF relation cross-sections also available via web-interface.

Conclusion

This experiment indicated that grid software needs additional high-level API for scientific applications, which is addressed by software described in this paper that allows users to port their own

applications to run on grid. Gridway with DRM4G and TORQUE makes a quick and easy solution for experimental grids.

References

Aleshin I. M., *Koryagin V. N.*, *Sukhoroslov O. V.*, *Kholodkov K. I.*, *Shogin A. N.* [in Russian] Инверсия сейсмических данных: высокоуровневый веб-интерфейс к инструментарию *Globus Toolkit* // Scientific and Technical Information Processing. — 2013. — Series 1, No. 7.

Aleshin I. M., Mishin *D. Yu.*., *Zhizhin M. N.*, *Koryagin V. N.*., Medvedev D. P., Novikov A. M., Peregoudov D. V. [in Russian] Применение распределенных вычислительных систем при определении параметров сейсмической анизотропии коры и верхней мантии // Geophysical Research. — 2009. — Vol. 10, No. 4. — P. 34–47.

DRM4G (Distributed Resource Management for Grid) [online] // Santander Meteorology Group, Instituto de Fisica de Cantabia — 2014 — URL: https://meteo.unican.es/trac/wiki/DRM4G (дата обращения: 17.12.2014);

Kholodkov K. I.. Implementation of seismic data inversion as grid-backed web service // Russian journal of Earth sciences. — 2013. — № 13.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 543–547

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК 004.4, 004.63

GridFTP frontend with redirection for DMlite

A. K. Kiryanov

Petersburg Nuclear Physics Institute, Orlova Roscha, Gatchina, 188300, Russia

E-mail: globus@pnpi.nw.ru

Получено 10 октября 2014 г.

One of the most widely used storage solutions in WLCG is a Disk Pool Manager (DPM) developed and supported by SDC/ID group at CERN. Recently DPM went through a massive overhaul to address scalability and extensibility issues of the old code.

New system was called DMLite. Unlike the old DPM that was based on daemons, DMLite is arranged as a library that can be loaded directly by an application. This approach greatly improves performance and transaction rate by avoiding unnecessary inter-process communication via network as well as threading bottlenecks.

DMLite has a modular architecture with its core library providing only the very basic functionality. Backends (storage engines) and frontends (data access protocols) are implemented as plug-in modules. Doubtlessly DMLite wouldn't be able to completely replace DPM without GridFTP as it is used for most of the data transfers in WLCG.

In DPM GridFTP support was implemented in a Data Storage Interface (DSI) module for Globus' GridFTP server. In DMLite an effort was made to rewrite a GridFTP module from scratch in order to take advantage of new DMLite features and also implement new functionality. The most important improvement over the old version is a redirection capability.

With old GridFTP frontend a client needed to contact SRM on the head node in order to obtain a transfer URL (TURL) before reading or writing a file. With new GridFTP frontend this is no longer necessary: a client may connect directly to the GridFTP server on the head node and perform file I/O using only logical file names (LFNs). Data channel is then automatically redirected to a proper disk node.

This renders the most often used part of SRM unnecessary, simplifies file access and improves performance. It also makes DMLite a more appealing choice for non-LHC VOs that were never much interested in SRM.

With new GridFTP frontend it's also possible to access data on various DMLite-supported backends like HDFS, S3 and legacy DPM.

Keywords: WLCG, Grid, GridFTP, DPM, DMLite, data storage, access protocol

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 543–547 (Russian).

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 543−547**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

Поддержка протокола GridFTP с возможностью перенаправления соединений в DMLite Title

А. К. Кирьянов

Петербургский институт ядерной физики им., Россия, 188300, Ленинградская обл., Гатчина, Орлова роща, ФГБУ ПИЯФ

Одним из наиболее широко используемых решений для хранения данных в WLCG является Disk Pool Manager (DPM), разрабатываемый и поддерживаемый группой SDC/ID в ЦЕРНе. Недавно старый код DPM был практически переписан с нуля для решения накопившихся проблем с масштабируемостью и расширением функциональности.

Новая система была названа DMLite. В отличие от DPM, который был реализован в виде нескольких демонов, DMLite выполнена в виде программной библиотеки, которая может быть непосредственно загружена приложением. Такой подход значительно повышает общую производительность и скорость обработки транзакций, избегая ненужного межпроцессного взаимодействия через сеть, а также узких мест в многопоточной обработке.

DMLite имеет модульную архитектуру, при которой основная библиотека обеспечивает только несколько базовых функций. Системы хранения данных, а также протоколы доступа к ним реализованы в виде подключаемых модулей (plug-ins). Конечно, DMLite не смогла бы полностью заменить DPM без поддержки протокола GridFTP, наиболее широко используемого для передачи данных в WLCG.

В DPM поддержка протокола GridFTP была реализована в виде модуля Data Storage Interface (DSI) для GridFTP сервера Globus. В DMLite было решено переписать модуль GridFTP с нуля, чтобы, во-первых, воспользоваться новыми возможностями DMLite, а во-вторых, добавить недостающую функциональность. Наиболее важным отличием по сравнению со старой версией является возможность перенаправления соединений.

При использовании старого интерфейса GridFTP клиенту было необходимо предварительно связаться со службой SRM на головном узле хранилища, чтобы получить Transfer URL (TURL), необходимый для обращения к файлу. С новым интерфейсом GridFTP делать этот промежуточный шаг не требуется: клиент может сразу подключиться к службе GridFTP на головном узле хранилища и выполнять чтение-запись используя логические имена файлов (LFNs). Канал передачи данных при этом будет автоматически перенаправлен на соответствующий дисковый узел.

Такая схема работы делает одну из наиболее часто используемых функций SRM ненужной, упрощает доступ к файлам и повышает производительность. Это также делает DMLite более привлекательным выбором для виртуальных организаций, не относящихся к БАК, поскольку они никогда не были особо заинтересованы в SRM.

Новый интерфейс GridFTP также открывает возможности для хранения данных на множестве альтернативных систем, поддерживаемых DMLite, таких как HDFS, S3 и существующие пулы DPM.

Ключевые слова: БАК, Грид, хранилище данных, протокол доступа

One of the most widely used storage solutions in WLCG is a Disk Pool Manager [DPM..., 2015] developed and supported by SDC/ID group at CERN. It was started in 2005 and over the time built a strong install base on more than 200 sites. Unfortunately some of the architectural decisions taken at the very beginning of the project resulted in scalability and extensibility issues such as:

– Monolithic code base, that was hard to maintain and add features to.
– IPC between multiple daemons even for simple operations.
– Reliance on SRM which turned out not to be attractive outside of HEP community.

Eventually it was decided to do a massive overhaul of the old code and effectively build a completely new system that would be free of DPM shortcomings while maintaining backwards compatibility with old clients: DMLite [DMLite..., 2015]. Unlike the old DPM that was based on daemons, DMLite is arranged as a library that can be loaded directly by an application. This approach greatly improves performance and transaction rate by avoiding unnecessary inter-process communication via network as well as threading bottlenecks.

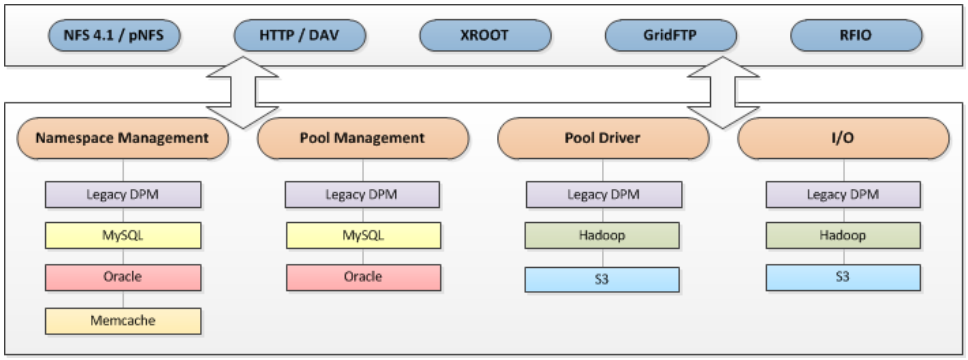


Fig. 1. DMLite modular architecture

DMLite has a modular architecture (fig.1) with its core library providing only the very basic functionality. Backends (storage engines — on bottom) and frontends (data access protocols — on top) are implemented as plug-in modules. While HTTP and XROOT are gaining momentum in inter-site data transfers, doubtlessly DMLite wouldn't be able to completely replace DPM without GridFTP [GridFTP, 2015] as it is still used for most of the data transfers in WLCG.

In DPM GridFTP support was implemented in a Data Storage Interface (DSI) module for Globus Toolkit GridFTP server. In DMLite an effort was made to rewrite GridFTP module from scratch in order to take advantage of new DMLite features and also implement new functionality. The most important improvement over the old version is redirection: an ability to authenticate clients and accept data transfer requests at one point (host) but seamlessly perform an actual data exchange with another one.

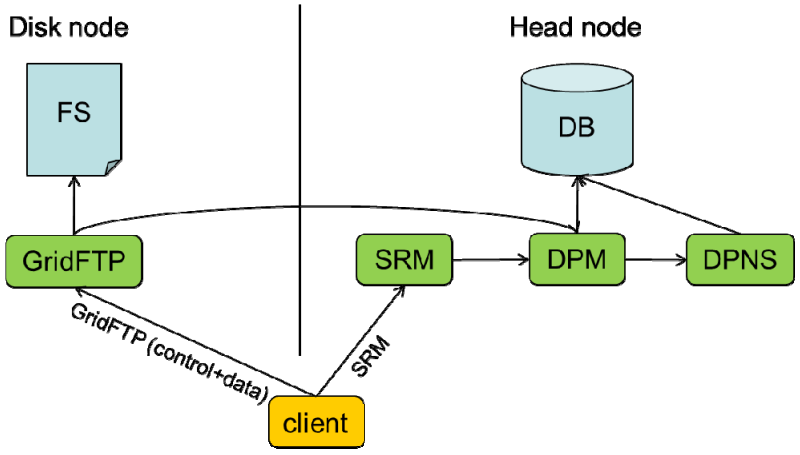


Fig. 2. GridFTP access with DPM

With old DPM GridFTP frontend a client needed to contact SRM on the head node in order to convert logical filename (LFN) to a transfer URL (TURL) before reading or writing a file (fig. 2). With new GridFTP frontend this is no longer necessary: a client may connect directly to the GridFTP server on the head node and perform file I/O using only logical file names (LFNs). Data channel is then automatically redirected to a proper disk node (fig. 3). This renders the most often used part of SRM unnecessary, simplifies file access and improves performance. It also makes DMLite a more appealing choice for non-LHC VOs that were never much interested in SRM.

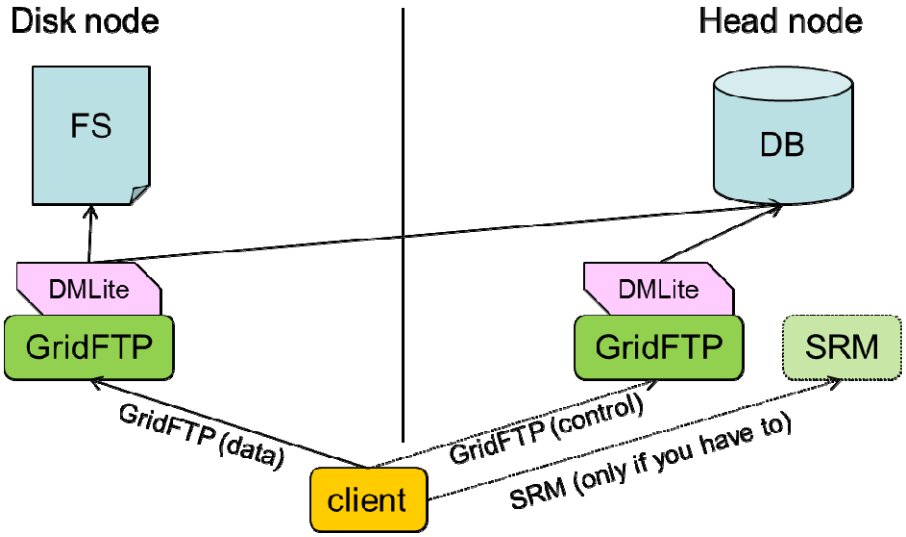


Fig. 3. GridFTP access with DMLite

GridFTP redirection was implemented thanks to the Delayed Passive connection mode available as one of the GridFTP v2 extensions available in Globus Toolkit. With legacy Passive mode a server had to provide connection endpoint parameters (address and port) before a file request, at a time point when file name was not yet known. With Delayed Passive a server postpones its response until a client actually initiates a file transfer, which makes it possible to redirect a client to different disk nodes depending on file name (fig. 4).

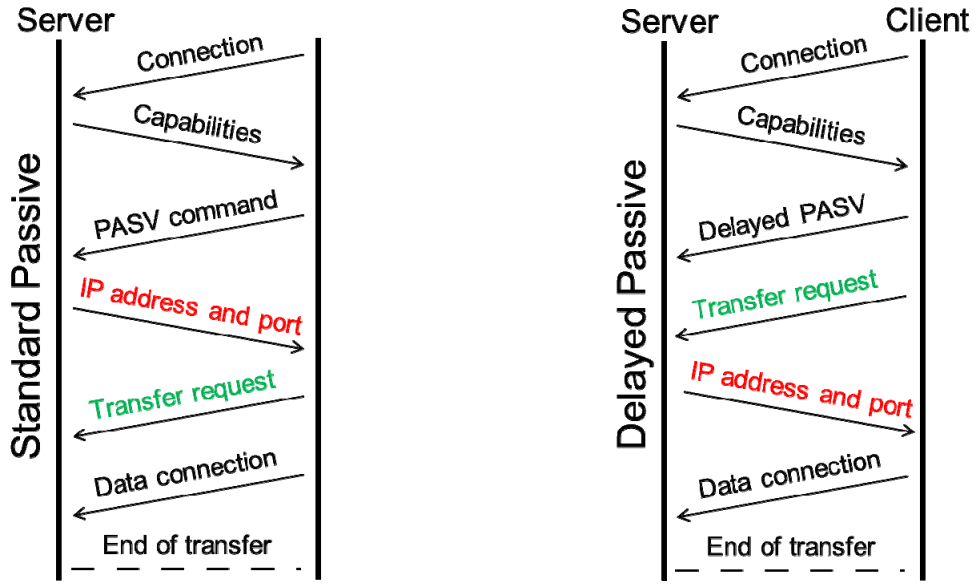


Fig. 4. Passive mode vs. Delayed Passive mode

New GridFTP frontend supports two deployment scenarios: stand-alone and redirecting. Stand-alone scenario is essentially identical to the way GridFTP was deployed with old DPM: GridFTP servers are configured independently and a client has to know which server to connect to. This may be used as drop-in replacement for DPM+SRM installations.

Redirecting scenario is somewhat different: head node is configured as a redirector, and disk nodes are put in “backend mode” which forbids direct client connections. By this scenario a client always has to contact head node and does not have to worry about obtaining TURLs to a proper disk node. The only limitation of this scenario is that clients have to support Delayed Passive mode for optimal performance.

GridFTP frontend for DMLite is in production since mid-2014. It is supported by GFAL2 library and FTS3 file transfer service which is widely used on WLCG for scheduled file transfers. With new GridFTP frontend it's also possible to access data on various DMLite-supported backends like HDFS, S3 and legacy DPM.

References

DPM (Disk Pool Manager) [online] // — 2015 — URL: https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Dev/Dmlite (дата обращения: 17.01.2015);

DMLite [online] // — 2015 — URL: https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Dev/Dmlite (дата обращения: 17.01.2015);

GridFTP [online] // — 2015 — URL: http://home.web.cern.ch/about/computing/worldwide-lhc-computing-grid (дата обращения: 17.01.2015).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 549–558

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.43, 004.94

OpenCL realization of some many-body potentials

A. A. Knizhnik[1,2,a], A. S. Minkin[1,b], B. V. Potapkin[1,2,c]

[1]National Research Center "Kurchatov Institute", Kurchatov Sq. 1, Moscow, 123182, Russia
[2]Kintech Lab Ltd, Kurchatov Sq. 1, Moscow 123182, Russia

E-mail: [a]knizhnik@kintechlab.com, [b]amink@mail.ru, [c] potapkin@kintechlab.com

Получено 27 октября 2014 г.

Abstract. — Modeling of carbon nanostructures by means of classical molecular dynamics requires a lot of computations. One of the ways to improve the performance of basic algorithms is to transform them for running on SIMD-type computing systems such as systems with dedicated GPU. In this work we describe the development of algorithms for computation of many-body interaction based on Tersoff and embedded-atom potentials by means of OpenCL technology. OpenCL standard provides universality and portability of the algorithms and can be successfully used for development of the software for heterogeneous computing systems. The performance of algorithms is evaluated on CPU and GPU hardware platforms. It is shown that concurrent memory writes is effective for Tersoff bond order potential. The same approach for embedded-atom potential is shown to be slower than algorithm without concurrent memory access. Performance evaluation shows a significant GPU acceleration of energy-force evaluation algorithms for many-body potentials in comparison to the corresponding serial implementations.

Реализация алгоритмов межатомного взаимодействия с использованием технологии OpenCL

А. С. Минкин[1], А. А. Книжник[1,2], Б. В. Потапкин[1,2]

[1]Национальный исследовательский центр "Курчатовский институт", Россия, 123182, г. Москва, пл. Академика Курчатова, д. 1
[2]Кинтех Лаб, Россия, 123182, г. Москва, пл. Академика Курчатова, д. 1

Моделирование углеродных наноструктур методом классической молекулярной динамики требует больших объемов вычислений. Один из способов повышения производительности соответствующих алгоритмов состоит в их адаптации для работы с SIMD-подобными архитектурами, в частности, с графическими процессорами. В данной работе рассмотрены особенности алгоритмов вычисления многочастичного взаимодействия на основе классических потенциалов Терсоффа и погруженного атома с использованием технологии OpenCL. Стандарт OpenCL позволяет обеспечить универсальность и переносимость алгоритмов и может быть эффективно использован для гетерогенных вычислений. В данной работе сделана оценка производительности OpenCL алгоритмов вычисления межатомного взаимодействия для систем на базе центральных и графических процессоров. Показано, что использование атомарных операций эффективно для вычисления потенциала Терсоффа и неэффективно в случае потенциала погруженного атома. Оценка производительности показывает значительное ускорение GPU реализации алгоритмов вычисления потенциалов межатомного взаимодействия по сравнению с соответствующими однопоточными алгоритмами.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 549–558 (Russian).

Introduction

Atomistic modeling is computationally intensive problem. Serial realization of algorithms can be done only for modeling of small system of atoms. Cluster type computing systems used in most cases to solve the problem of high computational complexity can be good but sometimes have problems with long time waiting for resource. Besides creating effective MIMD algorithms is a real challenge. The alternative way is optimization of existing algorithms for SIMD/SIMT hardware devices such as graphic processing units (GPU). GPU is a part of contemporary clusters and heterogeneous systems. Using such devices allows increasing the performance of algorithms along with decreasing of the power consumption. Such approach is good in sense higher availability of computing resources as some hardware such as personal computer with dedicated GPU or GPU-based GRID systems can be built.

The main problem of using heterogeneous parallel systems is creating of the effective algorithms by means of appropriate memory access patterns and load balancing. But the optimal algorithms should take into account the architecture of computing system so we are coming to development of algorithm for specific devices. That way is proposed by CUDA technology [Jason Sanders, Edward Kandrot, 2010]. CUDA is designed for NVidia GPUs only and does not assume free support of CPU computation. Such approach is very tedious and one prefers to have an algorithm for all devices. There are also some new technologies. For example, such technology as OpenACC give OpenMP-like pragmas for GPGPU adaptation but actually it is not free. OpenMP 4.0 is supported by gcc but it is under development and not available for all platforms. The only crossplatform technology with free SDKs available for all popular operating systems is OpenCL [Gaster et al, 2011]. It is supported by most of hardware vendors, gives access to most of GPU features via the frontend and makes it possible to use the same kernel code with different devices. So we come to a compromise between the performance and universality. We use OpenCL technology for CPU-based systems and for GPU programming as GPUs are the devices with the best ratio of performance and power consumption. The latter is also important in Russia as the reforms of 90-th result in no hope that electrical energy would ever be less expensive.

GPU are basically computer graphics devices and they are optimal for rendering. But with unified architecture we have a capability of GPGPU computation. There are the following advantages of GPU over traditional CPU systems:

- Large number of processor cores.
- Threads are lightweight and large number of threads hides global memory latency.
- The L1 cache (local memory) is available for direct access by programmer.
- Large memory bandwidth.
- Complete RISC architecture without CISC converter.

These advantages can be got for good use in computing as long as appropriate technology would be applied. OpenCL was chosen as such GPGPU technology.

There are some algorithms that can be effectively done by means of SIMD-similar hardware. One of such problems is molecular dynamics simulation. The critical aspect of classical molecular dynamics [Allen, Tildesley, 1990] and Monte-Carlo methods [Bielajew, 2001] is the choice of an appropriate energy function (potential) for describing the interatomic interactions. Our objective is modeling of carbon nanostructures such as graphene, fullerenes and nanotubes. So we need special potentials for adequate computation of their energetic. The most of these potentials are many-body. In this work we evaluate the performance of OpenCL algorithms of Tersoff potential in comparison to EAM potential for modeling of metallic systems.

Complicated form and larger set of parameters of many-body potentials results in large computational requirements. So the acceleration of existing algorithms of interatomic interaction is especially needed.

Interatomic potentials

In atomistic modeling quantum chemistry is the best way to reproduce experimental results but it is usually used for simulation of relatively small systems of atoms. We have about 10^{23} atoms in 2

gramms of carbon. So we need Exascale supercomputer for modeling of real patterns and that become almost impossible by now to do that via quantum mechanical approach. Classical and semi-classical simulations are a real compromise and they are used for modeling of relatively large atomic systems.

Interatomic potential can be described by the potential function. The potential function $U(\mathbf{r}_1,\mathbf{r}_2, \ldots, \mathbf{r}_N)$ is the dependence of the potential energy of N-atom system on their coordinates. The forces in MD simulation are defined by the potential,

$$\mathbf{F}_l = -\frac{\partial U(\mathbf{r}_1,\mathbf{r}_2,...,\mathbf{r}_N)}{\partial \mathbf{r}_l}, \tag{1}$$

where $\mathbf{r}_l$ is a position (3D vector) of the atom l, $\mathbf{F}_l$ — force (3D vector) acting on the atom l.

The most popular types of potentials used in numerical software are pair potentials. The interaction of any pair of atoms depends only on their spacing and is not affected by the presence of other atoms. Full potential energy in that case is the sum of pair interactions:

$$U(\mathbf{r}_1,\mathbf{r}_2,...,\mathbf{r}_N) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1\,(j\neq i)}^{N} U_{ij}^{pair}(r_{ij}), \tag{2}$$

where r_{ij} is a distance between atom i and atom j. The most popular pair potential implemented in the most MD software is Lennard-Jones (LJ) potential.

Though it can be effectively implemented on GPU [Anderson, Lorenz, 2010] it is not good for modeling of specific systems such as carbon nanostructures and metallic systems. Indeed, low density structures of covalent systems are instable with pair potentials.

That is why for modeling of covalent crystals, carbon nanostructures, metallic systems and polymers more complicated many-body potentials [Erkoc, 1997] are widely used. The set of parameters of many-body potentials are usually derived via quantum mechanical methods [Kumagai et al., 2009, Lebedeva et al., 2012]. That is why many-body potentials are much more accurate and can reproduce mechanical and transport phonon properties of carbon and metallic nanostructures.

A general form of many-body potentials such as Tersoff potential [Tersoff, 1989] and embedded-atom potential [Murray, Foiles, 1993] can be written as a sum of pair and many-body terms [Brenner, 1989]:

$$U_i = \frac{1}{2}\sum_{j\neq i} U_{ij}^{pair}\left(r_{ij}\right) + U_i^{mb}, \tag{3}$$

where U_i is an energy per atom i.

Embedded-atom potential takes into account many-body interaction by means of nonlinear embedding function in energy functional:

$$U_i^{mb} = E_i(\rho_i), \rho_i = \sum_{j\neq i} c_j\left(r_{ij}\right), \tag{4}$$

where ρ_i is an electron density per atom i, E_i is embedded energy per atom i, $c_j(r_{ij})$ represents the influence of atom j to the electron density (ρ_i) of atom i.

The main advantage of many-body potentials over pair potentials is the ability to describe the variation of the bond strength with coordination. Tersoff and Tersoff-Brenner potentials can reproduce the same mechanism by means of bond order formalism. Bond order function includes angle dependence that makes Tersoff potential many-body. The simplest functional form of Tersoff potential can be written in the following form:

$$U = \sum_i U_i = \frac{1}{2}\sum_i\sum_{j\neq i} U_{ij},$$

$$U_{ij}^{pair}\left(r_{ij}\right)=V_R\left(r_{ij}\right),$$

$$U_i^{mb}=-\frac{1}{2}\sum_{j\neq i}b_{ij}V_A\left(r_{ij}\right),$$

$$U_{ij}=U_{ij}^{pair}+U_{ij}^{mb}=V_R\left(r_{ij}\right)-b_{ij}V_A\left(r_{ij}\right),$$

$$V_R\left(r_{ij}\right)=f_{ij}\left(r_{ij}\right)A_{ij}\exp\left(-\lambda_{1,ij}r_{ij}\right), \qquad \text{[repulsive term]}$$

$$V_A\left(r_{ij}\right)=f_{ij}\left(r_{ij}\right)B_{ij}\exp\left(-\lambda_{2,ij}r_{ij}\right), \qquad \text{[attractive term]}$$

$$b_{ij}=\left[1+\beta^n\varsigma_{ij}^n\right]^{-\frac{1}{2n}}, \qquad \text{[bond order function]}$$

$$\varsigma_{ij}=\sum_{k(\neq i,j)}G_i\left(\theta_{ijk}\right)f_{ik}\left(r_{ik}\right)\exp\left[\lambda_{3,ijk}^m\left(r_{ij}-r_{ik}\right)^m\right],$$

$$G_i\left(\theta_{ijk}\right)=\gamma_{ijk}\left[1+\frac{c^2}{d^2}-\frac{c^2}{d^2+\left(\cos\theta-\cos\theta_0\right)^2}\right], \qquad \text{[angle dependence function]}$$

$$f_{ik}\left(r_{ik}\right)=\begin{cases}1, \text{if } r_{ik}<R-D,\\ \frac{1}{2}\left[1-\sin\left[\frac{\pi\left(r-R\right)}{2D}\right]\right], \text{if } R-D\le r_{ik}\le R+D,\\ 0, \text{if } r_{ik}>R+D.\end{cases} \qquad \text{[cutoff function]}$$

The Tersoff potential is short-ranged as it uses the cutoff function. So the local environment has limited number of atoms. That is why neighbor list can be used to represent the local environment [Allen, Tildesley, 1990]. Neighbor lists allow to speed up the computation of interatomic interaction by eliminating of N^2 search. That is especially important for computational expensive many-body potentials.

So many-body potentials have the following advantages:

- That is more adequate model of interatomic interaction than pair potentials. So more macroscopic parameters and physical constants can be represented by many-body potentials as they have more complicated parametric form.
- Appropriate parallel algorithms can be effectively designed because of interaction locality.

The disadvantages of many-body potentials are concerned with the need to match a lot ofconstants customized just for specific chemical compounds and their computational complexity.

The problem of high computational complexity of many-body potentials can be solved by the adaptation of the algorithms to such computing architectures as GPU and accelerators. Unfortunately the basic serial algorithms cannot be used on GPU without adaptation as some features of the accelerator architecture such as memory hierarchy are needed to be taken into account [Jason Sanders, Edward Kandrot, 2010, Gaster et al., 2011].

The following subsections are dedicated to description of various algorithms with concurrent memory access without it.

GPU algorithms for Tersoff potential

GPU algorithms for potential can be represented by computation of per atom energy along with force acting on atom. So we have parallel thread for each atom in systems. The more threads we have the best as global memory latency would be hided.

Parallel GPU algorithm for Tersoff potential can be represented in two forms:

1. Algorithm with atomic operations with memory (Algorithm A).

2. Algorithm without atomics (Algorithm WA).

Algorithm A can be represented as a transformation of the serial algorithm in the following form:

- Algorithm A is assumed to run for N work-items in 1D index space (*N* GPU threads).
- Each thread *i*

1. calculates U_i:

$$U_i = \frac{1}{2}\sum_{j\neq i} V_R\left(r_{ij}\right) - \frac{1}{2}\sum_{j\neq i} b_{ij} V_A\left(r_{ij}\right);$$

where *j* are the numbers of neighboring atoms of atom *i* (*j*≠*i*);

2. calculates $\mathbf{F}_i$:

$$\mathbf{F}_i = \mathbf{F}_i - \sum_j \frac{\partial V_R\left(r_{ij}\right)}{\partial \mathbf{r}_i} + \frac{1}{2}\sum_j b_{ij}\frac{\partial V_A\left(r_{ij}\right)}{\partial \mathbf{r}_i} + \frac{1}{2}\sum_{j,k}\frac{\partial b_{ij}}{\partial \mathbf{r}_i} V_A\left(r_{ij}\right), \frac{\partial b_{ij}}{\partial \mathbf{r}_i} = p(\varsigma_{ij})\cdot\frac{\partial \varsigma_{ij}}{\partial \mathbf{r}_i},$$

3. Calculates and sums with atomic operations $\mathbf{F}_j$ и $\mathbf{F}_k$ (*j* and *k* are the numbers of neighboring atoms of atom *i* (*j*≠*i, k*≠*i*):

$$\mathbf{F}_j = \mathbf{F}_j - \frac{1}{2} b_{ij}\frac{\partial V_A\left(r_{ij}\right)}{\partial \mathbf{r}_i} + \frac{1}{2}\sum_k \frac{\partial b_{ij}}{\partial \mathbf{r}_j} V_A\left(r_{ij}\right), \frac{\partial b_{ij}}{\partial \mathbf{r}_j} = p(\varsigma_{ij})\cdot\frac{\partial \varsigma_{ij}}{\partial \mathbf{r}_j},$$

$$\mathbf{F}_k = \mathbf{F}_k + \frac{1}{2}\sum_k \frac{\partial b_{ij}}{\partial \mathbf{r}_k} V_A\left(r_{ij}\right), \frac{\partial b_{ij}}{\partial \mathbf{r}_k} = p(\varsigma_{ij})\cdot\frac{\partial \varsigma_{ij}}{\partial \mathbf{r}_k},$$

$$p(\varsigma_{ij}) = -\frac{1}{2}\cdot\left[1+\left(\beta\varsigma_{ij}\right)^n\right]^{\frac{1}{2n}-1}\left(\beta\varsigma_{ij}\right)^n / \varsigma_{ij};$$

Atomic operations are intended to summarize correctly the force contributions to the interaction in parallel streams. In this case, memory is a shared resource and atomicity is achieved in several stages: blocking of the resource, summation, release of the resource. Three of these steps provide exclusive access to a portion of memory and atomicity prevents wrong updates, i.e. atomic operation is either successful or returns the occupation of the shared resource. In the case of occupation, the update of the memory cell by the other stream is delayed, i.e. the access of multiple threads to the same memory location serializes, resulting in a relative decrease of parallel efficiency.

Algorithm A is the most evident modification of serial algorithm but it can be transformed in the form without atomic operations with memory. That is possible for most potentials but for the sake of increasing the number of operations inside the thread. We need atomic operations with memory only to compute forces. In case of energy no concurrent access is needed. But the force computation can also be done without atomics as following (Algorithm WA):

- Algorithm WA is assumed to run for N work-items in 1D index space (*N* GPU threads).
- Each thread *i*

1. Calculates

$$U_i = \frac{1}{2}\sum_{j\neq i} U_{ij}.$$

2. Calculates

$$\mathbf{F}_i = -\frac{1}{2}\sum_{j\neq i}\left(\frac{\partial U_{ij}}{\partial \mathbf{r}_i} + \frac{\partial U_{ji}}{\partial \mathbf{r}_i}\right) - \frac{1}{2}\sum_{j\neq i,k\neq i}\frac{\partial U_{jk}}{\partial \mathbf{r}_i};$$

$$\frac{\partial U_{ij}}{\partial \mathbf{r}_i} = \frac{\partial V_R(r_{ij})}{\partial \mathbf{r}_i} - b_{ij}\frac{\partial V_A(r_{ij})}{\partial \mathbf{r}_i} - \frac{\partial b_{ij}}{\partial \mathbf{r}_i}V_A(r_{ij}),\ \frac{\partial b_{ij}}{\partial \mathbf{r}_i} = p(\varsigma_{ij})\cdot\frac{\partial \varsigma_{ij}}{\partial \mathbf{r}_i};$$

$$\frac{\partial U_{ji}}{\partial \mathbf{r}_i} = \frac{\partial V_R(r_{ij})}{\partial \mathbf{r}_i} - b_{ji}\frac{\partial V_A(r_{ij})}{\partial \mathbf{r}_i} - \frac{\partial b_{ji}}{\partial \mathbf{r}_i}V_A(r_{ij}),\ \frac{\partial b_{ji}}{\partial \mathbf{r}_i} = p(\varsigma_{ji})\cdot\frac{\partial \varsigma_{ji}}{\partial \mathbf{r}_i};$$

$$\frac{\partial U_{jk}}{\partial \mathbf{r}_i} = -\frac{\partial b_{jk}}{\partial \mathbf{r}_i}V_A(r_{jk}),\ \frac{\partial b_{jk}}{\partial \mathbf{r}_i} = p(\varsigma_{jk})\cdot\frac{\partial \varsigma_{jk}}{\partial \mathbf{r}_i}.$$

The Algorithm WA has the following features:

1. The neighbor list must be calculated with double cut radius (two coordination radius) for correct calculation of $\partial b_{ji}/\partial \mathbf{r}_i$ as

$$b_{ji} = \left[1+\beta^n\varsigma_{ji}^n\right]^{-\frac{1}{2n}} = \left[1+\beta^n\left(\sum_{k(\neq j,i)} G_j(\theta_{jik}) f_{jk}(r_{jk})\exp\left[\lambda_{3,jik}^m (r_{ji}-r_{jk})^m\right]\right)^n\right]^{-\frac{1}{2n}}$$

has the functional dependence from the $f_{jk}(r_{jk})$ term, i.e. the whole neighbor list of the atom i needs also to include all atoms j with $r_{ij} > R+D$ and $r_{jk} < R+D$.

2. Additional calculations of b_{ij}, b_{ji}, $\partial b_{ij}/\partial \mathbf{r}_i$, $\partial b_{ji}/\partial \mathbf{r}_i$ and $\partial b_{jk}/\partial \mathbf{r}_i$ must be done inside of each thread. So we have more computations than in the Algorithm A.
3. The present version of the Algorithm WA has more local variable per kernel than the Algorithm A. The limitation of register number per thread (64 registers for NVidia Fermi architecture) results in significant register spilling.

So we need extra computations and global memory transactions for Tersoff potential for the sake of eliminating of atomic operations.

GPU algorithms for embedded-atom potential

Many-body embedded-atom potential can also be implemented on GPU with atomic operations (Algorithm A):

- Algorithm A is assumed to run for N work-items in 1D index space работ (N GPU threads).
- Each thread i
  1. Calculates

$$U_i = \frac{1}{2}\sum_{j\neq i} U_{ij}^{pair}(r_{ij}) + E_i\left(\sum_{j\neq i} c_j(r_{ij})\right).$$

  2. Calculates

$$\mathbf{F}_i = \mathbf{F}_i - \frac{1}{2}\sum_{j\neq i}\left(\frac{\partial E_i}{\partial \rho_i}\frac{\partial c_j(r_{ij})}{\partial \mathbf{r}_i} + \frac{\partial U_{ij}^{pair}(r_{ij})}{\partial \mathbf{r}_i}\right).$$

  3. Calculates and sums with atomic operations all $\mathbf{F}_j$ (j are the numbers of neighboring atoms of atom i, $j\neq i$):

$$\mathbf{F}_j = \mathbf{F}_j + \frac{1}{2}\left(\frac{\partial E_i}{\partial \rho_i}\frac{\partial c_j(r_{ij})}{\partial \mathbf{r}_i} + \frac{\partial U_{ij}^{pair}(r_{ij})}{\partial \mathbf{r}_i}\right).$$

The more natural way is to compute embedded-atom potential without atomic operations:

- Algorithm WA is assumed to run for N work-items in 1D index space работ (*N* GPU threads).
- Each thread *i*
  1. Calculates

$$U_i = \frac{1}{2}\sum_{j\neq i} U_{ij}^{pair}\left(r_{ij}\right) + E_i\left(\sum_{j\neq i} c_j\left(r_{ij}\right)\right).$$

  2. Calculates

$$\mathbf{F}_i = -\sum_{j\neq i}\left(\frac{\partial E_j}{\partial \rho_i}\frac{\partial c_i}{\partial \mathbf{r}_i} + \frac{\partial E_i}{\partial \rho_j}\frac{\partial c_j}{\partial \mathbf{r}_i} + \frac{1}{2}\frac{\partial U_{ij}^{pair}\left(r_{ij}\right)}{\partial \mathbf{r}_i}\right).$$

Results

The performance of the described algorithms for Tersoff and embedded-atom potentials is evaluated via the following computer systems:

- CPU: Intel Core i5 760, GPU: GeForce GTX 470, Windows 7, Visual Studio 2008, *.
- CPU: Intel Xeon E5450, Linux, **.
- CPU: Intel Xeon X5650, GPU: Tesla M2050, Linux, ***.

For the following testing configurations AMD APP SDK 2.8 was used as a realization of OpenCL CPU platform and CUDA Toolkit (CUDA 4.2.1) as a realization of OpenCL GPU platform. All GPU computations and comparisons were done on one node of computer system mostly with single precision arithmetic.

The main point of the comparison is to find the impact of the atomic operations with memory on the performance of calculation of the forces due to many-body interaction between the atoms. The results of benchmarks are summarized in Table 1. The corresponding optimal algorithm is highlighted with italic (lower is better).

Table 1. Performance of the OpenCL algorithms for many-body interaction

Execution time ratio \ Number of atoms	**1000**	**2000**	**4000**	**8000**	**16000**
Tersoff **CPU**, Algorithm WA / *Algorithm A**	33.43	33.49	33.54	32.8	33.68
Tersoff **GPU**, Algorithm WA / *Algorithm A**	9.67	9.62	10.42	10.68	11.94
Tersoff **GPU**, Algorithm WA / *Algorithm A****	48.89	54.25	55.37	60.29	70.55
EAM **CPU**, *Algorithm WA* / Algorithm A*	0.66	0.86	0.76	0.69	0.49
EAM **GPU**, *Algorithm WA* / Algorithm A*	0.51	0.48	0.47	0.48	0.49
EAM **GPU**, *Algorithm WA* / Algorithm A***	0.63	0.41	0.09	0.23	0.20

The next point is a speedup comparison. The following variants of energy and force computation algorithms are considered:

- Serial CPU algorithm;
- OpenCL algorithm with atomic operations;
- OpenCL algorithm without atomic operations.

The speedup of OpenCL algorithms is estimated by comparison of their execution time with the corresponding execution time of the serial algorithms (Fig. 1 and 2).

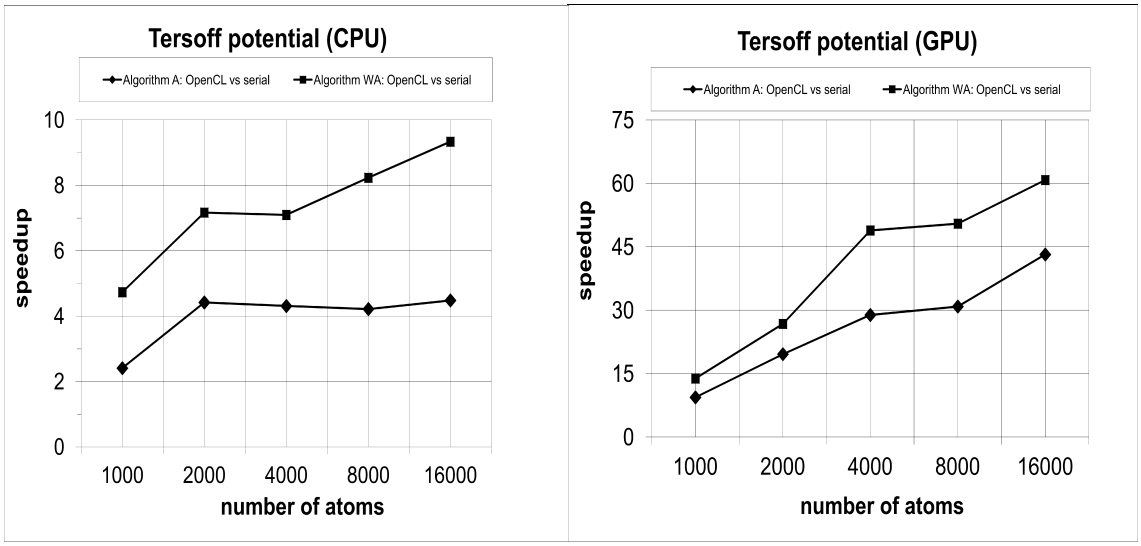


Fig. 1. Comparison of speedup of Tersoff potential (evaluated on the testing platform *)

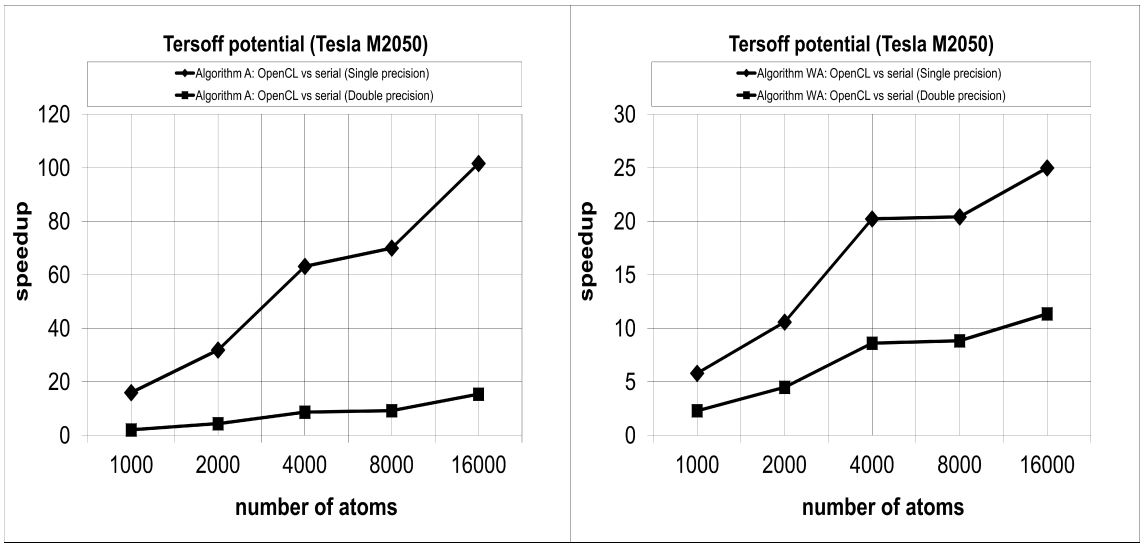


Fig. 2. Comparison of speedup of EAM potential (evaluated on the testing platform *)

One can see that Algorithm WA for the Tersoff potential provides greater acceleration than Algorithm A, both for the CPU and GPU. That result is mostly associated with a large number of arithmetic operations in the Algorithm WA. However, the higher value of the acceleration relative to the serial implementation does not mean the optimality of the algorithm itself according to the Table. 1. Algorithm A for Tersoff potential works 10 times faster on average in spite of atomic operations with memory.

Embedded-atom potential contains less arithmetic operations compared to Tersoff potential. The algorithms with and without atomic operations have approximately the same computational complexity. In that case, the effect of atomic operations with memory results in decrease of the performance. Thus, embedded-atom potential can be effectively implemented without atomic operations and the optimal algorithm is similar to that for pair potentials. So the result for embedded-atom potential is completely opposite to Tersoff potential (Table 1).

The last point is to note the influence of double precision arithmetic on the performance of GPU computations. The corresponding comparison for Tersoff potential (evaluated on the testing platform***) can be seen on Fig. 3. The speedup of algorithm depends on the floating point precision used

for performance evaluation and the hardware. For Algorithm A large values of speedup can be obtained for single precision arithmetic. For double precision the speedup is not as large and is similar to the Algorithm WA. For Algorithm WA the difference in speedup between single and double precision is not so clearly pronounced.

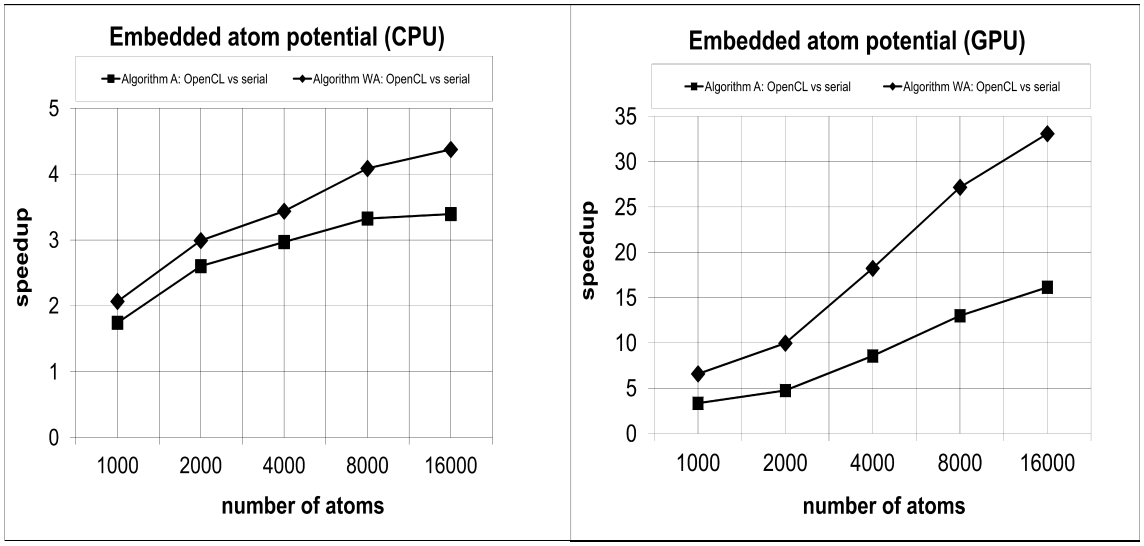


Fig. 3. The influence of floating point precision on the performance of Tersoff potential

Conclusions

Adequate modeling of nanostructures using classical methods requires accounting effects of the many-body interaction which leads to the increase of the computational complexity of the algorithms. Using GPU is the effective way of their acceleration. The performance of algorithms is greatly affected by the memory access patterns, floating-point arithmetic and some hardware features. Atomic operations are one of such patterns that are good or bad depending on the algorithm. As for the interatomic potentials one should always take into account their types and the architecture of the computing system for implementation. Some general notes can be seen as a conclusion.

Tersoff potential requires a significant amount of computation compared to pair potentials and embedded-atom potential. Using atomic operations is the optimal approach to the implementation of Tersoff potential. That gives significant reduction of the computational complexity. So the theoretical analysis and performance evaluation show that using atomic operations with memory does not always lead to poor performance. That is the case for Tersoff potential and can be assumed for other bond order potentials.

Embedded-atom potential gives the opposite result. The best way to compute EAM does not mean using atomic operations and critical sections. In that work such implementation was given just to show that difference.

Performance evaluation shows a significant acceleration of the GPU algorithms for many-body potentials. The average performance of the OpenCL algorithms is about 50 times compared to serial implementations. Using GPU is a good way for accelerating algorithms for interatomic interaction and particularly for the many-body interactions. OpenCL technology is universal tool to run the same algorithm on different hardware architectures. GPU algorithms for interatomic potentials can be used as building block of general molecular dynamics implementation for supercomputer systems.

References

Allen M. P., Tildesley D. J. Computer simulation of liquids. Oxford University Press, New York, 1990.

Anderson J. A., Lorenz C. D., Travesset A. General Purpose Molecular Dynamics Simulations Fully Implemented on Graphics Processing Units // J. Comp. Phys. — 2010. — 227(10). — P. 5342–5359.

Bielajew A. F. Fundamentals of the Monte Carlo method for neutral and charged particle transport. 2001. [online]: http://www-personal.umich.edu/~bielajew/MCBook/book.pdf

Brenner D. W. Relationship between the embedded-atom method and Tersoff potentials // Phys. Rev. Lett. — 1989. — 63(9). — P. 1022–1022.

Erkoc S. Empirical Many-Body Potential Energy Function Used In Computer Simulations Of Condensed Matter Properties // Physics Reports. — 1997. — 278(2). — P. 79–105.

Gaster B., Howes L., Kaeli D. R., Mistry P., Schaa D. Heterogeneous Computing with OpenCL // Morgan Kaufmann. — 2011. — 296 p.

Jason Sanders, Edward Kandrot. CUDA by Example: An Introduction to General-Purpose GPU Programming // Addison-Wesley Professional. — 2010.

Kumagai T., Hara S., Choi J., Izumi S., Kato T. Development of empirical bond-order-type interatomic potential for amorphous carbon structures // Journal of Applied Physics. — 2009. — 105(6). — P. 064310

Lebedeva I. V., Knizhnik A. A., Popov A. M. Potapkin B. V. Ni-Assisted Transformation of Graphene Flakes to Fullerenes // J. Phys. Chem. *C*. — 2012. — 116(11). — P. 6572–6584.

Murray S. Daw, Foiles S. The embedded-atom method: a review of theory and applications // Mat. Sci. Reports. — 1993. — 9.

Tersoff J. Modeling solid-state chemistry: Interatomic potentials for multicomponent systems // Phys. Rev. B. — 1989. — 39(8). — P. 5566–5568.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 559−563

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.63

Distributed dCache-based storage system of UB RAS

E. Yu. Kuklin[1,3,a], A. V. Sozykin[1,3], A. Yu. Bersenev[1,3], G. F. Masich[2]

[1] Institute of Mathematics and Mechanics UB RAS, Sofia Kovalevskaya Str. 16, Yekaterinburg, 620990, Russia
[2] Institute of Continuous Media Mechanics UB RAS, St. Academ. Koroleva 1, Perm, 614013, Russia
[3] Ural Federal University, St. Mira 19, Yekaterinburg, 620002, Russia

E-mail: key@imm.uran.ru[a]

Получено 10 октября 2014 г.

Abstract. — The approach to build territorial distributed storage system for high performance computing environment of UB RAS is presented. The storage system is based on the dCache middleware from the European Middleware Initiative project. The first milestone of distributed storage system implementation is includes the data centers at the two UB RAS Regions: Yekaterinburg and Perm.

Keywords: network attached storage, parallel NFS, GRID, HPC

Распределенная система хранения УРО РАН на основе dCache

Е. Ю. Куклин[1,3], А. В. Созыкин[1,3], А. Ю. Берсенёв[1,3], Г. Ф. Масич[2]

[1] Институт Математики и Механики УрО РАН, Россия, 620990, г. Екатеринбург, ул. Софьи Ковалевской, д. 16
[2] Институт Механики Сплошных Сред УрО РАН, Россия, 614013, г. Пермь, ул. Акад. Королёва, д. 1
[3] Уральский Федеральный Университет, Россия, 620002, г. Екатеринбург, ул. Мира, д. 19

Представлен подход к созданию территориально-распределенной системы хранения данных для нужд среды высокопроизводительных вычислений УрО РАН. Система основывается на промежуточном программном обеспечении dCache из проекта European Middleware Initiative. Первая очередь реализации системы охватывает вычислительные центры в двух регионах присутствия УрО РАН: г. Екатеринбург и г. Пермь.

Ключевые слова: сетевые системы хранения, parallel NFS, ГРИД-технологии, параллельные вычисления

Supported by the grant of UB RAS 15-7-1-26 and by the RFBR grant 14-07-96001r_ural_a.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 559–563 (Russian).

Introduction

Works on creating a distributed high-performance computing environment based on GRID technologies are under way at the Ural Branch of the Russian Academy of Sciences. One of the main components of this environment is a distributed data storage system, which aims at integrating storage systems in the Ural regions [Goldshtein et al., 2013]. The system connects various resources, such as computing clusters, supercomputers and experimental setups of Ural institutes and universities. The participants of this storage system project are Institute of Mathematics and Mechanics in Yekaterinburg (computational recourses and storage system) and Institute of Continuous Media Mechanics in Perm (backbone networks).

Middleware selection

The approach to building territorial distributed storage system based on the dCache [dCache..., 2015] middleware from the European Middleware Initiative (EMI) [EMI..., 2015] project is presented. dCache is a distributed storage system focused on storing large amounts of experimental data. It can run on commodity hardware and allows the construction of storage facilities in hundreds of terabytes, with all the files in it logically organized into a single virtual file system tree. In addition, dCache assumes a simple extension of your storage by adding new nodes and can work with tape libraries. dCache supports a wide range of access protocols. Together with common standard protocols FTP, WebDAV, NFS, grid protocols SRM and GRIDFTP, as well as its own protocol dCap is used.

EMI includes three projects for building distributed storage systems: dCache, Disk Pool Manager and Storage Resource Manager. They applied in different projects to build GRID infrastructures, includes WLCG [WLCG..., 2015]. For UB RAS storage dCache was chosen, as it provides support for both GRID and Internet protocols, has a high quality documentation, as well as it is easy to install and administer. An additional reason for choice was the fact that there is a dCache-based store at the Joint Institute for Nuclear Research, and the only one distributed Tier1 center — Nordic Data Grid Facility, which storage nodes are located in different Nordic countries [Behrman et al., 2008].

Current stage

The first milestone of the implementation of distributed storage system running dCache 2.6 has been currently completed. Figure 1 shows its general plan. For data storing were selected servers by Supermicro. Now there are 4 servers running Scientific Linux 6.5 with usable capacity of 210 TB. We decided to store our data in the XFS file system by Silicon Graphics. It has shown good results in tests for read/write/access to data, and as well as EXT4, is native for Linux and actively developing file system. For example, our test using bonnie++ [Bonnie..., 2015] showed that there is no the distinct advantages between file systems (except random delete, what exactly our system does not imply). So, taking into account the turn of RHEL 7 to XFS, it was decided to leave the XFS.

The storage nodes are located in two computing centers: in Institute of Mathematics and Mechanics in Yekaterinburg (3 nodes) and Institute of Continuous Media Mechanics in Perm. The computing centers are separated by a 450 km distance and are joined by a dedicated communication channel. Channel performance provides by DWDM equipment of ECI-Telecom Company. Installed platforms allows transmission of two λ-channels with 10 Gigabit Ethernet technology. Setup and maintenance of backbone network is engaged by Laboratory of Telecommunication and Information Systems in Institute of Continuous Media Mechanics.

As shown at the storage system plan, Institute of Mathematics and Mechanics has at its disposal a supercomputer "URAN", which occupies the 9th position in Top50 CIS; Institute of Continuous Me-

dia Mechanics also has its own cluster. Connection to storage system was performed using the NFS protocol version 4.1 with Parallel NFS [Parallel NFS..., 2015]. On cluster's nodes mounted, as dCache structure suggests, one of the storage system servers (broker host). Parallel NFS allows direct connection between computational and storage nodes for data transfers, removing the traditional NFS-server bottleneck.

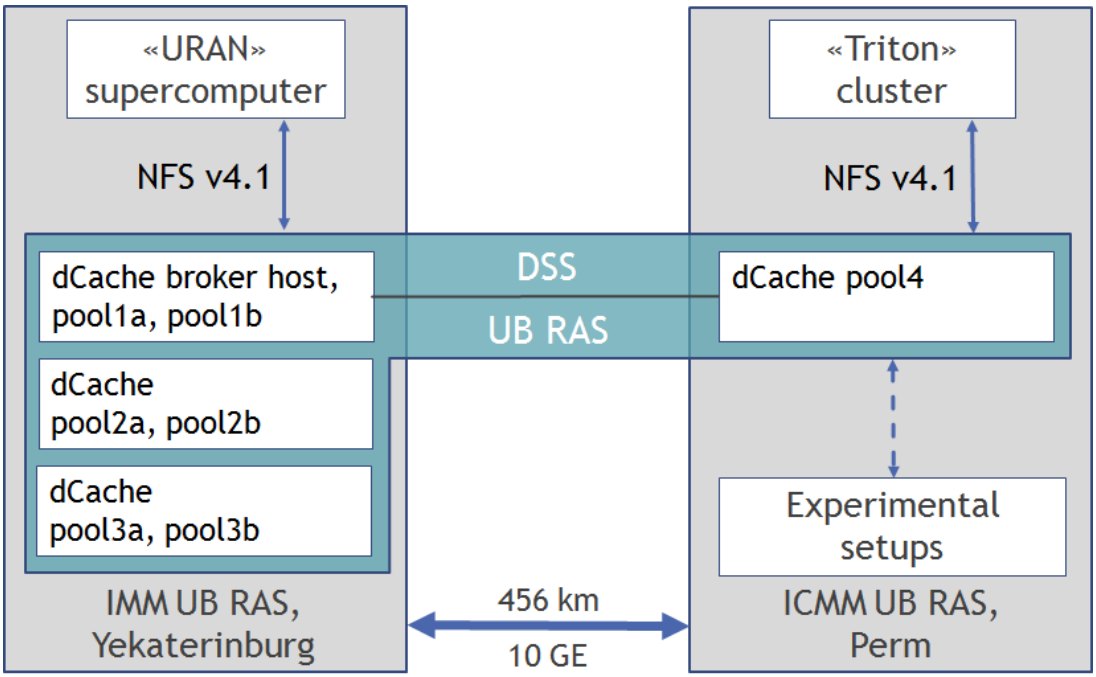


Fig. 1. General plan of UB RAS DSS

There are some unresolved for now drawbacks: how NSF will behave when transmitting data over a long distance, or how to optimize the OS network stack configuration for 10-Gigabit networks (the efficiency of Intel network cards on storage nodes leave much to be desired). Also, there are problems with mounting dCache on servers with a custom build kernel.

Network benchmarks

First of all, testing has shown that the evident bottleneck is the gigabit network between the storage system and supercomputer "URAN". 10-Gigabit equipment (with switch by Extreme Networks) for the internal network allowed fixing it, and significantly increasing the speed of data exchange. Figure 2 shows the results of the data recording via a dedicated channel between the data centers before and after attaching an additional storage node in Perm. It can be seen that the presence of the local server slightly improves storage system performance. These results were obtained with the IOR benchmark [IOR..., 2015].

Figure 3 displays an attempt to optimize the TCP/IP stack parameters on storage nodes. Our colleagues from Perm conducted tests using iperf [Iperf..., 2015] and various optimization algorithms. As can be seen, the best result was given by bic algorithm, but it could not overcome the threshold of 6 Gbit/s. This is expected has been given by Intel network cards installed on the servers.

dCache allows to create multiple copies of files that balance loading between the nodes and optimize the use of storage space. Figure 4 shows that automatic replication does not affect the system performance, so with the hardware RAID arrays, it will increase the reliability of the system, because we do not have at our disposal a tape library for archiving data.

Figure 2. Performance recording on DSS from ICMM client with and without an additional storage node in Perm

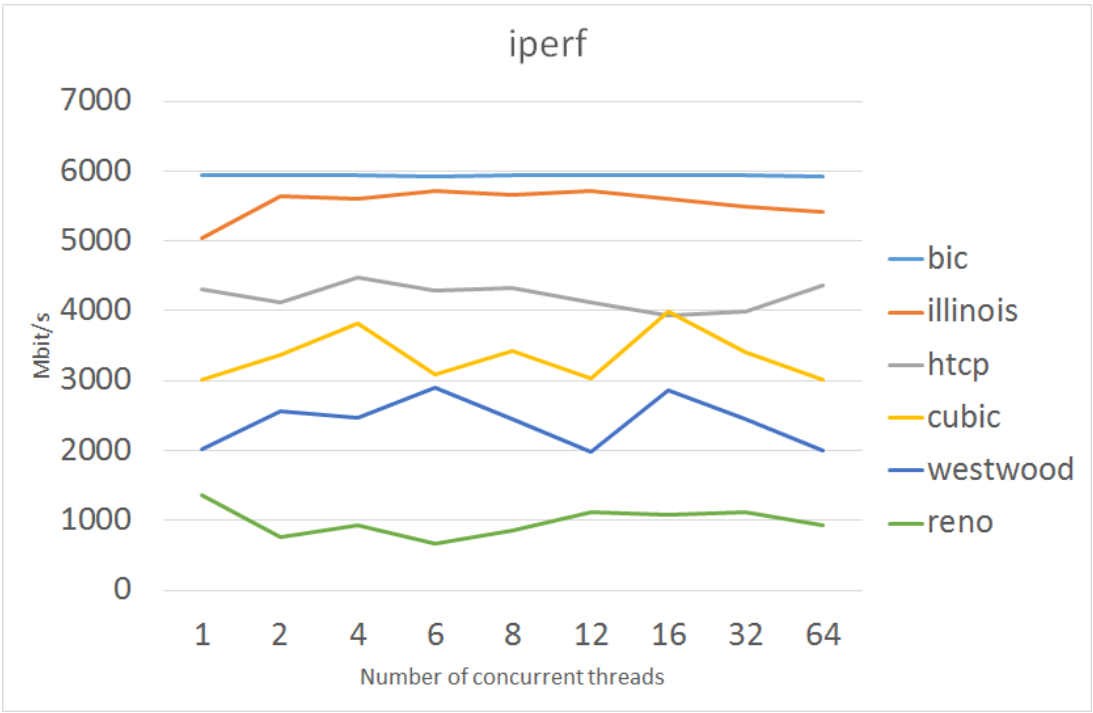


Fig. 3. Comparison of congestion control algorithms for TCP/IP networks

Conclusion

Now Institute of Mathematics and Mechanics conducted performance and fault tolerance benchmarks of considered storage system. Among them was investigated the effect of additional storage node and replication on the performance of the storage system. Also was selected congestion control algorithm to optimize the network parameters. In the nearest future we plan to pay more attention on security and monitoring. The next stage of implementation will be increasing of storage system

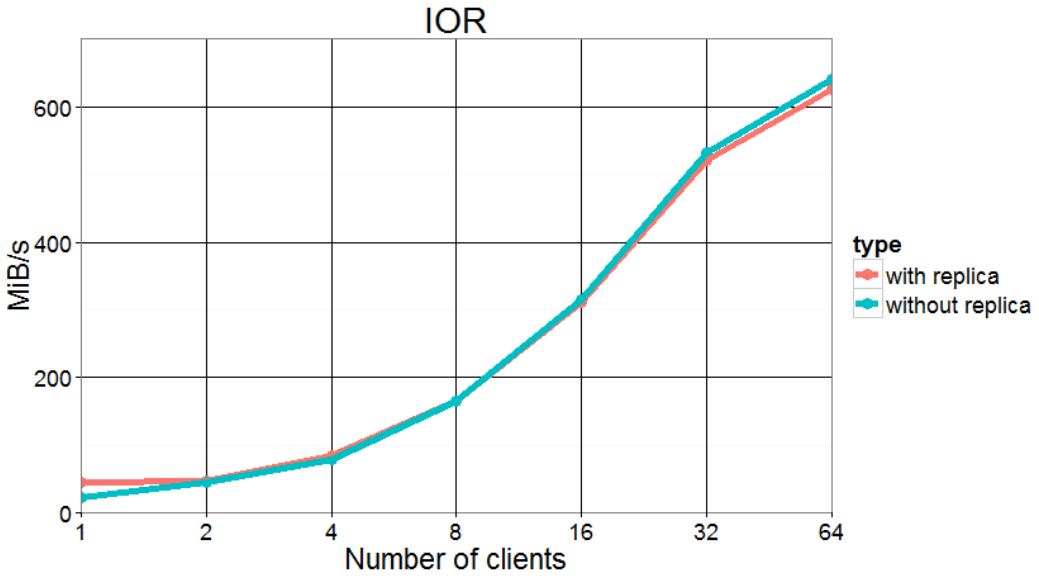


Fig. 4. Influence of replication on DSS performance

capacity and attaching the experimental setups in Institute of Continuous Media Mechanics in Perm and Ural Federal University. Data obtained from them will be recorded to the storage system and processed remotely by supercomputer "URAN", including the ability of process in real time and control the experiments. Further the connection of entire distributed computing environment to international GRID infrastructure Worldwide Large Hadron Collider Computing Grid is planned. We can provide for general use the resources of distributed storage system and computing clusters resources.

References

Behrman G. , Fuhrmann P. , Gronager M. and Kleist J. A distributed storage system with dCache // Journal of Physics: Conference Series, 119, 2008;

Bonnie++ [online] // — 2015. — URL: http://www.coker.com.au/bonnie++ (дата обращения: 16.01.2015);

dCache web-page [online] // — 2015. — URL: http://www.dcache.org (дата обращения: 16.01.2015);

EMI - European Middleware Initiative [online] // — 2015. — URL: http://www.eu-emi.eu (дата обращения: 16.01.2015);

Goldshtein M.L., Sozykin A.V., Masich G.F., Masich A.G. "Computing resources of UB RAS. Status and prospects." // Parallel Computational Technologies (PCT'2013): proceedings of the international scientific conference. Chelyabinsk, publishing center of SUSU, 2013. P. 330-337;

IOR [online] // — 2015. — URL: http://sourceforge.net/projects/ior-sio (дата обращения: 16.01.2015);

Iperf [online] // — 2015. — URL: https://iperf.fr (дата обращения: 16.01.2015);

Parallel NFS [online] // — 2015. — URL: http://www.pnfs.com (дата обращения: 16.01.2015) ;

WLCG — Worldwide LHC Computing Grid [online] // CERN, Switzerland — 2014 — URL: http://wlcg.web.cern.ch (дата обращения: 16.01.2015);

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 565−571

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Defining volunteer computing: a formal approach

A. Cs. Marosi[a], R. Lovas[b]

Institute for computer science and control, Hungarian Academy of Sciences,
1518 Budapest, P.O.Box. 63., Hungary

E-mail: [a]marosi.attila@sztaki.mta.hu, [b]lovas.robert@sztaki.mta.hu

Получено 28 января 2015 г.

Volunteer computing resembles private desktop grids whereas desktop grids are not fully equivalent to volunteer computing. There are several attempts to distinguish and categorize them using informal and formal methods. However, most formal approaches model a particular middleware and do not focus on the general notion of volunteer or desktop grid computing. This work makes an attempt to formalize their characteristics and relationship. To this end formal modeling is applied that tries to grasp the semantic of their functionalities — as opposed to comparisons based on properties, features, etc. We apply this modeling method to formalize the Berkeley Open Infrastructure for Network Computing (BOINC) [Anderson D. P., 2004] volunteer computing system.

Keywords: BOINC, ASM, Formalism, Volunteer Computing

Определение добровольных вычислений: формальный подход

А. К. Мароши, Р. Ловаш

Институт информатики и управления, Венгерская Академия Наук (MTA SZTAKI),
Венгрия, 1518, г. Будапешт, почтовый офис. 63.

Добровольные вычисления напоминают частные desktop гриды, тогда как desktop гриды не полностью эквивалентны добровольным вычислениям. Известны несколько попыток отличить и категоризировать их, используя как неофициальные, так и формальные методы. Однако, наиболее формальные подходы моделируют специфическое промежуточное ПО (middleware) и не сосредотачиваются на общем понятии добровольного или desktop грид. Эта работа и есть попытка формализовать их характеристики и отношения. Для этой цели применяется формальное моделирование, которое пытается охватить семантику их функциональных возможностей — в противоположность сравнениям, основанным на свойствах, особенностях, и т. п. Мы применяем этот метод моделирования с целью формализовать добровольную вычислительную систему Открытой Инфраструктуры Беркли для сетевых вычислений (BOINC) [Anderson D. P., 2004].

Ключевые слова: BOINC, ASM, Формализм, Добровольные Вычисления

The research presented in this paper was supported by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 312297 (IDGF-SP).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 565–571 (Russian).

1. Introduction

Desktop Grids (DGs) and Volunteer Computing (VC) utilize the idle computing cycles of desktop computers to solve embarrassingly parallel type of compute-intensive problems, such as Monte Carlo simulations or master-worker type applications. Publicly operated ones using mostly volunteer resources are referred as volunteer computing, or recently as "crowd computing". Contrary, private desktop grids are operated within an organization (i. e., university or company) using the computing resources and applying their local policies. There are several attempts to distinguish and categorize DGs and VC using informal and formal methods [Characterizing and Classifying Desktop Grid, 2007; A Taxonomy of Desktop Grids..., 2008; Wang Y., He H., & Wang Z., 2009]. However, most formal approaches model a particular middleware and do not focus on the general notion of volunteer or desktop grid computing. In this work formal modeling is applied that tries to grasp the semantic of their functionalities — as opposed to comparisons based on properties, features, etc. The result of this work is a formal model of BOINC that aims at serving as a foundation for formalizing other volunteer computing systems and helps categorizing existing middleware. The model is developed using the abstract state machines (ASMs) framework and builds on a model that formalized (service) Grid Computing in general. The paper is organized as follows: the next section summarizes the Abstract State Machines framework. Section 3 discusses related work. Section 4 details the formal model for BOINC, and finally section 5 concludes the paper.

2. Abstract State Machines

The Abstract State Machine is a mathematically well-founded framework for high-level system design and analysis [Borger E. & Stark R. F., 2003] originally introduced as evolving algebras by Gurevich [Gurevich Y., 1993]. ASM allows hiding easily the non-important details at the high-level design phase by formulating the model on a conceptional level rather than based on implementation details and attributes. Lower detail characteristics can be added to the models later gradually. It is an agent based modeling system where the system is described from the perspective of an agent. In ASM states are represented as algebras, i. e., basic sets (called universes) with functions and relations interpreted on them. A signature (or vocabulary) is a finite set of function names each with fixed arity. It also contains the usual Boolean operators (e.g., $\wedge,\vee$) and the symbols true, false, = and undef. A state S of signature Υ is a nonempty set X together with interpretations of function names in Υ on X. X is called the super universe. A nullary function name is interpreted as an element of X this corresponds to the notion of variables. An r-ary function name is interpreted as a function from $X^r \to X$. A location of S is a pair l=(f,x), where f is a function name of arity r in vocabulary Υ and x is an r-tuple of elements of X. The element $f(x)$ is the content of location l. An update is a pair z=(l,y), where l is a location and y is another element of X. Firing x at state S means putting y into the location l while other locations remain unchanged. The resulting state is S' (the sequel of S), thus the interpretation of a function f at argument x has been modified producing a algebra, i. e., a new state. The special nullary Self-function is used to represent the agent and also allows to identify itself amongst other agents. Different agents interpret it differently. This Self-function can never be the subject of updates. ASM models are defined as a set of transition rules.

3. Related work

Choi et al. [Characterizing and Classifying Desktop Grid, 2007; A Taxonomy of Desktop Grids..., 2008] state that DGs have received increased attention for executing high throughput workloads as resources are becoming less expensive. They argue that DGs are different from service grids in many aspects, but there is no taxonomy or survey on DGs. They categorize DGs based on organiza-

tion (centralized or distributed), platform, scale (Internet or LAN) and resource providers (volunteer or enterprise) characteristics. They also compare VC (they refer it as volunteer desktop grids) and DGs (referred as enterprise desktop grids by the authors) to service grids on an informal per attribute basis, and provide no insight what the relation between the DGs and service grids could be. Wang et al. [Wang Y., He H., & Wang Z., 2009] uses a formal method inspired by Mobile Ambients to build a formal model for VC by identifying the different roles for hosts in VC and describing their relation and interaction. The model is derived mainly based on the characteristics of XtremWeb(-HEP) [Computing on large-scale distributed systems..., 2005]. They state that their model can help to lay a strong foundation for further research on formalisms of VC. However their model does not distinguish between DGs and VC and seems generic in an extent that most DG systems could fit it as well. Also it seems their generic model is derived from a single specific middleware: XtremWeb-HEP (XWHEP). They do not validate their assumption that XWHEP indeed is a volunteer computing middleware. For example in [Characterizing and Classifying Desktop Grid, 2007] Choi et al. state: "Lack of trust: In Desktop Grid, anonymous nodes can participate as a resource provider. Some malicious resource providers tamper with the computation and then return corrupted results. *A scheduler should guarantee the correctness of results". In their comparison of volunteer and enterprise DGs result certification is listed for VC. However based on the documentation* [XtremWeb-HEP documentatio, 2014] XWHEP does not provide this functionality: "Result certification: The XWHEP middleware does not propose anything on this field. It is the end user responsibility to verify the results of her jobs". This contradicts the assumptions for the model by Wang et al. [Wang Y., He H., & Wang Z., 2009].

A formal model for (service) grids based on ASM was presented by Nemeth et al. in [Németh Z., & Sunderam V., 2003] and was refined later by Kertesz et al. [Kertész A., & Németh Z., 2009]. Originally Nemeth et al. compared Grids with other distributed systems based on operational differences. They proposed a definition for Grids based on (runtime) semantics of the systems rather than comparing their static characteristics.

In their ASM model Nemeth et al. consider an application (members of universe APPLICATION) as consisting of several processes (universe PROCESS). All processes are owned by a user (USER) and need resources to perform work. Abstract resources are present in resource request and are represented by the ARESOURCE universe, while the PRESOURCE universe represents physical resources allocated to processes. Processes execute a specific task (universe TASK). The physical representation of a task is a static realization of a running process, thus it must be present on the same node (universe NODE) where the process is. This is represented by the *installed: TASK* × *NODE*→ {*true, false*} relation. Nodes, tasks and resources have certain attributes (universe ATTR). A subset of ATTR is the architecture type represented by universe ARCH. The relation *compatible: ATTR* × *ATTR* → {*true, false*} denotes whether to attributes are compatible according to some reasonable definition. A user can login to certain nodes if *CanLogin: USER* × *NODE*→ {*true, false*} evaluates to true. A user is authorized to use given resource if the *CanUse: USER* × *PRESOURCE*→ *{true, false}* relation evaluates to true. The model is centered on processes and their life cycle is described by their states using the state *PROCESS* → *{running, waiting, receive_waiting}* function. In grids the resource requests can be satisfied from various nodes in various ways. The user and the application has no information about the state of the pool of resources a new agent executing module is needed that handles the mapping between them, thus the *resource mapping* functionality is introduced that provides the mapping via the *PROCESS* × *ARESOURCE* → *PRESOURCE* function. It does not specify how resources are actually chosen (it is rather an implementation detail), only assures that compatible physical resources are mapped to each resource request using the compatible: *ATTR* × *ATTR* → *{true, false}* relation. In grids the fact that a user can access the pool of resources does not mean that she can login to the nodes providing the resources $\forall u \in USER, \forall r \in PRESOURCE$, $\forall n \in NODE : CanUse(u,r) \not\Rightarrow CanLogin(u,n)$. Resources are granted by the operating system to processes on the same node, thus a process of the application — belonging to the user — must be present on the node. However users are not authorized to login and start processes. This contradiction is resolved by providing a mapping between the real person, the user who has credentials to access to the

resources of the pool (*globaluser*) and the user — not necessarily a real person — who has login rights on the node (*localuser*). The *user mapping* functionality provides this mapping. The model is a distributed multi-agent ASM where the agents are processes, i. e., elements of the PROCESS universe. It is depicted from the perspective of the processes, where the *Self*-function is represented as $p \in PROCESS$, i. e., different agents interpret p differently.

4. A formal model for BOINC

The here presented model for BOINC is part of a series of models introduced in [Marosi A. Cs., & Nemeth Z., 2013]. It builds on a previous model for volunteer computing (VC) in general, which is shown as MVC-VOTE on Fig. 1/c. The firs model in the series ($M_{GROUND\text{-}DG}$) is based on the model presented in [Németh Z., & Sunderam V., 2003; Kertész A., & Németh Z., 2009]. In this paper only the BOINC model (M_{BOINC}) is discussed, but where necessary details from the previous models are included. The model presented here is a multi-agent ASM model where agents are jobs (i. e., elements from the JOB universe). The nullary *self* function $j \in JOB$ allows an agent to identify itself among other agents. The different agents interpret it differently. Its rules form a module, i. e., a single-agent model that is executed by each agent. Due to space constraints the model transition rules — including the initial state — are going to be detailed in a future paper.

BOINC is a widely used volunteer computing framework with more than 70 public deployments around the world. A deployment of BOINC is generally referred as a project. The formal model presented here aims to capture the major semantics of BOINC as follows: (i) BOINC follows centralized client-server architecture. (ii) Applications cannot be submitted as part of jobs. Jobs must refer a previously at the project deployed application. An administrator must register applications by hand at the project application repository. (iii) BOINC implements a result certification mechanism based on comparing returned finished job instances. Result validators must be supplied on a per application basis. It is an application specific task to determine whether to job instances can be considered as matching or not. (iv) Its owner based on a per application basis can restrict access to a host. Each donor is able to filter the applications of the different BOINC project she contributes to. Finally (v) as reward and incentive donors are awarded virtual credits for each job instance they successfully complete based on the amount of contributed processor time.

Fig. 1/a summarizes the universes for the model only the new and changed components compared to [Németh Z., & Sunderam V., 2003; Kertész A., & Németh Z., 2009] are discussed here. The PLATFORM universe is introduced to represent the different preset combination of operating system and system architecture requirements of applications in BOINC. The REPOSITORY universe represents all application repositories where applications are deployed. Other new universes are as follows. USER: A user of BOINC is the entity that submits jobs and retrieves results. There might be multiple users each responsible for their own jobs. NODE: Umbrella term for user interfaces, managers and hosts (see below). UI: user interface (UI). A node type from where users can submit jobs to BOINC. It acts as a gateway and the pool of resources can be accessed through it. MANAGER: The main component of a desktop grid. It is a type of node that manages resources and allocates Jobs to hosts. Jobs are submitted through UIs to managers. HOST: provides physical resources (PRESOURCE) for jobs and thus, executes them. Hosts are computers of a lab, office and etc. resources for BOINC. The worker component is installed on the hosts. This worker acts as a handler on behalf of the host for BOINC. Since all hosts have workers installed these won't be distinguished in the model rather, only the host referenced with different context. TASK: The physical representation of a job installed on a host. All processes of a specific job executing on a host are represented by a task. A unit of work represented by the UNITOFWORK universe incorporates all data and metadata that can be specific for a job (e.g., command line parameters or input data and required libraries). The DONOR universe is introduced to represent the owners of the hosts who donate theirs to the BOINC project (i. e., *the volunteers*). Fig. 1/b maps the universes of the model to their counterparts in BOINC. Based on the identified major semantics the following functionalities compose the model:

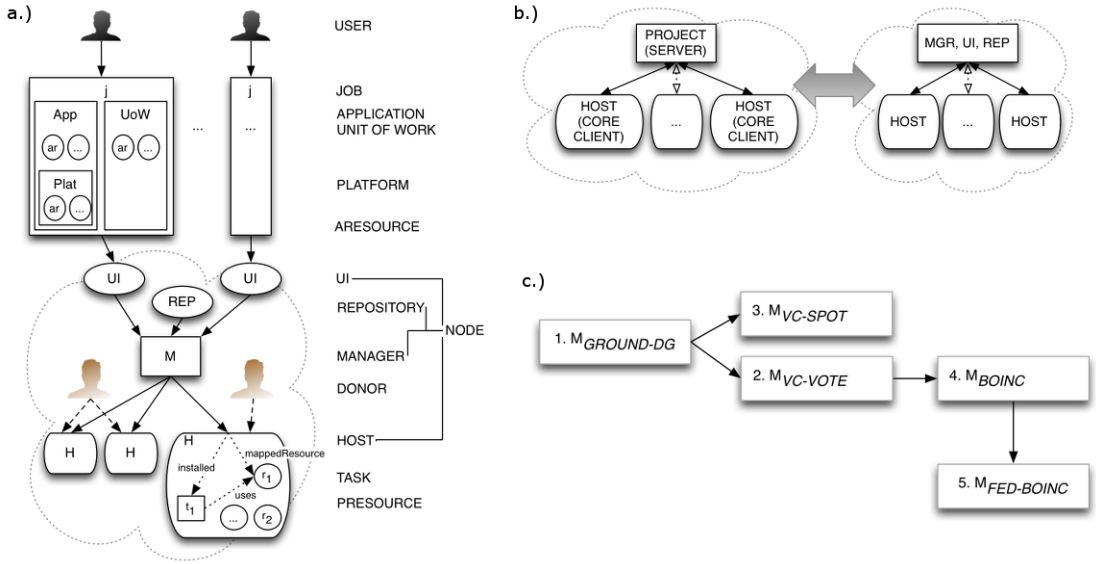


Fig. 1 a.: universes used in the model, b.: the components used in the model and their counterparts in BOINC, and c.: the place of the BOINC model in the series of formal models for DGs and VC

(1) *Resource mapping*. Resource requests of applications are consolidated into platforms in BOINC. During resource mapping it must be ensured that the host supports one of the platforms the application has available. Applications can have multiple implementations, each for a different platform, thus the selected platform is rather mapped to the job instead of the application. Still *unitofworks* have resource requests and mapped resources unchanged. The *supportsPlatform: APPLICATION* × *PLATFORM* → *{true, false}* function tells if an application has an instance for the given platform. This must platform must match the platform the host is reporting. The mapped platform of the job is represented by the mappedplatform: *JOB* → *PLATFORM* function. The platfotm: *{HOST, JOB}* → *PLATFORM* returns a platform the entity supports (there is no restriction that only a single one can be supported). The donor has the privilege to select allowed applications for her hosts as she wishes. BOINC projects usually run a single application aimed at solving some (grand) scientific challenge. However there are *umbrella* projects that host different applications. In this case the donor is given the freedom to disable application that she does not wish to support by accepting their jobs. This is represented by the *appAllowed: DONOR* × *APPLICATION* → *{true, false}* relation.

(2) *User abstraction*. BOINC provides a client program — the worker — that is installed every host and provides the *global user* to *local user* mapping. The worker acts on behalf of the host as a handler for the physical resources and provides the local part for the user mapping. In BOINC the *globaluser* → *localuser* mapping is straightforward since all users of BOINC have access to all hosts. The only restriction is imposed by the allowed applications list of donors.

(3) *Delayed jobs*. BOINC relies on non-dedicated, volatile resources, thus the possibility when a job is delayed for any reason (or even lost) must be taken in account. The cause for such delay can be numerous, e.g., resource requirements cannot be satisfied, or the host the job is mapped is claimed by its owner or is permanently shut down. This is represented by the high-level *jobDekayed: JOB*→ *{true, false}* monitored function. In BOINC a deadline is set for each job (result) to finish. Once the deadline passes the job is overdue and it is aborted by sending the ABORT event to the job: event (*j*):=ABORT.

(4) *Mapped resources become unavailable*. Owners of the resources are prioritized in BOINC and thus resources mapped and assigned to jobs can get unavailable for shorter or longer periods of time (i. e., the owner claims her computer). Here the presource: *JOB* → *PRESOURCE* function denotes the mapped physical resources of the job. The $pr \in PROCESS, h \in HOST : provides(pr, h) = false$ function denotes that the host h (through the worker software) can not provide (some or all of) the mapped

physical resources for the job any more. In this case the job must be suspended using the suspend event. The actual revocation (*users* (*j, presource (j*)):=*false*) happens in the state transition rule for suspend. It is assumed that the unavailability is a temporal state and not all of the resources need to be released, thus the job only needs to be suspended while so. If the resources are available again (represented by provides (*pr,h*)) = *true*) and the job is not using the resources (represented by *users* (*j, pr*):=*false*) task can be started.

(5) *Donor interaction*. Donors have direct control over the host(s) they donate to BOINC. They can suspend, resume computation in general and force to start, stop and abort specific tasks on their hosts. These are represented as events in the model. Specific events can be sent to jobs and the transition rules interpret these. Also *the maskEvents: JOB* $\times$ 2^{EVENT} $\times$ *MASKENTITY* $\rightarrow$ *{true, false}* relation allows to mask specific events from a given job. This allows e.g., if a user suspended all computation on her host, then no task will receive the start event, thus none is allowed to start. Not just the donor can generate events, but also the host, the MASKENTITY tuple represents these two entities (MASKENTITY ={donor, host}).

(6) *Result validation*. Comparing returned results is an application specific task in BOINC since it depends on the application which results can be considered matching and which not. The validator components provide this functionality. There must be a validator provided for each application. The comparison of results is performed by the checkset and checkpair functions. Each of the validator modules provides an implementation for these functions that check the successful results of a single application. BOINC also provides validators for some common cases that can be reused by applications, e.g., a bitwise validator that compares results bit-by-bit. The functions are used by the doValidation macro that provides the common functionality for all validator modules. Ultimately the module determines if the successful finished results indeed produced correct outputs. This state is represented by the validateState function of the results: valid, invalid or inconclusive if no decision could be made. Validation is achieved in two ways depending on if a representative result ("canonical" result denoted by the *canonicalresult: JOB* $\rightarrow$ *JOB function*) is already found for the work unit. If it was already found (*canonicalResult(j)* $\neq$ *undef*) then all new results are compared against it (using the checkPair function). The outcome of this comparison can be that either the results match thus the new one is valid (*validateState(r) = valid*) or they mismatch and the result is invalid. If there is no canonical result available yet then first it is checked whether there are enough successful results available to form a quorum and select one (the number is determined by the minQuorum function). Next a check is run on the set of results with checkSet. This function compares all results, decides whether they are valid or invalid and selects a canonical result from the valid ones. It is still a possibility that no canonical result is found (*validateState(r) = inconclusive* for all results). In this case the validation procedure is rerun later when a new successful result is returned. However if the limit for successful results is reached and still there is no consensus on the validation the work unit is considered failed. If there is no consensus but the limit is not reached then the targetNResults is increased for the work unit and a new result (job instance) will be created. If a canonical result is found by checkset then there is no need to send the unsent results to clients, thus abort event is generated for them. However already in progress results should be accepted (and validated) when they are returned. For valid results the validator grants credit (represented by the grantCredit function).

The here presented functionalities (with the transition rules and initial state) form a formal model of BOINC.

5. Conclusions

In this paper a formal model for BOINC was presented using the ASM method. The model (see M_{BOINC} on Fig. 1/c) is based on a series of models for Desktop Grid and Volunteer Computing [Marosi A. Cs., & Nemeth Z., 2013] and a formal model for Service Grids defined in [Németh Z., & Sunderam V., 2003; Kertész A., & Németh Z., 2009]. This model has three goals. First a validation for the previous models in the series: a real VC system can be modeled using them. Second it aims to be

a foundation for formalizing other volunteer computing systems. Finally the model acts as a basis for the next model in the series (see $M_{FED\text{-}BOINC}$ on Fig. 1/c), which models a novel method for federating distinct volunteer computing projects and enables workload sharing.

References

Anderson D. P. BOINC: A System for Public-Resource Computing and Storage. In R. Buyya (Ed.) // Fifth IEEE/ACM International Workshop on Grid Computing. — 2004. — P. 4–10.

Borger E. & Stark R..F. Abstract State Machines: A Method for High-Level System Design and Analysis. Secaucus, NJ, USA: Springer-Verlag New York, Inc. 2003.

Cappello F., Djilali S., Fedak G., Herault T., Magniette F., Neri V. & Lodygensky O. Computing on large-scale distributed systems: XtremWeb architecture, programming models, security, tests and convergence with grid // Future Generation Computer Systems. — 2005. — 21(3). — P. 417–437.

Choi S., Buyya R., Kim H., Byun E. & Baik M. A Taxonomy of Desktop Grids and its Mapping to State-of-the-Art Systems // ACM Computing Surveys. — 2008. — V. 1–61.

Choi S., Kim H., Byun E., Baik M., Kim S., Park C., & Hwang C. Characterizing and Classifying Desktop Grid // Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid '07). — 2007. — P. 743–748 (doi:10.1109/CCGRID.2007.31).

Gurevich Y. Evolving algebras: An attempt to discover semantics // Current Trends in Theoretical Computer Science. — 1993. — P. 1–27.

Kertész A., & Németh Z. Formal Aspects of Grid Brokering // Electronic Proceedings in Theoretical Computer Scienceю — 2009. — 14. — P. 18–31 (doi:10.4204/EPTCS.14.2).

Marosi A. Cs., & Nemeth Z. Two Sides of a Coin: Formalizing Volunteer and Desktop Grid Computing. In M. Bubak, M. Turała, & K. Wiatr (Eds.) // Proceedings of the Cracow Grid Workshop. — 2013. — P. 69–70. Kraków: ACK CYFRONET AGH.

Németh Z., & Sunderam V. Characterizing grids: Attributes, definitions, and formalisms // Journal of Grid Computing. — 2003. — P. 9–23.

Wang Y. He, H, & Wang Z. Towards a formal model of volunteer computing systems // In 2009 IEEE International Symposium on Parallel & Distributed Processing (p. 1–5). IEEE. — 2009 (doi:10.1109/IPDPS.2009.5161137).

XtremWeb-HEP documentation. http://www.xtremweb-hep.org/lal/doc/xwhep-intro-1.4.0.pdf, P. 30. Last accessed on 2014-02-01.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 573–579

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.94, 004.43

3D Molecular Dynamic Simulation of Thermodynamic Equilibrium Problem for Heated Nickel

V. O. Podryga[a], S. V. Polyakov

Keldysh Institute of Applied Mathematics Russian Academy of Sciences, Miusskaya sq., 4, Moscow, 125047, Russia,

E-mail: [a]pvictoria@list.ru

Получено 2 октября 2014 г.

This work is devoted molecular dynamic modeling of the thermal impact processes on the metal sample consisting of nickel atoms. For the decision of this problem the continuous mathematical model on the basis of the classical Newton mechanics equations is used, the numerical method using in the basis the Verlet scheme is chosen, the parallel algorithm is offered and its realization within the MPI and OpenMP technologies is executed. By means of the developed parallel program the investigation of thermodynamic equilibrium of nickel atoms system under the conditions of heating a sample to desired temperature was executed. In numerical experiments both optimum parameters of a calculation procedure, and physical parameters of analyzed process are defined. The received numerical results are well corresponding to known theoretical and experimental data.

Keywords: molecular dynamic simulation, nickel, EAM, temperature, thermostat, Newton mechanics equations, parallel algorithm and program, MPI and OpenMP technologies

Трехмерное молекулярно-динамическое моделирование термодинамического равновесия нагретого никеля

В. О. Подрыга, С. В. Поляков

[1] Институт прикладной математики им. М. В. Келдыша РАН,
Россия, 125047, г. Москва, Миусская пл., д. 4

Представленная работа посвящена молекулярно-динамическому моделированию процессов термического воздействия на металлический образец, который состоит из атомов никеля. Для решения этой задачи используется континуальная математическая модель, основанная на уравнениях классической механики Ньютона, выбран численный метод, использующий в основе схему Верле, предложен параллельный алгоритм и осуществлена его реализация в рамках MPIи OpenMP. С помощью разработанной параллельной программы было проведено исследование термодинамического равновесия атомов никеля при условии нагрева образца до желаемой температуры. В численных экспериментах определены оптимальные параметры методики расчета и физические параметры исследуемого процесса. Полученные численные результаты хорошо согласуются с известными теоретическими и экспериментальными данными.

Ключевые слова: молекулярно-динамическое моделирование, никель, ППА, температура, термостат, уравнения Ньютона, параллельные алгоритмы и программы, MPI, OpenMP

This work was supported by Russian Fund for Basic Researches (projects No. 13-01-12073-ofi_m, 14-01-00663-a).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 573–579 (Russian).

1 Introduction

Modern computer equipment makes it possible to simulate very complex systems and processes. Currently, the need for complex systems simulation arises in many branches of knowledge, including in the framework of the nanotechnology implementation in industry. The study of micro- and nano-processes often leads to problems of molecular dynamics of large-scale systems with a large set of uncertain parameters and a variety of conditions that simulate the physical experiment. In this regard, in the last decades new approaches to the modeling of large systems at the molecular level actively develop.

One of the most effective approaches having important fundamental and applied relevance is the method of molecular dynamics (MD) [Kaplan, 1982; Hockney, Eastwood, 1989; Sutmann, 2002; Allen, 2004; Haile, 1992]. The MD method based on a model representation of a polyatomic molecular system in which all the atoms are material points and move according to the laws of classical mechanics. Important areas for the application of molecular dynamics simulation are the design, manufacture and maintenance of the various technical systems, which consist of the metal elements.

In the present work the practical task of computing bases development and carrying out preliminary molecular dynamics simulation of the thermal effects on the sample composed of nickel atoms was set. The novelty of the problem stems from the fact that for nickel new interaction atoms potentials were designed that reproduces more realistically its equilibrium and non-equilibrium states. It gives the chance of carrying out calculations of many technological processes in the microsystems containing nickel coverings at qualitatively new level. In our case, with the help of the research described below preparation for the study of dynamic processes in the micro-nozzles and micro-channels with internal nickel-based alloys will be executed.

To solve the problem of nickel thermodynamic equilibrium calculation the molecular dynamics model was chosen, which is based on the algorithms described in work [Podryga, 2011], the necessary methods of numerical analysis were studied, the estimated parallel program based on specially selected algorithms was created.

2 Problem formulation

In classical molecular dynamics the researched system is represented set of interacting particles. Movements and interactions of the particles are described by Newton's equations. If the particle system is closed, the forces acting on the particles are determined only by the interactions of the particles with each other. These forces are expressed in terms of the gradient of the potential energy with the opposite sign. In case of the external impact on the system presence the forces are the sum of internal and external influences.

In the context of the chosen task it is necessary to research heating of system. For this purpose heat is supplied externally to the system. Thus, the system is not closed. External influence allows to change the system’s temperature. The potential energy of system is the sum of the partial interaction energies of particles pairs. Calculation of the pair forces is based on the formulas of the selected interaction potential.

At the initial time the positions and velocities of all particles are set. After that motion equations are solved. For this purpose, on each step forces, new coordinates and velocities of the particles are calculated taking into account boundary conditions and external effects on the system. Solution of the equations system is carried out using finite difference scheme Verlet [Verlet, 1967].

3 Numerical technique

Initial conditions include the distribution of the particle coordinates which has an accurate structure for the solid materials and the distribution of velocities given by the selected starting temperature

of the sample. Nickel has a face-centered cubic lattice (fcc) with the parameter (the edge of the unit cell) which needs to be determined on the basis of the problem conditions (temperature, pressure).

The considered nickel system represents a parallelepiped with the sides Lx, Ly, Lz. $\mathbf{L}(Lx, Ly, Lz) = \mathbf{k} \cdot rcr$ — sizes of the considered area on axes x, y, z respectively, where $\mathbf{k}(k_x, k_y, k_z)$ — quantity of unit cells. Thus, we have a parallelepiped from $k_x \cdot k_y \cdot k_z$ nickel crystals.

The purpose of computer simulation is to obtain estimates of molecular systems consisting of a large number of particles behavior. In case of model calculations are usually restricted to reviewing of the given volume area characteristics at the given density for what enter a molecular and dynamic cell and set periodic conditions on its boundaries. In this work the dynamics of the system examines, in which the axis z of the model is finite, on the other two axes periodicity conditions with periods Lx, Ly of axes x, y respectively are superimposed. In terms of frequency axis z direction is not.

Values of the initial velocity vectors are generated from a Maxwell distribution corresponding to the desired temperature value, for the problem of determining the system relaxation state for the initial it is possible to take value close to the desired value.

The potential energy is represented as a function depending on the coordinates of the particles and describing the interaction between the particles in the system. Selection of a specific type of interaction potential is based on a comparing of the mechanical properties of a computer model of potential and real material. As part of the problem is selected interaction model using the embedded atom model [Daw, Baskes, 1984]. As a function of the pair interaction and density functions were chosen form proposed in [Zhou, Johnson, Wadley, 2004].

Special algorithms — thermostats — are used to maintain the temperature of the system near the set point. Also thermostats are used to achieve the desired temperature and for the relaxation of the system to ensure continuity of the MD calculations at the initial stage. With the relaxation of the system in the case of steady thermodynamic equilibrium temperature of the thermostat and the average temperature of the molecular system must match.

In this paper, Berendsen thermostat and Langevin thermostat were considered to achieve the desired temperature of the system. Berendsen thermostat [Berendsen et al., 1984] is based on the introduction to calculation of sign-variable friction. In modeling the interaction with the heat reservoir is not considered explicitly, but it is defined by the force of friction. Change of kinetic energy is modeled by rescaling the velocities of the system atoms at each step.

Langevin thermostat [Kheerman, 1990] is based on the motion equations of Brownian dynamics. Interaction of system with the heat reservoir is carried out through the introduction of two additional force components comprising a random effect causing heating of the particles and friction forces to compensate for the temperature and prevent overheating of the particles.

An important parameter of the simulation by using the heating thermostat is the interaction time with the heat bath. As a result, consideration of different values of this parameter is determined by the optimal time interaction with the reservoir in the conditions of the problem, due to which the system will reach the equilibrium state in a shorter period of time compared with other values of this parameter.

4 Parallel realization

To implement the developed numerical approach the concept of parallel programming, developed in [Polyakov et al., 2012] was used. It is based on the principles of geometric and functional parallelism. In our case the estimated area breaks into local domains of identical power. Power of the domain is measured in number of elementary boxes, in each of which molecules surely interact with each other. Partition on domains is made within topology a three-dimensional grid. Each estimated domain gets on the calculator as whom the node of a cluster or a supercomputer is used. Distribution of domains on calculators is realized by means of MPI library.

Within a node (calculated domain) is a certain amount of elementary boxes grouped into three-dimensional sublattice. Such a structure is used for further calculations on the distribution of the trade central processing units (CPUs).

The main algorithm of calculation looks as follows.

Step 1 — Read the original data MPI-application and initialize data structures for each calculator.

2nd stage — basic calculations in the cycle time.

3rd stage — the implementation of the resulting calculations and deinitialization.

In the first phase the determination of the molecules number in computational domain, the generation of their location (within the face-centered lattice of nickel), as well as the generation of uniform at the angles and the Maxwell modulus distribution of pulses, the calculation starting forces are carried out.

As part of the main loop the following steps are executed on time. First, the new values of the coordinates are calculated. Next their correction by means of periodic boundary conditions is made. After this particles exchange between the concrete boxes is made, which is carried out within the calculated domain and between domains calculated using the functions MPI.

Further the calculation of the forces on the basis of which the correction is made of particle velocities and calculated all the necessary integral characteristics are calculated. When you reach the control time points the necessary data are stored in files.

Details of the computer implementation are explained more detail in [Podryga, Polyakov, 2014].

Basic test calculations were performed on the Keldysh Institute of Applied Mathematics RAS budget cluster with processors Intel Xeon X5650 @ 2.67GHz and network speed to 2 Gbit/s. The results of acceleration and parallelization efficiency calculations are presented in Table 1. In the calculation of these characteristics the total number involved in the calculation of parallel threads (NT) was used, although the parallelization was performed using hybrid technology that combines MPI and OpenMP. As might be expected, streaming parallelism was not so effective than parallelization across nodes. However, collectively it was succeeded to reduce the time of calculations dozens of times in a relatively small maximum configuration.

Table 1. Data on efficiency of calculations when using technology MPI+OpenMP

Size crystal	24x24x24 (55296 particles)		48x48x48 (442368 particles)		96x96x96 (3538944 particles)	
NT	ACC	EFF	ACC	EFF	ACC	EFF
Budget Cluster						
1	1.000	100.000	1.000	100.000	1.000	100.000
2	1.811	90.529	1.806	90.300	1.844	92.207
4	2.406	60.149	2.496	62.394	2.440	61.009
8	3.074	38.427	2.611	32.637	2.660	33.247
12	**2.943**	**24.528**	**2.917**	**24.306**	**2.967**	**24.727**
16	3.383	21.144	3.141	19.632	3.098	19.361
24	3.495	14.564	3.436	14.315	3.709	15.455
48	7.278	15.162	7.253	15.111	6.936	14.451
96	**13.679**	**14.249**	**12.864**	**13.400**	**13.572**	**14.137**
192	**24.220**	**12.615**	**25.391**	**13.225**	**27.116**	**14.123**
384	**28.812**	**7.503**	**48.385**	**12.600**	**50.572**	**13.170**
Supercomputer K100						
1	1.000	100.000	1.000	100.000	1.000	100.000
12	**5.003**	**41.691**	**5.056**	**42.135**	**5.066**	**42.220**
96	**33.726**	**35.131**	**35.933**	**37.431**	**34.733**	**36.180**
192	**55.386**	**28.847**	**63.010**	**32.818**	**62.627**	**32.618**
384	**77.276**	**20.124**	**97.550**	**25.404**	**107.265**	**27.933**

5 Simulation results

In this section we present some results of the calculations. Determination of equilibrium macroparameters of nickel system (average temperature, pressure and pulse) and also determination of optimum parameters of numerical algorithm and thermostat were the purpose of calculations. Most of this research has been performed in the framework of [Podryga, Polyakov, 2014]. As a result in numerical experiments optimum parameters of a calculation procedure (an integration step, time of interaction of system with the thermal tank) and also physical parameters of the modeled process were determined (in particular, internal pressure of a sample). The obtained numerical results were compared with known theoretical and experimental data and confirmed a good agreement with the last.

In this paper along with the efficiency of the parallel implementation we were interested in the distribution of pressure and temperature on the sample height depending on the thickness of the controlled temperature layer. For this purpose we selected the sample which is in vacuum with sizes 24x24x24n, where n — number of vertical layers. For example, we took $n = 5$ and considered the situations when thermostat wasn't use and also when the thermostat (in this case Berendsen thermostat was selected) located in the center of a sample and occupied 1, 3 и 5 layers. Length of a crystal edge was equal $a = 0.35311$ nm. The initial temperature of a sample and temperature of the thermostat matched and were equal 273 K. To obtain a stable crystal it is necessary before simulation to work on the determination of the edge length of the unit cell corresponding to the researched temperature and the used interaction potential model. As a result of such work the value of an edge $a = 0.35311$ nm was obtained.

Data of calculations are shown in Fig. 1, 2. Digits 1–4 denote the curves corresponding to the absence of a thermostat (curve 1), and the cases when number of thermostated layers is equal 1, 3 and 5 (curves 2, 3, 4). The figures show that the pressure in the sample is close to zero (i.e., the crystal is stable) irrespective of the thickness of thermostated zone of sample. The temperature in the sample has a parabolic profile corresponding to the thickness of the controlled temperature zone. These data are in agreement with the theoretical concepts of thermal distribution in a bulk sample. In case of simulation of near-surface interactions of a metal sample with an external environment temperature control should be applied in its inside layers not to distort an interaction pattern.

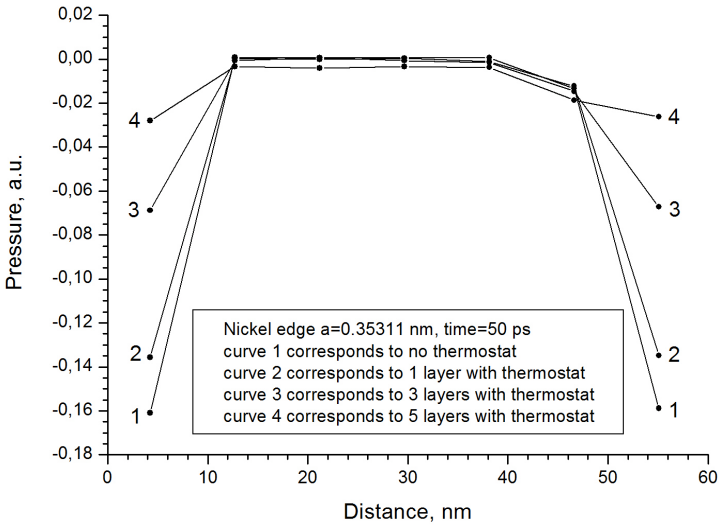


Fig. 1. Pressure distribution over the height of the sample

6 Conclusion

To solve the problem for the establishment of thermodynamic equilibrium in the nickel plate molecular dynamics model was selected, the necessary methods of its numerical analysis were studied,

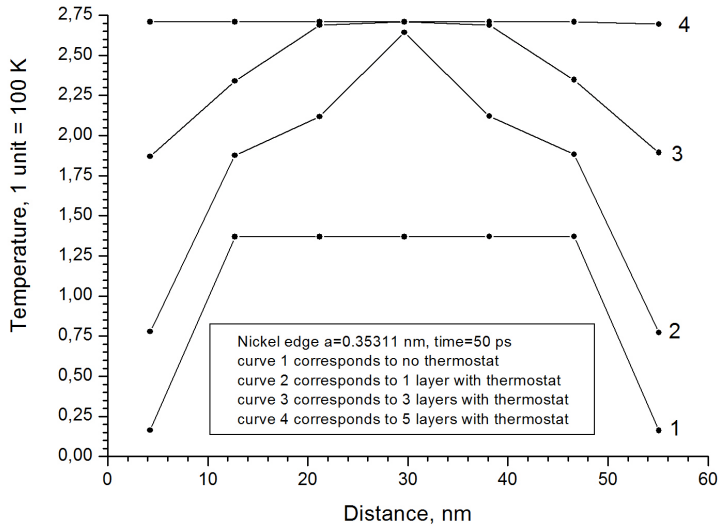


Fig. 2. Temperature distribution over the height of the sample

a parallel program was created. The developed program was carried out full-scale computational experiment whose purpose was to research the process of atoms thermodynamic equilibrium establishment in the sample heated to a predetermined temperature. Obtained in the calculation results demonstrated the adequacy of the proposed numerical approach to modern theoretical concepts of the simulated physical process. Comparing of the received results with the experiment was conducted on the basis of the known tabular data about the properties of nickel under normal conditions.

References

Allen M. P. Introduction to Molecular Dynamics Simulation // Computational Soft Matter: From Synthetic Polymers to Proteins. Julich: John von Neumann Institute for Computing, NIC Series. — 2004. — Vol. 23. — P. 1–28.

Berendsen H. J. C., Postma J. P. M., van Gunsteren W. F. et al. Molecular dinamics with coupling to an external bath // J. Chem. Phys. — 1984. — Vol. 81. — P. 3684–3690.

Daw M. S., Baskes M. I. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals // Physical Review B. — 1984. — Vol. 29, No. 12. — P. 6443–6453.

Haile J. M. Molecular Dynamics Simulation. Elementary Methods. New-York: John Wiley & Sons, Inc., 1992. — 490 p.

Hockney R. W., Eastwood J. W. Computer simulation using particles. Adam Hilger, IOP Publishing Ltd, 1989. — 523 p.

Kaplan I. G. Vvedenie v teoriyu mezhmolekulayarnykh vzaimodeystviy. — M.: Nauka, 1982. — 312 p. (in Russian).

Kheerman D. V. Metody kompyuternogo eksperimenta v kompyuternoy fizike. — M.: Nauka, 1990. — 176 p. (in Russian).

Podryga V. O. Modelirovanie processa ustanovleniya termodinamicheskogo ravnovesiya nagretogo metalla // Matematicheskoe modelirovanie. — 2011. — Vol. 23, No. 9. — P. 3–17 (in Russian).

Podryga V. O., Polyakov S. V. Molekulyarno-dinamicheskoye modelirovanie processa ustanovleniya termodinamicheskogo ravnovesiya nagretogo nikelya // Preprints of Keldysh Institute of Applied Mathematics. — 2014. — No. 41. — 20 p. (in Russian). URL: http://library.keldysh.ru/preprint.asp?id=2014-41

Polyakov S. V., Karamzin Yu. N., Kosolapov O. A., Kudryashova T. A., Sukov S. A. Gibridnaya superkompyuternaya platforma i razrabotka prilozheniy dlya resheniya zadach mekhaniki

sploshnoy sredy setochnymi metodami // Izvestiya YUFU. Tekhnicheskie nauki. — 2012. — No. 6(131). — P. 105–115 (in Russian).

Sutmann G. Classical molecular dynamics // Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms / Eds. Grotendorst J. et al. Julich: NIC. — 2002. — Vol. 10. — P. 211–254.

Verlet L. Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules // Physical Review. — 1967. — Vol. 159. — P. 98–103.

Zhou X. W., Johnson R. A., Wadley H. N. G. Misfit-energy-increasing dislocations in vapor-deposited CoFe/NiFe multilayers // Physical Review B. — 2004. — Vol. 69. — P. 144113.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 581−586

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.27

A CPU benchmarking characterization of ARM based processors

R. G. Reed[a], M. A. Cox, T. Wrigley, B. Mellado

School of Physics, University of the Witwatersrand. 1 Jan Smuts Avenue, Braamfontein, Johannesburg, 2000, South Africa

E-mail: [a]robert.reed@cern.ch

Получено 30 сентября 2014 г.

Big science projects are producing data at ever increases rates. Typical techniques involve storing the data to disk, after minor filtering, and then processing it in large computer farms. Data production has reached a point where on-line processing is required in order to filter the data down to manageable sizes. A potential solution involves using low-cost, low-power ARM processors in large arrays to provide massive parallelisation for data stream computing (DSC). The main advantage in using System on Chips (SoCs) is inherent in its design philosophy. SoCs are primarily used in mobile devices and hence consume less power while maintaining relatively good performance. A benchmarking characterisation of three different models of ARM processors will be presented.

Keywords: High data throughput, Computing, Big Data, ARM System on Chips, Benchmarking

Характеристика тестирования центрального процессора на базе процессоров ARM

Р. Г. Рид, М. Кокс, Т. Ригли, Б. Мелладо

[1] *Отделение Физики, Университет Витватерсранда, Южная Африка, 2000, Йоханнесбург, 1 Ян Смут Авеню*

Большие научные проекты генерируют данные на всё более возрастающих скоростях. Типичные методы включают в себя хранение данных на диске, после незначительного фильтрования, а затем их обработку на больших компьютерных фермах. Производство данных достигло той точки, когда требуется обработка в режиме on-line, чтобы отфильтровать данные до управляемых размеров. Потенциальное решение включает в себя использование низко затратных процессоров ARM с маленькой мощностью в больших массивах для обеспечения массивного распараллеливания для вычислений потока данных (DSC). Главное преимущество в использовании систем на одном кристалле (SoCs) присуще самой философии этой разработки. Системы на микросхеме, прежде всего, используются в мобильных устройствах и, следовательно, потребляют меньше энергии при своей относительно хорошей производительности. Дано описание тестирования трех различных моделей процессоров ARM.

Ключевые слова: высокая вычислительная пропускная способность, большие данные, система на ARM чипе, эталонные тесты

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 581–586 (Russian).

1. Introduction

The term ”Big Data” has caught on in the mainstream media and science worlds. While this word in now ubiquitous and almost exhausted in it’s use it still identifies an important issue in the science community. Processing data is getting more difficult due to the shear amount being produced. In the year 2022 the ATLAS detector will be upgraded and in doing so will produce in the order of Petabytes per second of raw data [ATLAS C 2012 Letter of Intent..., 2012]. There is no feasible way to process this much data in a reasonable amount of time. This is largely due to external Input/Output (I/O) bottlenecks present in current super computing systems. A team at the University of the Witwatersrand, Johannesburg is actively involved in the development of a computing system which is both cost-effective and able to provide high data throughputs in the order of Gigabits per second. There are four widely accepted computing paradigms. The first, and most commonly known, is the High Performance Computing paradigm (HPC) which is focused on the raw number of calculations performed per second. The second is the Many Task Computing (MTC) which focusses on the number of jobs that can be completed in a given amount of time. Real Time Computing (RTC) involves very strict restrictions on execution times (such as air-bag sensors or process controls). Finally, a fourth paradigm called Data Stream Computing (DSC) involves the processing of large amounts of data with no off-line storage. The processing unit that the team at the University of the Witwatersrand is designing falls under this DSC paradigm and provides the motivation for this paper. In order to design a processing unit that is capable of handling high throughputs the system must be very well balanced. A better understanding of the SoCs is needed in order to achieve this. Presented below is a CPU benchmarking characterisation of three ARM based SoCs.

2. Hardware

Three ARM system on chips will be characterised. The Cortex-A7, Cortex-A9 and Cortex-A15 are available on the Cubieboard2, Wandboard and Odroid-XU+E platforms [Cubieboard 2013 Fedora 19..., 2013; Freescale..., 2009; Hardkernel,...2013]. The specifications of each board can be found in Tab. 1.

Table 1: Specifications of the ARM platforms.

	Cortex-A7	**Cortex-A9**	**Cortex-A15**
Platform	Cubieboard A20	Wandboard Quad	ODROID-XU+E
SoC	Allwinner A20	Freescale i.MX6Q	Samsung 5410
Cores	2	4	4 (+ 4 Cortex-A7)
Max. CPU Clock (MHz)	1008	996	1600
L2 Cache (kB)	256	1024	2048
Floating Point Unit	VFPv4 + NEONv2	VFPv3 + NEON	VFPv4 + NEONv2
RAM (MB)	1024	2048	2048
RAM Type	432 MHz 32 bit DDR3	528 MHz 64 bit DDR3	800 MHz 64 bit DDR3
Ethernet (Mb/s)	100	400	100
PCI-Express (Gb/s)	-	5	-
2014 Retail (USD)	65	129	169

3. CoreMark

CoreMark was developed by The Embedded Microprocessor Benchmark Consortium (EEMBC) and has been proposed as the replacement for Drystone by ARM Holdings [Dunn and Marini, 2009]. The benchmark is specifically designed for Embedded Microprocessors which makes it an ideal benchmark to use. There are numerous pros to using CoreMark and they are summarised by Eric Schorn, VP marketing, Processor Division, ARM ”We believe that CoreMark represents a significant

improvement on the current Dhrystone benchmarks by measuring processor behaviour that could more realistically be expected in a real application. Combined with greater access to the results, this new benchmark should enable developers to obtain an unambiguous representation of processor performance enabling comparisons between competing processors to be made." [Dunn and Marini, 2009].

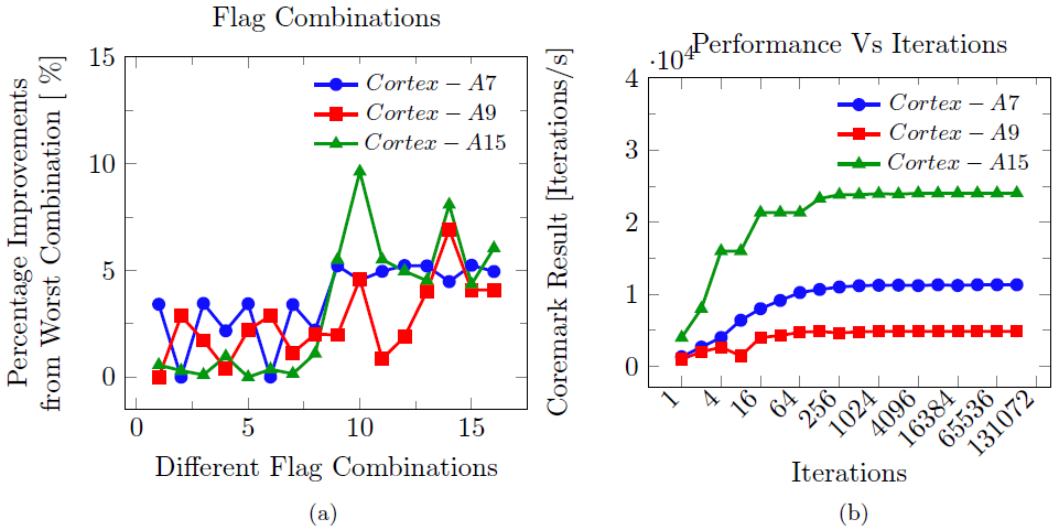


Fig. 1: a) Different flag combinations for compiling and b) Coremarks as a function of Iteration counts

CoreMark uses four common algorithms found in realistic applications such as matrix manipulation, linked list manipulation, state machine operations and cyclic redundancy checks. This provides an overall ''realistic" performance of the chips. Additionally, Coremark has strict online result submission guidlines. This provides a trustworthy and strong database of results with which to compare your own chips. The result is reported as the number of iterations of these four common algorithms per second. Figure 1a) shows the performance of different combinations of compiler flags. The best performing flag combinations can be seen in Tab. 2

Table 2: Best Performing Flag Combinations

Architecture	Flag Combination
Cortex-A7	-mfloat-abi=hard -ffastmath -O3 -mfpu=neon-vfpv4 -march=armv7-a -mtune=cortex-a7
Cortex-A9	-mfloat-abi=hard -ffastmath -O3 -mfpu=neon -march=armv7-a
Cortex-A15	-mfloat-abi=hard -ffastmath -O3 -mfpu=neon-vfpv4 -march=armv7-a

Figure 1b) shows the performance rise to a plateau for increasing iteration count. A particular criteria for result submission is to run the test for at least 10 seconds. This will be at approximately 2048 iterations and is well onto the plateau which illustrates why the database results are a fair comparison.

Figure 2 shows CoreMark results for various different systems. The first three bars represent the Cortex-A7, A9, and A15 results that we measured ourselves and the last four are from the CoreMark online database [EEMBC OnlineDatabase...]. The chosen systems are: a low powered Intel Atom 330, Intel Atom N2800, mid range Intel i7 2600 and a high end Intel i7 3930k. From Figure 2a) it can been seen that the high end Intel i7's are much more powerful per core. The Cortex-A9 is similar to the Atom 330 and like wise the Cortex-A15 is similar to the Atom N2800. Both sets of chips were manufactured around the same time. It must be noted that the Atom N2800 only has two cores while the Cortex-Al5 has four, which means that the overall performance of the Cortex-Al5 will still be better. Looking at the performance per watt in Fig 2b) the complete opposite is observed with the Cortex-A15 being over 3 times more efficient than the Intel i7 3930k. The power consumption measured for the Cortex chips included all peripherals while the Thermal Design Power is quoted for the Intel chips. This means the power consumption of the Intel chips would most likely increase when taking into account all peripherals.

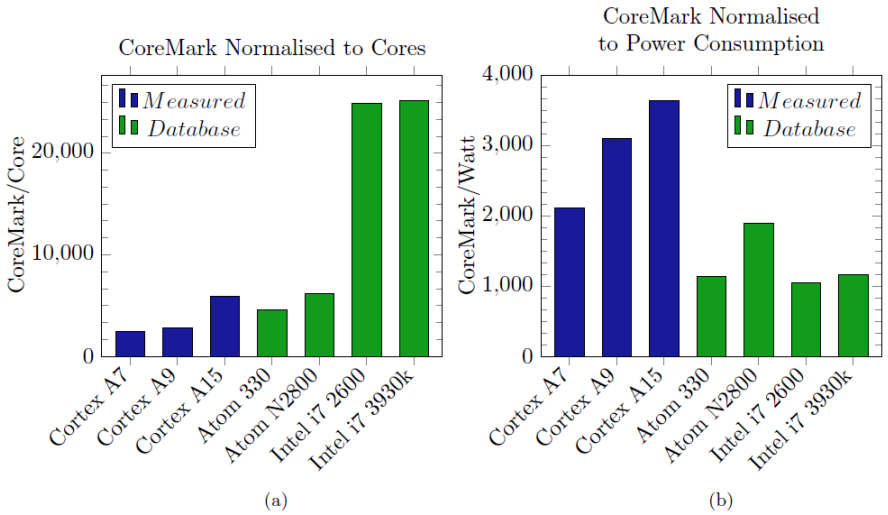


Fig. 2: a) CoreMark results for various systems normalised to the number of cores and b) CoreMark per Watt for various systems

4. High Performance Linpack

Coremark gives an overall performance of a system but to understand the computing capability one must use a benchmark like High Performance Linpack. This benchmark uses matrix manipulations to give the number of Floating Point Operations Per Second (FLOPS) that a system can achieve in double precision [Dongarra et all, 2003]. It was introduced by Jack Dongarra in l979 and first reported in the LINPACK Users Guide [Dongarra et all, 1979]. HPL is currently being used on the TOP500 Super-Computing List [TOP500, 2013] which makes the benchmark a necessity when characterising the ARM CPUs since its largely accepted and understood. HPL is scalable and specifically targeted at distributed memory clusters. An important measurement is the number of FLOPS measured per Watt of the system. This can be used to compare to the GREEN500 [GREEN500, 2013] which is a super-computing list based on efficiency rather than pure performance with results measured in GFLOPS/Watt.

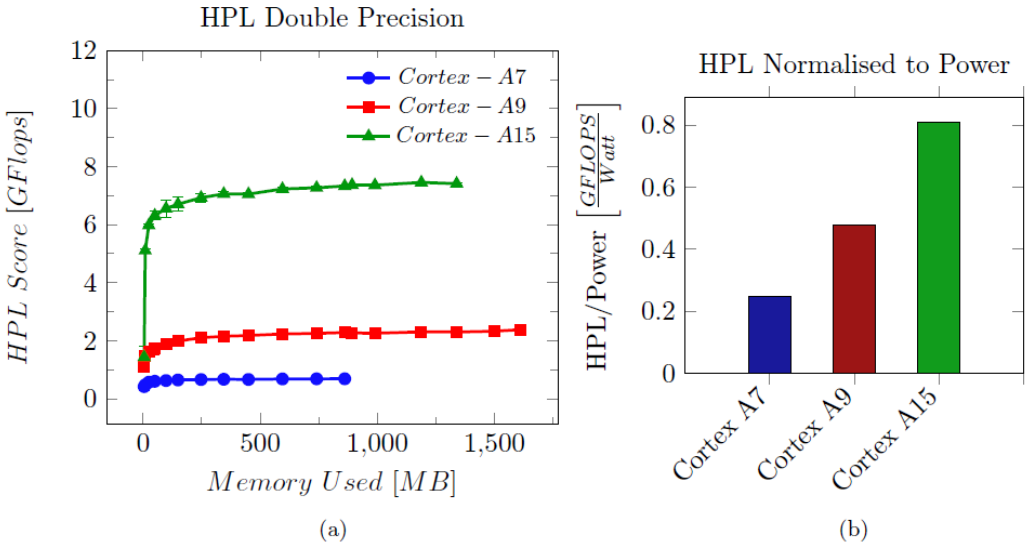


Fig. 3: Double precision High Performance Linpack results for Cortex-A7, Cortex-A9 and Cortex-A15

Figure 3a) shows the typical results reported by HPL for double precision floating point numbers. For clarity the array size (x-axis) is shown as the amount of memory used rather than the actual NxN matrix size. The Cortex-A15 performs the best with a GFLOP score over 3 times higher than the Cortex-A9. The Cortex-A7 is designed to be low performance with low power consumption so it's not surprising that it achieves approximately 0.8 GFLOPS. For very small array sizes (small sizes in RAM) there are large overheads when HPL is running. This is expected to be due to the calculation on given small arrays being shorter than the time taken to populate and fetch data from RAM.

Figure 3b) shows the double precision HPL Efficiency expressed as GFLOPS/Watt. The best performing result is taken. The Cortex-A15 is not 3 times the efficiency of the Cortex-A9. It is only a factor of 1.8. The Cortex-A15 attains a peak of 0.87 GFLOPS/Watt. This would be placed at 110th spot in the Green500 list [Green500-Top200, 2013]. Its interesting to see that the performance efficiency doubles from one architecture to the next. It is surprising to see that the Cortex-A7 has such low efficiency but this is attributed to all the external peripherals that increase the power consumption.

5. FFTW

The Fastest Fourier Transform in the West (FFTW) is a benchmark based on the discrete Fourier transform [Rajovic et all, 2013]. This type of transform is unique in that it has a finite number of elements and thus can be solved computationally. The transform allows a change of domain for a given set of data. The FFTW reports data in MFLOPS = (5Nlog10N)/t, where N is the size of the FFT and t is the time taken to compute the FFT. This is a theoretical performance but found to be quite accurate as it can be compared to results obtained for HPL. A second result reported is the throughput. Using the size of the FFT and the time taken to compute the MB/s throughput can be calculated. This is useful in the context of Data Stream Computing.

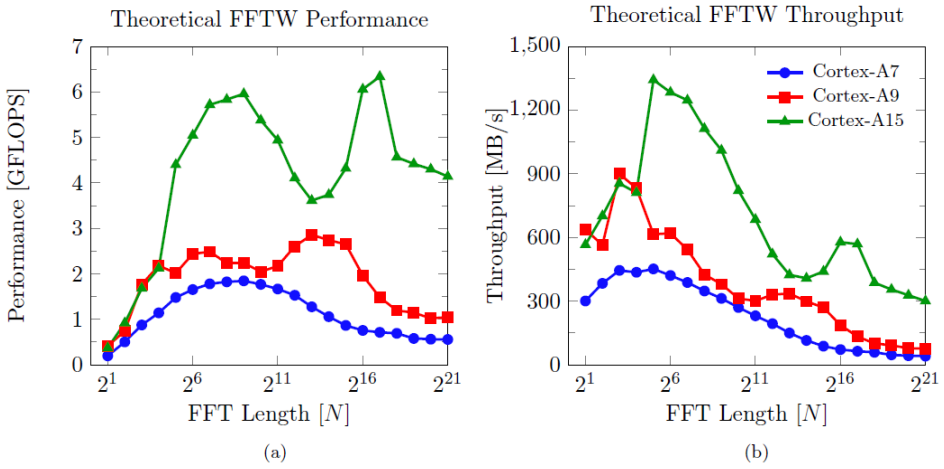


Fig. 4: FFTW benchmarks for the ARM Cortex-A7, A9 and A15 showing best-case multi-core and multi-process performance (a) and theoretical FFT throughput (b)

Figure 4a) shows the performance for various FFT sizes, N. It can be seen that the peak performances are similar to that of HPL in Fig. 3a). The shape is not smooth and this is due to the CPU specific components such as cache sizes, memory controllers and architectures. Figure 4b) shows the theoretical throughput that could be achieved if there were no bottle necks. Unfortunately the I/O performance is drastically limiting. Table 1 shows the connectivity of each SoC. The Cortex-A15 and Cortex-A7 both are limited to 1 Gb Ethernet while the Cortex- A9 has a PCIe lane at 5 Gb. For larger N values around 2^{16} we see the SoCs reaching the 1 Gb throughput range. The Cortex-A15 has a secondary peak at this point in the order of 5 Gb. This indicates that a Cortex-A15 could saturate a single PCIe lane for complex FFT problems.

6. Conclusion

The performance of the Atom is comparable to the Cortex A9 but looking at the power consumption it can be seen that the Atoms are less power efficient. The power consumption of the Atoms is based on their chip specifications alone so taking into account the need for a motherboard and all peripherals this is expected to increase. The Cortex-A7 does not have sufficient performance as seen in the HPL results in Fig 3a. There are newer ARM cores available such as the Cortex-A50 series [Holdings A Cortex-A50...] and new Atoms such as the Z34XX Series [Intel..., 2014] which will usher in the 64 bit architectures. The new Atoms are SoCs and the power consumption will be better than the Atoms shown in this paper. These new ARM architectures will have double precision NEON extensions which will drastically increase their performance as current NEON extensions are only single precision. Intel have opened their fabrication plants to the new Cortex-A53 on their 12 nm trigate technology [Martenson, Altera, 2013] so the expected performance of these ARM CPUs and possibly the newer Atoms will increase significantly and could offer a cheaper alternative for parallel computing. The same can be said for the new generation of Atoms.

References

ATLAS C 2012 Letter of Intent for the Phase-II Upgrade of the ATLAS Experiment URL: http://cds.cern.ch/record/1502664?ln=en

Cubieboard 2013 Fedora 19 For Cubieboard(A20) is Available URL: http://cubieboard.org/2013/07/19/fedora-19-for-cubieboarda20-is-available/

Dongarra J. J., Luszczek P. and Petitet A. Concurrency and Computation: Practice and Experience 15. 2003. 803-820 ISSN 1532-0626 URL: http://doi.wiley.com/10.1002/cpe.728

Dongarra J., Bunch J., Moler C. and Stewart G. LINPACK: users' guide (SIAM). 1979. URL: http://books.google.ch/books?id=AmSm1n3Vw0cC\&lpg=PR5\&ots=EDFdqJhr8x\&dq=infoV/o3AhttpV/o3AV/02F\°/02Fs3da3171290b34600.scholar.google.comV/o2F0\&lr\&pg=SL2-PA1\#v=onepage\&q\&f=false

Dunn L. and Marini C. ARM Announces Support For EEMBC CoreMark Benchmark — ARM. 2009 . URL: http://www.arm.com/about/newsroom/25152.php

EEMBC OnlineDatabase — The Embedded Microprocessor Benchmark Consortium URL: http://www.eembc.org/coremark/

Freescale 2009 i. MX 6 Series of Applications Processors Tech. rep. URL: http://www.freescale.com/webapp/sps/site/taxonomy.jsp?code=IMX6X_SERIES

GREEN500. The Green500 List. 2013. URL: http://www.green500.org/

Green500-Top200 The Green500 List — November 2013 — The Green500 URL http://www.green500.org/lists/green201311\&green500from=101\&green500to=200

Hardkernel 2013 ODROID XU+E URL: http://hardkernel.com/main/products/prdt_info.php?g_code=G137463363079\&tab_idx=2

Holdings A Cortex-A50 Series - ARM URL: http://www.arm.com/products/processors/cortex-a50/

Intel 2014 Intel Atom Processor Z34XX Series for Smartphones and Tablets URL: http://www.intel.com/content/www/us/en/processors/atom/atom-z34xx-smartphones-tablets-brief.html

Martenson Sue, Altera A. Newsroom — Altera Announces Quad-Core 64-bit ARM Cortex-A53 for Stratix 10 SoCs. 2013. URL: http://newsroom.altera.com/press-releases/nr-altera-arm-a53.htm

Rajovic N., Carpenter P. M., Gelado I., Puzovic N., Ramirez A. and Valero M. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis on — SC '13. New York, USA. 2013. P. 1–12. ISBN 9781450323789

TOP 500. November 2013 — TOP500 Supercomputer Sites. 2013. URL: http://www.top500.org

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 587–592

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.4

An automated system for program parameters fine tuning in the cloud

S. A. Smirnov[a], A. S. Tarasov

Institute for Information Transmission Problems of the Russian Academy of Science, Kharkevich Institute, Bolshoy Karetny per. 19, build.1, Moscow 127051 Russia

E-mail: [a] sasmir@gmail.com

Получено 25 декабря 2014 г.

The paper presents a software system aimed at finding best (in some sense) parameters of an algorithm. The system handles both discrete and continuous parameters and employs massive parallelism offered by public clouds. The paper presents an overview of the system, a method to measure algorithm's performance in the cloud and numerical results of system's use on several problem sets.

Keywords: algorithmic parameter optimization, parameter tuning, cloud computing

Автоматическая облачная система подстройки параметров алгоритмов

С.А. Смирнов, А.С. Тарасов

Институт проблем передачи информации им. А. А. Харкевича Российской академии наук, Россия, 127051, г. Москва, Большой Каретный переулок, д. 19, стр. 1

В работе представлена система, обеспечивающая подбор наилучших в смысле времени выполнения настроек алгоритма. В качестве алгоритма был взят пакет решения задач частично-целочисленного линейного и нелинейного программирования SCIP. Возможность параллельного перебора множества вариантов настроек обеспечивается кластером из виртуальных машин, автоматически создаваемых в облаке. Представлены результаты работы системы на нескольких наборах задач.

Ключевые слова: оптимизация параметров алгоритмов, облачные вычисления

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 587–592 (Russian).

Introduction

Growing number of Infrastructure as a service (IaaS) providers we observe today is a direct effect of computation costs getting cheaper and of infrastructure automatization levels getting higher. Cloud services make it possible to automate more programmer's work making him more productive. It may be considered as another step in continuous process of adding more abstraction levels to a computer system: high-level programming languages, interactive debugging, automatic build systems, etc. For developers it allows for rapidly creating development and test sandboxes, quickly provisioning virtual machines with needed software, testing load and scalability.

There are lots of problems that can be automated by the use of clouds. One of such problems is fine tuning an algorithm to make it work better in some sense, for example, faster. It can be done in many ways: modifying hard coded parameters inside a program, smart analysis of the program's source code, adjusting parameters inside configuration files of program's modules. In our study we have chosen the last described way: fine tuning configuration parameters for the SCIP (Solving Constraint Integer Programs) solver. SCIP [Achterberg Tobias, 2009] is currently one of the fastest non-commercial solvers for mixed integer programming (MIP) and mixed integer nonlinear programming (MINLP). It is also a framework for constraint integer programming and branch-cut-and-price. It allows for total control of the solution process and the access of detailed information down to the guts of the solver. Although SCIP is a very fast solver even with default parameters, it should be possible to fine tune the parameters for one's work. It is quite simple if there are a couple of parameters and not many test problems. However SCIP is very configurable having more than a thousand parameters. Apparently having such a large set of configuration parameters makes fine tuning it quite time consuming. That is why we tried to automate this process making a system choosing best configuration setting for a set of problem instances. Due to vast number of SCIP runs needed we had to use a cloud to make the process quick.

In this paper we use the following terminology. A program has configuration *parameters* controlling how the it works. Every parameter has a *value* assigned. A set of parameter values is called the *settings*. When we run a solver it is given settings in form of a configuration file and a problem instance. The goal of our system may be thought as finding the settings yielding the shortest running time on a set of problem instances.

Other works on general optimization of algorithmic parameters include Selection Tool for Optimization Parameters (STOP) [Baz et al., 2007] based on intelligent sampling of settings throughout the space and OPAL framework [Audet, Dang, Orban, 2012] based on mesh adaptive direct search. The former tool works with a small set of parameters having discrete values. Our aproach allows working with large numbers of parameters and thair values.

Implementation

Let us begin with a brief overview of the system. One begins using it by specifying the number of computing hosts in the Vagrant's [Vagrant] configuration file and then starting the system by running init-virtualbox.sh or init-digitalocean.sh script. After a while one has a cluster of a master host and the specified number of slave hosts where Simple Linux Utility for Resource Management (SLURM) [Yoo, Morris, and Mark, 2003] and other essential software are installed and running. Then one connects to the master host by issuing *vagrant ssh* command where one can manage the system with *optctl.py* command.

Currently the settings optimization process consists of three phases: time check, big step and inter step. During the first phase every problem's instance is evaluated once on each computing node with default settings. The main goal of this step is to get an estimate of maximum time allowed for a problem instance to run until it's killed by SLURM. The big step phase is the most computationally intensive one. On this step huge number of settings with only one parameter different from defaults is evaluated. As a result the big step allows us to sort parameter values based on their impact on solving time.

Next, on the inter step phase, four best parameter values from the head of the big step's sorted list are chosen and all their possible combinations are evaluated. After this step we have the best settings in terms of running time. This step is not very time consuming and can be repeated multiple times.

Measuring running time in the cloud

Measuring programs's running time in the cloud reliably is not very simple. Naive approach like wall-clock time or processor cycles are not reliable due to computer resources overcommit by a cloud provider. Depending on the load other virtual machines express on the hypervisor host, program's running time can change dramatically. There is a better approach: one can measure the number of instructions executed by the CPU while running the program. Of course, different instructions may need different numbers of cycles to complete so it may be hard to correlate running time to the number of instructions executed. Instruction count becomes handy when comparing performance the same program expresses with the same input but with different settings.

In x86 CPUs instructions can be counted in hardware by the Performance Monitoring Unit (PMU). One can use PAPI or perf tool to set up and access the hardware counters. Not every hypervisor supports PMU virtualization, e.g. VirtualBox does not. However modern KVM releases has such support.

In our system we used perf tool to measure user space instruction count which gives very stable results independent of the hypervisor host's load.

Here is a sample run of SCIP under perf-stat. Six runs of SCIP were made, average counter values and their standard deviations can be observed:

```
$ perf stat -r 6 -e cpu-clock,task-clock,cycles,instructions,instructions:u,\
instructions:k scipampl TSP_Uniform_50_10.nl
 Performance counter stats for 'scipampl TSP_Uniform_50_10.nl' (6 runs):
      81154.175629 cpu-clock                                       ( +-  2.32% )
      81154.063870 task-clock           # 0.999 CPUs utilized     ( +-  2.32% )
   175,626,392,898 cycles               # 2.164 GHz               ( +-  0.13% )
   267,235,503,611 instructions         # 1.52  insns per cycle   ( +-  0.00% )
   265,101,243,265 instructions:u       # 1.51  insns per cycle   ( +-  0.00% )
     2,134,260,346 instructions:k       # 0.01  insns per cycle   ( +-  0.10% )

      81.224668399 seconds time elapsed                            ( +-  2.32% )
```

Instaructions:u counter gives much more stable results than software counters or cycles counted in hardware.

Same single CPU virtual machine with two SCIP instances running simultaneously:

```
 Performance counter stats for 'scipampl TSP_Uniform_50_10.nl':
      82580.457064 cpu-clock
      82579.334255 task-clock           # 0.493 CPUs utilized
   181,566,274,355 cycles               # 2.199 GHz
   267,300,821,128 instructions         # 1.47  insns per cycle
   265,099,385,783 instructions:u       # 1.46  insns per cycle
     2,201,435,345 instructions:k       # 0.01  insns per cycle

     167.578326122 seconds time elapsed

 Performance counter stats for 'scipampl TSP_Uniform_50_10.nl':
      82581.195083 cpu-clock
      82580.104484 task-clock           # 0.493 CPUs utilized
   181,589,302,031 cycles               # 2.199 GHz
   267,299,923,846 instructions         # 1.47  insns per cycle
   265,099,381,995 instructions:u       # 1.46  insns per cycle
     2,200,541,851 instructions:k       # 0.01  insns per cycle

     167.589704033 seconds time elapsed
```

Again, instructions:u are much more accurate.

As we can see from this examples, instructions:u is the most stable event counter at least with SCIP. It even allows for running multiple solver instances simultaneously with acceptable timing accuracy.

Numerical results

We have performed testing with two different sets of problem instances. One of the sets was tested with two version of SCIP: 3.0.2 and 3.1.0. Throughout the tests, 48 computing nodes with identical virtual machines were used.

First problem set consisted of ten randomly generated traveling salesman problem instances of the same size. SCIP 3.1.0 was used. Big step for this set consisted of 28810 jobs and took six hours and a half to complete while total CPU time consumed was 296 hours, as if 46 machines were used. After one inter step optimal settings were obtained. Second inter step showed no improvement. If we compare the sums of running times for default settings and for optimized ones, we oserve 3x speedup with the latter (see Table 1). Optimized settings consisted of only one parameter value:

```
lp/scaling = FALSE
```

Table 1. Traveling salesman problem, learning data set

Problem instance	Default, sec.	Optimized, sec.
TSP_Uniform_50_1	3,0	2,5
TSP_Uniform_50_2	10,4	6,4
TSP_Uniform_50_3	29,4	16,1
TSP_Uniform_50_4	6,8	8,7
TSP_Uniform_50_5	39,9	36,6
TSP_Uniform_50_6	37,9	7,6
TSP_Uniform_50_7	105,8	31,3
TSP_Uniform_50_8	102,8	13,6
TSP_Uniform_50_9	31,4	10,9
TSP_Uniform_50_10	80,7	13,2

For testing purposes more TSP instances were generated and run with the same optimized parameters (see Table 2), the speedup is just 1,41x here.

Table 2. Traveling salesman problem, control data set

Problem instance	Default, sec.	Optimized, sec.
TSP_Uniform_50_11	52,6	27,3
TSP_Uniform_50_12	11,5	9,2
TSP_Uniform_50_13	1,4	2,2
TSP_Uniform_50_14	7,0	5,2
TSP_Uniform_50_15	270,8	230,6
TSP_Uniform_50_16	64,0	10,5
TSP_Uniform_50_17	4,0	3,7
TSP_Uniform_50_18	27,1	14,4
TSP_Uniform_50_19	5,0	12,0

Second problem set consisted of five instances which solved quickly with SCIP 3.0.2 and very slowly with SCIP 3.1.0.

An attempt was made to find parameters making SCIP 3.1.0 working on the problem as good as 3.0.2. We took all parameters that changed their default values, were renamed or added in 3.1.0, which resulted in 186 parameters (against 1547 total parameters). One of the instances

(w6_t19_test_8) was dropped after the time check phase due to hitting memory limit (512 MB RAM in VM). Big step consisted of 1260 jobs for the first four instances and took five hours and a half to complete. Total CPU time spent in SCIP was 237 hours which equals to 43 hosts working. After one inter step optimized settings were obtained. Second inter step showed no improvement. If we compare the sums of running times for default settings and for optimized ones, we oserve 1,65x speedup when the latter is used. It should be noted that optimized settings also improved time for the problem w6_t19_test_8 that was not involved in the tests due to memory limitation. As we can see, our system was not able to find settings making SCIP 3.1.0 perform as good as SCIP 3.0.2 for this problem, however a noticeble speedup was obtained. Optimized settings:

```
heuristics/rins/minnodes = 25
lp/checkdualfeas = FALSE
lp/disablecutoff = 1
```

Table 3. Load balancing problem

Problem	3.1.0, def., sec.	3.1.0, opt., sec.	3.0.2, def., sec.	3.0.2, opt., sec.
w6_t15_test_4	5,95	3,11	1,97	0,92
w6_t18_test_4	586,18	186,8	70,3	32,5
w6_t19_test_4	1420,4	823,9	223,9	178,3
w6_t19_test_5	941,7	737,6	138,7	119,3
w6_t19_test_8	11596,3	7077,7	382,6	319,7

We also tried optimizing settings for this problem set in SCIP 3.0.2 on all its parameters. During big step 13200 jobs were run in 13 hours and a half, 594 hours were spent in the solver as if 44 hosts were working. After two inter steps optimized settings were obtained, third interstep yielded no improvement. Here 1,26x speedup was obtained. Optimized settings after first inter step:

```
constraints/linear/upgrade/setppc = FALSE
lp/solvefreq = 0
```

After the second inter step:

```
constraints/linear/upgrade/setppc = FALSE
lp/solvefreq = 0
conflict/preferbinary = TRUE
heuristics/fracdiving/freqofs = 1
heuristics/veclendiving/freq = -1
```

Conclusion

As a result of the study the system described was made. It uses Vagrant for virtual machine management, SLURM for batch job processing, Python [Sanner, 1999] for automation, Virtualbox [Oracle VM…] for debugging and DigitalOcean [DigitalOcean…] as a cloud provider. It was tested on a number of problem classes and noticeable speedup was shown.

It is possible to extend the system on other solvers e.g. CBC or Ipopt. Another possible improvement may be made by making the system accessible on the Web. It is also planned to publish the source code on GitHub after some cleanup.

In conclusion we expect that the service may become popular among SCIP users. Another conclusion is that cloud computing is very convenient and cheap nowadays which is definitely a good driver for developing new and nonconventional approaches.

References

Achterberg Tobias SCIP: solving constraint integer programs // Mathematical Programming Computation. — 2009. — Vol. 1, No. 1. — P. 1–41.

Audet C., Dang K. C., & Orban D. Optimization of algorithms with OPAL // Mathematical Programming Computation. — 2012. — P. 1–22.

Baz M., Hunsaker B., Brooks P., & Gosavi, A. Automated tuning of optimization software parameters // University of Pittsburgh Department of Industrial Engineering Technical Report. — 2007. — 7.

DigitalOcean cloud hosting, https://digitalocean.com/

Oracle VM VirtualBox, https://www.virtualbox.org/

Sanner Michel F. Python: a programming language for software integration and development // J. Mol. Graph. Model. — 1999. — 17.1. — P. 57–61.

Vagrant, http://www.vagrantup.com/

Yoo Andy B., Morris A. Jette, and Mark Grondona. SLURM: Simple linux utility for resource management // Job Scheduling Strategies for Parallel Processing. — Springer Berlin Heidelberg, 2003.

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 593–599**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75, 004.45

Development of distributed computing applications and services with Everest cloud platform

O. V. Sukhoroslov[a], A. O. Rubtsov, S. Yu. Volkov

Institute for Information Transmission Problems of the Russian Academy of Science, Kharkevich Institute, Bolshoy Karetny per. 19, build.1, Moscow, 127051, Russia

E-mail: [a] oleg.sukhoroslov@gmail.com

Получено 30 сентября 2014 г.

Everest is a cloud platform for researchers supporting publication, execution and composition of applications running across distributed computing resources. The paper presents current state of the platform, recent developments and remaining challenges.

Keywords: distributed computing, cloud platform, web services, REST, integration of computing resources, application composition

Создание распределенных вычислительных приложений и сервисов на базе облачной платформы Everest

О. В. Сухорослов, А. О. Рубцов, С. Ю. Волков

*Институт проблем передачи информации им. А. А. Харкевича Российской академии наук,
Россия, 127051, г. Москва, Большой Каретный переулок, д. 19, стр. 1*

Everest — облачная платформа для сервис-ориентированных научных сред, реализующая публикацию, выполнение и композицию вычислительных приложений в распределенной среде. В статье рассматриваются текущая реализация платформы, новые разработки и направления дальнейших исследований.

Ключевые слова: распределенные вычисления, облачная платформа, веб-сервисы, REST, интеграция вычислительных ресурсов, композиция приложений

The work is supported by the Russian Foundation for Basic Research (grant No. 14-07-00309 A).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 593–599 (Russian).

Introduction

The ability to effortlessly use and combine existing computational tools and computing resources is an important factor influencing research productivity in many scientific domains. However, scientific software often requires specific expertise in order to install, configure and run it that is beyond the expertise of an ordinary researcher. This also applies to configuration and use of computing resources to run the software. Finally, researchers increasingly need to combine multiple tools in order to solve complex problems, which brings an important issue of application composition.

While existing approaches and technologies provide solutions to some of these problems, they also have several drawbacks. Grid middleware uses service-oriented architecture to implement generic web services to access computing resources, however it is too low-level and hard to use for unskilled researchers. Scientific portals provide remote access to scientific applications and computing resources via convenient web interfaces, however they do not support programmatic access to applications thus limiting opportunities for application reuse and composition. Scientific web service toolkits support programmatic access by exposing applications as web services, however they lack common practices and require an infrastructure for hosting services.

Everest [Sukhoroslov, Afanasiev, 2014; Everest] is a cloud platform that addresses these problems by supporting publication, sharing and reuse of scientific applications across distributed computing resources. The underlying approach is based on combining the strengths of related approaches while eliminating the mentioned drawbacks by using modern web technologies and cloud computing models.

In contrast to traditional software, Everest follows the Platform as a Service (PaaS) cloud delivery model by providing all its functionality via remote web and programming interfaces. A single instance of the platform can be accessed by many users in order to create, run and share applications with each other without the need to install additional software on users' computers. Any application ported to Everest can be accessed via web user interface or unified REST API. The latter enables integration with external systems and composition of applications. Another distinct feature of Everest is the ability to run applications on arbitrary sets of external computing resources.

The paper presents current state of Everest, recent developments and remaining challenges. Section 2 provides an overview of Everest architecture and its main components. Section 3 discusses abstract model and implementation of Everest applications. Section 4 describes integration of external computing resources with Everest and binding of resources to applications. Section 5 presents Python API that enables Everest users to write programs that access applications and combine them in arbitrary workflows. Section 6 concludes and discusses future work.

Everest Architecture

A high-level architecture of Everest is presented in Figure 1. The server-side part of the platform is composed of three main layers: REST API, Applications layer and Compute layer. The client-side part includes web user interface (Web UI) and client libraries.

REST API is the platform's application programming interface implemented as a RESTful web service [Richardson, Ruby, 2007]. It includes operations for accessing and managing applications, jobs, resources and other platform entities. REST API serves as a single entry point for all clients, including Web UI and client libraries.

Applications layer corresponds to a hosting environment for applications created by users. Applications are the core entities in Everest that represent reusable computational units that follow a well-defined model described in the next section. Each application created by user is automatically exposed as a RESTful web service via the platform's REST API. This enables remote access to the application both via Web UI and client libraries. An application owner can manage the list of users that are allowed to run the application.

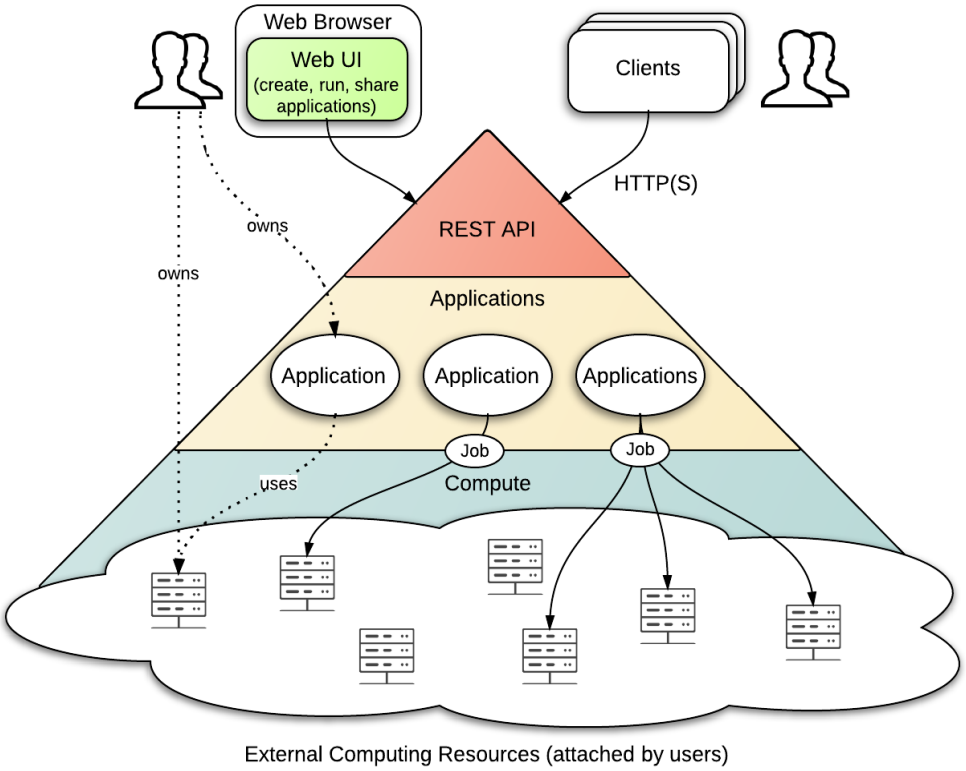


Fig. 1. High-level architecture of Everest

Everest doesn't provide its own computing infrastructure to run applications, nor does it provide access to some fixed external infrastructure like grid. Instead Everest enables users to attach to it any external computing resources and to run applications on arbitrary sets of these resources.

Compute layer manages execution of applications on remote computing resources. When an application is invoked via REST API it generates a job consisting of one or more computational tasks. Compute layer manages execution of these jobs on remote resources and performs all routine actions related to staging of task input files, submitting a task, monitoring a task state and downloading task results. Compute layer also monitors the state of resources attached to the platform and uses this information during job scheduling.

Web UI provides a convenient graphical interface for interaction with the platform. It is implemented as a JavaScript application that can run in any modern web browser. Web UI provides access to all functionality of the platform. It is built directly on top of the REST API, i.e., it uses the same interface as all other platform clients.

Client libraries simplify programmatic access to Everest via REST API and enable users to easily write programs that access applications and combine them in arbitrary *workflows*. At the moment, a client library for Python programming language is implemented.

Applications

Applications are the main entities in Everest - any computation is performed in the context of some application. Clients interact with applications by sending requests and receiving back results.

All Everest applications follow the same abstract model. An application has a number of *inputs* that constitute a valid request and a number of *outputs* that constitute a result of computation corresponding to some request. It is convenient to think of an application as a "black box" with some input and output ports or as a "function" with some arguments and return values. Just like pure functions, applications usually process each request independently from other requests in a stateless fashion.

The described model makes it possible to define a uniform web service interface for accessing applications [Afanasiev, Sukhoroslov, Voloshinov, 2013] which is essential in order to support application composition. This interface is implemented in Everest as a part of REST API.

From the user's viewpoint running an application basically means sending it a request containing input values and waiting for a result containing corresponding output values. For each request Everest performs the following actions:

1. Authenticate and authorize the client.
2. Parse and validate input values.
3. Create a new *job* which can be used to track the status of the request and to collect the result.
4. Translate input values to one or more *tasks* that represent units of computation.
5. Run job tasks on specified computing resources.
6. Translate job results to output values returned to the client.

Steps 1, 3 and 5 can be implemented in a similar fashion for all applications. However steps 2, 4 and 6 are application dependent. In order to simplify creation of applications, Everest provides generic implementations of these steps, the so called *application skeletons*, that can be configured by users. This declarative approach helps to avoid programming while adding applications to Everest.

In order to add an application to Everest a user should provide an application description that consists of two parts:

- *public information* that is used by clients in order to discover application and interact with it, including specification of inputs and outputs (also used to automatically implement Step 2 above).
- *internal configuration* that is used by Everest in order to process requests to the application and generate results (configuration of the application skeleton, application files and resource binding).

Currently Everest implements a single application skeleton for command-line applications which is suitable for porting to Everest existing applications. This skeleton generates jobs consisting of a single task. The internal configuration for this skeleton includes:

– a string template for mapping input values to a task command,
– input mappings that define how input values map to task input files,
– output mappings that define how task output files map to output values.

Additional skeletons for distributed applications with jobs consisting of multiple tasks, such as bag-of-tasks applications, composite applications (workflows), etc., are planned to be implemented in the future. The internal Compute layer of Everest already supports such jobs which is demonstrated by developing a generic application for running parameter sweep experiments [Volkov, Sukhoroslov].

Integration with Computing Resources

As was mentioned before, instead of providing it's own computing infrastructure Everest enables users to attach to it external computing resources and to run applications on arbitrary sets of these resources. From this point of view Everest can be seen as a multitenant metascheduling service.

Currently the preferred method for attaching a resource to Everest is based on using a developed program called *agent*. The agent runs on the resource and acts as a mediator between it and Everest. This method has one drawback - it requires deployment of the agent on each resource. However, it also brings a number of advantages in comparison to plain SSH access such as supporting resources behind a firewall and more strict security policies. Also the agent has minimal requirements (Python 2.6+) and is easy to install and run by an unprivileged user.

The agent supports integration with various types of resources via adapter mechanism. At the moment the following adapters are implemented:

– *local* — running tasks on a local server,
– *torque* — running tasks on a TORQUE cluster (agent is running on a submission host),
– *slurm* — running tasks on a SLURM cluster (agent is running on a submission host),
– *docker* — running tasks on a local server inside Docker containers.

The communication between an agent and the platform is implemented through the WebSocket protocol [Fette, Melnikov, 2011]. Upon startup an agent initiates connection with the platform to es-

tablish a bidirectional communication channel. This channel is used only for control and status messages. Authentication of an agent is performed by passing a secret token issued by Everest. Job data transfer is performed by an agent via the HTTP protocol.

In order to attach a resource to Everest a user should obtain a new resource token via Web UI, install the agent on the resource and run the agent with configuration including the obtained token. After the agent is connected to Everest it starts to send information about the resource state that is displayed in Web UI.

Besides standalone servers and clusters supported via the described agent, Everest also supports integration with the European Grid Infrastructure (EGI). This integration is implemented via EMI User Interface (UI). A user can attach as a new resource a specific EGI VO by providing a valid proxy certificate.

As with applications, a resource owner can manage the list of users that are allowed to use the resource to run applications. In order to run an Everest application it should be *bound* to at least one available resource. Everest implements flexible binding of resources to applications. Application owner can configure a static set of resources that should be used by Everest to run application tasks. In this case an application owner implicitly allows application users to run the application on these resources. Application owner can also enable *dynamic resource binding* when a user can manually select a resource for running his job. In both static and dynamic binding it is possible to specify multiple resources and let Everest to schedule application tasks across these resources.

Flexible resource binding opens new possibilities, but also brings some challenges. For example, if an application has commercial value or special hardware requirements, the application owner can restrict the application to run only on specific resources and disable dynamic binding. However, in such case the owner has to maintain resources used by application. On the other hand, if the application owner wants to share the application but doesn't have resources to support application users, he can enable dynamic binding and let users run the application on their resources. However, this brings another challenge of making the application portable across heterogeneous resources.

Programmatic Access

Running Everest applications via Web UI is easy and convenient, but it has some limitations. For example, if a user wants to run an application many times with different inputs, it is inconvenient to submit many jobs manually via web form. In other case, if a user wants to produce some result by using multiple applications, she has to manually copy data between several jobs. Finally, Web UI is not suitable if one wants to run an Everest application from his program or some other external application.

To support all these cases, from automation of repetitive tasks to application composition and integration, Everest implements a REST API. It can be used to access Everest applications from any programming language that can speak HTTP protocol and parse JSON format. However REST API is too low level for most of users, so it is convenient to have ready-to-use client libraries built on top of it. For this purpose a client library for Python programming language called Python API was implemented.

Figure 2 contains an example of program using Python API. It implements a simple diamond-shaped workflow (depicted in the top right corner of the picture) that consists of running four different applications — *A*, *B*, *C* and *D*.

At the beginning the program imports *everest* module which implements Python API and creates a new *session* by using a *client token*. Each client accessing Everest should present such token with its request in order to authenticate itself. In order to access applications the program creates a new *App* object for each application by passing application ID and the session.

The program initiates requests to applications by invoking *run()* method of the application object. Inputs are passed as a Python dictionary with keys and values corresponding to parameter names and values respectively. The *run()* method returns a *Job* object that can be used to check the job state and obtain the result. Note that *run()* method doesn't block the program until the job is done. Instead Py-

thon API performs all job related activities in the background thus allowing the program to continue its execution.

```
import everest

session = everest.Session(
  'https://everest.distcomp.org', token = '...'
)

appA = everest.App('52b1d2d13b...', session)
appB = everest.App('...', session)
appC = everest.App('...', session)
appD = everest.App('...', session)

jobA = appA.run({'a': '...'})
jobB = appB.run({'b': jobA.output('out1')})
jobC = appC.run({'c': jobA.output('out2')})
jobD = appD.run({'d1': jobB.output('out'), 'd2': jobC.output('out')})

print(jobD.result())

session.close()
```

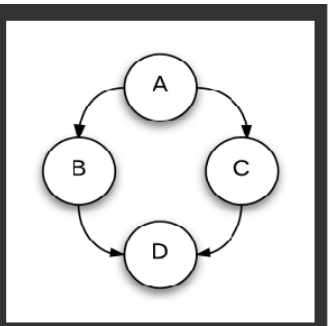


Fig. 2. Example program in Python API

In the presented example all jobs except *jobA* cannot be submitted immediately because they have data dependencies. Note how the program refers to the output values of the *jobA* by using *output()* method of the job and specifying the output names *out1* and *out2*. This method also doesn't block the program until the output value is available. Instead Python API will wait in the background until the *jobA* is completed, read the output values and then submit *jobB* and *jobC*. Similarly the *jobD* will be actually submitted to Everest only after *jobB* and *jobC* are completed.

The nonblocking semantics of Python API makes it simple to describe arbitrary workflows without requiring a user to implement boilerplate code dealing with waiting for and passing job results. This approach also implicitly supports parallel execution of independent jobs such as *jobB* and *jobC*.

After all jobs are created (while possibly not submitted) the program waits for a final result by calling the *result()* method on *jobD*. This method blocks the program until the job is completed and returns the job result. The result is returned as a Python dictionary with keys and values corresponding to output names and values respectively. Finally the program closes the session by invoking *close()* method. This terminates all background activities and ensures that the program exits normally.

Conclusions

The paper presented current state of Everest, a cloud platform supporting publication, execution and composition of applications running across distributed computing resources. The platform implementation provides all described functionality and is available online to all interested users [Everest]. Future work will address the remaining challenges such as implementation of advanced scheduling across multiple resources, publication of composite applications, supporting other types of applications (parallel, distributed, data-intensive) and integration with other types of computing resources (clouds, desktop grids).

References

Afanasiev A., Sukhoroslov O., Voloshinov V. MathCloud: Publication and Reuse of Scientific Applications as RESTful Web Services // Lecture Notes in Computer Science. — 2013. — Vol. 7979. — P. 394–408.

Everest. http://everest.distcomp.org/

Fette I., Melnikov A. The WebSocket Protocol. RFC 6455, Internet Engineering Task Force, 2011.

Richardson L., Ruby S. RESTful Web Services. O'Reilly, 2007.

Sukhoroslov O., Afanasiev A. Everest: A Cloud Platform for Computational Web Services // 4th International Conference on Cloud Computing and Services Science (CLOSER 2014). — P. 411–416.

Volkov S., Sukhoroslov O. Running Parameter Sweep Applications on Everest Cloud Platform // In these Proceedings.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 601−606

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75, 004.45

Running Parameter Sweep applications on Everest cloud platform

S. Yu. Volkov[a], O. V. Sukhoroslov[b]

Institute for Information Transmission Problems of the Russian Academy of Science, Kharkevich Institute, Bolshoy Karetny per. 19, build.1, Moscow 127051 Russia

E-mail: [a] fizteh.volkov@gmail.com, [b] oleg.sukhoroslov@gmail.com

Получено 30 сентября 2014 г.

Parameter sweep applications are a very important class of applications, which are typically defined as a set of computational experiments over a set of input parameters, each of which is executed with its own parameter combination. These computations arise in many scientific contexts. This article introduces the Parameter Sweep web service that runs such applications in distributed computing environment. Also discussed is the Everest cloud platform, on which this service is built.

Keywords: parameter sweep experiments, distributed computing, web services, cloud platform

Реализация запуска многовариантных расчетов на платформе Everest

С. Ю. Волков, О. В. Сухорослов

Институт проблем передачи информации им. А. А. Харкевича Российской академии наук, Россия, 127051, г. Москва, Большой Каретный переулок, д. 19, стр. 1

Многовариантные расчеты являются чрезвычайно важным классом приложений, обычно определяемых как набор вычислительных задач, определенных на множестве входных параметров и запускаемых с различными значениями данных параметров. Необходимость такого рода вычислений возникает во многих научных областях. Данная статья рассматривает веб-сервис, реализующий запуск данных приложений в распределенной вычислительной среде, а также облачную платформу Everest, на базе которой реализован данный сервис.

Ключевые слова: многовариантные расчеты, распределенные вычисления, веб-сервисы, облачная платформа

The work is supported by the Russian Foundation for Basic Research (grant No. 14-07-00309 A).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 601–606 (Russian).

Introduction

Parameter sweep applications are becoming extremely important in science and engineering. As an example, one can explore the behavior of the airfoil by running its model multiple times, depending on its properties, such as speed, angle attack, shape and so on. Parameter sweep applications address this kind of computations. They may be extremely time-consuming and require enormous amount of processor time. Therefore, this class of applications is an ideal class for distributed computing.

Parameter sweep applications involve some input set of computational parameters and files. Each parameter has its range of values, such as different angle attack values in the above example. Multiple computations, or tasks, are then run for different combinations of each parameter values. Each file may be the input of two or more tasks. Each task is supposed to have some output, typically in the form of the model's output parameters, describing the computed characteristics of this model, depending on the input parameters. The resulting set of all task outputs represents the result of the whole parameter sweep experiment.

The presented parameter sweep service has been influenced by the NimrodG system [Bethwaite et al., 2010; Buyya, Abramson and Giddy, 2000]. This system has the so-called plan file, describing the whole experiment, including parameters, input and output files and the command to be executed for each task. These tasks are generated for each combination of the parameters using the cartesian product. Our service takes the cartesian product as well, but allows users to impose some restrictions on each parameters combination in the form of the *constraints directive* (see **Plan file structure** for more details). It also allows to filter the output results and to introduce the computation's criterion to find out the 'best' (in terms of criterion) values of output parameters.

This service has been written on the Everest platform. Before we proceed to the service itself, let us introduce this cloud platform.

Everest Platform

Everest [Sukhoroslov, Afanasiev, 2014; Sukhoroslov, Rubtsov, Volkov; Everest] is a cloud platform that supports publication, sharing and reuse of scientific applications as web services. The underlying approach is based on a uniform representation of computational web services and its implementation using REST architectural style.

In contrast to traditional service development tools, Everest follows the Platform as a Service cloud delivery model by providing all its functionality via remote interfaces. A single instance of the platform can be accessed by many users in order to create, run and share services with each other without the need to install additional software on users' computers.

Another distinct feature of Everest is the ability to connect services with external computing resources. That means that service developer can provide computing resource for running service jobs. A service user can also override the default resource by providing another resource for running her jobs.

While the platform doesn't provide its own infrastructure to run compute jobs as classic PaaS examples, it can handle the problems of resource allocation, job management, data transfer and so on without the interference of users.

Everest is work in progress. The platform is currently undergoing experimental evaluation and pilot deployment.

Parameter sweep service

In order to initiate the parametric computation, the user needs to submit two files. **Figure 1** shows, how this service looks like in the client's web browser. The first one is the plan file, while the second one is the archive with the experiment's input files (currently supported are *tar.gz* and *zip* formats). If the computation succeeds, the result the user can download from the server is the archive with all of the re-

sults, satisfying both the criterion and the filter, if any. These results represent folders, containing the current task's output files as well as a parameter file, filled with the corresponding parameters values.

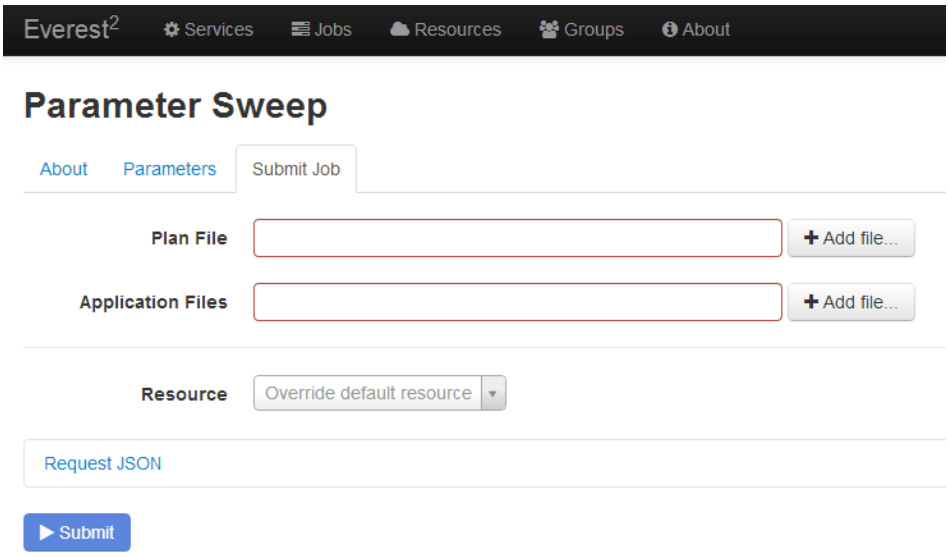


Fig. 1. Parameter Sweep web service

Let us delve a little bit into how this service works. **Figure 2** shows the architecture of the service. It's a package named *parametric*, written in the Scala language, as well as the so-called parametric application, which is implemented as a new Everest extension and has three standard methods: *prolog*, *epilog* and *taskStateChanged*. Its *prolog* method generates computational tasks (in the platform's internal format) and returns these tasks for subsequent execution by the platform on distributed computing resources. This is where the parametric extension requires plan tasks, generated from the plan file by the parametric package, and converts them to the internal format, mentioned above. The *taskStateChanged* method is invoked once some task's state has changed, for example if some task is completed or cancelled. This is a place, where the parametric application handles the task's output parameters, processing them through filters and computing the criterion's value (see **Plan file structure** for more details). Finally, the *epilog* method is invoked once all of the tasks have been completed. This is where the criterion part of the plan file, if any, is applied and the 'best' tasks are chosen and returned to the client.

Plan file structure

The syntax for our plan file has been inspired by the Nimrod's plan file. It is a simple text file, describing the experiment. It has the following directives: parameter, constraint, input, substitute, command, output, filter and criterion. The parameter, input, command and output directives are required. The constraint, substitute, filter and criterion directives are optional. Each directive, except for command, could span multiple lines. This includes either the line break, or the new line with the same directive. The directives order is significant and should be as described below.

The description of the whole computational experiment uses the standard *$var* and *${var}* substitute syntax. In our service, the curly braces '{' and '}' are optional with only one exception. If one of the substitute variables is the prefix of another, the longest one must be enclosed into curly braces. As an example, if we have two variables, *var* and *var1*, and don't enclose the second one (i.e. leave it as *$var1*), only the *$var* part will be substituted, which is obviously not what we expect.

Parameter directive. This directive describes our experiment's parameters. It has the following syntax:

parameter {name} {range}

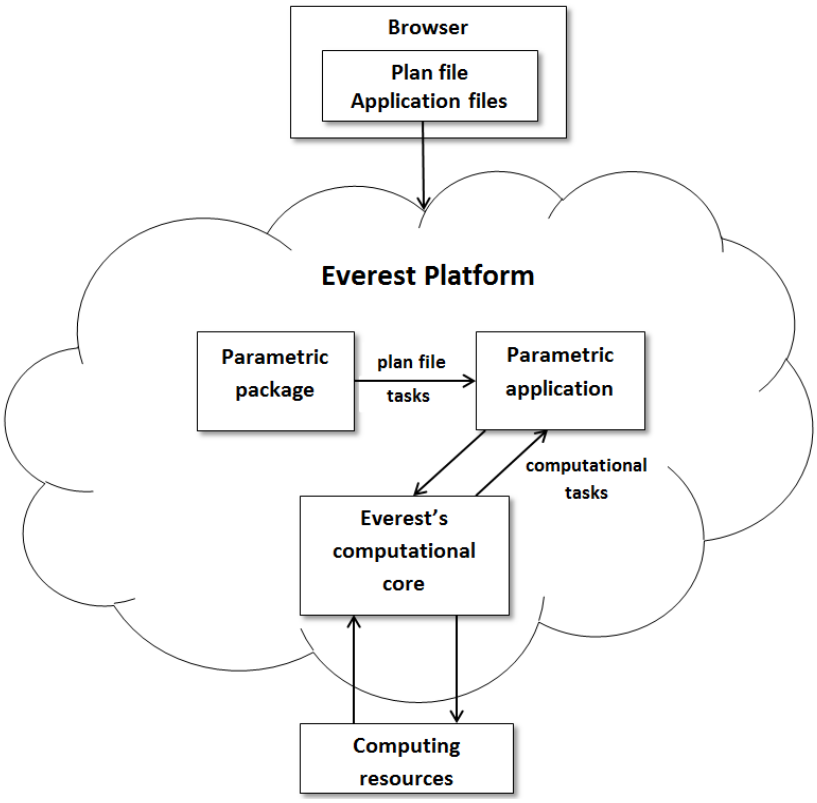


Fig. 2. Architecture of the Parameter Sweep service

The {name} is parameter's name and {range} represents the range of its values. The {range} part has the following syntax:

from {value} to {value} step {value} **or**

{list of values, separated by at least one whitespace}

The first syntax is for integers and floating point numbers only. Their values are simply integers and floating point numbers respectively. Otherwise, all of the values, separated by whitespace, should be listed. If the value itself contains at least one whitespace, it should be enclosed into quotation marks.

Constraint directive. This is an optional directive. Its goal is to impose restrictions on experiment's parameters. Its syntax is as follows:

constraint {type} {constraint expressions, separated by a comma}

The constraint expressions are math expressions. They support the following operators:

+ (addition), — (subtraction), * (multiplication), / (division), ^ (exponentiation), % (remainder operator), < (less than), <= (less than or equal to), > (greater than), >= (greater than or equal to), = (equal to), != (not equal to), and (conjunction), or (inclusive or), not **or** ! (negation).

They can also contain parenthesis, all standard functions like 'sin', 'cos', 'log', 'abs', 'sqrt' etc. and the names of parameters to be substituted, prefixed by the '$' sign.

The constraint's {type} specifies, whether the parameters' values or these values' indices are substituted into the constraints' expressions. It should be 'value' or 'index' respectively.

Input Files directive. This directive lists each task's input files. Here is its syntax:

input files {file names or paths, separated by at least one whitespace}

As with the **parameter** directive, these file names or paths must be enclosed into quotation marks if they contain whitespaces. Please, note that they could also be parameterized. File paths are directories inside the parameter sweep's input archive and could contain file masks using the standard * sign.

Some file names could be prefixed by the '@' sign. Such files are called *substitution files* and should contain parameters' names (specified in the **parameter** directive) with the *$* or *${}* substitution

notation. Let us assume this file is a script file, written in Scala. Here's what some part of this file may look like:

val v1 = $i
val v2 = $d
val result = someFunction(v1, v2)

Here *i* and *d* are parameters, specified in the **parameter** directive. This file will be looked through in the search of the *$var* or *${var}* syntax. If found, *$i* and *$d* will be substituted with current task's values of parameters *i* and *d*. For example, if some task has the values *i = 7* and *d = -123.32*, this part of the script will be transformed into

val v1 = 7
val v2 = -123.32
val result = someFunction(v1, v2)

It's important to notice that the parser has no idea, where this parameter could be used. If it is an *Int* or a *Boolean*, it needs to be substituted as is. However, if it is a *String*, it needs to be substituted with quotation marks. And since no one but the client himself knows, how his script works, it's up to him to take types into consideration. For example, the piece of Scala code above works if *v1* is an *Int* or a *Boolean*, but if it's a *String*, this code should be modified like val *v1 = "$i"*.

Command directive. This is the command to be executed on the computational nodes. Its syntax is the following:

command {command}

There is only one command line, which should contain only one command. Since it will be executed on the computational nodes, the Nimrod 'node: execute' declaration has been replaced with 'command'. This directive could also be parameterized.

Output Files directive. These are the task's output files. They files must be the output of the command in the previous directive, which is usually some script. As with previous directive, they could be parameterized. Here is the syntax of this directive:

output_files {file names, separated by at least one whitespace}

Some of these files could be prefixed with the '@' sign. Such files should contain the task's outputs and have the following structure:

output1 = output1Value
output2 = output2Value **and so on**.

It's important to notice that the names of the outputs must be unique. Different output files must list different outputs.

Filter directive. This directive is optional. The values of task's output parameters, described in the previous section, may be processed by the filters. Here is this directive's syntax:

filter {filter expressions, separated by a comma}

Filter expression is essentially the same as constraint expression, with only one difference: instead of parameter names to be substituted it has the outputs to be substituted. Since the names of the outputs must be unique, this directive will simply look through all of the output files until it finds the corresponding outputs. Only the tasks, satisfying all of the filter expressions, will be taken into consideration and processed by the criterion directive.

Criterion directive. This directive is optional as well. It is only applied to the tasks, satisfying the filter directive. Here is its syntax:

criterion {type} {criterion function}

The {type} part must be either *max* or *min* (case-sensitive). Criterion function is a math expression over the output parameters. The results of this directive are all of the tasks, maximizing or minimizing (depending of the {type} part) the criterion function.

Extensive example

Let us consider an example of the aforementioned directives using a well-known program of molecular docking *Autodock Vina* [Autodock Vina]. In our Parameter Sweep computation we will run 10

docking tasks with different ligands and find tasks with minimum affinity (energy). Here is the appropriate plan file:

```
parameter n from 1 to 10 step 1
input_files @run.sh vina write_score.py protein.pdbqt ligand${n}.pdbqt config.txt
command ./run.sh
output_files ligand${n}_out.pdbqt log.txt @score
criterion min $affinity
```

In this example we use **parameter** directive to define parameter *n* that refers to ligand number and takes integer values from 1 to 10.

In the **input files** directive we define input files per task. Note how *${n}* is used in the name of ligand file to refer to the value of parameter *n*. That means that task for *n=1* will use input file *ligand1.pdbqt*, the task for n=2 will use file *ligand2.pdbqt*, and so on.

In the **substitute files** directive we specify that we want to substitute all strings *$n* or *${n}* inside file *run.sh* with the value of parameter *n*. This is done on a per task level, so each task will use a different run script.

The **command** directive simply runs the script *run.sh*.

The **output files** directive lists each task's output files. In this example these are the output ligand file, the log as well as the score file, which lists the *affinity* output parameter. As already stated, this file must have the structure *affinity = affinityValue.*

Finally, we use **criterion** directive to specify that we are interested in results with minimal affinity value. The criterion function simply refers to this value as *$affinity*. This directive will look through the task's output files until it finds the *affinity* output parameter.

Conclusions and future work

This paper has covered the class of parametric applications, extremely important in many areas, such as science or engineering. We have introduced the Parameter Sweep service, built on the Everest cloud platform. Compared to prior works, it has a number of advantages, such as the possibility to filter the output results just as the client sees fit. Moreover, since it's a web service, the clients don't have to download, install and run any software. However, the work is still in progress. The main goal is to make the service more user-friendly. Among other things, this includes the web interface for building a plan file, which is definitely much more convenient than typing this file in a text editor and less error-prone. Future work will also address the problem of efficient scheduling of parameter sweep computations across multiple heterogeneous resources.

References

Autodock Vina. http://vina.scripps.edu/

Bethwaite, B., Abramson, D., Bohnert, F., Garic, S., Enticott, C., Peachey, T., "Mixing the Grid and Clouds: High-throughput Science using the Nimrod Tool Family", "Cloud Computing: Principles, Systems and Applications", Eds Antonopoulos and Gillam, Springer, pp 219-237, ISBN: 978-1-84996-240-7, 2010.

Buyya R., Abramson D. and Giddy J. "Nimrod/G: An Architecture of a Resource Management and Scheduling System in a Global Computational Grid", HPC Asia 2000, May 14–17, 2000. — P. 283–289, Beijing, China.

Everest. http://everest.distcomp.org/

Sukhoroslov O., Afanasiev A. Everest: A Cloud Platform for Computational Web Services // 4th International Conference on Cloud Computing and Services Science (CLOSER 2014). — P. 411–416.

Sukhoroslov O.V., Rubtsov A.O., Volkov S.Yu. Development of Distributed Computing Applications and Services with Everest Cloud Platform // In these Proceedings.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 607–613

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.27

Memory Benchmarking Characterisation of ARM-based SoCs

G. T. Wrigley[a], R. G. Reed, B. Mellado

School of Physics, University of the Witwatersrand. 1 Jan Smuts Avenue, Braamfontein, Johannesburg, South Africa, 2000

E-mail: [a] thomas.wrigley@cern.ch

Получено 30 сентября 2014 г.

Computational intensity is traditionally the focus of large-scale computing system designs, generally leaving such designs ill-equipped to efficiently handle throughput-oriented workloads. In addition, cost and energy consumption considerations for large-scale computing systems in general remain a source of concern. A potential solution involves using low-cost, low-power ARM processors in large arrays in a manner which provides massive parallelisation and high rates of data throughput (relative to existing large-scale computing designs). Giving greater priority to both throughput-rate and cost considerations increases the relevance of primary memory performance and design optimisations to overall system performance. Using several primary memory performance benchmarks to evaluate various aspects of RAM and cache performance, we provide characterisations of the performances of four different models of ARM-based system-on-chip, namely the Cortex-A9, Cortex-A7, Cortex-A15 r3p2 and Cortex-A15 r3p3. We then discuss the relevance of these results to high volume computing and the potential for ARM processors.

Keywords: ARM, Memory, Benchmarks, Throughput-oriented computing, High-volume computing

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 607–617 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 607–613

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

Описание тестирования памяти однокристальных систем на основе ARM

Г. Т. Ригли, Р. Г. Рид, Б. Мелладо

Отделение Физики, Университет Витватерсранда, Южная Африка, 2000, Йоханнесбург, 1 Ян Смут Авеню

Мощность вычислений традиционно находится в фокусе при разработке крупномасштабных вычислительных систем, в большинстве случаев такие проекты остаются плохо оборудованными и не могут эффективно справляться с ориентированными на высокую производительность рабочими нагрузками. Кроме того, стоимость и вопросы энергопотребления для крупномасштабных вычислительных систем всё ещё остаются источником беспокойства. Потенциальное решение включает в себя использование низко затратных процессоров ARM с маленькой мощностью в больших массивах в манере, которая обеспечивает массивное распараллеливание и высокую пропускную способность, производительность (относительно существующих крупномасштабных вычислительных проектов). Предоставление большего приоритета производительности и стоимости повышает значимость производительности оперативной памяти и оптимизации проекта до высокой производительности всей системы. Используя несколько эталонных тестов производительности оперативной памяти для оценки различных аспектов производительности RAM и кэш-памяти, мы даем описание производительности четырех различных моделей однокристальной системы на основе ARM, а именно Cortex-A9, Cortex-A7, Cortex-A15 r3p2 и Cortex-A15 r3p3. Затем мы обсуждаем значимость этих результатов для вычислений большого объема и потенциала для ARM- процессоров.

Ключевые слова: ARM-процессор, память, эталонные тесты, вычисления, ориентированные на высокую производительность, вычисления большого объема.

1. Introduction and Background

The volume of data generated by the wide array of available computing services in the consumer, industrial, academic and other spheres is vast and ever-increasing and the challenge posed by this is often called 'Big Data' — a term which is rapidly approaching ubiquity, with its wide array of potential applications generating a great deal of interest across many fields [Manyika et al., 2011]. Large-scale computing systems have traditionally been designed with computationally-intensive tasks as their primary focus. These systems are often highly inefficient for the purposes of throughput-oriented computing. A computing paradigm called High Volume Computing (HVC) has been proposed by Zhan et al [Zhan et al., 2012], which they define as a large number of loosely-coupled, throughput-oriented workloads, with increasing throughput volume being a principal goal of such system designs. A potential HVC solution involves the use of ARM processors, which are low-power, low-cost and low-energy consumption system-on-chips (SoCs), in large arrays which would provide very high levels of parallelisation. ARM-based SoCs, which are commonly used in mobile devices such as smartphones and tablets, are low-cost, mass-produced and potentially highly energy-efficient [Aroca, Gonçalves, 2012], all of which bodes well for both system affordability and energy efficiency. Although large scale computing has traditionally placed its primary focus on processor performance, there is an increasing shift towards including memory performance in this focus [Dongarra, Heroux, 2012; Ang et al., 2010]. Memory performance is a key component of overall system performance and is particularly important for throughput rates, memory bottlenecks could potentially affect energy-efficiency and cost through under-utilisation of existing system hardware. Using ARM-based SoCs in any proposed solution therefore requires that the performance of ARM-based SoCs be properly characterised and understood.

2. Experimental Configuration

The primary memory (i.e. RAM and cache) performance of four models of ARM SoC-based development boards were evaluated. Commercially available development boards were used for the purposes of benchmarking. The technical specifications of these boards are listed in Table 1 below.

Table 1. ARM development board hardware specifications

	Cortex-A7	**Cortex-A9**	**Cortex-A15 r2**	**Cortex-A15 r3**
Platform	Cubieboard2	Wandboard Quad	Odroid-XU+E	Jetson TK1
SoC	Allwinner A20	Freescale i.IMX6Q	Samsung Exynos 5410	NVIDIA Tegra K1
ARM Core Revision	r0p4	r2p2	r3p2	r3p3
Cores	2	4	4	4
Power-saver cores	0	0	4 Cortex-A7	1
Max. CPU Clock (MHz)	1008	996	1600	2300
L1 Cache (kB)	32	32	32	32
L2 Cache (kB)	256	1024	2048	2048
RAM Size (MB)	1024	2048	2048	2048
DDR3 RAM Type	432 MHz 32 bit	528 MHz 64 bit	800 MHz 64 bit	933 MHz 64bit DDR3L
Approx. 2014 Retail Price (USD)	65	129	169	192
Operating System	Ubuntu	Linaro	Ubuntu	Ubuntu

A Linux-based distribution was installed on all four board models. Three benchmarking software programmes were used to evaluate the memory performance of these four boards, namely the LMBench benchmark suite, the STREAM benchmark and the Parallel Memory Bandwidth Benchmark

(*pmbw*). The LMBench benchmarking suite analyses several aspects of memory performance — this study focuses on the measures of memory latency. The STREAM benchmark provides a measure of sustained memory bandwidth. STREAM works by generating an array of random numbers of a specified size (which is then stored in RAM) and performs four types of operations, namely copy, scale, add and triad. Measures of sustained bandwidth are then produced for each of these four tests. The *pmbw* benchmark is similar to STREAM in that it also provides a measure of memory bandwidth, but is also strongly influenced by memory latency. The *pmbw* benchmark consists of 14 separate subtests, each performing a slightly different operation. There are 5 variables which distinguish the 14 subtests, namely: (1) sequential scanning or a random access (permutation walking) test, (2) write or read operation, (3) bit size transferred in each operation, (4) pointer-based iterations vs index-based array access, and (5) number of operations per loop (1 — Simple vs 16 — Unroll) [Bingmann, 2013]. Two of the subtests involve Multiroll Loops and are not analysed here. The benchmark generates an array and runs one of the subtest routines. The allocated array size is then increased and the subtest routine is then repeated. This is repeated until the highest power of 2 able to fit onto the system's RAM is reached. These steps are repeated for each one of the subtest routines. *pmbw* is useful because it measures both bandwidth and latency and can potentially offer deeper insight into memory performance.

3. Results and Discussion

3.1. STREAM and LMBench

For the STREAM benchmark, which measures sustained memory bandwidth, the two Cortex-A15-based systems are clearly shown to be the best-performing of the four systems, with the r3p3 (Jetson TK1) obtaining the highest absolute bandwidth and the r3p2 (Odroid) obtaining the high bandwidth efficiency (i.e. percentage of theoretical maximum obtained). The Cortex-A7 displays reasonable bandwidth efficiency, while the Cortex-A9, which is the oldest of the four systems, achieves very low bandwidth efficiency, reaching only 16% of its theoretical maximum. In the case of RAM and cache latencies, the Cortex-A7, Cortex-A15p2 and Cortex-A15p3 all perform well, recording low latencies, with a clear correlation between CPU clock frequency and cache latency. The latency of the Cortex-A9 is also significantly higher the other three SoCs. For both of these benchmarks, a clear positive correlation can be seen between age of SoC design and performance. Table 2 below summarises the results obtained from both LMBench and STREAM for all four boards.

Table 2. LMBench and STREAM Benchmark Results

	Cortex-A7	**Cortex-A9**	**Cortex-A15 r2**	**Cortex-A15 r3**
Copy (MB/s)	1996	1329	6066	6430
Scale (MB/s)	1444	1110	6114	6403
Add (MB/s)	757	1448	5413	5358
Triad (MB/s)	702	1290	5275	5302
RAM (Theoretical MB/s)	3296	8054	12 207	14 236
Ave. RAM B/W Efficiency (%)	37	16	47	41
L1 Latency (ns)	3.02	4.02	2.51	1.73
L2 Latency (ns)	9.2	30.8	13.8	9.95
RAM Latency (ns)	58.5	119.8	104.8	115.6

3.2. The pmbw benchmark

The design of the *pmbw* benchmark means that each subtest routine generates several hundred sets of observations — between 200 and 300 observations in the case of the four systems tested here. Because there are several hundred observations per subtest and 12 subtests which are analysed here,

the volume of data produced by this benchmark for each system is very large — numbering around several thousand observations. For this reason, statistical tools are useful for extracting meaning from these data sets. A statistical test known as analysis of variance (ANOVA) was used to analyse the results of this benchmark. ANOVA is used to compare multiple datasets and determine whether the individual means of these datasets are equal to one another. More specifically, ANOVA compares the variance within each of these datasets to the variance which is present between these datasets and determines whether statistically significant differences exist between these datasets [Larson, 2008]. If statistically significant differences between these datasets do exist, various *post hoc* tests and analyses can then be used to gain greater insight into the distribution and nature of these differences.

In this case, each subtest (with its 200-300 observations per system) represents a dataset and ANOVA is used to determine whether these individual subtests are statistically similar to one another. A two-way analysis of variance showed that significant differences existed between the subtest groups for all four boards — i.e. at least one pair of means was different from one another. *Post hoc* analysis was then conducted to gain greater insight into the nature and distribution of these results. This analysis revealed the results generated by the 12 subtests appear to be distributed into five general groupings, with each grouping being made up of two, three or four subtests. As each subtest results from a combination of the benchmark's five function variables, the existence of these five groupings gives a greater level of insight into which of these characteristics appear to have the greatest impact on performance — insights which allow for memory performance to be better understood. The types of subtests which make up each grouping are briefly detailed in Table 3 below.

Table 3. Subtests contained *pmbw* in general result groupings

Group no.	Subtest types in group	Abbreviation
1	Random Pointer Permutations (Perm)	Random Pointer Permutation
2	Sequential Reading — 32 bit Simple Loop	SeqRead32Simple
3	Sequential Write — 32 bit Simple & Unroll Loop	SeqWrite32 Simple+Unroll
4	Sequential 32 bit Unroll & 64 bit Simple Loop	Seq32Unroll+64Simp
5	Sequential 64 bit Unroll Loop	Seq64Unroll

Based on the subtest result groupings determined above, the average of the two/three/four RAM bandwidth results for each of the five groupings was plotted. These bandwidth results are shown in Fig. 1 below. The first grouping (Random Pointer Permutation) is substantially lower than the other four groupings. This is, however, consistent with expectations, as this benchmark is based on a random pointer permutation and is essentially a measure of raw bandwidth and latency for one memory fetch cycle, while the other four are measures of sustained memory bandwidth for sequential scanning [Bingmann, 2013]. These results indicate that the Cortex-A7 (Cubieboard2) produces the lowest performance, while the Cortex-A9 (Wandboard) performing approximately 50% better. The Cortex-A15p2 (Odroid) performs approximately 50% better than the Wandboard, while the Cortex-A15p3 (Jetson TK1) in turn performs approximately 50% better than the Odroid. The Wandboard performing better than the Cubieboard2 appears to be inconsistent with the memory latency and sustained memory bandwidth results obtained by LMBench and STREAM, which showed the Cortex-A7 performing better than the Cortex-A9 in both cases. While these two random pointer permutation subtests are not solely dependent on memory latency, this would be expected to have some effect on random memory access performance. It is not immediately clear why the results produced by pmbw appear to conflict with the trends implied by the obtained LMBench results, although factors such as the Cortex-A9 SoC's 64 bit RAM bus width compared to the Cortex-A7 SoC's 32 bit RAM bus width may influence this result. This question must be further investigated in future work.

Groupings 2, 3, 4 and 5 are all based on sequential scanning rather than random memory access. This means that these four groupings offer some measure of sustained memory bandwidth. The very bandwidth measurements shown in Fig. 1 below are not directly comparable to STREAM as the measurements below represent average bandwidth, while the measurements in Tab. 2 for STREAM

are for sustained main memory (i.e. RAM) bandwidth. The general profile of the first three groups (i.e. Cortex-A7, Cortex-A9 & Cortex-A15p2) is consistent with the results obtained by the STREAM benchmark. The performance of the Cortex-A15p3 (NVIDIA Jetson TK1) is, however, more than twice that of the Cortex-A15p2 (Odroid-XU+E), while the Jetson TK1 only marginally outperforms the Odroid-XU+3 on the STREAM benchmark. This is most likely due to the influence higher clock frequency of the Jetson TK1 and its subsequently lower L1 and L2 cache latencies.

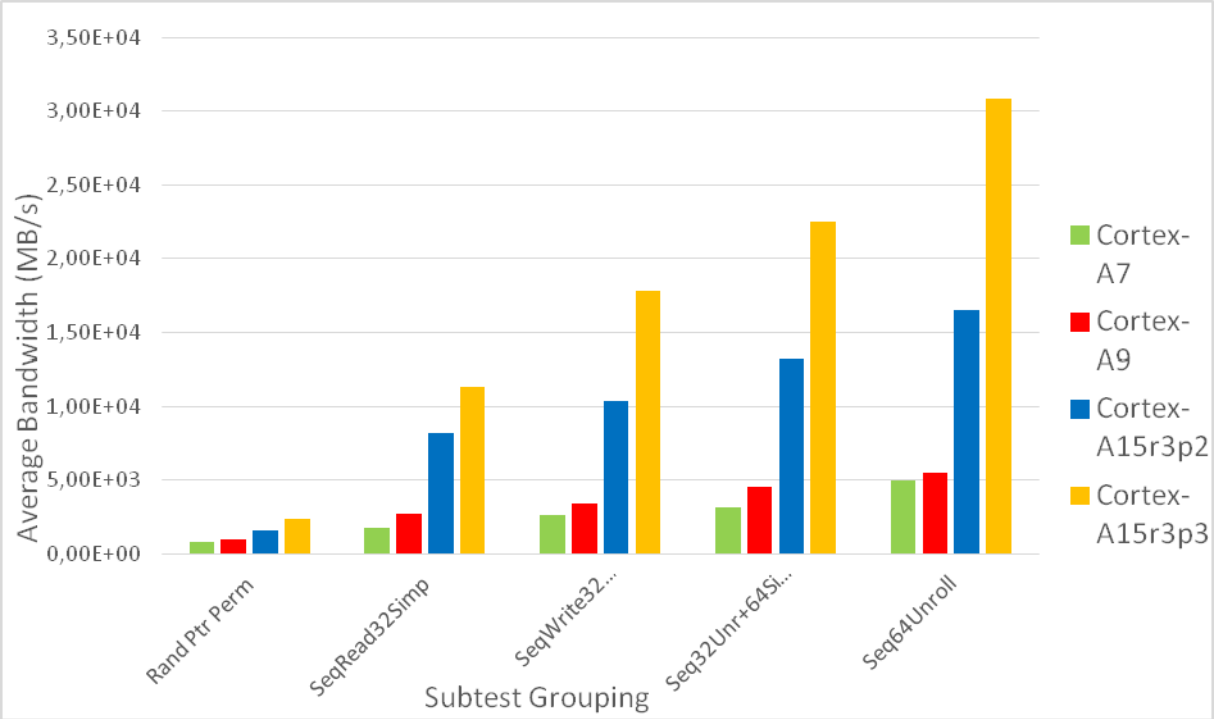


Fig. 1. *pmbw* Bandwidth Grouping Results

3.3. Discussion and Analysis

A clear correlation between age of SoC design and overall memory performance is clearly demonstrated, with the newest SoC, the Cortex-A15p3 performing the most effectively and the oldest SoC, the Cortex-A9 performing the least effectively. The bandwidth efficiency of the newest board (41%) is lower than the second-newest board (47%), which is something that can be improved upon in future board and SoC designs. Preliminary results presented at the 2014 South Africa Institute of Physics Conference by Mitchell Cox [Cox, 2014] show that it is possible to obtain I/O connection rates of approximately 300 MB/s between two Cortex-A9 SoCs. This suggests that memory performance is not the primary source of throughput rate bottlenecks for relatively simple algorithms (i.e. where CPU performance is not the bottleneck), as this figure is approximately 5 times lower than the sustained memory bandwidth measured for the Cortex-A9. As I/O connection rates continue to improve, this low sustained memory bandwidth may present an obstacle to overall throughput rates. The Cortex-A9 design tested here is, however, more than six years old. The performance improvements of the newer SoCs mean that I/O capacity is more likely to be the primary cause of throughput rate bottlenecks, particularly for algorithms which are not computationally intensive. These improvements are expected to continue as newer ARM-based SoCs are released, particularly with the soon-to-be released ARMv8 architecture 64 bit SoCs (such as NVIDIA's Project Denver). The potential of ARM-based SoCs for use in HVC systems therefore remains strong. Intel Atom-based SoCs hardware (i.e. development boards) will be procured in due course, in order to evaluate their potential for use in HVC.

Conclusion

In summary, the memory performance of four ARM SoC-based development boards was evaluated using three separate memory benchmarks. Of the four boards, the Cortex-A15r3p3 NVIDIA TK1 — the newest SoC design — was the best both in terms of sustained memory bandwidth and cache latency, reaching 6.4 GB/s for the former. Throughput-oriented workloads are thus unlikely to saturate memory, particularly for tasks which are more computationally complex than the STREAM benchmark. Although bandwidth efficiency for the newest board is lower than for the second-newest board, the general improvement in memory performance of the newer SoC designs displayed by these benchmarks suggest that memory performance will continue to improve in the near future. This suggests that ARM-based SoCs are viable candidates for use in HVC.

References

Ang, J. A., Barrett B. W., Wheeler K. B., Murphy R. C. 2010. Introducing the graph 500. No. SAND2010-3263C (Albuquerque, NM: Sandia National Laboratories).

Aroca R. V., Gonçalves L. M. G. Towards green data centers: A comparison of x86 and ARM architectures' power efficiency // Journal of Parallel and Distributed Computing. — 2012. — **72**. — P. 1770–1780.

Bingmann, T. pmbw — Parallel Memory Bandwidth Benchmark/Measurement. 2013. Retrieved from: http://panthema.net/2013/pmbw/

Cox, M. The development of a general purpose Processing Unit for the upgraded electronics of the ATLAS detector Tile Calorimeter, *SAIP 2014*. 2014 (submitted).

Dongarra, J., Heroux M. A. 2013. Toward a new metric for ranking high performance computing systems. No. SAND2013-4744 312 (Albuquerque, NM: Sandia National Laboratories).

Larson, M. G. Analysis of variance // Circulation. — 2008. — **117.1**. — P. 115–121.

Manyika, J., Chui M., Brown B., Bughin B., Dobbs R., Roxburgh C., Hung Byers A. 2011. Big data: The next frontier for innovation, competition and productivity. (New York City, NY: McKinsey Global Institute).

Zhan, J., Zhang L., Sun N., Wang L., Zhen J., Luo C. High volume throughput computing: Identifying and characterising throughput oriented workloads in data centers // Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International. — 2012. — P. 1712–1721.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 615–620

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.45

Computational task tracking complex in the scientific project informational support system

N. V. Yuzhanin, Yu. A. Tipikin, I. G. Gankevich, V. I. Zolotarev

Saint Petersburg State University, University ave. 35, St. Petersburg, Peterhof, 198504, Russia

E-mail: {yuzhanin, iutipikin, igankevich, viz}@cc.spbu.ru

Получено 4 декабря 2014 г.

This work describes the idea of the system of informational support for the scientific projects and the development of computational task tracking complex. Due to large requirements for computational experiments the problem of presentation of the information about HPC tasks becomes one of the most important. Nonstandard usage of the service desk system as a basis of the computational task tracking and support system can be the solution of this problem. Particular attention is paid to the analysis and the satisfaction of the conflicting requirements to the task tracking complex from the different user groups. Besides the web service kit used for the integration of the task tracking complex and the datacenter environment is considered. This service kit became the main interconnect between the parts of the scientific project support system and also this kit allows to reconfigure the whole system quickly and safely.

Keywords: service desk, task tracking, HPC, web service

Комплекс слежения за вычислительными задачами в системе информационной поддержки научных проектов

Н. В. Южанин, Ю. А. Типикин, И. Г. Ганкевич, В. И. Золотарев

Санкт-Петербургский государственный университет, Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

В данной работе рассматривается идея системы информационной поддержки научных проектов и построение комплекса слежения за вычислительными задачами. Ввиду больших потребностей в вычислительных экспериментах предоставление информации о вычислительных задачах на HPC-ресурсах становится одной из важнейших проблем. В качестве решения этой проблемы предлагается нестандартное использование системы service desk — построение на ее базе комплекса слежения за выполнением вычислительных задач на распределенной системе и ее сопровождения. Особое внимание в статье уделено анализу и удовлетворению противоречивых требований к комплексу со стороны разных групп пользователей. Помимо этого, рассмотрена система веб-служб, служащая для интеграции комплекса слежения с окружением датацентра. Данный набор веб-служб является основным связующим компонентом системы поддержки научных проектов и позволяет гибко изменять конфигурацию системы в целом в любое время с минимальными потерями.

Research was carried out using computational resources provided by Resource Center "Computer Center of SPbU" (http://cc.spbu.ru/)

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 615–620 (Russian).

Introduction

As a result of the growth of an HPC industry many powerful supercomputers appeared in the last years. Their peak performance seems to be large in terms of GFLOPS * hour, but one should always try to keep resources loaded at an acceptable level in order to achieve good average utilization coefficient of the resources. This implies an application of the Capacity Management to the supercomputer [Gayduchok et al., 2012]. Such approach consists of monitoring, application sizing, analysis and capacity planning, and it could be a really challenging task in modern hybrid systems including the heterogeneous resources (different CPU architectures, specialized accelerators like NVIDIA Tesla or Intel MIC). More than that multi-user environment brings the following problems: load balancing and security, logging and accounting tasks.

Problems mentioned above are solved partly by the resource managers such as Portable Batch System (PBS). There are many implementations of Portable Batch System that differ from each other in detail but with the same main idea. Such systems start jobs in accordance with a schedule that is based on user preferences, capacity plan, resources load and availability. The scheduler can be a part of the PBS system or can be installed from a separate software package. It makes a decision about job start time depending on its rules based on the capacity plan and system data collected by PBS. This data updates constantly with logging in the PBS log files. PBS also store log files with information about jobs: owner, timeframes, requested and used resources, etc. So, in order to get accounting data one needs to have a log parser that will retrieve the data from large PBS log files.

The PBS system has a possibility to send reports by e-mail to users when their jobs are started or stopped (with job status that indicates successful execution or error). But this is inconvenient too: there are too many e-mails with such information (users usually run many jobs), so such a large list of emails is hard to analyze. Console PBS commands usually require some Linux skills, so they are inconvenient for inexperienced users too. In this article some approaches for controlling of the tasks on shared HPC resources, which are quite convenient but not traditional, are reviewed.

These approaches are reviewed as a part of the scientific project informational support system. The support system regulates all relations between the supercomputer center and scientists that execute the projects. Such approach allows using information about computational jobs in analysis and continuous service improvement.

Conflict of interests

Center for collective use of HPC resources as any social institute involves working with groups of users having different goals. Question of retrieving monitoring information and computational job tracking is answered differently by the two main groups — users and staff of HPC center which have different capabilities and different goals. As a result, they need different representations of the monitoring information about the computational jobs. But within these groups we can also distinguish smaller groups. Users can be divided into executors and leaders while computer center staff consists of administrators and managers. These subgroups have different requirements for the job tracking too.

For example, user who executes scientific task and directly submits computational jobs to a PBS cluster needs to know which of his jobs are completed and investigate job failures. He needs to know execution time of finished jobs and amount of consumed resources. Important factor is user-friendly, clear representation of the information. At the same time user does not need to know such an information as allocated slots or other technical details.

Unlike the executor, research work principal is not interested in every particular job. He must control executors, plan the use of the computational resources necessary for their research. Last, he must report to grantors. That is why he needs statistics about each subordinate: he needs statistics with the same or lower level of detail as executor and at the same time he needs general reports reflecting the overall progress of the research work of his scientific group. For him the tables are more suitable than graphs and diagrams. He usually needs more parameters than executor as he ought to assess the

overall process. He can also find automatic reports generation and prediction of resources usage very useful because he has to estimate the efficiency of the work. Also this way he can generate and submit capacity plans for oncoming quarter. He also has to submit requests for additional resources in case of lack of computational power.

On the other side of computational process there is the staff that supports the Computing Center services: system administrators and managers of the computer center. Administrators should have the possibility to get any available information about any job. At the same time they need to get selections from overall job statistics about cluster usage. For example, it can be necessary for capacity planning and load balancing. Also administrators have to track erroneous jobs and solve any problems related to jobs, system and hardware. So, they need a monitoring system that meets these requirements and also a service desk system which can be used to facilitate work with users and improve work flow. Such system can speed up overall work process, clear interaction with users by creating tickets and speed up the incident processing by registering the known errors and problems.

Manager of computational center, in turn, wants to get information about overall resources usage. At the same time, he wants the possibility to assess the resource usage of specific user groups and he does not need detailed reports but an overview of total resource usage during the different time frames. Also, he needs an opportunity to sort reports by each resource and each scientific group. In case of operation with many different supercomputers such separation of the statistics improves the capacity planning and work with computation services level.

As one can see, the requirements of mentioned groups have both intersections and differences. Universal system corresponding to the needs of all groups is difficult to implement and the problem is intended to be solved with a system based on a set of services attached to the service desk software.

Web services as an interconnect

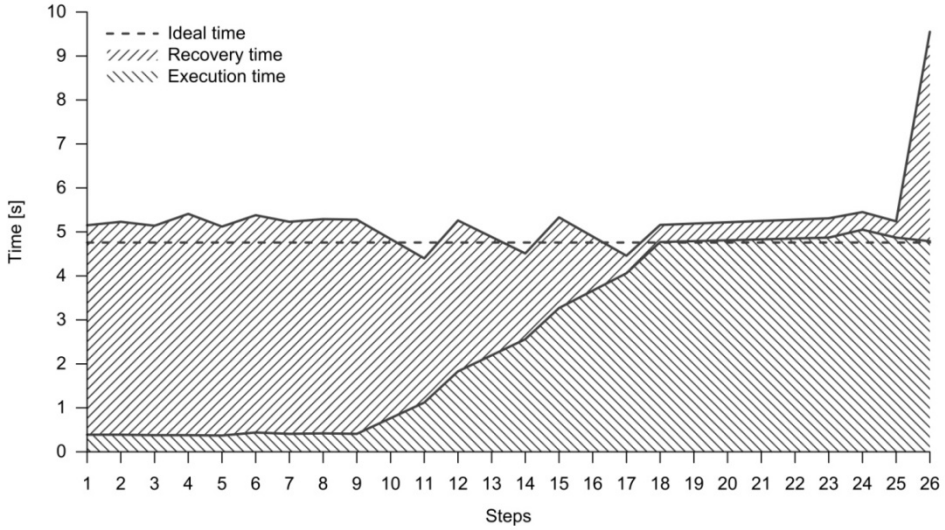
In case of complex information system the problem of connection between modules becomes crucial. Goals of an interconnect are not only correct functioning of data transfer but the ensuring the reliability and timeliness of command execution. Information systems in scientific HPC environment are sensitive to the speed of inside operation. In our case the collective usage center is considered. In such center efficiency of resource usage has a first priority. For the interconnect of information system of such center the best solution will be the web service set. This technology allows to connect the modules flexibly using de facto common network protocols. Also, the advantage of this approach is the desire of developers to provide an API of its products as web services, which facilitates the development of the complex as a whole. It is necessary to clarify that in this case we use web services architecture RESTful, it means strict adherence to the protocol HTTP / 1.1 and the idea of resource availability in the network [Fielding, 2000]. Special attention was paid to the speed of response to the occurrence of the problem while creating complex for task tracking. For this purpose we have developed an automated system for web service call processing. The basic idea is to reduce the recovery time of the settlement process through a mechanism of transactions.

Principle of the transactions of RESTful web services is similar to that of the work of the transaction in ACID model, but with important exceptions. Thus, the atomic system can be achieved if we have control over the operation of a specific web service, i.e. it is important to have the possibility of handling the information/command received. Common basis of all web services relating to a tracking complex achieves consistency properties, but not at any point in time. But as for the isolation property, everything is much more complicated.

This property can not be achieved in practice in case of asynchronous web requests. So, taking into attention these exclusions, we have a reliable system that can work with common RESTful web services.

In this case, each of the states of such an intermediate system is written synchronously into the journal, it allows recreate the state at the time of the emergency stop precisely. In this case, the system start the execution with the last stable point.

In case of failure of the transaction system, time that spent on the query execution is just slightly different from the time spent on the request without the recovery process, as illustrated in the Fig. 1. In the case presented the operation is restored, it makes 26 notes in journal in case of the successful implementation of 26 records. The time is compared with that obtained without the recovery process. The Fig. 1 shows that the recovery time is not enough. With regards to the peak on the 26th step, it is due to the imperfection of the algorithm of analysis at the time of testing — restarts laden branches.



Implementation

In its usual setting the service desk software provides the main point of incident and service request registration, ticket workflow for them and archiving after the solution was found and the ticket was closed [Kácha, 2010]. Also the service desk software provides the role-based permissions on the ticket workflow steps to the helpdesk staff, accounts the incident or service request solution time, provides reminders, notifications, ticket escalation mechanism, etc. Service desk system also usually has the knowledge base which contains most frequent incidents and its quick solutions.

The basis of our research is a model of supercomputer center which provides scientists with computational resources. Scientific work requires detailed reports to the grantors, that is why such supercomputer center has to provide detailed reports and statistics to its users. In such center the service desk system can be the central point of the interaction between supercomputer center staff and users. All correspondence about the incidents and service requests can be processed by service desk software.

On the assumption of such position of the service desk it is reasonable to make the website based on the open source CMS as a frontend and the service desk as a backend of the user support system. In our virtual testbed OTRS service desk and the website based on the Drupal CMS were integrated using the web services and the corresponding interfaces of both systems [Bakker et al., 2013]. To track the computational tasks on the supercomputers the idea of receiving the reports of PBS as tickets by the service desk was used.

It seems to be simpler to solve the problem without service desk and use direct transfer of the PBS reports to the CMS database instead with email transfer. In this way the processing can be done by a custom website module. However, the usage of service desk software in this case has some advantages. On the one hand the database schema development is already done. On the other hand the service desk software has a convenient object of ticket with set of fields and automatic state machine. This object allows to open a ticket by the keyword in the email header or field in data set received by web service automatically and to close it the same way. And finally the service desk allows the supercomputer center staff to watch the overall picture of the batch system and to operate with the PBS error as with an incident ticket.

Also usage of the service desk system as the core of the information system of supercomputer center corresponds the good practices of the ITIL [Potgieter, Botha, Lew, 2005]. The integration of service desk with PBS and the website realizes a part of the Capacity Management Process.

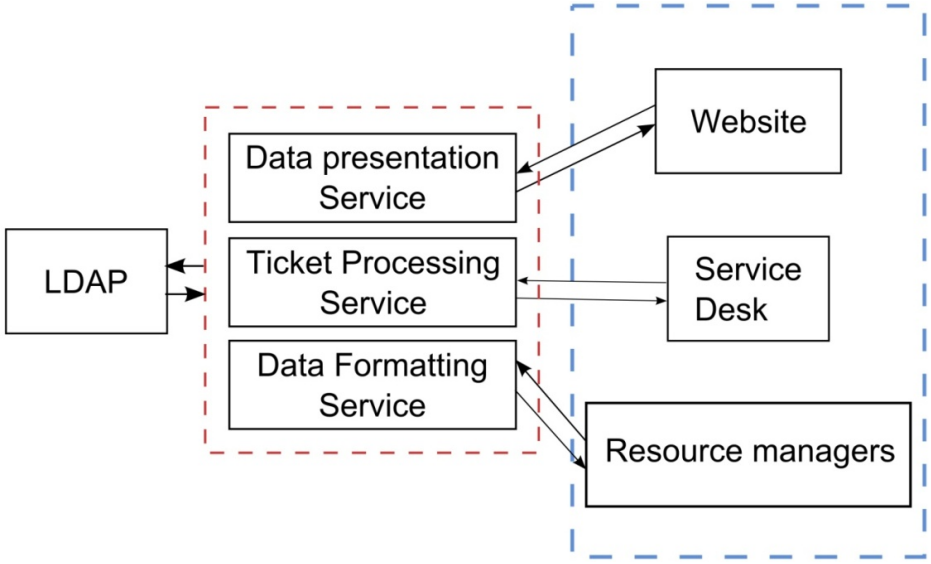


Fig. 1. Task tracker scheme

The first goal of our tracking complex is collecting an information about computational jobs, so the OTRS service desk and the PBS managers were connected via web services, placed on the duplicated transaction server. Every message from PBS becomes an article in a ticket: opening or closing one depending on job status in the message.

The next problem is the report and statistics generation. Relevant information for the user depends on this user role: executor, research work principal, system administrator, computer center manager. Such dependency can be realized by the LDAP technology. LDAP account can contain role information and information about user's priority for the batch system. The OTRS, PBS and either the website uses the common LDAP server to associate the data relative with one user account. For the statistic report creation OTRS Stats module can be used. This tool can be controlled via SOAP, so user can choose reported parameters on the website and generate the report as a *.csv or *.pdf at any time interacting with simple GUI.

The question of ordering and recording the activities of various groups of users on the physical computing resources is not a simple question, especially when organizing access to the resource through a heterogeneous system of services. Stats module can be used to gather statistics, but it is not flexible tool and does not allow changing the configuration of the output data "on the fly". For fast and fine-tuning data output, a web service connected to OTRS via the corresponding API can be used.

Personal account is implemented as a module of Drupal CMS interacting with databases (CouchDB, ActiveDirectory) through REST web services and direct requests using Sag library. The summary report of work is made using datacenter resources contains resource utilization figures, however, datacenter systems that log activity of applications running on cluster are not linked directly to this account. To display on-demand computing resource usage statistics, a service that would collect all metrics and represent this data in the user account in accessible graphical form is required.

Information about tasks executed on a supercomputer is stored as chains of tickets: one for the beginning of the computational task and one for the end. This information is attached to personal account through PHP extensions for CMS and different criteria can be chosen to be displayed. PHP extension then submits request to the web service that contains the data and then pre-generated chart is received as a static picture.

Based on received data web service creates a chart and returns it as a bitmap. The main difficulty is the fact that the request to the database has to be completed in a short period of time. That is why

the distributed database is a requirement. For chart creation free Java library JFreeChart is used. Such approach allows meeting the needs of executors and research work principal for the visual representation of data in the form of graphs and charts.

Graphical representation gives users more options for an estimation of resource consumption, which further helps to avoid excessive (or lack of) resource allocation, increases efficiency of computational services and prevents from conflicts between customers and computer center staff.

Conclusions

The solution of the task of integrating multiple information systems with user-friendly environment for computational task tracking allows an executor of scientific project to make necessary quantitative evaluations of his work on the supercomputer. For example such system can be useful to generate the reports about numerical simulations performed within the timeframe of a research work or to track the computational resource consumption and make prognosis of future resource utilization.

Our system will be especially useful for research work principals. Using the system they can evaluate the overall progress of numerical simulations within the research work, the productivity of every subordinate, the measure and the uniformity of quotas consumption. Also the automated report generator eliminates the need for manual creation of report after the end of research work. Metrics of total resource consumption and its distribution into the timeframe are included into the report automatically.

Supercomputer center staff also will have their profit. Capacity Management can easily receive information about the total computational resource consumption and make the predictions and a capacity plan for services. System keeps the data in the service desk database and allows archiving and making backups to protect the information. Also thanks to the database the reports can cover any time period. Such an opportunity is very useful for the computing center managers.

Another advantage of solution based on the service desk system is a possibility to increase the efficiency of solving the program errors on the supercomputers. The time of error recognizing decreases thanks to automatic incident registration. That is why overall time costs also decreases and the helpdesk staff performance indicators increases.

References

Bakker R. et al. OTRS 3.3 — Admin Manual. 2013. http://ftp.otrs.org/pub/otrs/doc/doc-admin/3.3/en/pdf/otrs_admin_book.pdf (retrieved on 2014-04-07).

Fielding R. T. Architectural styles and the design of network-based software architectures: Ph. D. thesis / University of California. 2000. P. 162.

Gayduchok V. Yu., Bogdanov A. V., Degtyarev A. B., Gankevich I. G., Zolotarev V. I. Virtual Workspace as a Basis of Supercomputer Center // Proceedings of the 5th Intern. Conf. “Distributed Computing and Grid-Technologies in Science and Education” (Dubna, 16–21 July, 2012) / Joint Institute for Nuclear Research (Dubna). — 2012. — P. 60–66.

Kácha P. OTRS: CSIRT WorkFlow Improvements. — CESNET, Tech. Rep., 10 2010.

Potgieter B. C., Botha J. H., Lew C. Evidence that use of the ITIL framework is effective // 18th Annual conference of the national advisory committee on computing qualifications, Tauranga, NZ. 2005. — P. 160–167.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 621−630

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Ресурсный центр обработки данных уровня Tier-1 в национальном исследовательском центре «Курчатовский институт» для экспериментов ALICE, ATLAS и LHCb на Большом адронном коллайдере (БАК)

А. Я. Бережная[a], В. Е. Велихов, Ю. А. Лазин[b], И. Н. Лялин, Е. А. Рябинкин, И. А. Ткаченко

Национальный исследовательский центр «Курчатовский институт»,
Россия, 123182, г. Москва, пл. Академика Курчатова, д. 1

E-mail: [a]gridops@grid.kiae.ru, [b]Yury.Lazin@grid.kiae.ru

Получено 10 декабря 2014 г.

Представлен обзор распределенной вычислительной инфраструктуры ресурсных центров коллаборации WLCG для экспериментов БАК. Особое внимание уделено описанию решаемых задач и основным сервисам нового ресурсного центра уровня Tier-1, созданного в Национальном исследовательском центре «Курчатовский институт» для обслуживания ALICE, ATLAS и LHCb экспериментов (г. Москва).

Ключевые слова: высокопроизводительные вычислительные системы, системы распределенного массового хранения данных, системы распределенной обработки данных, грид

The Tier-1 resource center at the National Research Centre "Kurchatov Institute" for the experiments, ALICE, ATLAS and LHCb at the Large Hadron Collider (LHC)

A. Ya. Berezhnaya, V. E. Velikhov, Ya. A. Lazin, I. N. Lyalin, E. A. Ryabinkin, I. A. Tkachenko

National Research Centre "Kurchatov Institute", 1 Kurchatov Sq., Moscow, 123182, Russia

Abstract. — The review of the distributed computing infrastructure of the Tier-1 sites for the Alice, ATLAS, LHCb experiments at the LHC is given. The special emphasis is placed on the main tasks and services of the Tier-1 site, which operates in the Kurchatov Institute in Moscow.

Keywords: high-performance computing systems, mass storage distributed system, distributed data processing, grid

Вычисления выполнялись на компьютерных ресурсах ЦКП «Комплекс моделирования и обработки данных исследовательских установок мегакласса», поддерживаемого соглашением с Минобрнауки России о предоставлении субсидии № 14.621.21.0006.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 621–630 (Russian).

Введение

Начиная с 2004 года вычислительные ресурсы НИЦ «Курчатовский институт» интегрированы в глобальную грид-систему — WLCG (Worldwide LHC Computing Grid, или Всемирный грид для Большого адронного коллайдера)[1] [LHC..., 2015] в Европейской организации ядерных исследований (ЦЕРН)[2] [CERN..., 2015].

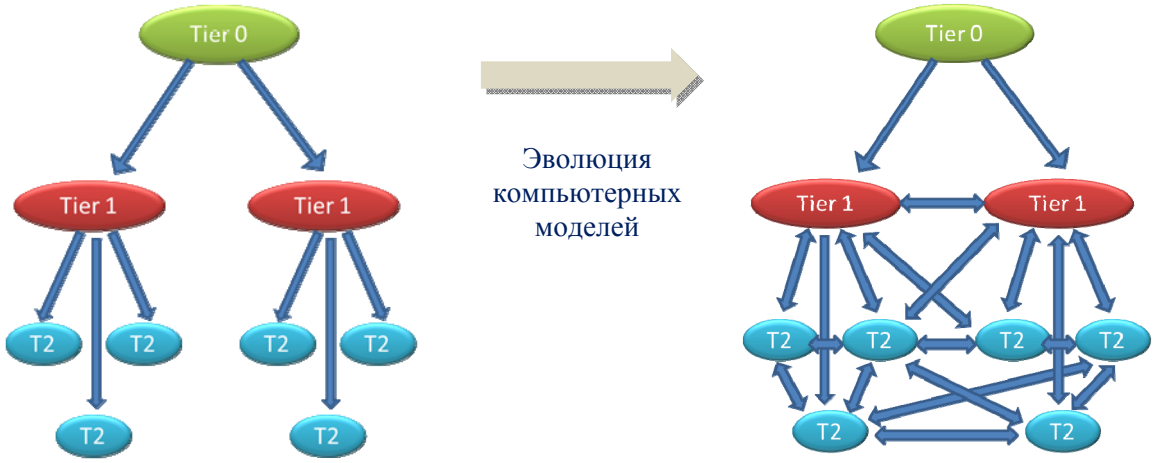
Ресурсный центр уровня Tier-1 НИЦ «Курчатовский институт» представляет собой высокоорганизованный вычислительный комплекс, включающий систему кондиционирования и систему резервированного питания с дизельной генераторной установкой (ДГУ).

Tier-1 НИЦ «Курчатовский институт» вносит вклад в распределенную обработку Больших Данных, получаемых в БАК-экспериментах ATLAS, ALICE и LHCb, и обеспечивает проведение полного цикла обработки экспериментальных и смоделированных событий, включающего в себя этапы приема/передачи исходных «сырых» данных, их последующую обработку, анализ и защищенное долговременное хранение. Доступ к ресурсам центра предоставляется всем участниками международной коллаборации WLCG, которая является самой большой академической распределенной вычислительной средой в мире, что обеспечивает полноправное участие российских исследователей в исследовательской и публикационной активности экспериментов БАК [The Large Hadron Collider, 2015].

Данная работа посвящена описанию инфраструктуры грид-компьютинга в международной коллаборации WLCG (разделы 1, 2). Раздел 3 включает описание ресурсного центра Tier-1 в НИЦ «Курчатовский Институт» (г. Москва).

1. Распределенная вычислительная инфраструктура грид-центров для БАК

Со времени создания (1998 год) по настоящий момент структура компьютерных моделей БАК-экспериментов претерпела эволюционные изменения (рис. 1).



(а) (б)

Рис. 1. Эволюция структуры компьютерных моделей БАК-экспериментов от строго иерархической до Mesh-топологии

Суть распределенной модели архитектуры компьютерной системы состоит в том, что первичная информации с детекторов БАК после обработки в реальном времени и первичной ее

[1] http://wlcg.web.cern.ch/
[2] http://www.cern.ch/

реконструкции в Tier-0 направляется для дальнейшей обработки, анализа и резервного хранения в региональные центры Tier-1, которые, в свою очередь, подключают к процессу распределенной обработки ресурсные центры уровня Tier-2.

Иерархия и соответствующие задачи центров каждого уровня определены в Меморандуме WLCG (Memorandum of Understanding for Collaboration in the Deployment and Exploitation of the Worldwide LHC Computing Grid) [Worldwide LHC..., 2014].

2. Центры уровня Tier-1

В настоящее время внутри WLCG-коллаборации функционируют следующие ресурсные центры Tier-1 (табл. 1).

Таблица 1

Ресурсный центр	***Обслуживаемые эксперименты***			
	ALICE	***ATLAS***	***CMS***	***LHCb***
Канада, TRIUMF		X		
Франция, CC-IN2P3	X	X	X	X
Германия, KIT	X	X	X	X
Италия, CNAF	X	X	X	X
Голландия LHC/Tier1	X	X		X
Скандинавские страны (NDGF)	X	X		
Республика Корея, GSDCatKISTI	X			
Россия, НИЦ «Курчатовский Институт»	X	X		X
Россия, ОИЯИ, г.Дубна			X	
Испания, PIC		X	X	X
Тайпей, ASGC		X	X	
Великобритания, RAL	X	X	X	X
США, BNL		X		
США, FNAL			X	

Для грид-пользователей ресурсные центры Tier-1 обеспечивают следующие сервисы, перечень которых официально определен в Меморандуме о взаимопонимании и согласован между коллаборацией WLCG и НИЦ «Курчатовский институт» [Memorandum of Understanding, 2014]:

1) предоставление управляемого дискового пространства, обеспечивающего постоянное и/или временное хранение данных для файлов и баз данных;
2) обеспечение доступа к хранимым данным со стороны других центров WLCG;
3) обеспечение работ конечных пользователей по анализу объектов данных;
4) предоставление других сервисов, например моделирования столкновений в соответствии с согласованными требованиями экспериментов;
5) обеспечение необходимой пропускной способности для обмена данными с центрами Tier-1 по согласованному между экспериментами и заинтересованными центрами Tier-1 плану.

Каждый из трех обслуживаемых в ресурсном центре Tier-1 НИЦ «КИ» экспериментов БАК имеет свою модель обработки экспериментальных данных, что, в свою очередь, определяет потребность в ресурсах центра (рис. 2, 3, 4) [LHC..., 2015]:

Обработка исходных данных состоит из автономной калибровки и обновления условий получения данных, а затем реконструкции и создания ESDs, AODs и обеспечения качества (QA — Quality Assurance) объектов. На рис. 2 показаны форматы данных и шаги по снижению их объема в процессе обработки. Типы данных и акронимы определены ниже.

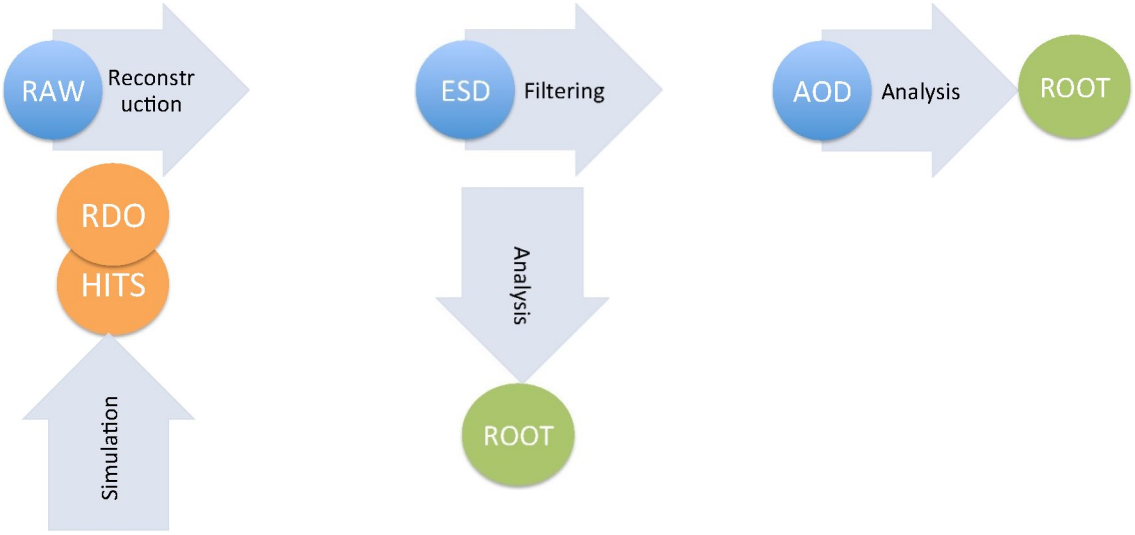


Рис. 2. Форматы данных и стадии их обработки в ALICE

Обработка смоделированных методом Монте-Карло событий и RAW-данных возвращает два анализа-ориентированных объекта: ESDs и AODs. ESDs содержит всю информацию, необходимую для любого анализа, последующие калибровки и проверки QA, а AODs содержит данные, подходящие для большинства задач анализа. Размер ESDs составляет 15–30 % от соответствующих исходных данных, в то время как AODs — около 30 % от соответствующего размера ESD. В основном для анализа физики предпочитают AODs-объекты. Некоторые очень специфические типы анализов выполняются на ESDs-данных. После обновления калибровки и программного обеспечения обновленные версии AODs получают через процедуру извлечения из ESDs с учетом новых условий.

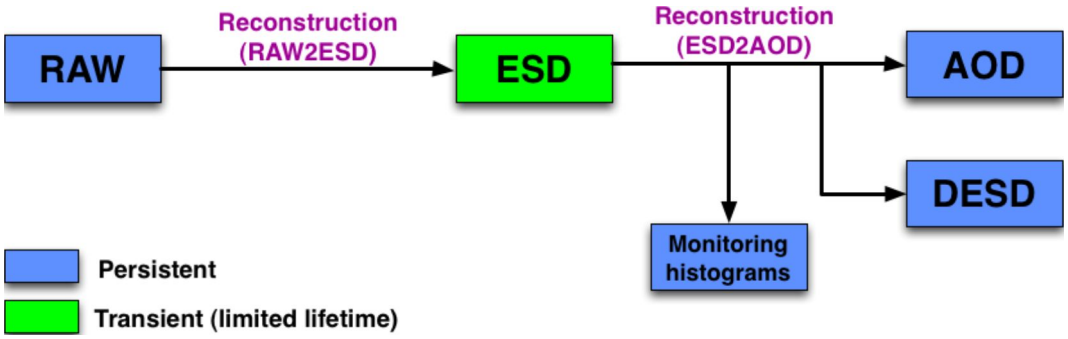


Рис. 3. Процесс реконструкции данных ATLAS

В ATLAS (рис. 3) первоначальная реконструкция данных выполняется на Tier-0. Исходные данные обрабатываются в два этапа в течение одного рабочего задания, вначале получается ESDs, а затем, на втором этапе, — AODs и DESDs. Данные RAW и выходы реконструкции экспортируется в ATLAS-грид-хранилище в соответствии с политикой репликации, то есть в Run-1 Tier-0 использовался для оперативной реконструкции данных вместе с быстрой калибровкой и оперативным определением качества данных и использовался как основное ленточное хранилище для RAW-данных. Ожидается, что в Run-2 роль Tier-0 останется без изменений, однако предполагается, что в Run-2 часть оперативной реконструкции данных может быть передана центрам Tier-1 в случает больших нагрузок на Tier-0.

Изменения в программном обеспечении ATLAS, а также изменения в распределенной вычислительной среде позволили центрам Tier-2 выполнять некоторые процессы, которые до это-

го были закреплены только для Tier-1 (репроцессинг, групповой анализ, реконструкция Монте-Карло) уже к концу Run-1. В Run-2 планируется совершенствовать это преимущество для оптимизации пропускной способности и нагрузки на сайты.

Центры Tier-1 по-прежнему будут иметь особую роль в качестве основного хранилища данных, а также предоставления второй копии RAW-данных с лент.

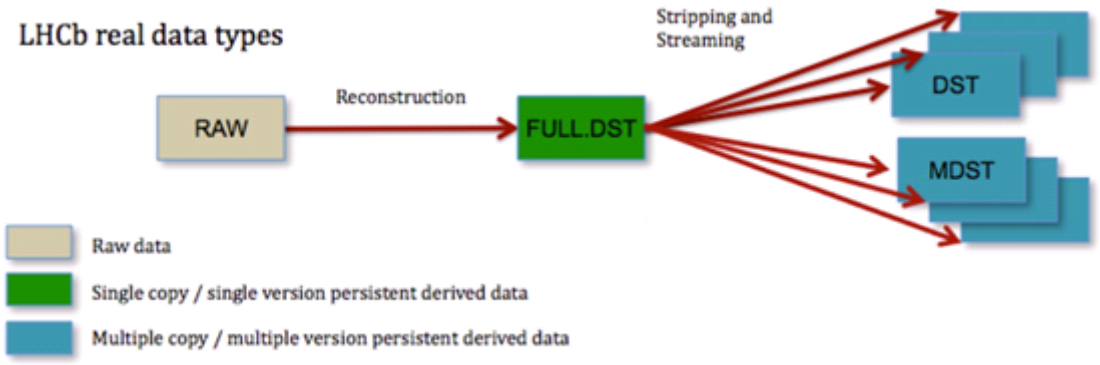


Рис. 4. Модель обработки данных LHCb

В LHCb (рис. 4) быстрая реконструкция и репроцессинг проходят идентично. На первом этапе необработанные данные восстанавливаются для получения FULL.DST, которые могут быть использованы в качестве входных данных для всех дальнейших действий по обработке данных. Второй этап состоит из выполнения приложения («зачистки»), которое выбирает события для физического анализа или калибровки; выполняется разборка нескольких сотен независимых потоков, каждый из которых связан с одним из десятков выходных потоков; в зависимости от анализа может использоваться, поток формата DST или MicroDST (MDST). Каждому MDST планируется также иметь одну копию DST (так называемую MDST.DST, не показанную на рис. 4), содержащую все события, соответствующие MDST-потокам, и от которой MDST может легко и быстро быть получен повторно. Это временная мера призванная стимулировать переход к использованию MDSTs путем предоставления «страховки», на случай если и когда будет найдена дополнительная информация, которая потребуется во время анализа. Это добавляет где-то 5–10 % к общему объему ленточного хранилища, необходимого для реальных данных, которые будут восстановлены, когда MDST-миграция завершится.

Поскольку шаг зачистки не зависит от шага восстановления, повторная зачистка может быть выполнена начиная с FULL.DST. Оба процесса — и реконструкция, и зачистка — организованы и планируются централизованно. Физические группы имеют доступ только к разбору выходных наборов данных. Ниже перечислены различные форматы данных:

RAW — необработанные («сырые») данные;

ESD — данные полной реконструкции событий (EventSummaryData);

AOD — данные, содержащие характеристики восстановленных физических объектов и используемые для физического анализа;

HITS — смоделированные данные чувствительности детектора;

RDO — смоделированные «сырые» данные;

DESD — сокращенные ESD для определенных целей;

FULLDST — полный выход реконструкции для всех физических событий после зачистки от шума;

DST — выход после зачистки: события, выбранные по критериям физики, полная копия восстановленного события плюс дерево(ья) распада частиц, которые вызвали отбор события;

MDST — данные, как DST, но содержащие только подмножество события (треки, PID, которое вызвало отбор события, и минимальные исходные данные).

3. Ресурсоемкость и функциональность центра уровня Tier-1 в НИЦ «Курчатовский институт»

В соответствии с требованиями, определенными соглашением по WLCG (MoU) [Memorandum of Understanding, 2014], ресурсный центр уровня Tier-1 в НИЦ «Курчатовский институт» в настоящий момент предоставляет для коллаборации WLCG ресурсы представленные в таблице 2.

Таблица 2

НИЦ КИ	2014
ЦПУ (HEP-SPEC06)	22700
Диски (ТБ)	2600
Ленты (ТБ)	2000

На данных ресурсах обеспечивается полная функциональность ресурсного центра уровня Tier-1, а именно:

– возможность управления большими объемами данных на высоких скоростях передачи данных;

– обеспечение необходимых характеристик для всех центральных процессоров и хранилищ;

– организация доступа к данным тысяч пользователей;

– обеспечение надежного долгосрочного хранения архивных данных.

Кроме того, вычислительная среда управляет обработкой реальных и моделируемых данных, а также предоставляет данные для пользовательских отчетов. При обработке данных для каждого эксперимента запускаются стандартные программы на все статистические наборы данных, в то время как для пользовательского анализа необходимый набор данных определяется только потребностями различных физических и аналитических команд и особенностями отчета, который будет выполняться.

Схема работы Tier-1 НИЦ «КИ» с указанием функций показана на рис. 5.

Tier-1 НИЦ «КИ» выполняет в автоматическом режиме следующие функции:

1) запись необработанных данных, получаемых из Tier-0 (ЦЕРН), и хранение их на ленточных накопителях;

2) запись обработанных данных, получаемых из Tier-0 (ЦЕРН), и хранение их на дисковых массивах;

3) предоставление хранимых данных другим центрам — Tier-1 и Tier-2;

4) обработку «сырых» (первичных) данных, получаемых на трех экспериментах — ATLAS, Alice, LHCb;

5) расчеты событий физического моделирования;

6) аутентифицированные и авторизованные запросы на загрузку и выгрузку экспериментальных данных;

7) аутентифицированные и авторизованные запросы на запуск вычислительных задач для обработки данных экспериментов БАК;

8) мониторинг загрузки всех структурных узлов системы и оперативное информирование обслуживающего персонала о приближении параметров мониторинга к критическим значениям.

Качество реализации функций обеспечивает полное выполнение входящих в их состав операций и задач и гарантирует корректную с точки зрения предметной области обработку данных и представление результатов.

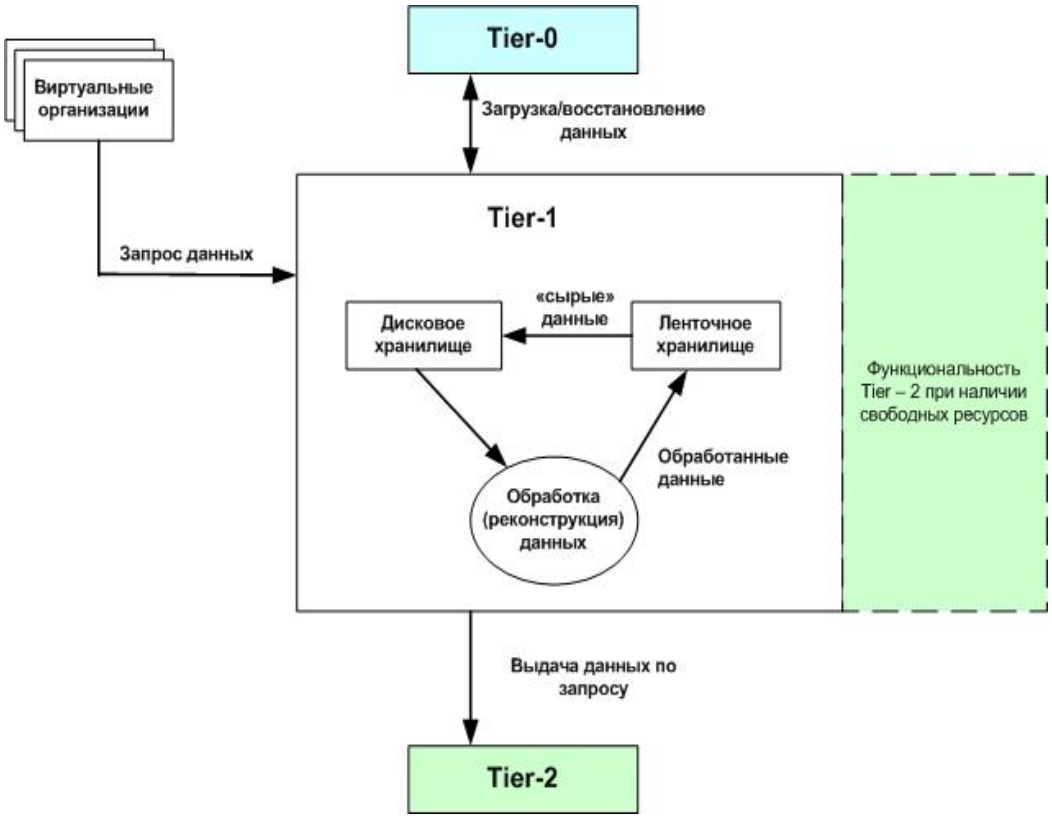


Рис. 5. Общая схема работы Tier-1 НИЦ «КИ»

Для надежного и эффективного управления инфраструктурой ресурсного центра осуществляется постоянный мониторинг его состояния средствами как внешнего, так и внутреннего мониторинга, как то:

– постоянное наблюдение за состоянием сервисов в грид-среде, как общих для всей инфраструктуры, так и сервисов в каждом ресурсном центре;

– сбор информации о количестве ресурсов (число процессоров, дисковое пространство) и их состоянии (свободные/занятые ресурсы);

– мониторинг выполнения заданий, передачи данных, запуск задач;

– отслеживание состояния каналов связи.

Центр работает в следующем режиме: в течение 24 часов, 7 дней в неделю, 365 дней в году.

Структурно комплекс технических средств ресурсного центра Tier-1 является слабосвязанным кластером на базе Intel-совместимых процессоров и сетей передачи данных, построенных с применением технологии Ethernet. Структура комплекса такова:

- **внешняя сеть кластера**, обслуживающая маршрутизируемые IP-адреса:

– устройства уровня ядра сети, обеспечивающие каналы передачи данных во внешние системы и являющиеся резервированными на уровне отказа единичных коммутаторов;

– устройства уровня доступа, подключающиеся к нескольким устройствам уровня ядра сети и обеспечивающие порты для подключения конечных устройств;

- **внутренняя сеть кластера**, обслуживающая немаршрутизируемые IP-адреса:

– устройства уровня ядра сети, обеспечивающие коммутацию пакетов между своими портами в неблокируемом режиме и являющиеся отказоустойчивыми на уровне выхода из строя отдельных компонентов коммутатора: блока питания, модуля портов, образа операционной системы;

– устройства уровня доступа, подключающиеся в ядро сети и обеспечивающие порты для доступа конечных устройств;

- **вычислительное поле**, состоящее из серверов, которые обслуживают вычислительные задачи, приходящие на Tier-1 через сервис CREAM CE;
- **сервисы хранения данных на дисках**, состоящие из серверов и дисковых массивов, подключенных к серверам по протоколам SAS и iSCSI;
- **сервисы хранения данных на лентах**, состоящие из:

– ленточных роботов и библиотеки, хранящей ленточные накопители и устройства считывания;

– серверов, обслуживающих устройства считывания, подсоединенные к машинам посредством интерфейсов FibreChannel;

– серверов и дисковых массивов, обеспечивающих дисковый буфер для ленточного хранилища;

- **сервисы gLite**/UMD-3: CREAM CE, site-BDII, APEL, Logging & Bookkeeping, top-BDII, VOBOX;
- **инфраструктурные сервисы**:

– сервисы фильтрации трафика и преобразования сетевых адресов,

– сервисы доменной системы имен,

– сервисы кеширования HTTP-запросов,

– сервисы локального мониторинга,

– сервисы терминального доступа для системных администраторов и обслуживающего персонала.

Схема структуры центра с указанием функциональных модулей показана на рис. 6:

Основные программные компоненты ресурсного центра Tier-1 В НИЦ «КИ»

С точки зрения программного обеспечения (ПО) структура ресурсного центра Tier-1 в НИЦ «КИ» состоит из следующих подсистем и сервисов.

- **Подсистема управления** компьютерными ресурсами на базе свободного ПО «Puppet [9, 10].
- **Подсистема передачи данных:**
  - сервисы передачи и хранения данных (dCache, EOS, Enstore).
- **Подсистема управления загрузкой:**
  - сервис управления счетными заданиями и пулом вычислительных узлов (CREAM CE);
  - сервис пакетного планирования запуска и выполнения заданий (Torque + MAUI);
  - информационные сервисы по ресурсам сайта (site-BDII, top-BDII);
  - сервис журналов (Logging и Bookeeping), хранящий информацию о задании.
- **Подсистема информационного обслуживания и мониторинга грид:**
  - сервисы сбора, хранения и предоставления информации этой подсистемы (Nagios, MonALISA, Panda, DIRAC).
- **Подсистема безопасности и контроля прав доступа:**
  - сервис выдачи и поддержки сертификатов.
- **Подсистема учета:**
  - сервис регистрации и учета вычислительных ресурсов;
  - сервис регистрации и учета ресурсов хранения данных.
- **Подсистема прикладного программного обеспечения экспериментов.**

Оперативное обслуживание и поддержка центров уровня Tier-2

Центр Tier-1 в НИЦ «КИ» функционирует в соответствии с вычислительными моделями БАК-экспериментов и согласно требованиям WLCG [LHC..., 2015]. С середины 2013 года центр участвует в процессе обработки полученных в Run-1 данных на поддерживаемых БАК-

экспериментах и осуществляет прием и хранение согласованных объемов экспериментальных данных и данных моделирования обеспечивая доступа к ним из других центров уровня Tier-1/ Tier-2 инфраструктуры WLCG.

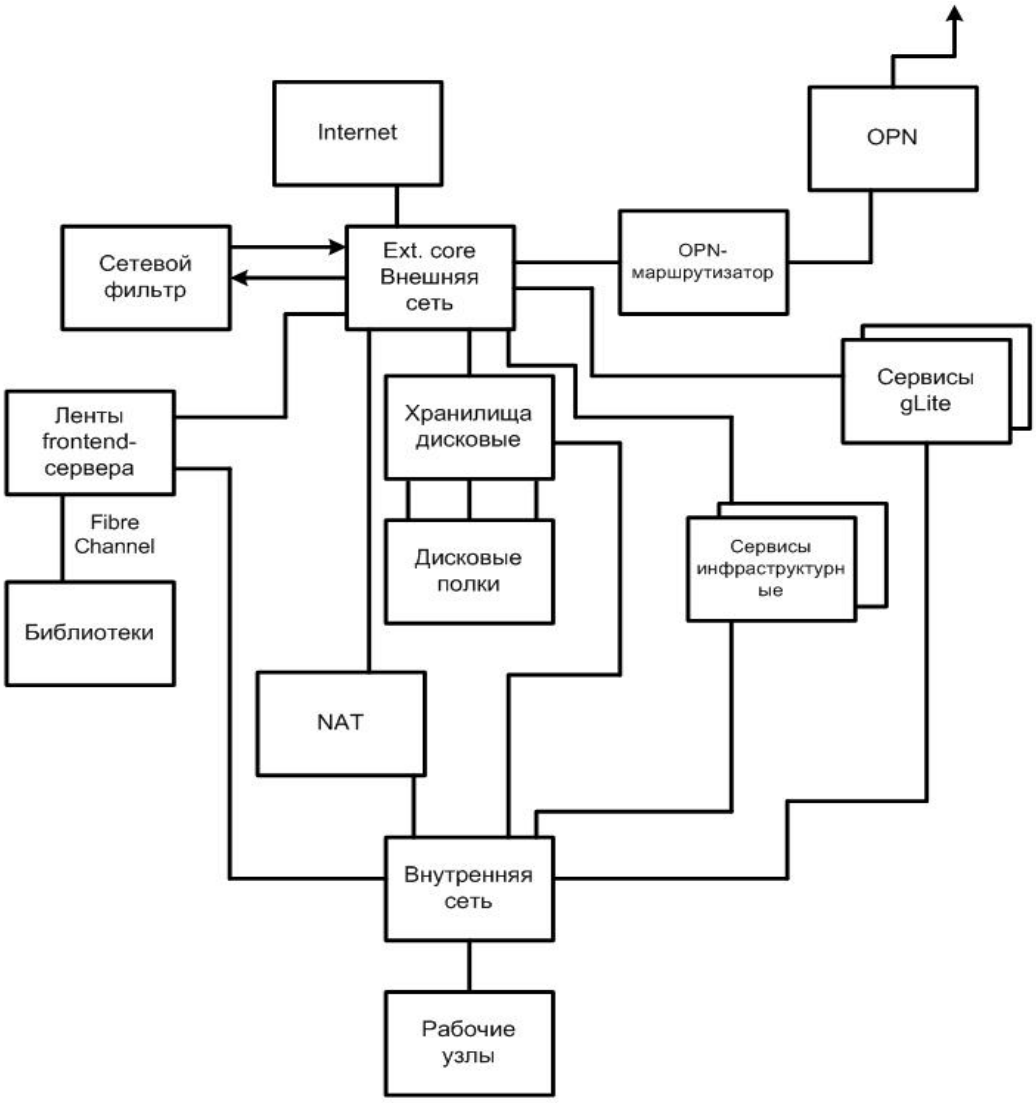


Рис. 6. Схема структуры центра Tier-1 НИЦ «КИ»

Во время Run-2 (с апреля 2015 года) на БАК Tier-1 НИЦ «КИ» будет выполнять основные функции центра, определенные в MoU [Worldwide LHC..., 2014], а также осуществлять:

- собственное оперативное обслуживание;
- поддержку региональных центров уровня Tier-2;
- поддержку грид-пользователей, включая консультации и помощь по специфическим проблемам грид-сервисов и выполнению счетных заданий;
- поддержку в разрешении инцидентов, связанных с безопасностью.

Список литературы

Климентов А., Кореньков В. Распределенные вычислительные системы и их роль в открытии новой частицы // Суперкомпьютеры. — 2012. — № 3 (11). — С. 7–11.

Ткаченко И. А. Опыт использования «Puppet» для управления вычислительным грид-кластером Tier-1 в НИЦ «Курчатовский Институт».

Aderholz M. et al. Models of Networked Analysis at Regional Centers for LHC Experiments (MONARC) — Phase 2 Report // CERN/LCB, 2000–2001 (2000).

ATLAS Collaboration: Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC // Phys. Lett. B. — 2012. — Vol. 716. — P. 1–29.

CERN [электронный ресурс] // CERN, Switzerland. — 2015. — URL: http://public.web.cern.ch/public (дата обращения: 06.02.2015).

Dobre M., Stratan C. Monarc simulation framework // Proceedings of the RoEduNet International Conference, Buletinul Stiintific al Universitatii “Politehnica” din Timisoara, Romania, Seria Automatica si Calculatoare Periodica Politechnica, Transactions on Automatic Control and Computer Science. — 2004. — Vol. 49 (63). — P. 35–42. — ISSN 1224-600X.

LHC Computing Grid Technical Design Report. CERN-LHCC-2014-014 [электронный ресурс] // CERN, Switzerland. — 2015. — URL: http://lcg.web.cern.ch/LCG/public/default.htm (дата обращения: 17.01.2015).

Memorandum of Understanding [электронный ресурс] // CERN, Switzerland. — 2014. — URL: https://espace2013.cern.ch/WLCG-document-repository/MoU/countries/Russia/MoU-CERN-NRCKI_17JUL2014.pdf?Web=1 (дата обращения: 26.12.2014).

Puppet Labs [электронный ресурс] // London, United Kingdom. — 2014. — URL: http://puppetlabs.com/ (дата обращения: 26.12.2014).

The Large Hadron Collider [электронный ресурс] // CERN, Switzerland. — 2015. — URL: http://public.web.cern.ch/public/en/lhc/lhc-en.html (дата обращения: 06.02.2015).

Worldwide LHC Computing Grid Memorandum of Understanding [электронный ресурс] // CERN, Switzerland. — 2014. — URL: http://wlcg.web.cern.ch/collaboration/mou (дата обращения: 26.12.2014).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 631–639

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.02, 004.94

Метод представления дифракционных изображений XFEL для классификации, индексации и поиска

С. А. Бобков[1,a], А. Б. Теслюк[1,b], О. Ю. Горобцов[1,2], О. М. Ефанов[3], Р. П. Курта[2], В. А. Ильин[1,4], М. В. Голосова[1], И. А. Вартаньянц[2,5]

[1] Национальный исследовательский центр «Курчатовский институт»,
Россия, 123182, г. Москва, пл. Академика Курчатова, д. 1
[2]Немецкий электронный синхотрон ДЕЗИ,
Германия, D-22607, г. Гамбург, Ноткесштрассе, д. 85
[3]Научный центр лазеров на свободных электронах,
Германия, D-22607, г. Гамбург, Ноткесштрассе, д. 85
[4] Московский государственный университет им. М. В. Ломоносова,
Россия, 119991, г. Москва, ГСП-1, Ленинские горы, д. 1-52
[5]Национальный исследовательский ядерный университет «МИФИ»,
Россия, 115409, г. Москва, Каширское шоссе, д. 31

E-mail: [a]s.bobkov@grid.kiae.ru, [b]anthony.teslyuk@grid.kiae.ru

Получено 21 января 2015 г.

В работе представлены результаты применения алгоритмов машинного обучения: метода главных компонент и метода опорных векторов для классификации дифракционных изображений, полученных в экспериментах на лазерах на свободных электронах. Показана высокая эффективность применения такого подхода с использованием модельных данных дифракции лазерного пучка на капсиде аденовируса и вируса катаральной лихорадки, в которых учтены условия реального эксперимента на лазерах на свободных электронах, такие как шум и особенности используемых детекторов.

Ключевые слова: метод главных компонент, метод опорных векторов, когерентная визуализация

Вычисления выполнялись на компьютерных ресурсах ЦКП «Комплекс моделирования и обработки данных исследовательских установок мегакласса», поддерживаемого соглашением с Минобрнауки России о предоставлении субсидии № 14.621.21.0006.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 631–639 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 631−639

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

XFEL diffraction patterns representation method for classification, indexing and search

S. A. Bobkov[1], A. B. Teslyuk[1], O. Yu. Gorobtsov[1,2], O. M. Yefanov[3], R. P. Kurta[2], V. A. Ilyin[1,4], M. V. Golosova[1], I. A. Vartanyants [2,5]

[1] National Research Center "Kurchatov Institute", 1 Kurchatov Sq., Moscow 123182, Russia
[2]Deutsches Elektronen-Synchrotron DESY, 85 Notkestraße, D-22607 Hamburg, Germany
[3]Center for Free-Electron Laser Science, 85 Notkestraße, D-22607 Hamburg, Germany
[4] Lomonosov Moscow State University,GSP-1, 1-52 Leninskie Gory, Moscow, 119991, Russia
[5] National Research Nuclear University MEPhI, 31 Kashirskoe highway, 115409, Moscow, Russia

Abstract. — The paper presents the results of application of machine learning methods: principle component analysis and support vector machine for classification of diffraction images produced in experiments at free-electron lasers. High efficiency of this approach presented by application to simulated data of adenovirus capsid and bluetongue virus core. This dataset were simulated with taking into account the real conditions of the experiment on lasers free electrons such as noise and features of used detectors.

Keywords: principle component analysis, support vector machine, coherent diffraction imaging

I. Введение

При традиционных исследованиях малых биологических образцов с применением лазерного излучения возникают два ограничения. Во-первых, большинство белковых макромолекул не кристаллизуются, что препятствует использованию надежно отработанных методов рентгеновской кристаллографии. Во-вторых, для некристаллических образцов излучение вызывает повреждения, которые ограничивают возможное разрешение до нескольких десятков нанометров. Новый подход к визуализации с использованием одночастичной когерентной дифракции может позволить преодолеть второе ограничение и увеличить разрешение биологических объектов в субнанометровом диапазоне [Gaffney and Chapman, 2007; Seibert et al., 2011].

Идентичные образцы в случайной ориентации вводятся в луч лазера. Сами образцы разрушаются после выстрела за счет кулоновского взрыва [Neutze et al., 2000], но их дифракционные изображения регистрируются. Высокая мощность лазеров на свободных электронах (FELs) с фемтосекундными импульсами позволяет проводить эксперименты по определению структуры отдельных воспроизводимых частиц [Gaffney and Chapman, 2007; Mancuso, Yefanov, and Vartanyants, 2010]. Впоследствии по этим 2D-изображениям может быть восстановлена 3D-структура частицы. Данный метод уже был применен для исследования нанокристаллов [Chapman et al., 2011].

Тем не менее при таком подходе возникают дополнительные трудности. Одной из них является проблема ориентации частиц, т. к. положение, соответствующие каждому дифракционному изображению, изначально неизвестно. Правильные ориентации могут быть определены из множества измерений [Loh and Elser, 2009; Yefanov and Vartanyants, 2013; Fung et al., 2009] с учетом того факта, что все дифракционные изображения представляют собой сечения сферы Эвальда одинаковым трехмерным распределением интенсивности в обратном пространстве. Однако необходимо иметь достаточную выборку для того, чтобы определить ориентации изображений друг относительно друга, по крайней мере необходимо несколько сотен измерений. Это число может быть увеличено до нескольких тысяч, если учесть тот факт, что сигнал от одной макромолекулы является относительно слабым.

В процессе измерений возникает еще одна сложность. Не все получаемые изображения содержат дифракцию от образцов: большинство изображений пустые, а некоторые из них могут содержать, например, дифракцию от капель воды, несколько частиц некоторой примеси. Необходимо классифицировать изображения перед процедурой восстановления и использовать только те, которые соответствуют исследуемому образцу.

Можно выполнить классификацию вручную, но это занимает много времени. Недавно были предложены вычислительные методы для сортировки на основе метода главных компонент [Yoon et al., 2011].

В обработке изображений широко распространены различные алгоритмы определения характерных признаков. Они направлены на сокращение объемов данных, необходимых для точного описания исследуемых данных. Наиболее известной областью применения таких алгоритмов является машинное обучение: распознавание лиц [Yang et al., 2004], компьютерное зрение [Viola, Jones, 2001], лингвистика [Sebastiani, 2002] или интеллектуальный анализ данных [Berkhin, 2006].

Кроме того, существуют общие алгоритмы кластеризации, такие как метод главных компонент. Попытки применить эти алгоритмы непосредственно к дифракционным изображениям не принесли желаемых результатов. Для улучшения анализа дифракционных изображений требуется метод, который бы учитывал следующие составляющие дифракционной физики: процесс распространения лазерного импульса, характеристики луча, пространственные особенности молекул и т. д.

В этой статье представлен метод сортировки, основанный на сочетании метода главных компонент [Jolliffe, 2002] и корреляции угловой интенсивности дифракционных изображений [Altarelli, Kurta, and Vartanyants, 2010]. Для описания дифракционных изображений использовались характеристические векторы, которые вычисляются на основе угловой корреляции интенсивности. Связь между локальной структурой частиц и угловой корреляцией интенсивности была теоретически обоснована в недавней публикации М. Альтарелли и др. [Altarelli, Kurta and Vartanyants, 2010].

Наборы векторов для дифракционных изображений исследовались с помощью метода главных компонент. Сочетание двух подходов позволило выявить внутренние связи между дифракционными изображениями и добиться кластеризации данных в соответствии с исходными молекулами. Метод был разработан и проверен на основе данных моделирования, в которых учены условия реального эксперимента на лазерах на свободных электронах, такие как шум и особенности используемых детекторов.

II. Описание исходных данных

Для разработки метода был использован набор дифракционных изображений, полученный для молекул трех типов: капсид аденовируса [Zubieta, Blanchoin and Cusack, 2006], ядро вируса катаральной лихорадки (bluetongue virus core, 2BTV) [Grimes et al., 1998] и капля воды диаметром 10 нм.

Данные частицы имеют сравнимый размер, что усложняет классификацию. Капсид аденовируса обладает гексагональной симметрией, а 2BTV не обладает симметрией. Было сгенерировано по 1000 изображений для каждого типа частицы.

Весь набор модельных данных из 3000 изображений был случайно перемешан, затем из него был выделен обучающий набор. Контрольные данные о классификации полного набора изображений использовались только для сравнения итоговых результатов. Так как использовались данные моделирования, нельзя говорить о погрешности определения исходного типа молекул в контрольной классификации.

Размер обучающего набора должен удовлетворять двум условиям: содержать достаточную информацию для классификации полного набора и иметь как можно меньший размер, так как в экспериментальных условиях он составляется вручную. Варьируя размер обучающей выборки, мы установили, что для классификации полного набора данных из 3000 изображений с помощью метода опорных векторов достаточным является размер в 40 изображений. При меньшем размере появляются ошибки классификации, больший обучающий набор избыточен.

При моделировании использовались следующие параметры эксперимента: детектор установлен на расстоянии 100 мм от образца и имеет размеры 100×100 мм^2; разрешение 224×224 пикселя. Длина волны излучения равнялась 0.3 нм. Угловой размер спекла составлял 0.06 радиана.

Падающий луч имел гауссово распределение, ширина на полувысоте равнялась 150 нм, плотность потока — 107 фотонов/нм^2.

К изображениям был добавлен пуассоновский шум и beamstop с диаметром 10 пикселей. Моделирование проводилось в программе MOLTRANS, разработанной в DESY (Deutsches Elektronen-Synchrotron), которая учитывает особенности экспериментов по дифракции лазерного излучения на отдельных макромолекулярных объектах для лазеров на свободных электронах.

Дифракционные изображения, соответствующие разным образцам, показаны на рис. 1. Дифракционные изображения для капель воды заметно отличаются, тогда как изображения двух других типов сложно отделить друг от друга. Основная трудность — классифиировать BTV и аденовирус и показать, что различия в дифракционных изображениях могут быть использованы для успешной классификации.

III. Метод автоматической классификации дифракционных изображений

A. Характеристический вектор для дифракционных изображений

Для классификации нам необходимо извлечь из дифракционных изображений набор признаков, которые наилучшим образом связаны со структурой изучаемого объекта. Используя

факты о взаимосвязи спектра угловой кросскорреляционной функции интенсивности дифракционной картины и спектра электронной плотности, мы предлагаем в качестве характеристических векторов использовать спектр угловой автокорреляционной функции (1). В автокорреляционной функции содержится существенно меньшее количество информации, однако мы покажем, что этого достаточно для классификации изображений различных молекул.

$$C(q,\Delta)=\left\langle I(q,\phi)I(q,\phi+\Delta)\right\rangle_{\phi}. \tag{1}$$

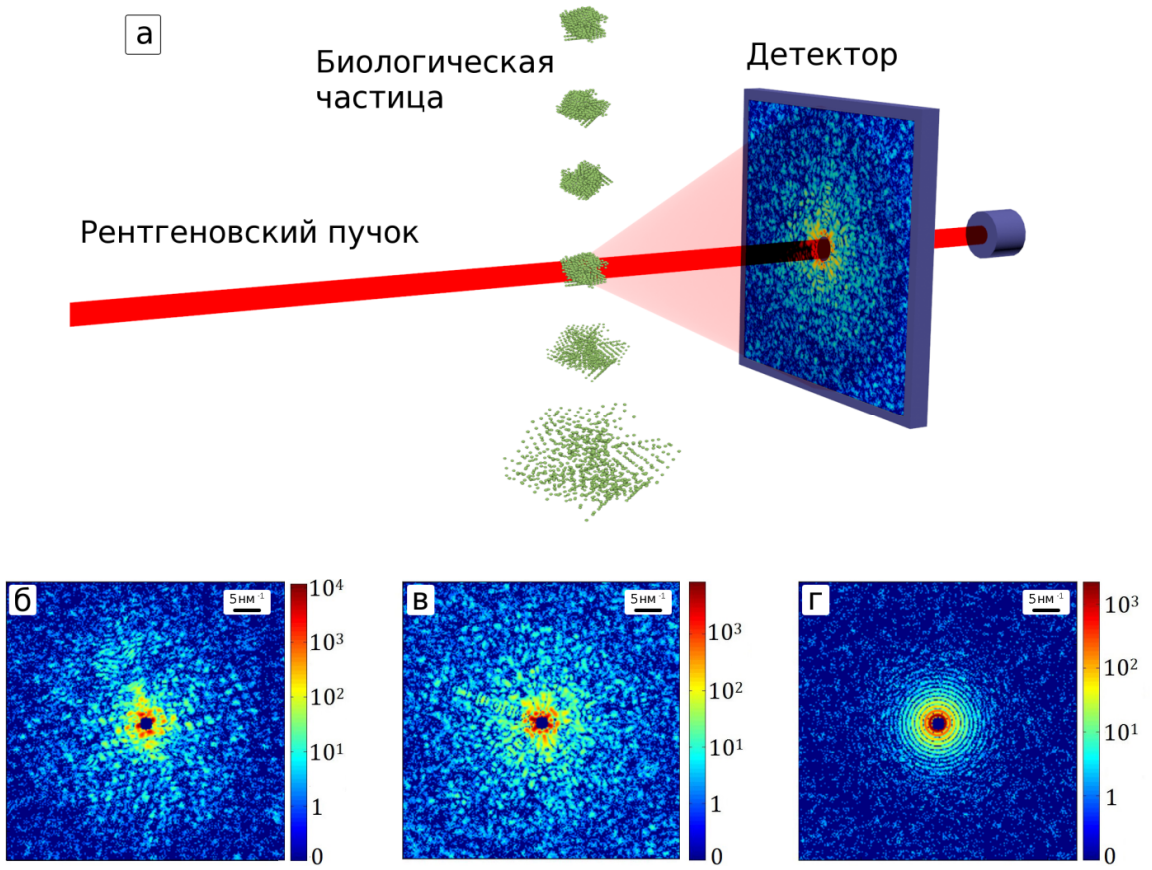


Рис. 1. (а) Схема эксперимента; (б), (в), (г) характерные дифракционные изображения различных молекул: (б) BTV; (в) аденовирус; (г) вода

Таким образом, каждому изображению будет соответствовать вектор:

$$F=\left\langle \bar{C}_q^1,\ldots,\bar{C}_q^n\right\rangle. \tag{2}$$

B. Кластеризация характеристических векторов с помощью метода главных компонент

Для классификации изображений мы уменьшаем размерность фазового пространства, определяемую размерностью характеристических векторов до плоскости, где точки, соответствующие отдельным дифракционным изображениям, могут быть визуально проанализированы и разделены, если возможно. Для этого мы применяем метод главных компонент (PCA).

PCA строит новый базис собственных векторов для матрицы ковариации данных, которые называются главными компонентами, и эти векторы упорядочены по убыванию соответствующих собственных значений. Затем ограниченное число первых главных компонент выбирается

как новое фазовое пространство. После этого анализируется проекция начального фазового пространства в новое пространство главных компонент.

Одна из особенностей PCA состоит в том, что пространство главных компонент, построенное с использованием первых n компонент, имеет максимальную дисперсию среди всех возможных ортонормальных базисов размерности n в фазовом пространстве. Таким образом, плоскость, построенная на основе первых двух главных компонент, будет содержать максимальную возможную дисперсию проекции данных.

Чтобы использовать метод главных компонент, мы строим матрицу: $A = \left\| a_{ij} \right\| = \left\| CCF_i^{(j)} \right\|$, где i соответствует изображению, а j — компоненте преобразования косинусов от угловой корреляции. Чтобы найти базис метода главных компонент, вычисляются собственные векторы матрицы ковариации данных. Самый быстрый способ сделать это — центрировать по столбцам матрицу A:

$$\overline{a}_{ij} = a_{ij} - \frac{1}{N}\sum_{k=0}^{N} a_{ki} \tag{3}$$

и затем, используя сингулярное разложение, записать $\left\| \overline{A} = \overline{a}_{ij} \right\|$ как

$$\overline{A} = U \Sigma V^T, \tag{4}$$

где U, V — унитарные матрицы, а Σ — диагональная. Легко показать, что столбцы матрицы V — это главные компоненты, которые мы ищем. Затем мы берем проекции для векторов, соответствующих каждом изображению и получаем координаты на плоскости (PC1, PC2).

Применяя метод главных компонент к набору характеристических векторов, мы получаем двумерную плоскость, где каждое дифракционное изображение описывается точкой. Если обработка изображений настроена правильно, разные типы изображений будут образовывать отдельные группы.

Если характеристические векторы, соответствующие разным группам, могут быть разделены в пространстве первых N главных компонент некоторым разделяющим правилом, то мы будем использовать это правило для классификации типов изображений.

Изображения, соответствующие каплям воды, группируются в одну точку, т. к. представляют собой концентрические окружности, и кросскорреляционная функция для таких изображений является константой, поэтому изображения для капель воды точно классифицируются по спектру кросскорреляционной функции. Результат применения метода главных компонент к набору модельных данных для аденовируса и 2BTV представлен на рис. 2. Видно, что дифракционные изображения образуют кластеры в многомерном пространстве. Данный результат подтверждает связь кросскорреляционных коэффициентов и пространственной структуры молекул, однако для классификации полного набора требуется более подходящий алгоритм.

C. Кластеризация характеристических векторов с помощью метода опорных векторов

Метод главных компонент (PCA) разделяет данные на основе среднего положения отдельных групп, а свойства данных требуют сфокусировать метод разделения на границе между группами. Этот подход был реализован с использованием линейного метода опорных векторов (Linear SVM).

SVM строит гиперплоскость в пространстве характеристических векторов высокой размерности, которая затем используется для разделения. Качество разделения достигается максимизацией расстояния от плоскости до ближайшей точки обучающего набора. Пусть есть обучающий набор (x_i, y_i), где x_i — это характеристический вектор и y_i — либо 1, либо –1 в зависимости от типа изображения, SVM находит решение проблемы оптимизации для всего обу-

чающего набора:

$$y_i\left(\mathbf{w}\cdot x_i-\mathbf{b}\right)\geq 1, \tag{5}$$

где ***w*** вектор нормали к гиперплоскости, а **b** определяет положение плоскости относительно начала координат вдоль вектора нормали **w**. Получившаяся гиперплоскость используется для классификации. Результат скалярного произведения характеристического вектора и **w** дает вероятность для частицы принадлежать определенному типу.

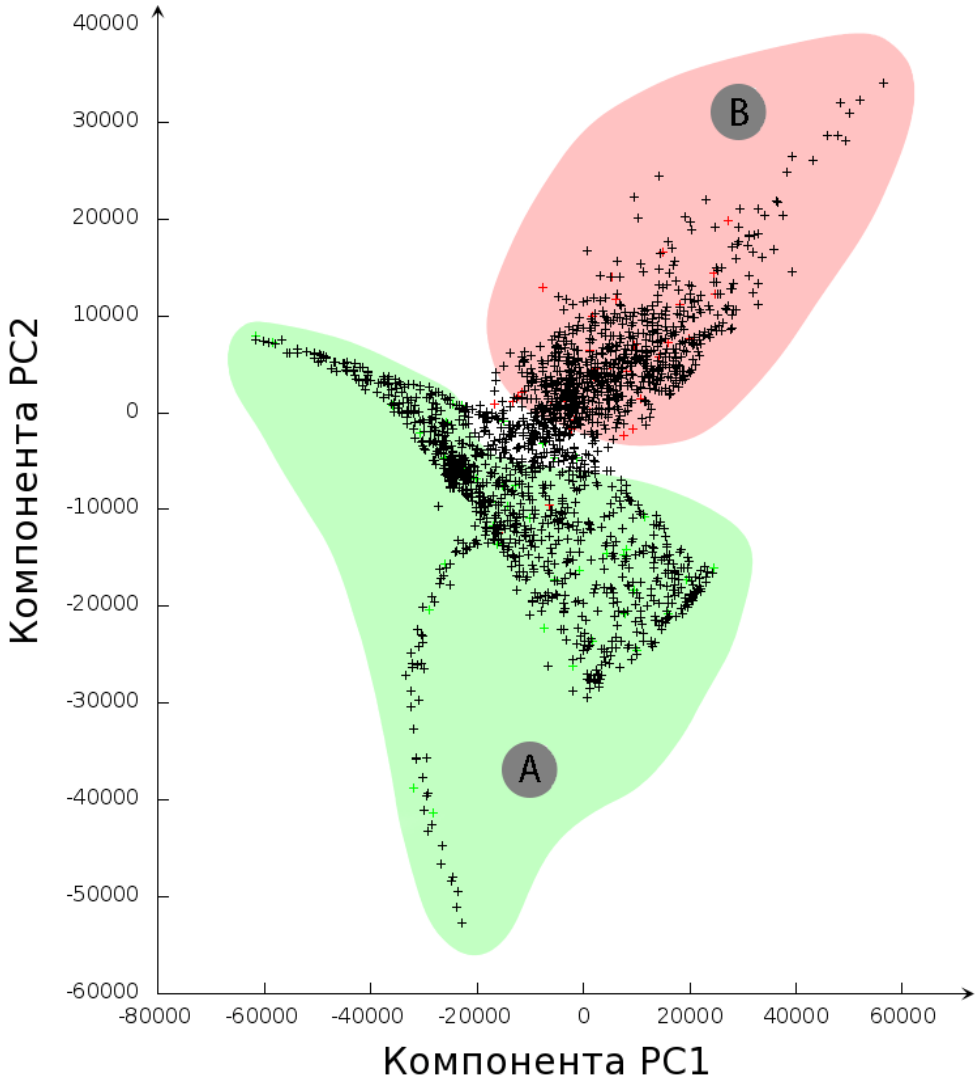


Рис. 2. Кластеризация модельных данных с помощью метода главных компонент. Каждому изображению соответсвует точка с координатами на плоскости первых двух главных компонент: PC1–PC2. Область A содежит изображения тренировочного набора, соответствующие только молекулам аденовируса, а область B — только молекулам 2BTV. Изображения между областями требуют более точной классификации

Существует набор различных расширений для метода опорных векторов, таких как мультиклассовый SVM или нелинейный SVM. Однако линейный SVM для двух классов был выбран как наиболее подходящий и имеющий наибольшее качество. Разделение для нескольких типов было реализовано последовательным применением метода опорных компонент к разделению определенного типа против всех остальных. Порядок выбора типа для разделения определяется сложностью отделения типа из априорных соображений.

Результат классификации набора модельных данных с помощью метода опортных векторов показан на рис. 3. При сравнении полученной классификации с точной классификаций набора модельных данных тип исходной молекулы был определен верно для 99 % дифракционных изображений.

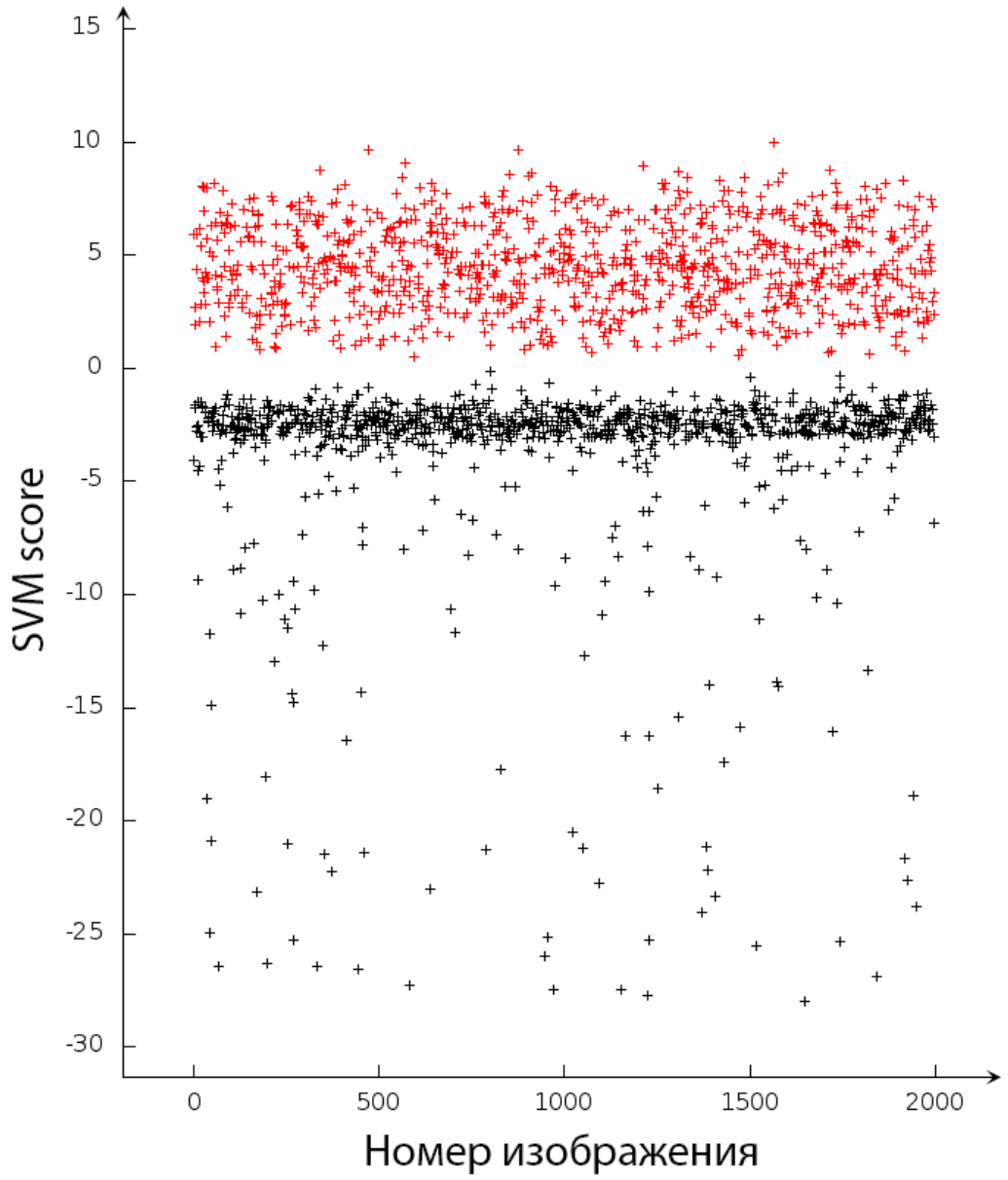


Рис. 3. Классификация на основе метода опорных векторов. Изображения аденовируса находятся в отрицательной части и имеют черный цвет, а изображения 2BTV — красный цвет. Параметр SVM score характеризует положение изображения относительно границы между двумя классами

Методы были реализованы с помощью языка Python, библиотеки матричных вычислений Numpy, математической библиотеки Intel Math Kernel Library, библиотеки Scikit-learn.

Высокая производительность вычислений достигалась с помощью применения технологии параллелизации вычислений OpenMP. Для вычислений использовался высокопроизводительный кластер центра коллективного пользования «Комплекс моделирования и обработки данных от исследовательских установок мегакласса» НИЦ «Курчатовский институт».

IV. Выводы

В работе мы представили два метода классификации дифракционных изображений на базе метода главных компонент (PCA) и на базе метода опорных векторов (BVM). Для модельных данных PCA позволяет показать кластеризуемость данных, однако не позволяет точно определить тип исходных молекул для всех изображений. Эффективность классификации SVM близка к 100 %. Превосходство SVM над PCA можно объяснить следующими соображениями: PCA ищет линейные комбинации свойств анализируемых объектов, которые наилучшим образом отличают все объекты друг от друга, в то время как SVM направлен на поиск оптимальной границы между двумя классами объектов, что больше соответствует задаче классификации классов изображений.

Наш метод классификации может быть применен в автоматизированной системе анализа данных, для поиска изображений, содержащих дифракционную картину исследуемых объектов, а также для индексирования наборов данных и поиска дифракционных изображений интересующих объектов. На его основе возможно построение самообучающихся алгоритмов классификации.

Список Литекратуры

Altarelli M., Kurta R., and Vartanyants I. Physical Review B 82, 104207. — 2010.

Berkhin P. Grouping Multidimensional Data, Recent Advances in Clustering. — 2006. — P. 25–71.

Chapman H. N. et al. Femtosecond X-ray protein nanocrystallography // Nature. — 2011. — Vol. 470. — P. 73.

Fung R., Shneerson V., Saldin D. K., and Abbas O. // Nature Physics. — 2009. — Vol. 5. — P. 64.

Gaffney K. J. and Chapman H. N. Imaging atomic structure and dynamics with ultrafast X-ray scattering // Science. — 2007. — Vol. 316. — P. 1444.

Grimes J. M. et al. // Nature. — 1998. — Vol. 395. — P. 470.

Jolliffe I. T., ed. Principal Component Analysis. — Springer. — 2002.

Loh N.-T. D. and Elser V. // Phys. Rev. — 2009. — E 80, 026705.

Mancuso A. P., Yefanov O. M., and Vartanyants I. A. // J. Biotechnology. — 2010. — Vol. 149. — P. 229.

Neutze R., Wouts R., Van der Spoel D., Weckert E., and Hajdu J. Potential for biomolecular imaging with femtosecond X-ray pulses // Nature. — 2000. — Vol. 406. — P. 752.

Sebastiani F. ACM Computing Surveys (CSUR) Surveys 34, 1. — 2002.

Seibert M. M. et al. // Nature. — 2011. — Vol. 470. — P. 78.

Viola M., Jones P. Computer Vision and Pattern Recognition, CVPR. 1, I. — 2001.

Yang J., Zhang D., Frangi A., and Yang J.-Y. Pattern Analysis and Machine Intelligence. — 2004. — Vol. 26. — P. 131.

Yefanov O. M. and Vartanyants I. A. // J. Phys. B: At. Mol. Opt. Phys. — 2013. — 46, 164013.

Yoon C. H. et al. // Optics Express. — 2011. — Vol. 19. — 16542.

Zubieta C., Blanchoin L., and Cusack S. 273, 4336 — 2006.

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 641–648**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.02

Интерактивный графический инструментарий глобального вычислительного эксперимента в службе морских оперативных прогнозов

А. В. Богданов[a], Я. А. Дегтярева, Е. А. Захарчук, Н. А. Тихонова, В. Р. Фукс, В. Н. Храмушин

Санкт-Петербургский государственный университет,
Россия, 199034, г. Санкт-Петербург, Университетская наб., д. 7-9

E-mail: [a]bogdanov@csa.ru

Получено 11 февраля 2015 г.

Эффективность и полнота численного моделирования в океанологии и гидрометеорологии всецело обусловливаются алгоритмическими особенностями построения интерактивного вычислительного эксперимента в масштабах Мирового океана с адаптивным покрытием закрытых морей и прибрежных акваторий уточненными математическими моделями, с возможностью программного распараллеливания уточняющих расчетов вблизи конкретных — защищаемых участков морского побережья. Важной составляющей исследований представляются методы непрерывной графической визуализации в ходе вычислений, в том числе осуществляемой в параллельных процессах с общей оперативной памятью или по контрольным точкам на внешних носителях. Результаты вычислительных экспериментов используются в описании гидродинамических процессов вблизи побережья, учет которых важен в организации морских служб контроля и прогноза опасных морских явлений.

Ключевые слова: вычислительный эксперимент, интерактивные графические комплексы, инженерные системы, океанология, гидрометеорология, тензорная математика

Исследования выполняются при поддержке грантами РФФИ (№ 13-07-00747), СПбГУ (№ 9.38.674.2013, № 0.37.155.2014, № 18.37.140.2014) и «Комплексная программа ДВО РАН «Дальний Восток» (№ 15.3312-III-СО-08-023), с использованием вычислительной техники Ресурсного центра «Вычислительный центр СПбГУ».

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 641–648 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 641−648

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

Interactive graphical toolkit global computer simulations in marine service operational forecasts

A. V. Bogdanov, Ya. A. Degtyreva, E. A. Zakharchuk, N. A. Tikhonova, V. R. Foux, V. N. Khramushin

Saint-Petersburg state university, 7-9 Universitetskaya naberezhnaya, St. Petersburg, 199034, Russia

Abstract. — Efficiency and completeness of the numerical simulation in oceanography and hydrometeorology entirely determined by algorithmic features of the construction of an interactive computer simulations in the scale of the oceans with adaptive coated closed seas and coastal waters refined mathematical models, with the possibility of specifying software parallelization calculations near the concrete — the protected areas of the sea coast. An important component of the research are continuous graphical visualization techniques in the course of calculations, including those undertaken in parallel processes with shared RAM or test points on the external media. The results of computational experiments are used in the description of hydrodynamic processes near the coast, which is important in keeping the organization of sea control services and forecasting marine hazards.

Keywords: computational simulation, interactive graphics systems, engineering systems, oceanography, hydrometeorology, tensor mathematics

Введение

Современные информационно-вычислительные системы вполне удовлетворяют минимальным требованиям по производительности и способны оперировать глобальными объемами общегеографических, океанографических и гидрометеорологических данных для вовлечения прямых вычислительных экспериментов в регламентную работу морских оперативных служб. Актуальной задачей видится реализация контроля текущего состояния с обеспечением своевременных прогнозов опасных морских явлений, прибрежных экстремальных течений и нетиповых астрономических приливов, штормовых нагонов и сейсмических цунами, численное моделирование которых основано на вовлечении динамических граничных условий по огромным океанским акваториям, в том числе с использованием различных эмпирических источников потенциальной опасности в удаленных районах Мирового океана.

Новизна исследований определяется возможностью анализа алгоритмических особенностей интерактивного вычислительного эксперимента для моделирования длинноволновых процессов в масштабах Мирового океана и частично закрытых морей, в том числе под управляемым распараллеливанием уточняющих расчетов вблизи конкретных участков побережья. Графическая визуализация служит средством контроля качества моделирования гидродинамических процессов в подобластях с плохо обусловленными аппроксимационными критериями и востребуется в оценках вычислительной эффективности при оптимизации физической постановки задачи, в том числе с должным учетом архитектурных особенностей вычислительной техники. С этой целью рассматриваются особенности функциональной среды прямых вычислительных экспериментов в гидромеханике, формулируются оптимальные подходы к реализации объектно ориентированной среды программирования, включая согласованные методы построения числовых объектов и операций с явным или функциональным (рекурсивным) описанием больших сеточных областей и массивов общегеографических исходных данных.

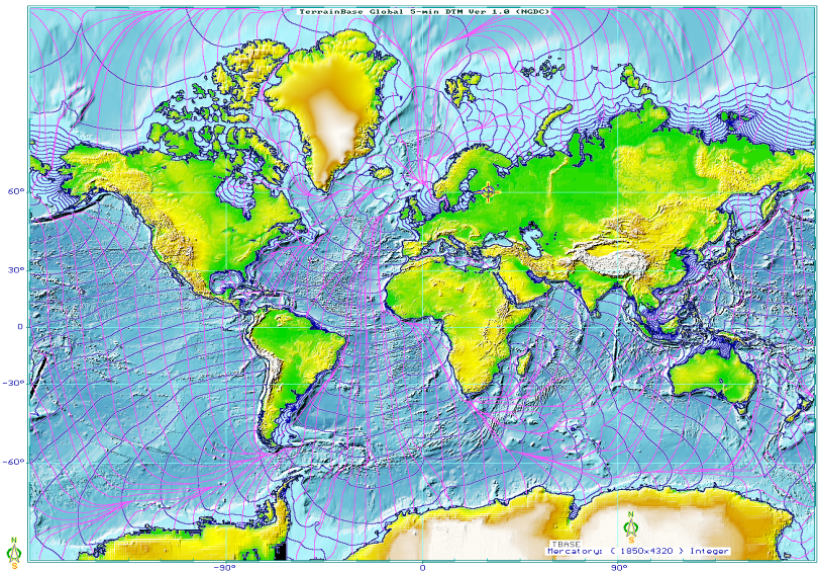


Рис. 1. Цифровой массив высот и глубин по Земному шару в меркаторской проекции с шагом 5 морских миль на средней приведенной широте. Океан покрыт сеткой ежечасных изохрон и лучей для распространения длинноволнового импульса со стороны г. Санкт-Петербурга

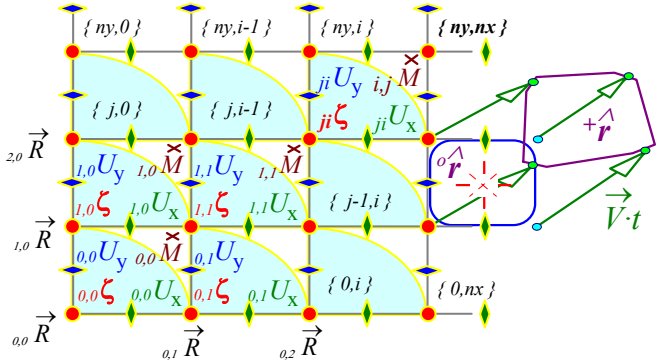
Приведенная на рис. 1 цифровая батиметрическая модель Земного шара адаптирована для глобального моделирования длинноволновых процессов в Мировом океане, в том числе под динамическим воздействием геопотенциала Луны и Солнца, с учетом придонного трения, сферичности и вращения Земли [Храмушин, 2010], при этом скорость вычислений, на виртуальной машине с графической станцией и восьмью CPU-Xeon по 3 ГГц, в два раза быстрее реального

времени. Формально резерв скорости необходим, к примеру, для дополнительного учета стратификации водных потоков или для включения в вычислительный эксперимент атмосферных циклонов и приводных ветров, по необходимости для приостановки расчетов с целью адаптации вычислений к иным внешним гидродинамическим воздействиям типа сейсмических цунами или подводных оползней и вулканов.

Вычислительная задача

Длинноволновая вычислительная модель представляется замечательным образцом обобщенной инженерной задачи для всестороннего изучения и освоения интерактивных графических комплексов при реализации совместных вычислительных экспериментов в гидромеханике океана и атмосферы, в которых распараллеливание алгоритмов по моделированию в масштабах всего Земного шара должно сочетаться с необходимостью параллельной отработки множества синхронных сценариев для прогнозного и уточняющего моделирования гидрофизических и гидродинамических процессов вблизи морского побережья, на акваториях рейдов и гаваней морских портов.

Исходная математическая модель в канонической форме (рис. 2, выражения (I) и (III)) представляется системой дифференциальных уравнений первого порядка [Храмушин, 2010], для которых возможно применение явных численных схем с разделением этапов решения по независимым физическим процессам. В алгоритмическом плане это означает возможность распараллеливания вычислительных операций вплоть до каждой отдельно взятой сеточной ячейки — частицы жидкости — и, соответственно, доступность сквозного контроля физического состояния моделируемой среды с целью динамического выбора адекватной математической, асимптотической или эвристической модели течения.



$$\delta\vec{U} = \mathrm{g}\,H\cdot\frac{\partial\zeta}{\partial\vec{r}}\,\delta t \qquad \text{I}$$

$$\delta\tilde{\vec{U}} \cong \frac{\partial\vec{V}}{\partial\vec{r}}\vec{V}\cdot H\,\delta t \qquad \text{II}$$

$$\delta\zeta = \left(\frac{\partial\tilde{\vec{U}}}{\partial\vec{r}}\right)\delta t \qquad \text{III}$$

Рис. 2. Вычислительная схема континуально-корпускулярного представления длинноволновой модели динамики океана: U — вектор полного потока; H — глубина моря; ζ — отклонение уровня от равновесного положения; V — скорость смещения частицы относительно неподвижной сетки; R — вектор местоположения и r — тензор формы частицы; I — уравнение движения (внешнее воздействие); II — деформация крупной частицы жидкости (конвективность — интерполяция к исходным узлам); III — условие неразрывности (захват энергии движения)

Интерполяционный этап II (рис. 2) отвечает за большие смещения и деформации частиц жидкости и обычно не включается в длинноволновые вычислительные эксперименты. Вычислительная модель вполне может учитывать реальную высоту волны — для моделирования планового наката на пологом побережье, для чего должен усиливаться контроль сохранности частиц жидкости — корпускул на наклонной поверхности, наполняющих русла рек и растекающихся по пологим пляжам и др. В любом случае явные численные схемы не востребуют специальных регулярных расчетных сеток, и не приводят к необходимости задействования гигантских массивов оперативной памяти для порождения числовых матриц с заведомо плохой обусловленностью в подобластях с большими градиентами физических полей.

Вычислительная среда

Современное развитие систем дистанционного зондирования и телеметрического контроля состояния атмосферы и океана достигает минимально необходимого уровня информационной поддержки для контроля корректности и актуальности вычислительных экспериментов при непрерывном моделировании динамики океана, атмосферы и их взаимодействия в реальном масштабе времени.

Кинематика длинноволнового поля (рис. 1) показывает геометрическую соразмерность океанов и открытых акваторий, ограниченных лучевыми трубками между суточными и полусуточными изохронами. Это объясняет резонансное усиление лунно-солнечного геопотенциала в приливном отклике с 70 см до 12 м (Пенжинская губа Охотского моря) и более и служит обоснованием необходимости непрерывного численного моделирования приливного режима по всему Мировому океану в реальном времени, что, по крайней мере, обязательно востребуется в динамическом представлении граничных условий в вычислительных экспериментах с опасными морскими явлениями для локальных акваторий океана и отдельных участков побережья.

В явных численных схемах каждая расчетная ячейка-частица может обрабатываться по особым алгоритмическим правилам, например, в простейшем случае, для логического исключения из расчетов береговых или осушенных поверхностей и др. В программном комплексе Ani [Храмушин, 2010] обнаружено и задействовано важнейшее свойство прямых вычислительных моделей — это равнозначность физических уравнений движения (рис. 2, I) и неразрывности (рис. 2, III), для которых все внешние воздействия должны включаться в соответствии с концепцией половинных шагов во времени при разделении этапов моделирования по физическим процессам и столь же тщательно прописываться в условиях на свободных границах расчетных областей.

В вычислительной модели (рис. 2) плавно изменяются векторное поле полных потоков U и поле уровней моря ζ, при этом весовая функция — поле фазовых скоростей длинных волн $C=(\mathrm{g}\cdot H)^{1/2}$ — является принципиально негладким и недифференцируемым. Это требует особого внимания при алгоритмической балансировке численных схем, что, в свою очередь, проявляется в результатах заметным дисперсионным выполаживанием (красным смещением) волновых процессов во времени, что в целом соответствует наблюдениям за естественным распространением длинных волн над неровным морским дном.

В реализации вычислительного эксперимента необходимо согласование аппроксимационных критериев по пространству и во времени. «Во времени» — это критерий устойчивости, превышение которого приводит к взрывному разрушению вычислительных процессов. Существующее противоречие разрешается увеличением шага сетки в акваториях с большими глубинами, что значительно повышает эффективность и скорость проведения вычислительного эксперимента без потери качества и точности моделируемых гидродинамических процессов. Нарушение критериев пространственных аппроксимаций связано с уменьшением длины моделируемой волны до размеров сеточных ячеек, отчего возникают неугасаемые точечные источники волн, влияющие на общий декремент затухания гидродинамических процессов во всей расчетной области, и практически необнаружимо в результатах вычислительных экспериментов в виде мареграфных записей колебаний уровня моря и скоростей течений. Если в вычислительном эксперименте не задействуются гибридные алгоритмы для моделирования планового наката и вихревых течений на мелководьях, то обнаружение проблем с аппроксимационной гладкостью возможно лишь визуально, по графическому изображению скалярных полей уровня моря или векторных полей полных потоков скоростей течений.

Гидродинамические особенности проявления длинноволновой динамики моря вблизи побережья были выявлены в ходе проведения серии прямых вычислительных экспериментов. Результаты аналитических обобщений могут быть полезны в регламенте работы морских служб при выработке и обосновании оперативных предупреждений о проявлении потенциально опасных морских явлений. Проявление морских наводнений (*«эхо и реверберация» — в акустике*)

у конкретных участков морского побережья более всего обусловливается этапами прохождения прогрессивных волн и возбуждения сопутствующих резонансных проявлений длинноволновых процессов в относительно мелководных шельфовых акваториях (рис. 3) и всецело зависящих от геометрических особенностей прибрежного рельефа морского дна.

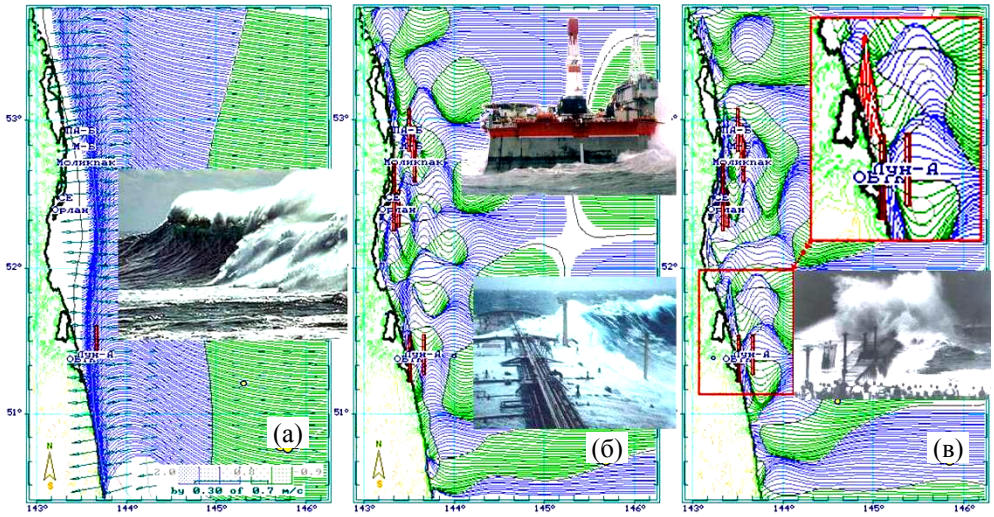


Рис. 3. Северо-восточный шельф острова Сахалин. Три типовых варианта трансформации длинных волн вблизи морского побережья: (а) — обрушение фронта первого вступления волны вблизи побережья, характерное для цунами; (б) — установившиеся колебания уровня моря под воздействием штормов или морской зыби; (в) — обрушение длиннопериодной морской волны на побережье в случае нарушения условий стационарности длинноволновых процессов на изобатах порядка 25–40 м

Если прибрежная акватория характеризуется монотонным наклоном морского дна, без каких-либо горизонтальных поверхностей и закрытых заливов, то это является условием отсутствия собственных длинноволновых колебаний вблизи такого побережья. Однако именно такие участки морского дна могут стать своеобразным проводником длинных волн с большой амплитудой до самого побережья, нерезонирующий «волновод» способен откачивать на себя длинноволновую энергию из близлежащих акваторий со значительным увеличением высоты падающей на берег прогрессивной волны (рис. 3*в*), с последующим рассеянием на прибрежных мелководьях и непосредственно на береговом урезе в виде высоких волновых групп.

Характер воздействия на побережье первых фронтов прогрессивных волн (цунами, рис. 3*а*) заметно отличается от постоянно существующих длинноволновых резонаторов в прибрежных мелководьях открытого моря (рис. 3*б*), зарождающихся под воздействием различных метеорологических факторов или вынужденных волновых колебаний, привносимых из удаленных штормовых акваторий. Многовековые геоморфологические процессы самостабилизируют собственные длинноволновые колебания вблизи побережья, однако если в штормовых условиях случается условно незначительное нарушение стандартного течения гидрофизических или гидрометеорологических процессов, то огромная энергия шельфовых колебаний уровня моря может быть перенаправлена в сторону побережья (рис. 3-С), что естественным образом послужит усиленному разрушению берегов и всей инженерной инфраструктуры, приведшей к нарушению стационарности, порождая опасные, но формально восстановительные процессы перестроения устойчивых состояний шельфовых резонансов в обширных прибрежных акваториях.

Аналогичные по характеру нестационарности наводнения могут вызываться после первого вступления одиночного или нетипичного фронта волны наводнения, что будет проявляться в форме подвижных пакетов волновых структур (*захваченной энергии*), перемещающихся строго вдоль побережья, и последовательно заливающего все пункты по маршруту следования.

Интенсивность длинноволновых процессов на морском шельфе и вызываемых ими морских наводнений на побережье в существенной степени зависит от гидродинамических условий

поддержания стабильности собственных длинноволновых колебаний уровня моря, проявление которых может быть с высокой достоверностью картировано по результатам серии длительных прямых вычислительных экспериментов.

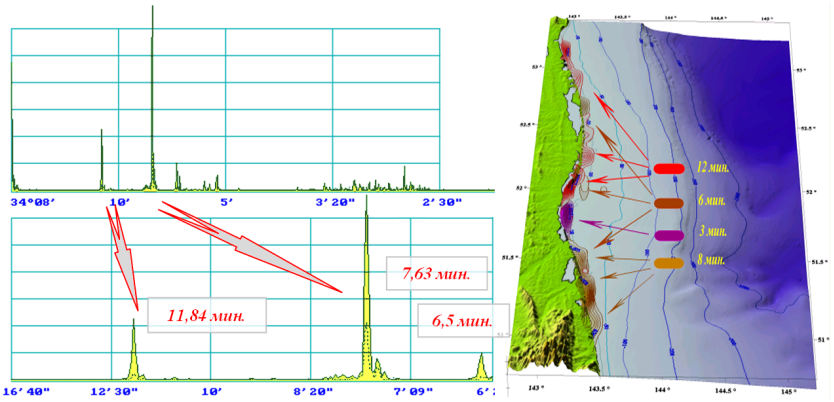


Рис. 4. Сахалинский шельф. Характерные периоды длинных волн, способных длительное время сохраняться или аккумулироваться на северо-восточном шельфе о-ва Сахалина. Существование прибрежных резонаторов принципиально подтверждается морскими экспедиционными измерениями

В прибрежных акваториях, и особенно в полузамкнутых бухтах и портовых гаванях, всегда присутствуют области со свойствами высокодобротных резонаторов, на которых аккумулируется энергия длинноволновых колебаний уровня моря на строго определенных периодах, определяемых ее топологическими особенностями. Своеобразные «волновые аккорды» или «полосовые спектры».

По рис. 4 отметим, что если пространственные экстремумы для собственных колебаний уровня моря находятся непосредственно вблизи побережья, то такой берег может быть признан небезопасным для строительства прибрежных инженерных сооружений. Если же на берег будет выходить узловая линия между пучностями стоячих волн, то у такого участка небезопасны строительство выносных причалов или организация якорных стоянок, даже в случае защищенности этих акваторий от прямого воздействия морской зыби и штормовых ветров. Такие заключения могут не учитываться только в одном случае — если по результатам серии вычислительных экспериментов доказывается принципиальная невозможность зарождения интенсивных гидродинамических процессов с означенными периодами длинных волн.

В прямых вычислительных экспериментах выявляются аккорды собственных длинноволновых колебаний, отображающиеся во всех прилегающих пунктах регистрации уровня моря в форме стоячих волн, с понижением энергетического уровня при удалении от зоны пучности — конкретного мелководного резонатора. Процессы на других периодах длинных волн, отличных от составляющих характерного аккорда близлежащих резонаторов, рассеиваются на границах открытого моря и быстро прекращают сколь-нибудь значимое воздействие на побережье.

Выводы

Картирование интенсивности длинноволновых процессов для защищаемых участков морского побережья [Симаков, Храмушин, Шевченко, 2012] может служить не только традиционным инженерным задачам районирования морских акваторий по проявлениям потенциально опасных морских наводнений и экстремальных течений. Одновременное формирование цифровой базы знаний для разнообразных исторических штормовых нагонов и цунами обязательно должно задействоваться как для быстрых оценок развития неблагоприятных синоптических или сейсмических событий на море, так и с целью быстрого формирования надежных сценариев для прямых вычислительных экспериментов для прогноза наиболее опасных или катастрофических последствий у побережья в действующем регламенте морских оперативных служб.

В случае развертывания системы наблюдения за состоянием моря на Дальнем Востоке России прямые вычислительные эксперименты обретут наивысшую эффективность, так как могут задействоваться по факту превышения пороговых значений в скорости изменения уровня моря или опасных скоростей течений, независимо от типа вызвавшего их источника.

По результатам большой серии вычислительных экспериментов важно отметить, что именно штормовые наводнения представляют наибольшую опасность на морском побережье, так как они способны вызывать непредвиденные подъемы уровня моря и экстремальные течения под воздействием барических градиентов атмосферного давления, ветровых напряжений на поверхности мелководных шельфовых акваторий. Опасность штормовых наводнений усугубляется действием длиннопериодной зыби, глубоко проникающей на побережье, что чрезвычайно опасно для всех морских инженерных сооружений на побережье и рейдовых мелководьях.

Заключение

Длинноволновые вычислительные эксперименты чрезвычайно актуальны в задачах контроля состояния и прогноза необратимых или опасных процессов вблизи морского побережья России, имеющего наивысшую гидрофизическую активность и наибольшую протяженность во всем Мире. Практическая реализация прямых вычислительных экспериментов служит высококачественным инструментом для инженерных изысканий в открытом океане и в прибрежных мелководьях, которые по уровню точности и детализации не поддаются ни теоретическому, ни экспериментальному аналитическому обобщению.

Список литературы

Симаков Е. Е., Храмушин В. Н., Шевченко Г. В. Сахалин — море — Расчетные и регистрируемые колебания уровня моря на Дальнем Востоке России // СахГУ, Роспатент № 2012620509. — 2012.

Храмушин В. Н. Ani — Прямые вычислительные эксперименты для моделирования цунами, штормовых нагонов, экстремальных течений и приливного режима в открытом океане и вблизи побережья // СахГУ. Роспатент № 2010615848. — 2010.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 649–655

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.65

Оптимизация запросов в РБД и распространение технологии облачных вычислений

А. В. Богданов, Тхурейн Киав Лвин[a]

Санкт-Петербургский государственный университет,
Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

E-mail: [a]trkl.mm@mail.ru

Получено 21 января 2015 г.

Рассматривается задача оптимизации запросов в распределенных базах данных. Анализируются традиционные подходы к оптимизиции и их обобщение в распределенных системах. Выясняются существенные проблемы при переносе стандартных подходов на такие системы. Предлагается использовать новые технологии облачных вычислений для решения таких проблем.

Ключевые слова: оптимизация запросов, облачные вычисления, распределенные базы данных

Query Optimization in Relational Database Systems and Cloud Computing Technology

A. V. Bogdanov, Thurein Kyaw Lwin

[1] *Saint Petersburg State University, University ave. 35, Peterhof, St. Petersburg, 198504, Russia*

Abstract. — The problem of query optimization in distributed databases is presented. The traditional approaches to optimization and their generalization in distributed systems are analyzed. The significant problems when porting standard approaches to such systems are illuminated. It is proposed to use some new approaches in cloud computing to solve such problems.

Keywords: query optimization, cloud computing technology, relational database systems

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 649–655 (Russian).

Оптимизация — это сердце для реляционных СУБД. Она анализирует SQL-заявления и определяет наиболее эффективный план доступа для удовлетворения каждого запроса. Оптимизация решает эту задачу и анализирует SQL-заявления, определяя, какие таблицы и столбцы должны быть доступны. Затем запросы информационной системы и статистические данные, хранящиеся в системном каталоге, определяют наилучший метод решения задач, необходимых для удовлетворения этой просьбы.

Оптимизация существенна и для экспертной системы при доступе к базе данных. Экспертная система представляет собой набор стандартных правил, который в сочетании с ситуационными данными может либо выдать рекомендацию, либо вернуться к мнению экспертов. Реляционный оптимизатор показывает влияние мнения экспертов о методах поиска данных. Понятие «оптимизация» для доступа к данным в СУБД дает очень мощный потенциал, который мы сегодня воспринимаем как должное. На сегодняшний день доступ к реляционным данным достигается путем запроса СУБД. Независимо от того, какие данные физически хранятся и обрабатываются, SQL могут быть использованы для доступа к данным. Такое разделение критериев доступа, физических характеристик и типа систем хранения называется физической независимостью данных, и оптимизация имеет решающее значение в достижении этой физической независимости.

Если индексы в таблице удаляются, вы все равно можете получить доступ к данным. Если столбец будет добавлен в таблицу, к которой осуществляется доступ, с данными все еще можно манипулировать без изменения программного кода. Все это возможно потому, что физические пути доступа к данным не заложены программистами в прикладной программе, а создаются оптимизатором.

Оптимизация выполняет сложные расчеты, основанные на множественной информации. Для упрощения функциональности оптимизатора мы можем разложить его деятельность на четыре этапа:

получение и проверка синтаксиса SQL-заявления;

анализ окружающей операционной среды и оптимизация методов удовлетворения SQL-заявления;

создание команды для выполнения оптимизированных SQL;

выполнение инструкции или сохранение ее для будущего исполнения.

Оптимизация SQL имеет множество стратегий. Производители СУБД не публикуют углубленные детали внутренней работы своих оптимизаторов, но хороший оптимизатор должен будет минимизировать потребные ресурсы (cost-based). Это означает, что оптимизатор всегда будет пытаться сформулировать путь доступа для каждого запроса, чтобы уменьшить общую стоимость транзакций. Чтобы достичь этого, оптимизатор запросов будет применять формулы, которые оценивают стоимость и вес многих факторов для каждого потенциального пути доступа, такие как стоимость процессора, I/O-стоимость, статистическая информация в системном каталоге, а также фактическое заявление SQL.

Без статистики, которая хранится в системном каталоге, оптимизация будет трудно осуществима. Эта статистика оптимизатора с информацией о состоянии таблиц, которые будут доступны в заявлении SQL, и данные о том, что в настоящее время уже оптимизировано. Типы статистической информации включают в себя:

информацию в таблицах, включающую общее число строк, информацию о компрессии и общее количество страниц;

информацию в столбцах, в том числе количество дискретных значений для столбца, и распределение диапазона значений, хранимых в столбце;

информацию в табличных пространствах, в том числе количество активных страниц;

текущее состояние индекса, включается ли индекс, существует или нет организация индекса (number of leaf pages and number of levels), число дискретных значений ключа индекса и делится ли индекс по группам;

информация табличного пространства и табуляторы.

Статистические данные собираются и хранятся в системном каталоге, и поэтому полезно выполнить операции обновления статистики (например, RUNSTATS или UPDATE STATISTICS). Для более эффективной работы с DBA важно контролировать, чтобы статистические данные накапливались в процессе выполнения транзакций, особенно в производственной среде.

В настоящее время происходит развитие и распространение технологии «облачных вычислений» [Никульчев, 2003; Плужник, Функционирование…, 2013]. Растущий спрос на услуги провайдеров, предлагающих широкий спектр услуг в области облачных вычислений для большого числа пользователей по всему миру, приводит к увеличению количества приложений, целью которых является обработка больших массивов данных. Функционирование баз данных в облачной среде приводит к необходимости поиска новых инструментов [Плужник, Слабоструктурированные…, 2013].

Оптимизация запросов в облачных БД SQL

Обработка запроса сводится к преобразованию высокоуровневого запроса в эквивалентную низкоуровневую форму, и основной трудностью при этом является обеспечение эффективности преобразования с учетом специфики облачных хранилищ. Стандартные SQL-запросы используют соединения (join), выборку (select), проекции (projections), группировки (group-by). В [Zhang, 2012] приводится описание архитектуры, предназначенной для обработки и хранения больших объемов данных на основе семантических алгоритмов поиска плана исполнения текущего запроса в глобальной схеме исполнения запросов (рис. 1).

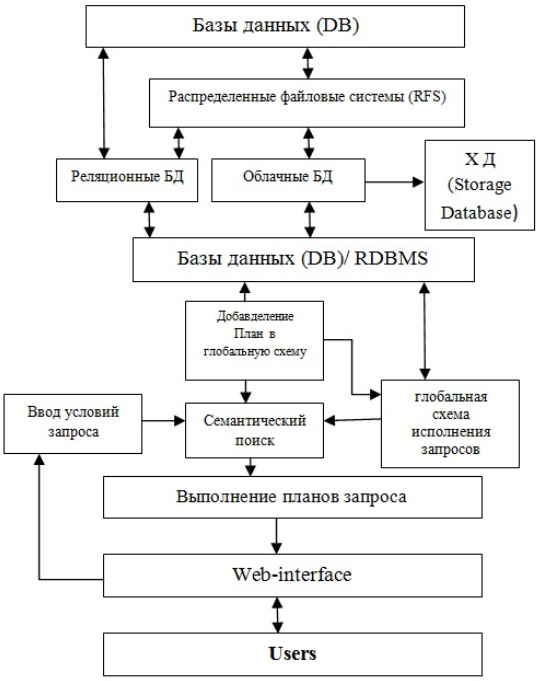


Рис. 1. Архитектура системы, основанной на SQL-запросах

Ключевые принципы указанной архитектуры состоят в следующем.

1. Все файлы хранятся в локальной файловой системе (например, файловая система Windows, Linux и т. д.).

2. Облачная БД предназначена для хранения и управления огромными массивами файлов индекса и метаданных. При этом следует отметить, что облачная база данных со всем содержимым развернута поверх распределенной файловой системы.

3. Ввод запросов и получение результатов выполняется посредством пользовательского веб-интерфейса.

4. После получения пользовательского запроса выполняется семантический поиск плана исполнения текущего запроса в глобальной схеме (как подмножество).

Архитектурно зависимые решения

Основной интерес указанного подхода заключается в том, что его целью является балансировка нагрузки на оборудование посредством миграции виртуальных машин в облачном окружении, что опосредованно приводит к повышению качества поиска. Зачастую для управления ресурсами внутри крупномасштабных центров обработки данных разрабатываются и внедряются централизованные решения, однако в этом случае возникновение сбоя на управляющем узле приводит к неработоспособности всей системы в целом.

Как показано на рис. 2, каждый активный узел в процессе функционирования выборочно с заданным интервалом отправляет собственный индекс загруженности некоторым узлам системы, в то же время получая индексы загруженности случайно выбранных активных узлов. При этом целевые узлы меняются на каждой итерации. Информация о загруженности других узлов добавляется в вектор загруженности текущего. Таким образом, средняя длина вектора загруженности узла равна количеству итераций отправки индекса. Информация о загруженности будет храниться децентрализовано, что позволит избежать неприятностей в случае выхода из строя части узлов, еще одним положительным моментом является то, что сетевой трафик будет распределен по всем активным узлам.

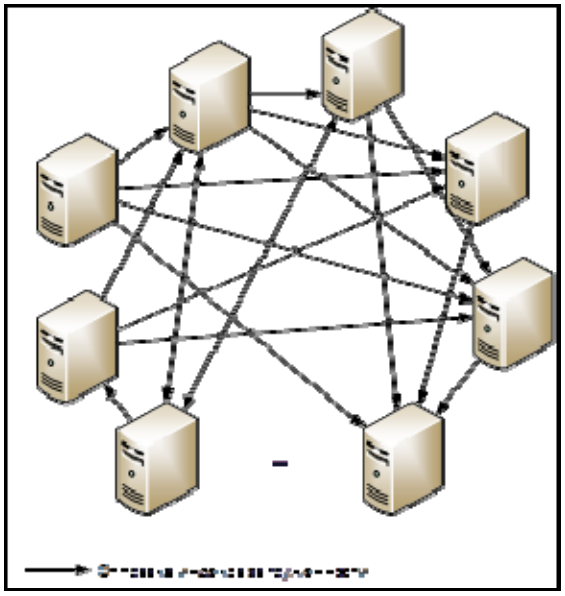


Рис. 2. Децентрализованный обмен индексом загруженности

Поскольку виртуальные машины служат хостом для развертывания разнообразных приложений с различающимися рабочими нагрузками на ЦПУ, то со временем загруженность физических ЦПУ может сильно меняться. При этом решение о миграции виртуальной машины может быть принято в двух случаях:

1) когда использование ЦПУ превышает определенный уровень (верхний порог); целью установления верхнего порога является сохранение дополнительных вычислительных мощностей на случай возникновения ситуаций с резким (незапланированным) повышением нагрузки;

2) когда использование ЦПУ ниже определенного уровня (нижний порог), узел используется недостаточно; цель установления нижнего порога состоит в том, чтобы по возможности

большее число физических узлов было переведено в «спящий» режим, что позволит снизить энергопотребление.

После того как принято решение о миграции виртуальной машины, стартует поиск узла назначения [Wang, 2013]. Для этого выполняется обход вектора загруженности текущего узла с целью обнаружения узла с наименьшим потреблением ЦПУ при условии попадания в заданные интервалы. Если такой узел обнаружить не удается, выполняется поиск такого узла, индекс загруженности которого при переносе на него выбранной ВМ не превышает нижней границы загруженности. Если же и в этом случае поиск не дает результатов, один из узлов, находящихся в «спящем» режиме, переводится в активное состояние и выполняется миграция.

Недостатки рассматриваемого подхода

1. Описываемые в 10 алгоритмы (псевдокоды) не гарантируют обязательности выбора ВМ для миграции даже при условии необходимости в этом и наличии свободных физических узлов.
2. Не рассматривается оптимальность выбора ВМ для миграции.
3. Поиск целевого узла осуществляется не на всем наборе узлов (согласно описанию подхода).
4. Не говорится о том, как часто выполняется проверка необходимости миграции.

Алгоритмы не учитывают продолжительности нахождения узла в состоянии повышенной загруженности: очевидно, что при непродолжительной загруженности длительность миграции может снизить ее эффективность.

В настоящее время ведутся направления по разработке динамических подходов, основанных на работах [Никульчев, 2008; Lemmon, 2012]. Данные работы используют динамические модели в форме системы конечно-разностных моделей на основе идентифицированных моделей [Никульчев, 2004].

В схеме (рис. 3) исполнения и оптимизации динамических распределенных запросов в облачных пиринговых сетях авторами также разработан фреймворк (DObjects) для работы с p2p-сетями.

Ключевым элементом обработки запросов в предлагаемом подходе является наличие ядра, способного динамически адаптироваться к условиям сети и источникам. При этом подходе вывод результатов и физические расчеты по плану выполняются динамически и итеративно.

Такой подход гарантирует лучшую реакцию на изменение нагрузки и позволяет снизить задержки в системе. Следует отметить, что оптимизация исполнения запросов на локальных БД в текущем подходе ложится на адаптеры и источники данных.

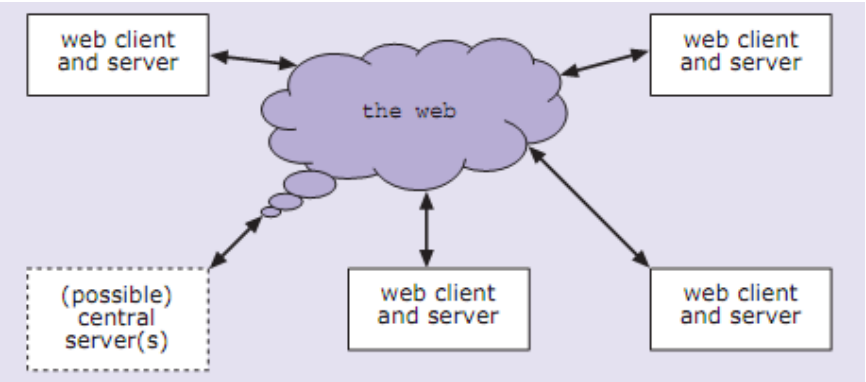


Рис. 3. Архитектура облачной пиринговой (p2p) сети

Исполнение и оптимизация запроса состоят из нескольких основных этапов.

1. В момент получения узлом запроса от пользователя генерируется высокоуровневый план исполнения.

2. На следующем шаге узел, исполняющий запрос, выбирает активные элементы плана сверху вниз в порядке следования. Однако исполнение активного элемента может быть делегировано любому узлу системы в целях достижения масштабирования нагрузки. Для выбора целевого узла исполнения в сети разворачивается модуль, способный адаптироваться к специфике сети и загруженности ресурсов. Если активный элемент передается на исполнение удаленному узлу, то управление его дочерними элементами также возлагается на этот узел. Удаленный узел, в свою очередь, может принять решение о перемещении дочерних узлов элемента плана на исполнение другим узлам либо выполнить локально.

Достоинством данного подхода является то, что он может быть внедрен в кратчайшие сроки, поскольку алгоритмы были реализованы в виде фреймворка. Однако облачные центры обработки данных, основанные на пиринговых сетях, на данный момент мало применяются на практике.

Заключение

Каждый из представленных методов обладает как достоинствами, так и недостатками. Общим для всех недостатком является синтетичность результатов, то есть то, что статистика по внедрению получена на искусственных системах, созданных только для тестирования подхода. Однако относительно большое количество исследований для довольно молодой области говорит о том, что проблема оптимизации актуальна и такие разработки в ближайшем времени будут востребованы. В настоящем сообщении рассматривается в широком смысле задача оптимизации обращений к базам данных и даются практические рекомендации, которые могут упростить обработку очень больших массивов данных.

Список литературы

Никульчев Е. В. Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре / Е. В. Никульчев, С. В. Паяин, Е. В. Плужник // Вестник Рязанского радиотехнического университета. — 2003. — № 3. — С. 54–57.

Никульчев Е. В. Использование групп симметрий для идентификации сложных систем / Е. В. Никульчев // Вычислительные технологии. — 2004. — Т. 9, № 3. — С. 72–80.

Никульчев Е. В. Построение модели загрузки каналов связи в сетях передачи данных на основе геометрического подхода / Е. В. Никульчев, С. В. Паяин // Известия высших учебных заведений. Проблемы полиграфии и издательского дела. — 2008. — № 6. — С. 91–95.

Плужник Е. В. Слабоструктурированные базы данных в гибридной облачной инфраструктуре / Е. В. Плужник, Е. В. Никульчев // Современные проблемы науки и образования. 2013. — № 4. URL: www.science-education.ru/110-9980 (дата обращения: 15.10.2013).

Плужник Е. В. Функционирование образовательных систем в гибридной облачной инфраструктуре / Е. В. Плужник, Е. В. Никульчев // Известия вузов. Проблемы полиграфии и издательского дела. — 2013. — № 3. — С. 96–105.

Fegaras L. An Optimization Framework for Map-Reduce Queries / L. Fegaras, C. Li, U. Gupta // Proc. of the 15th International Conference on Extending Database Technology. — ACM, 2012. — P. 26–37.

Jahani E. Automatic optimization for MapReduce programs / E. Jahani, M. J. Cafarella, C. Ré // Proceedings of the VLDB Endowment. — 2011. — Vol. 4, No. 6. — P. 385–396.

Jurczyk P. Dynamic query processing for p2p data services in the cloud / P. Jurczyk, L. Xiong // Database and Expert Systems Applications. — Springer Berlin Heidelberg, 2009. — P. 396–411.

Lemmon M. D. Towards a passivity framework for power control and response time management in cloud computing // In Proc. of 7th Intl. Workshop on Feedback Computing, San Jose, CA. 2012.

Wang X. A Decentralized Virtual Machine Migration Approach of Data Centers for Cloud Computing / X. Wang, X. Liu, L. Fan, X. Jia // Mathematical Problems in Engineering [Электронный журнал]. — 2013. — Vol. 2013. — Режим доступа: http://www.hindawi.com/journals/mpe/2013/878542/, свободный. — Дата обращения 15.10.2013.

Zhang G. Massive Data Query Optimization on Large Clusters / G. Zhang, Chao LI, Yong Zhang, Chunxiao Xing. // Journal of Computational Information Systems. — 2012. — Vol. 8. — С. 3191–3198.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 657–660

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 681.3.01
PACS: 02.70.-c

Основные направления развития информационных технологий Национальной академии наук Азербайджана

А. С. Бондяков

Лаборатория информационных технологий, Объединенный институт ядерных исследований,
Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6
Институт Физики НАН Азербайджана,
Азербайджан, AZ1143, г. Баку, ул. Г. Джавида, д. 33

E-mail: aleksey@jinr.ru

Получено 30 сентября 2014 г.

Грид-инфраструктура — компьютерная инфраструктура нового типа, обеспечивающая глобальную интеграцию информационных и вычислительных ресурсов. Грид-сегмент в Азербайджане был создан в 2008 году в Институте физики НАН при активной поддержке международных организаций ОИЯИ и CERN. Грид приобретает все большую популярность в научно-исследовательских и образовательных центрах Азербайджана. Среди основных направлений использования грид на данный момент можно выделить научные исследования в физике высоких энергий, физике твердого тела, энергетике, астрофизике, биологии, науках о Земле, а также в медицине.

Ключевые слова: грид-инфраструктура, информационные технологии, грид-сегмент, облачная инфраструктура

Basic directions of information technology in National Academy of Sciences of Azerbaijan

A. S. Bondyakov

Laboratory of Information Technologies, Joint Institute for Nuclear Research, 6 Joliot Curie st., Dubna, 141980, Russia
Institute of Physics, 33 H. Javid st., Baku, AZ1143, Azerbaijan

Abstract. — Grid — a new type of computing infrastructure, is intensively developed in today world of information technologies. Grid provides global integration of information and computing resources. The essence Conception of GRID in Azerbaijan is to create a set of standardized services to provide a reliable, compatible, inexpensive and secure access to geographically distributed high-tech information and computing resources a separate computer, cluster and supercomputing centers, information storage, networks, scientific tools etc.

Keywords: grid infrastructure, information technology, grid segment, cloud infrastructure

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 657–660 (Russian).

В настоящее время в Азербайджане при активной поддержке Национальной академии наук интенсивно развиваются информационные технологии. Можно выделить несколько основных направлений:

- грид-инфраструктура;
- облачная инфраструктура;
- информационная безопасность.

На данный момент дата-центр Института физики включает в себя 300 двухпроцессорных и 8 блейд-серверов. Все серверы — на базе процессора Intel Xeon. Общая память составляет 200 ТВ. Общее количество ядер 950. 200 ядер — для грид-сегмента. 150 ядер выделено для локального кластера. 600 ядер — для учебного класса. Вся внутренняя сеть дата-центра построена на базе 1GbE.

(а) (б)

Рис. 1. Серверное оборудование грид-сегмента.
(а) группа блейд-серверов, (б) хранилище данных)

Дата-центр Института физики НАН Азербайджана (ИФ) построен как единый информационно-вычислительный ресурс для всех направлений исследований, в которых участвуют сотрудники ИФ [Abdinov O. et al., 2013]. Все счетные ресурсы и ресурсы для хранения данных обслуживаются единым базовым программным обеспечением, позволяющим использовать ресурсы комплекса как в международных проектах распределенных вычислений, так и локально пользователями НАН. Системное программное обеспечение оптимизировано таким образом, чтобы обеспечить наиболее эффективное использование вычислительных ресурсов вместе с максимально защищенным и в то же время наиболее универсальным доступом к данным. Базовой операционной системой является ОС Linux. Дата-центр ИФ состоит в виртуальных организациях SEE-GRID (страны Черноморского бассейна) и EDU — учебно-исследовательская и тестовая грид-инфраструктура (ОИЯИ, Россия). Кроме того, в рамках виртуальной организации SEE-GRID дата-центр ИФ участвует в таких организациях, как EGI (European Grid Infrastructure) и NGI (National Grid Initiatives).

Серверное и сетевое оборудование представлено продукцией Supermicro и Cisco. Дата-центр Института физики работает в режиме 24/7. Защита оборудования от различных перепадов в электрической сети осуществляется средствами UPS и генератора. Климат-контроль обеспечивается прецизионными кондиционерами. Температура в машинном зале не превышает 18 °C.

Подготовлены необходимые условия для подключения к ресурсам грид-центра Института физики, научных и образовательных центров Азербайджана. Для этой цели используется сеть провайдера НАН–AZRENA.

Мониторинг ЦПУ, локальной сети и интернет-соединения в режиме реального времени осуществляется средствами GANGLIA и CACTI (рис. 4). Мониторинг грид-сервисов осуществляется средствами Nagios.

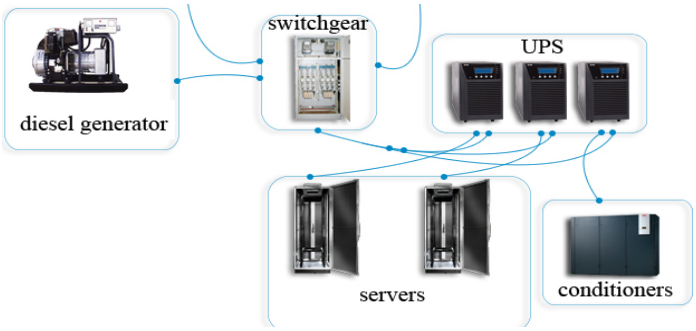


Рис. 2. Структура дата-центра Института физики

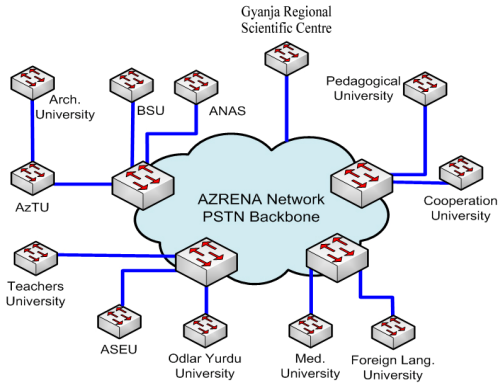


Рис. 3. Сеть провайдера НАН–AZRENA

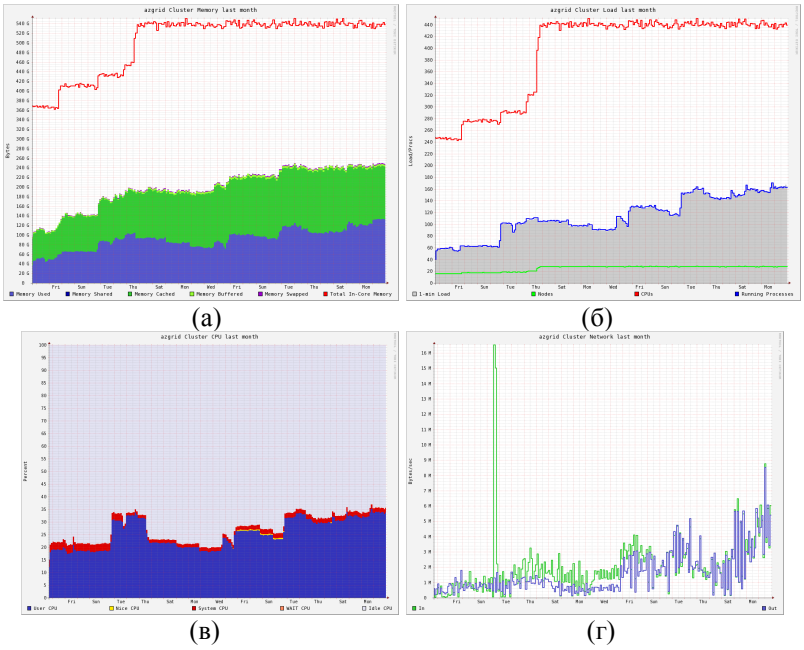


Рис. 4. Результаты мониторинга с 01.07.2014 по 01.08.2014: (а) ОЗУ; (б) выполняемые процессы; (в) ЦПУ; (г) локальная сеть и интернет-соединение

Для сложных математических расчетов были установлены следующие компиляторы: g77/gcc/g++ — GNU Fortran 77, C and C++ compilers version 3.4.6; gfortran/gcc4/g++4 — GNU Fortran 95, C and C++ compilers version 4.1.2; ifort/icc/icpc — Intel Fortran, C, C++ compilers version 11.1. GCC (GNU Compiler Collection).

Также были установлены следующие пакеты программного приложения:

Abinit — свободное программное обеспечение, распространяемое по GNU General Public License3 и предназначенное для расчетов полной энергии, электронной плотности и т. д. систем электронов и ядер (с использованием периодических граничных условий) в рамках метода функционала плотности с использованием базиса из плоских волн и псевдопотенциалов.

Quantum ESPRESSO — позволяет оптимизировать геометрию системы, минимизируя силы или напряжения, проводить молекулярно-динамическое моделирование, вычислять распределение электронной плотности, определять динамическую матрицу, эффективный заряд и многое другое.

Функциональность программ значительно расширяется имеющимися утилитами.

Авторами [Hashimzade et al., 2010; Huseinova et al., 2011; Allakhverdiev et al., 2012] с помощью пакета Abinit были рассчитаны зависимости структурных параметров кристаллической решетки, координаты атомов в зависимости от давления, линейные сжимаемости, а также фононные частоты слоистых кристаллов TlGaSe2 при гидростатическом давлении в диапазоне 0–5.0 GPa (рис. 5).

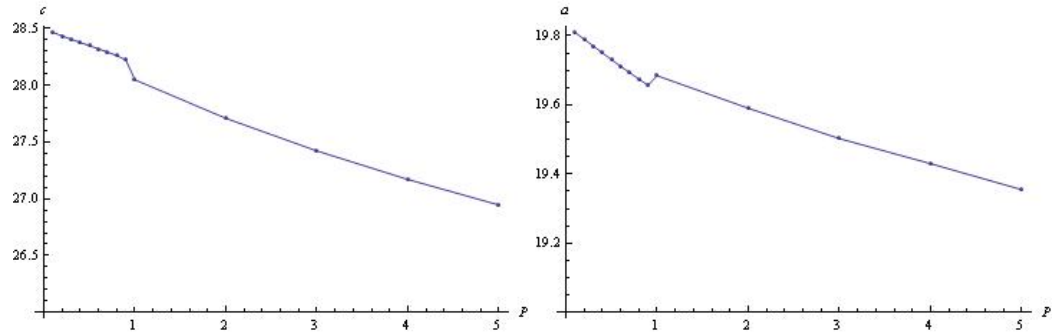


Рис. 5. Результаты расчета параметров решетки кристалла TlGaSe2

Список литературы

Abdinov O., Bondyakov A., Khalilova Sh., Orujova N. Conception GRID Infrastructure in Azerbaijan // XXIV International Symposium NEC 2013. — 2013. — P. 9–12.

Allakhverdiev K. R., Hashimzade F. M., Huseinova D. A., Nizametdinova M. A., Orudzhev G.S., Ulubey A. M., Kir M. H. Lattice dynamics of Ferroelectric TlInS2 crystal // Can. J. Phys. — 2012. — Vol. 90 — P. 407–412.

Huseinova D. A., Hashimzade F. M., Orudzhev G. S., Nizametdinova M. A., Allakhverdiev K. R. Abinitio Lattice Dynamics and Gruneisen Parameters of TlGaSe2 // Japanese Journal of Applied Physics. — 2011. — Vol. 50. — DOI: 10.1143/JJAP.50.05FE05.

Hashimzade F. M., Huseinova D. A., Orudzhev G. S., Nizametdinova M. A. Lattice dynamics of layered ferroelectric semiconductor compound TlGaSe // Materials Research Bulletin. — 2010. — Vol. 45. — P. 1438–1442.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 661−673

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.9

Технология формирования каталога информационного фонда

В. Н. Добрынин[1], И. А. Филозова[2, a]

[1] ГОУ ВПО «Международный университет природы, общества и человека «Дубна»,
Институт системного анализа и управления,
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19
[2] Объединенный институт ядерных исследований,
Лаборатория информационных технологий,
Россия, 141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6

E-mail: [a] fia@jinr.ru

Получено 30 сентября 2014 г.

В статье рассматривается подход совершенствования технологий обработки информации на основе логико-семантической сети (ЛСС) «Вопрос–ответ–реакция», направленный на формирование и поддержку каталожной службы, обеспечивающей эффективный поиск ответов на вопросы [Большой энциклопедический словарь, 1998; Касавин, 2009]. В основу такой каталожной службы положены семантические связи, отражающие логику изложения авторской мысли в рамках данной публикации, темы, предметной области. Структурирование и поддержка этих связей позволят работать с полем смыслов, обеспечив новые возможности для исследования корпуса документов электронных библиотек (ЭБ) [Касавин, 2009]. Формирование каталога информационного фонда (ИФ) включает: формирование лексического словаря ИФ; построение дерева классификации ИФ по нескольким основаниям; классификация ИФ по вопросно-ответным темам; формирование поисковых запросов, адекватных дереву классификации вопросно-ответных тем (таблица соответствия «запрос → ответ ↔ {вопрос–ответ–реакция}»); автоматизированный поиск запросов по тематическим поисковым машинам; анализ ответов на запросы; поддержка каталога ЛСС на этапе эксплуатации (пополнение и уточнение каталога). Технология рассматривается для двух ситуаций: 1) ИФ уже сформирован; 2) ИФ отсутствует, его необходимо создать.

Ключевые слова: информационный фонд, Большие Данные, информационный поиск, пертинентность, навигация, информационно-поисковая система, семантические связи, логико-семантическая сеть «вопрос–ответ–реакция»

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 661–673 (Russian).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 661−673

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

Cataloging technology of information fund

V. N. Dobrynin[1], **I. A. Filozova**[2]

[1] *Institute of System Analysis and Management, Dubna International University for Nature, Society, and Man*
19 Universitetskaya str., Dubna, Moscow region, 141980, Russia
[2]*Joint Institute for Nuclear Research, Laboratory of Information Technologies,*
6 Joliot Curie st., Dubna, 141980, Russia

Abstract. — The article discusses the approach to the improvement of information processing technology on the basis of logical-semantic network (LSN) Question–Answer–Reaction aimed at formation and support of the catalog service providing efficient search of answers to questions.

The basis of such a catalog service are semantic links, reflecting the logic of presentation of the author's thoughts within the framework this publication, theme, subject area. Structuring and support of these links will allow working with a field of meanings, providing new opportunities for the study the corps of digital libraries documents. Cataloging of the information fund includes: formation of lexical dictionary; formation of the classification tree for several bases; information fund classification for question–answer topics; formation of the search queries that are adequate classification trees the question–answer; automated search queries on thematic search engines; analysis of the responses to queries; LSN catalog support during the operational phase (updating and refinement of the catalog). The technology is considered for two situations: 1) information fund has already been formed; 2) information fund is missing, you must create it.

Keywords: information fund, Big Data, information search, pertinence, navigation, search engine, semantic relations, logic-semantic network «Question–Answer–Reaction»

Введение

Современные проблемы и задачи требуют для своего решения анализа больших объемов информации, распределенных в различных источниках. Время и ресурсы для решения проблем и задач, как правило, ограничены и зачастую несопоставимы с существующими механизмами поиска, селекции и аккумуляции требуемой информации по качеству. Неслучайно, что один из наиболее часто упоминаемых сегодня терминов в IT-области — это Big Data [Hilbert, López, 2011; Найдич, 2012; Якшонок, 2012].

Объемы научных фондов и их число растут с некоторой скоростью [Редькина, 2010]. Механизмы поиска становятся неэффективными по показателям «время», «деньги», «качество». С одной стороны, имеют место рост объема разнородной информации, рост в потребности в качественной информации. С другой — неэффективные поисковые информационные системы (ИПС) и общие вопросно-ответные системы. Специалисту в определенной области знаний важно иметь инструмент для эффективного исследования информации в массивах научных публикаций как основной продукции деятельности ученых и исследователей. В связи с этим возникает необходимость эффективных (высокий уровень релевантности, время поиска, большие объемы информации) ИПС и вопросно-ответных систем.

1. Данные–информация–знание

Существует множество взглядов на термин Big Data. McKinsey в отчете 2011 г. “Big data: The next frontier for innovation, competition, and productivity” определяет большие данные как такие объемы информации, которые выходят за рамки возможностей используемых в организации СУБД по их анализу и хранению. По мнению консорциума MIKE 2.0, термин означает не столько большой объем данных, сколько их сложность, вариативность, разнородность и неструктурированность. Источники Big Data разнообразны. Ими могут быть текстовые документы, файлы CAD-, CAM-приложений, показания датчиков, сенсоров, систем видеонаблюдения, системные журналы и пр. То есть это массивы данных, которые потенциально содержат ценную информацию, но в чем состоит ценность и как ее извлечь — непонятно. Таким образом, главными признакми данных категории Big Data можно назвать: а) затруднения в их обработке; б) трудность их интерпретации.

Характеристики 3V — объем/volume, скорость/velocity, многообразие/variety — требуют адекватных реакций информационных систем на решение задач поиска и обработки больших массивов с качеством Big Data. Это находит отражение на всех уровнях архитектуры современных систем: 1) аппаратное обеспечение; 2) системное и проблемно ориентированное обеспечение; 3) прикладное ориентированное обеспечение. Для решения задач такого рода активно задействуются параллельные распределенные архитектуры, грид, облачные инфраструктуры. Возможно, что появится совершенно новое решение в другой парадигме. Усиление аппаратной составляющей особенно актуально для систем массового обслуживания.

В характеристику «больших данных» *velocity* (динамичность, изменчивость, скорость изменения) может быть заложено изменение структуры данных. Хорошо структурированная информация может быть достаточно точно представлена данными. Слабо структурированная информация может быть представлена данными с высокой степенью неопределенности, что является следствием их изменчивости. Семантическое структурирование контента информационных фондов имеет целью формирование его смыслового поля и направлено на снижение степени неопределенности.

В настоящий момент информационные потребности пользователей направлены на получение новой информации и новых знаний из уже имеющихся массивов данных. Вычислительная нагрузка на компьютеры гораздо меньше, чем собственно обработка данных. То есть данные обрабатываются с целью получения информации, которую человек способен преобразовать в знание. Таким образом, актуальна цепочка «Данные–информация–знание».

Термин *знание* имеет несколько значений и толкований. Знание противопоставляется незнанию, т. е. отсутствию проверенной информации о чем-либо.

Знание — форма существования и систематизации результатов познавательной деятельности человека [Большой энциклопедический словарь, 1998]. Знание помогает людям рационально организовывать свою деятельность и решать проблемы, возникающие в ее процессе.

Знание (предмета) — уверенное понимание предмета, умение обращаться с ним, разбираться в нем, а также использовать для достижения намеченных целей.

Новое знание — совокупность сведений о существовании каких-либо объектов или их свойств, о процессах и явлениях действительности, ранее не известных науке и не входящих в существующую на данный момент систему человеческих представлений о мире [Касавин, 2009].

Пропущенное знание — знание, известное человечеству, но на данный момент не известное конкретному человеку (например, студенту, изучающему новый предмет образовательной программы).

Знание в широком смысле — субъективный образ реальности в форме понятий и представлений.

Знание в узком смысле — обладание проверенной информацией (ответами на вопросы), позволяющей решать поставленную задачу.

Знание в теории искусственного интеллекта (ИИ) и экспертных систем — совокупность информации и правил вывода (у индивидуума, общества или системы ИИ) о мире, свойствах объектов, закономерностях процессов и явлений, а также правилах использования их для принятия решений. Главное отличие знаний от данных состоит в их структурности и активности, появление в базе новых фактов или установление новых связей может стать источником изменений в принятии решений.

Целью научной деятельности является генерация нового знания, которому неизбежно предшествует информационный поиск (для изучения текущего состояния предметной области); обработка результатов поиска и генерация на ее основе новых данных. То есть прочтение, изучение любого научного текста происходит с определенной целью, формируемой в контексте задач, которые пользователи решают в процессе своей профессиональной деятельности. Знание фиксируется в определенных образах и знаках естественных и искусственных языков. Формой представления знаний может быть публикация различных жанров (статья, монография, препринт и т. д.), электронные архивы, репозитории, таблицы, базы данных и т. д.

2. Проблемы информационного поиска

Поиск — это сложная итерационная процедура, предполагающая уточнение запроса. Специалист в некоторой предметной области, осуществляющий поиск, имеет определенное представление о том, что из полученных результатов может являться ответом на его вопрос. Однако большинство классических информационно-поисковых и вопросно-ответных систем являются одноконтурными или двухконтурными (позволяющими осуществить поиск в найденном). Специалисту же удобнее уточнять запрос/вопрос не выходя из системы. Основные критерии оценки эффективности поисковых систем — скорость, точность и полнота ответов. Точность определяется тем, какая часть информации, выданной в ответ на запрос, является релевантной, т. е. относящейся к этому запросу. Полнота характеризуется соотношением между всей релевантной информацией, имеющейся в базе, и той ее частью, которая включена в ответ. Кроме этого, при оценке поисковых систем учитывается, с какими типами данных может работать та или иная система, в какой форме представляются результаты поиска и какой уровень подготовки пользователей необходим для работы в этой системе. Но для пользователя наиболее важна прагматическая характеристика информационного поиска, отражающая насколько результаты поиска удовлетворяют информационной потребности пользователя (пертинентность) — соответствие полученного результата информационной потребности пользователя независимо от того, как

полно и как точно эта информационная потребность была выражена с помощью информационного запроса или вопроса. Пертинентность измеряется степенью соответствия между ожиданиями пользователя и результатами поиска и определяется как отношение объема полезной для пользователя информации к общему объему полученной информации, найденной поисковой системой. Возможные причины низкой пертинентности информационного поиска изображены на рис. 1.

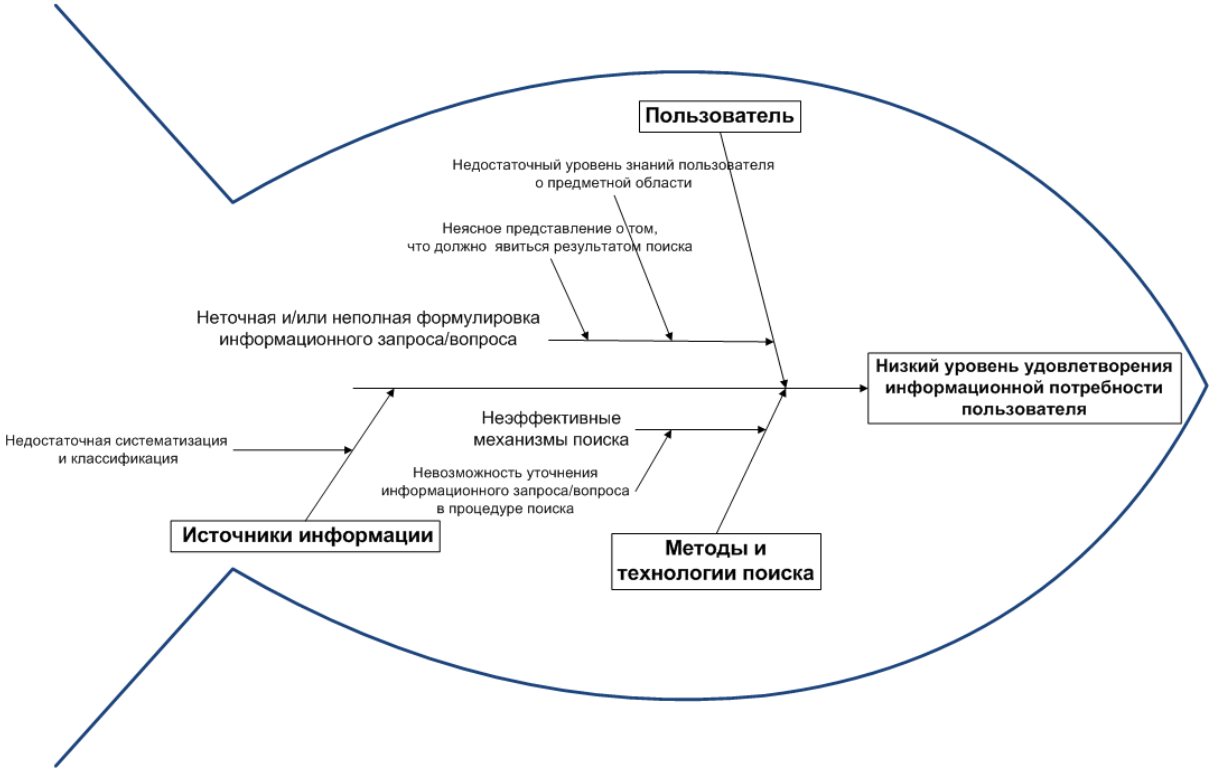


Рис. 1. Причины низкой пертинентности информационного поиска

Развитие методов организации информационного поиска подтверждает научно-практический интерес к решению этой проблемы.

3. Пути решения проблемной ситуации

Традиционные подходы к организации поиска информации можно разделить на три группы: методы индексного (или двоичного) поиска, статистические методы и методы, основанные на базах знаний. Для максимального удовлетворения информационных потребностей пользователей в настоящее время в поисковых системах широко применяются теории и методы семантических сетей, контент-анализа и интеллектуальный анализ текстов (Text Mining).

Индексный, или двоичный, поиск применяется главным образом со структурированными базами данных. Системы двоичного поиска имеют ограничения по точности, влияющие на возможность нахождения всей относящейся к запросу информации. В методах двоичного поиска не учитываются различные формы и значения слов; пользователю непросто угадать точные слова и фразы, которые были использованы авторами в документах. Системы двоичного поиска не могут также ранжировать документы по степени соответствия запросу, поэтому пользователь вынужден читать каждый документ, чтобы определить, насколько он соответствует запросу.

Статистические методы основываются на расчете различных частотных характеристик: частоты вхождения слова в документ, взвешенной частоты вхождения и частоты совместного

вхождения нескольких слов. Основной единицей информации, которой оперируют статистические методы, является отдельное слово, однако связи между словами рассматриваются исключительно с математической, а не с лингвистической точки зрения. В отличие от методов двоичного поиска статистические методы не требуют применения жесткого формального языка запросов. Они позволяют проводить ранжирование документов по степени соответствия запросу, что существенно повышает эффективность работы с поисковыми системами. Однако такие методы не всегда позволяют получить желаемые точность и полноту ответов, поскольку важность того или иного термина не напрямую связана с частотой его использования в документе.

Системы, основанные на базе знаний, занимаются поиском информации на основе некоторых внешних знаний. Они используют концептуальные отношения, которые не применяются при статистическом поиске. Одним из наиболее простых и распространенных способов представления знаний является файл синонимов. Другой подход к системам, основанным на базе знаний, использует иерархию терминов и понятий, создаваемую самими пользователями. Третий известен как подход на основе лингвистических правил.

Подход, использующий ссылочные документы, в том числе обычные словари и словари терминов, основан на смысловых значениях слов и называется семантической сетью. Как и словарь, семантическая сеть содержит множество определений для каждого хранимого слова. Однако определения родственных слов и понятий связываются между собой. Значения слов, наиболее подходящие для данного поиска, могут быть выбраны самим пользователем с целью повышения точности этого поиска. Подход на основе семантических сетей реально объединяет статистический поиск и поиск на основе базы знаний. При этом используются смысловые значения слов для определения и классификации отношений, которые статистический поиск не отслеживает.

Системы, основанные на базах знаний, гораздо удобнее тех, которые базируются на двоичном поиске. Однако сегодня лишь подход, основанный на построении семантических сетей, свободен от ограничений, присущих двоичному поиску; он обладает достаточной гибкостью, доступен для расширения и не слишком громоздок при эксплуатации.

Под контент-анализом в интернет-поисковиках понимают оценку структуры и материалов веб-ресурса с точки зрения поисковой оптимизации. При контент-анализе оцениваются такие смысловые единицы, как наличие контента (страниц), релевантного поисковому запросу; уникальность контента; удобство использования (дизайн, структура сайта, навигация); удобство контента для восприятия; качество html-кода.

Text Mining представляет собой множество методов обработки текста, в результате применения которых появляются новые, ранее не предполагавшиеся знания. Это междисциплинарная область, в которой используются базовые технологии Data Mining совместно с методами информационного поиска, извлечения информации, математической лингвистики, создания онтологий, классификации, кластеризации и др.

4. Логико-семантические сети «вопрос–ответ–реакция»

Работа специалиста-профессионала с информационными фондами предполагает наличие системы каталогизации и классификации материала. В зависимости от специализации контента информационные системы обеспечиваются электронными каталогами с целью описания ресурсов для их однозначной идентификации и обеспечения доступа к ним. В рамках заданной проблемной темы предлагается технология формирования и поддержки *каталожной* службы, которая обеспечивает эффективный поиск ответов на вопросы. Основой такой *каталожной* службы является упорядоченное открытое множество логико-семантических сетей (ЛСС) «вопрос–ответ–реакция» [Добрынин и др., 2014].

Любая научно-практическая область знаний включает предмет исследования, который может быть представлен проблемным полем (перечнем проблемных вопросов), являющимся основой для научной и практической деятельности. Проблемные вопросы могут быть пред-

ставлены в виде иерархического дерева по принципу «от общего к частному». Для некоторых вопросов уже существуют возможные альтернативные ответы и способы их реализаций (реакции). Для понимания вопроса также необходима определенная реакция. Ответы могут порождать, в свою очередь, вопросы. Таким образом, проблемный вопрос соотносится с определенной темой предметной области и раскрывается семантической структурой вопрос–ответ–реакция, которая, вообще говоря, является открытой (т. е. пополняемой, изменяемой) во времени. Другими словами, знания, накопленные в предметной области, могут быть представлены открытым множеством логико-семантических сетей (ЛСС), упорядоченных по предметным темам. Задача предметной области может быть сформулирована в форме вопроса. Выявление в вопросе таких смыслов, как тема вопроса, содержание вопроса, объем вопроса, позволяет найти релевантные ЛСС, в которых могут содержаться как ответы, так и необходимые объяснения (реакции). Ввод реакций помогает пользователю понять, получил ли он релевантный и пертинентный ответ на свой вопрос. В качестве реакций может выступать дополнительная информация по теме вопроса и ответа, иллюстрации, изображения, таблицы, ссылки на сайты, словари, рубрикаторы, каталоги и т. д. Такими реакциями может сопровождаться как вопрос, так и ответ, что позволит пользователю лучше и быстрее сориентироваться в предметной области.

4.1. Основные положения ЛСС

Под логико-семантической сетью будем понимать множество вопросов, ответов и связей между ними, образующее целостную систему. Под целостностью ЛСС имеется в виду следующее:

1) множество «вопрос–ответ» относится к определенной теме предметной области;

2) множество «вопрос–ответ» иерархически упорядочено по принципу «от общего к частному»;

3) на нечетном уровне иерархии находятся вопросы, на четном уровне — ответы;

4) вопросы *i*-го уровня иерархии связаны только и только с ответами *i* + 1-го уровня;

5) вопросы *i*-го уровня связаны с ответами *i* – 1-го уровня;

6) вопрос *i*-го уровня семантически связан с ответами i+1-го уровня если удовлетворяет условиям А или В. В случае удовлетворения условию А имеет место конечная вершина; В случае удовлетворения условию В из данного ответа следуют вопросы *i* + 2-го уровня;

7) на *i* = 1 уровне находятся вопросы, которые раскрываются множеством ответов *i* = 2-го уровня, частично или полностью охватывающее тему предметной области;

8) на *i* = 3-м уровне находятся вопросы, которые восполняют и уточняют ответы *i* = 2-го уровня.

Единицей ЛСС является логическая связка «вопрос–ответ» и связанные с ними реакции. Вопросы всегда опираются на уже известное знание, выступающее их базисом и выполняющее роль предпосылки вопроса. Постановка вопроса и поиск информации для формирования ответа составляют вопросно-ответную логическую форму развития знаний. Таким образом, ЛСС «вопрос–ответ–реакция» можно представить в виде направленного графа (рис. 2). Суть излагаемого подхода состоит в том, что любая задача или научно-технический текст может быть представлен в виде логической последовательности вопросов и ответов, которая дополняется полезной информацией.

Вопрос — это выраженный в форме вопросительного предложения запрос, направленный на развитие (уточнение) или дополнение знаний.

Ответ — это реализация познавательной функции вопроса в форме вновь полученного суждения. При этом по содержанию и структуре ответ должен строиться в соответствии с поставленным вопросом. Лишь в этом случае ответ расценивается как релевантный, т. е. как ответ по существу поставленного вопроса.

Реакция — это смысловое описание вопроса и ответа, характеризующее предпосылки вопроса и область поиска ответа. Реакция позволяет учитывать и использовать дополнительные знания о предметной области.

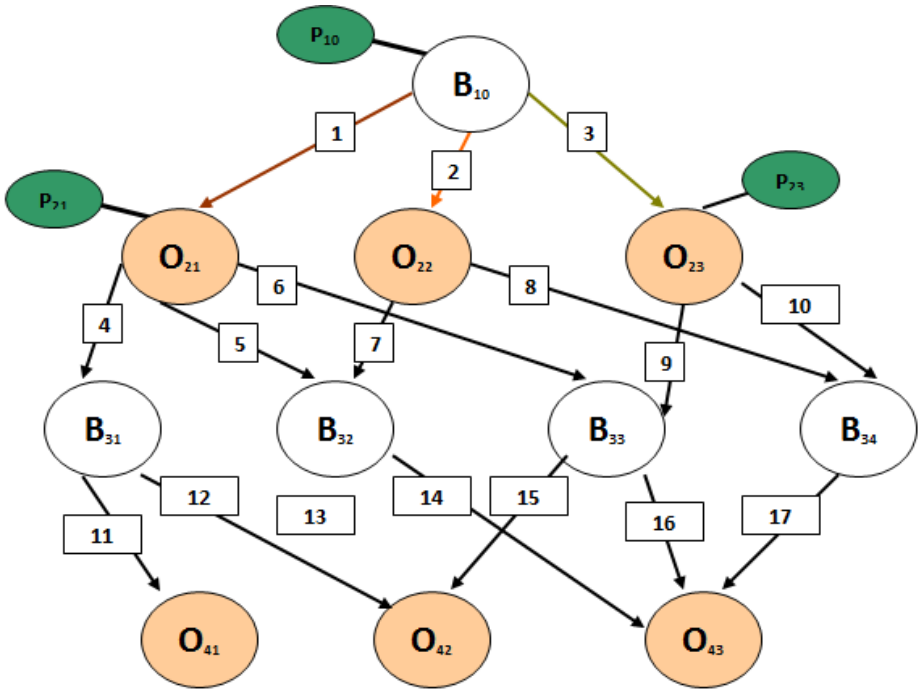


Рис. 2. Граф «вопрос–ответ–реакция»

Типы реакций

1. Реакции вопроса — это описание области предпосылки вопроса (для осознания обстоятельств и причин возникновения вопроса и дальнейшего установления смыслового соответствия с областью ответа). Реакция вопроса характеризует область вопроса — смысловое пространство, из которого аргументируется возникновение вопроса (хотя самой аргументации нет).

2. Реакции ответа — это описание области ответа (для осознания смысла вопроса и смысловой связи с ответом). Реакция ответа — смысловое пространство, имеющее связь с пространством вопроса, из которого следует ответ. Каркасом связи вопроса и ответа является ЛСС (рис. 3–5).

5. Электронные научные фонды & BigData

Сегодня в мире издается примерно 25 000 научных журналов, публикующих 1 млн статей в год, что соответствует ежедневному выпуску порядка 2700 публикаций [Редькина, 2010]. Для изучения таких объемов информации стандартными поисковыми технологиями специалисту потребуется значительное количество времени. Соответственно, важно иметь инструмент для эффективного исследования информации в массивах научных публикаций как основной продукции деятельности ученых и исследователей.

В электронных научных фондах количественные характеристики меняются быстро, качественные характеристики меняются медленнее, чем у сетевых систем. Качественное изменение научно-практических, технологических, технических знаний происходит скачкообразно и взаимно обусловлено и до определенной степени непредсказуемо. Можно считать, что на определенных временных периодах эти массивы знаний неизменны. В такие периоды исследователи работают традиционно, придерживаясь существующей парадигмы. А на периодах качественного перехода изменяются парадигмы и возникают новые знания, суть которых состоит в проявлении определенных косвенных качественных тупиковых (предельных) возможностей используемых знаний без желаемых эффектов. Такого рода деятельностью занимаются специалисты — эксперты, определяющие возможности будущих направлений научных, технологических и технических исследований и достижений.

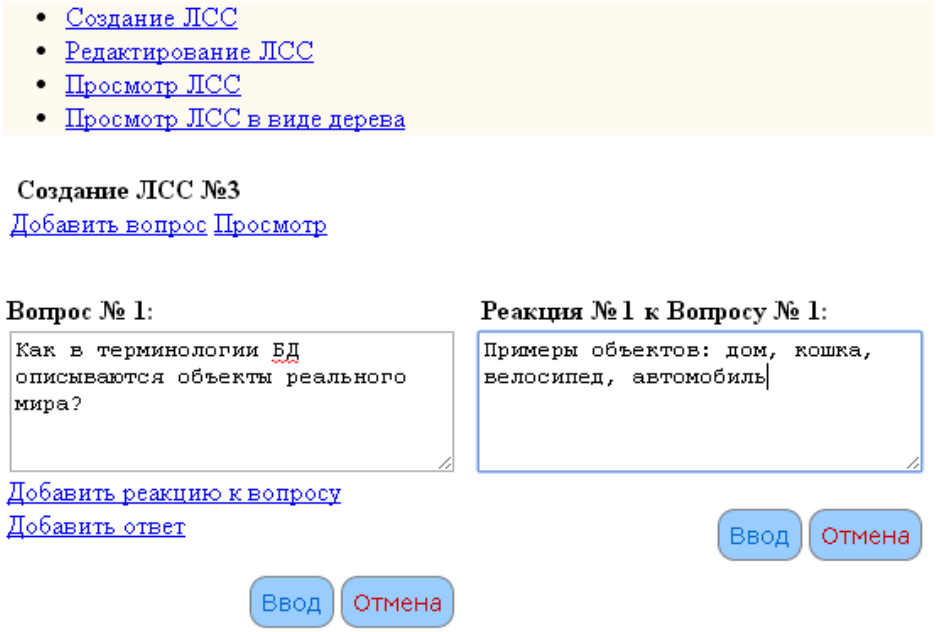


Рис. 3. АРМ аналитика. Режим создания ЛСС: формирование вопроса и реакции

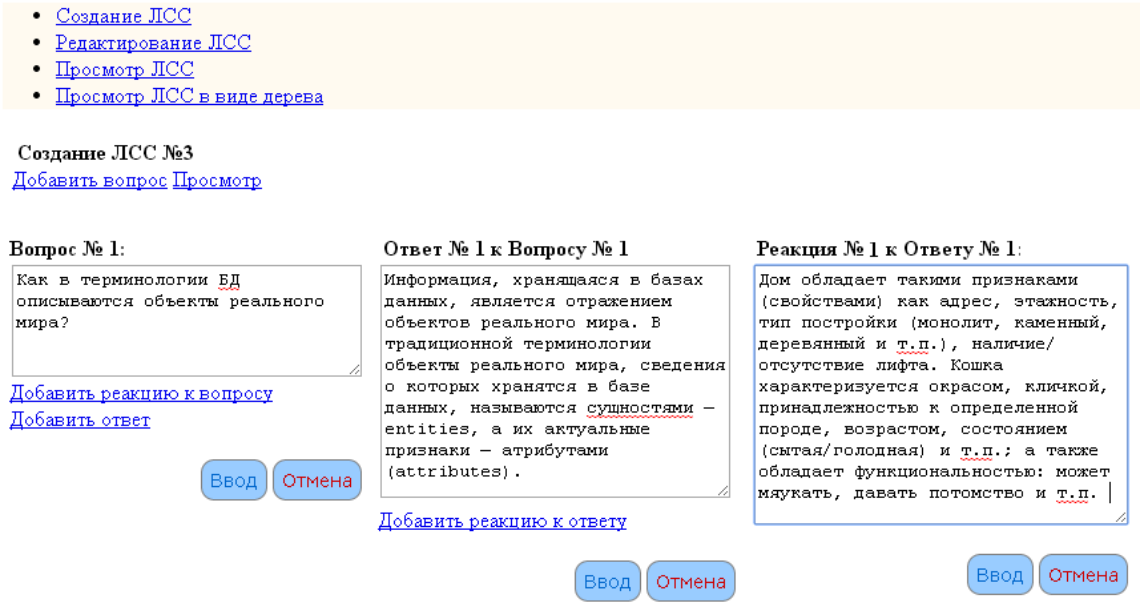


Рис. 4. АРМ аналитика. Режим создания ЛСС: формирование ответа и реакции

Но грань между пропущенными и новыми знаниями не очевидна. Ученые и исследователи часто прибегают к рассуждениям по аналогии. Аналогия — мощный инструмент в науке, обеспечивающий генерацию новых идей, гипотез и решений. По сути, аналогия является переносом, т. е. понятия, допущения, модели переносятся из одной области человеческого знания, где они показали свое эффективное применение, в другую область, в которой исследователь пытается разрешить некую проблемную ситуацию. Перенос идеи, уже успешно апробированной в другой области, подкрепляет убежденность ученых в эффективности используемых методов. С помощью аналоговых переносов устанавливаются взаимосвязи между новыми идеями и тем, что уже считается достоверным знанием. Есть мнение, что новое знание — это знание, которое

не имеет аналогии. Но история науки иллюстрирует, что самое радикальное новшество, как правило, проявляет неожиданные аналогии с уже имеющимися знаниями [Ивин, 2002].

Рис. 5. АРМ аналитика. Режим редактирования ЛСС: изменение текста реакции

Так, современная биология активно использует математический аппарат, в частности теорию дифференциальных уравнений, теорию вероятностей и статистику, теорию игр для формализации представлений о структуре, принципах функционирования и взаимоотношений живых организмов. Например, в популяционной динамике возникла математическая теория взаимодействия популяций одного трофического уровня (конкуренция) или разных трофических уровней (хищник–жертва), в которой сложные живые системы описываются при помощи систем обыкновенных дифференциальных уравнений. В математике дифференциальные уравнения не являются новым знанием, но перенос их в биологию позволяет не только моделировать, но и прогнозировать процессы в живых системах, что может привести биологов к новым знаниям — понимаю закономерностей соответствующих биологических процессов.

Логично предположить, что специалист, решающий некоторую профессиональную задачу в определенной области человеческого знания, будет действовать по принципу аналогового переноса, а именно, попытается выяснить, какие существуют успешные решения подобных задач. То есть первый этап его деятельности — информационный поиск. Из этого следует, что в системе поиска информации надо уделить внимание поиску пропущенных и/или новых знаний.

6. Направления исследования

Роль вопроса в процессе познания чрезвычайно важна. Совокупность вопроса и ответа формирует *единицу мысли*. В форме вопроса осуществляется постановка новых проблем в науке, с помощью вопросов люди получают новую информацию в социальной практике. Соответственно, любая задача может быть сформулирована в виде вопроса, а ее решение представлено как серия взаимосвязанных вопросов и ответов.

Разработка метода и механизма эффективного поиска множества релевантных ответов на вопрос включает:

1) разработку технологии формирования и поддержки *каталожной* службы информационного фонда, обеспечивающей эффективный поиск ответов на вопросы, на основе ЛСС «вопрос–ответ–реакция»;

2) создание инструментария (ПО) — АРМ аналитика для структурирования информационного фонда, предназначенного для создания и редактирования множества ЛСС.

Основой метода является способ описания научно-технической и образовательной информации множеством логико-семантических сетей «вопрос–ответ–реакция». Основой механизма поиска является способ движения по ЛСС, управляемый пользователем посредством выбора в ЛСС узлов — вопросов или ответов — на основе онтологической модели вопроса пользователя.

В рамках заданной проблемной темы предлагается технология формирования и поддержки *каталожной* службы, которая обеспечивает эффективный поиск ответов на вопросы. Стержнем такой *каталожной* службы является упорядоченное открытое множество логико-семантических сетей (ЛСС) «вопрос–ответ–реакция» [Добрынин, Филозова, 2010; Добрынин, Филозова, 2014]. С помощью специализированного навигатора специалист-профессионал, выполняющий поиск, может либо уточнять вопрос, либо его углублять, получая соответствующие связки «вопрос–ответ». Эта возможность достигается за счет введения Реакции, позволяющей учитывать и использовать дополнительные знания о предметной области. Тем самым пользователь от имеющихся знаний может получить расширенные знания, углубленные знания, уточненные знания или пропущенные знания. При этом за счет реакции пользователь может контролировать согласованность смыслового собственного понимания вопросов и ответов и понимания вопросов и ответов, заложенных в семантической поисковой системе. Поскольку система открытая, пользователь в процессе взаимодействия может уточнять и расширять саму ЛСС.

Данный подход позволяет заменить неопределенности, связанные с информационным поиском (когда не ясно точно, какая информация ищется и с какой целью) на более продуктивную технологию, ориентированную на пространство смыслов. Поисково-обрабатывающая система на основе ЛСС — это еще один путь совершенствования информационных технологий обработки информации.

7. Разработка АРМ аналитика для структурирования информационного фонда

Создание, наполнение и сопровождение такой информационной системы требует большой и серьезной работы, как технологической, так и организационной. Некоторую ее часть можно автоматизировать, предоставив соответствующее программное обеспечение аналитикам — АРМ аналитика для создания и редактирования множества ЛСС.

Формирование ЛСС базируется на методике анализа научных текстов, согласно которой текст исследуется каталогизатором с точки зрения [Filozova, 2012]:

1) смыслового соответствия заглавия и содержания;

2) набора фильтров:

F1 — общая часть: анализ проблемы, ее история, обзор, актуальность;

F2 — авторские понятия: вводимые авторами новые термины, обще употребляемые термины с авторской интерпретацией, сужающие семантику;

F3 — примеры и иллюстрации;

F4 — идея автора: описание и раскрытие основной авторской идеи;

3) формирования базовых вопросов, на которые отвечает текст.

На полученном таким образом материале далее строится ЛСС информационного ресурса: формулируются вопросы, ответы и реакции к ним.

В соответствии с вышесказанным АРМ аналитика предназначен для создания и редактирования множества ЛСС и обеспечивает следующую функциональность: 1) создание ЛСС (рис. 3, рис. 4); 2) редактирование ЛСС (рис. 5); 3) просмотр ЛСС (рис. 6).

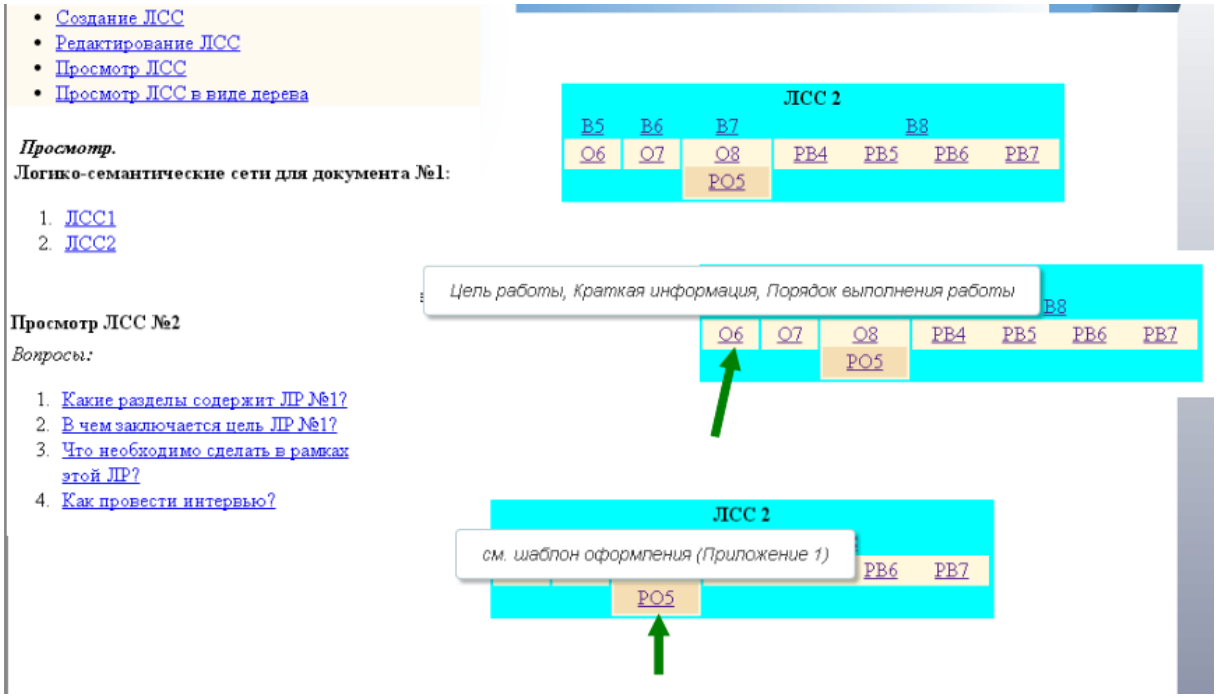


Рис. 6. АРМ аналитика. Режим просмотра ЛСС

Заключение

Одной из важных прикладных проблем эффективного поиска информации в условиях жестких временных ограничений — это проблема поиска новых и/или пропущенных, уточненных, углубленных знаний в точках бифуркации социотехнических систем. Информационные фонды (в том числе и электронные библиотеки), отражающие накопленные знания (теоретические, прикладные, прагматические), являются источниками генерации новых идей и формирования постановки и решения широкого спектра задач: исследование, экспертиза, инженерная задача, конструкторская задача и пр. Поиск необходимых для решения поставленной задачи знаний (в узком смысле) в некотором массиве информации может быть осуществлен посредством вопроса на языке предметной области.

В рамках заданной проблемной темы предлагается технология формирования и поддержки *каталожной* службы, которая обеспечивает эффективный поиск ответов на вопросы. Стержнем такой службы является упорядоченное открытое множество логико-семантических сетей «вопрос–ответ–реакция». Механизм навигации позволяет уточнять вопрос либо его углублять, получая соответствующие связки *вопрос–ответ*. Эта возможность достигается за счет введения *реакции,* позволяющей учитывать и использовать дополнительные знания о предметной области. Тем самым пользователь от имеющихся знаний может получить расширенные знания, углубленные знания, уточненные знания или пропущенные знания. При этом за счет *реакции* пользователь может контролировать согласованность смыслового собственного понимания вопросов и ответов и понимания вопросов и ответов, заложенных в семантической поисковой системе. Поскольку система открытая (пополняемая, изменяемая во времени), пользователь в процессе взаимодействия с системой может уточнять и расширять саму ЛСС. То есть пользователь при активном развитии системы становится соавтором смыслового пространства ЛСС. В этом состоит адаптация системы. Таким образом, поисково-обрабатывающая система на ос-

нове ЛСС — это еще один путь совершенствования информационных технологий обработки информации.

Список литературы

Большой энциклопедический словарь. 2-е изд., перераб. и доп. — М.–СПб.: Большая российская энциклопедия, 1998. — 1456 с.

Добрынин В. Н., Филозова И. А. Поиск в научной электронной библиотеке на основе логико-семантической сети «вопрос–ответ–реакция» // Труды XII Всероссийской научной конференции RCDL'2010 «Электронные библиотеки: перспективные методы и технологии, электронные коллекции». — Казань: Казанский университет, 2010. — С. 301–308. — Библиогр.: С. 308. — ISBN: 978-5-98180-838-8

Добрынин В. Н., Филозова И. А. Семантический поиск в научных электронных библиотеках // Информатизация образования и науки. — 2014. — № 2(22). — С. 110–110.

Ивин. А. А. Логика. Учебник для гуманитарных факультетов. — М.: ФАИР-ПРЕСС, 2002.

Касавин И. Т. Энциклопедия эпистемологии и философии науки. — М.: «Канон+», РООИ «Реабилитация», 2009.

Найдич А. Big Data: проблема, технология, рынок // КомпьютерПресс №1. 2012 [Электронный ресурс]. URL: http://www.compress.ru/article.aspx?id=22725&iid=1044

Редькина Н. С. Современное состояние и тенденции развития информационных ресурсов и технологий // Библиосфера. — 2010. — № 2. — С. 23–29.

Якшонок Г. Эффективный поиск и анализ научно-исследовательской информации в SciVerse: Scopus, Hub, ScienceDirect // МГИМО, 2012. [Электронный ресурс]. URL: http://mgimo.ru/files2/y03_2012/220642/MGIMO_March-2012.ppt

Filozova I. A. Technology of semantic structuring of the digital library content // Proceedings of the 5th International Conference "Distributed Computing and Grid-technologies in Science and Education". Dubna: JINR, 2012. — P. 117–122.

Hilbert M., López P. The World's Technological Capacity to Store, Communicate, and Compute Information // Science. — April. — 2011. — Vol. 332, no. 6025. — P. 60–65. — DOI: 10.1126/science.1200970.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 675–682

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Особенности методики обеспечения интероперабельности в грид-среде и облачных вычислениях

Е. Е. Журавлев[1,a], С. В. Иванов[2], А. А. Каменщиков[3], В. Н. Корниенко[3], А. Я. Олейников[3], Т. Д. Широбокова[3]

[1] Физический институт им. П. Н. Лебедева Российской академии наук, Россия, 119991, г. Москва, ГСП-1, Ленинский проспект, д. 53
[2] Российский новый университет, Россия, 105005, г. Москва, ул. Радио, д. 22
[3] Институт радиотехники и электроники им. В. А. Котельникова РАН, Россия, 125009, г. Москва, ул. Моховая, д. 11, корп. 7

E-mail: [a] derkien4life@yandex.ru

Получено 2 октября 2014 г.

В основу настоящей статьи положен доклад авторов, сделанный на конференции «GRID'2014», и содержит материалы, представляющие собой развитие наших результатов по проблеме интероперабельности в грид-среде и облачных вычислениях.

Ключевые слова: интероперабельность, грид, грид-среда, облачные вычисления, облака, методика, стандартизация

Aspects of methodology of ensuring interoperability in the Grid-environment and cloud computing

E. E. Zhuravlev[1], S. V. Ivanov[2], A. A. Kamenshchikov[3], V. N. Kornienko[3], A. Ya. Oleynikov[3], T. D. Shirobokova[3]

[1] P.N. Lebedev Physical Institute of the Russian Academy of Sciences, LPI, 53 Leninskij Prospekt, Moscow, 119991, Russia
[2] Russian New University, 22 Radio st., Moscow, 105005, Russia
[3] Kotel'nikov Institute of Radioengineering and Electronics of RAS, 11-7 Mokhovaya st., Moscow, 125009, Russia

Abstract. — The article is based on the report of the authors presented at the conference "GRID'2014" and contains materials including the development of our previous results on interoperability problem at grid and cloud environment.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 675–682 (Russian).

Введение

Грид-среда и среда облачных вычислений, состоящие из разнородных программно-аппаратных платформ, заведомо представляют собой гетерогенные среды, в которых неизбежно возникает проблема взаимодействия входящих в них систем, получившая название «проблемы интероперабельности». Актуальность данной работы обуславливается не только тем, что в нашей стране идет активное применение грид и облаков, но и тем, что вопросы развития принципов интероперабельности, стандартов и технологий открытых систем, а также развития технологий и стандартов грид включены в Программу фундаментальных исследований государственных академий наук в 2013–2020 г. Данная работа выполняется в рамках проекта РФФИ 12-0700261-а и Программы Президиума РАН № 14 и должна быть завершена к концу 2014 г.

Проблема интероперабельности

Согласно определению, приведенному в документе ISO/IEC FCD 24765-Systems and Software Engineering-Vocabulary, «интероперабельность — способность двух или более систем или элементов к обмену информацией и к использованию информации, полученной в результате обмена». Интероперабельность достигается за счет использования технологии открытых систем и согласованных наборов стандартов — профилей» [Технология…, 2004]. Построение профиля — лишь один из этапов определенной методики обеспечения интероперабельности. Проблема интероперабельности возникает в гетерогенной ИКТ-среде для информационных систем практически любого назначения и масштаба (от наносистем до грид-систем, систем облачных вычислений и сверхбольших систем — systems of systems). Эта проблема тем острее, чем выше уровень гетерогенности среды. Обеспечение интероперабельности — сложная научно-техническая задача, которой занимаются многие организации и исследователи; основными международными организациями в области грид-систем и систем облачных вычислений следует считать Open Grid Forum (OGF) и Open Cloud Consorcium (OCC). Этими вопросами занимается также IEEE.

Авторы ведут систематизированные работы по проблеме интероперабельности более 10 лет. Авторы выполнили ряд научных исследований по интероперабельности в области грид и облаков [Журавлев, Корниенко, Олейников, 2010; Журавлев, Корниенко, Олейников, 2012; Иванов, 2012] и разработали ряд национальных стандартов, указанных ниже.

Результаты работ авторов по интероперабельности

В своих работах по стандартизации авторы руководствовались федеральным законом РФ «О техническом регулировании», согласно которому на территории РФ должны применяться национальные стандарты, гармонизированные с международными. Процедура разработки национального стандарта достаточно сложная, занимает около 2 лет и требует согласования со всеми заинтересованными организациями [ГОСТ Р 1.2-2004, 2014].

В течение последних двух лет авторами была завершена разработка ряда документов в области интероперабельности, оформленных в виде национальных стандартов:

1) ГОСТ Р 55062-2012 «Информационные технологии. Системы промышленной автоматизации и их интеграция. Интероперабельность»;
2) ГОСТ Р 55022-2012 «Информационные технологии. Спецификация языка описания представления задач (JSDL). Версия 1.0»;
3) ГОСТ Р 55768-2013 «Модель открытой Грид-системы. Основные положения»;
4) ГОСТ Р «Информационные технологии. Архитектура служб открытой Грид-среды. Термины и определения», представляющий собой, по существу, глоссарий, касающийся интероперабельности грид среды (представлен к утверждению).

Первый из этих документов предназначен для систем широкого класса, документы 2 и 3 относятся к грид-системам. В настоящее время завершается НИР по созданию методики обеспечения интероперабельности в грид и облаках, которую планируется оформить как ГОСТ Р.

Методика обеспечения интероперабельности в грид и облаках

Методика (см. рис. 1), построена на основе единого подхода к обеспечению интероперабельности, зафиксированного в ГОСТ Р 55062-2012, и, по существу, использует принципы системной инженерии.

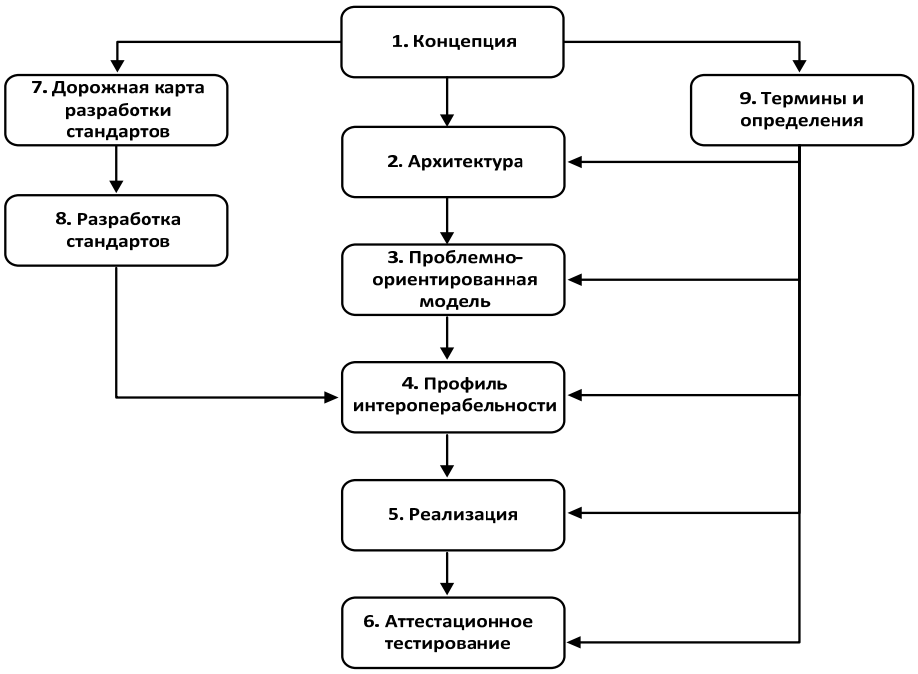


Рис. 1. Методика обеспечения интероперабельности грид и облаков

Методика содержит ряд этапов. К основным этапам относятся этапы 1–5, а к вспомогательным — этапы 6–9. Для обеспечения интероперабельности в случае грид и облаков должны быть выполнены все этапы, приведенные на рис. 1 с учетом специфики этих сред. Для этого приведем определения грид и облачных вычислений.

Согласно документу GFD 120 — Open Grid Services Architecture, разработанному международной организацией — Open Grid Forum:

«Грид — система, которая связана с интеграцией, виртуализацией и управлением услугами и ресурсами в распределенной, гетерогенной среде».

Для облаков приведем определение, данное The National Institute of Standards and Technology (NIST):

«Облачные вычисления — это модель предоставления повсеместного и удобного сетевого доступа по мере необходимости к общему пулу конфигурируемых вычислительных ресурсов (например, сетей, серверов, систем хранения, приложений и сервисов), которые могут быть быстро предоставлены и освобождены с минимальными усилиями по управлению и необходимостью взаимодействия с провайдером услуг (сервис-провайдером)».

Основные положения концепции

В области грид известны следующие концепции:

- Web Services Resource Framework (http://en.wikipedia.org/wiki/Web_Services_Resource_Framework);

- GridWise Interoperability Context-Setting Framework (http://www.caba.org/resources/Documents/IS-2008-30.pdf).

В области облачных вычислений известна концепция обеспечения семантической интероперабельности: Cloud4SOA (http://www.cloud4soa.eu/).

Согласно единому подходу к обеспечению интероперабельности, концепция интероперабельности должна содержать ряд основных положений. Рассмотрим эти положения (см. таблицу 1).

Таблица 1. Основные положения концепции грид и облаков

Грид	Облака
Предполагает объединение групп компьютеров и устройства хранения, позволяющее динамически выделять под определенные задачи необходимые ресурсы по мере появления потребности в них	Суть облачных вычислений заключается в удаленном предоставлении по требованию конечным пользователям динамического доступа к услугам (вычислительным ресурсам, приложениям, платформам и инфраструктурам) через локальную сеть или Интернет.
Интероперабельность в области грид означает способность двух или более грид-систем или их узлов обмениваться информацией и использовать эту информацию	Итероперабельность в области облачных вычислений означает способность двух или более облаков и их компонентов к обмену информацией и использованию информации, полученной в результате этого обмена
Цель обеспечения интероперабельности грид-систем — создание единой грид-среды, содержащей множество стандартизованных компонентов, благодаря которым возможно взаимодействие между отдельными частями грид-систем	Цель обеспечения интероперабельности облачных вычислений — создание единой облачной системы, раскрывающей истинный потенциал и преимущества облачных вычислений, заключающихся в возможности обмениваться понятными сообщениями, умении передавать и хранить данные в унифицированном формате, иметь возможность переносить образы виртуальных машин

Ключевая разница в концепциях — это способ предоставления вычислительных мощностей. В случае с грид это распределенная мощность и ресурсы. В случае с облаками эта мощность арендуется, и чем ее больше, тем больше приходится платить.

Архитектура

В таблицах 2 и 3 представлены архитектуры [Ian et al., 2008] для сравнения архитектур грид и облаков.

Таблица 2. Архитектура грид

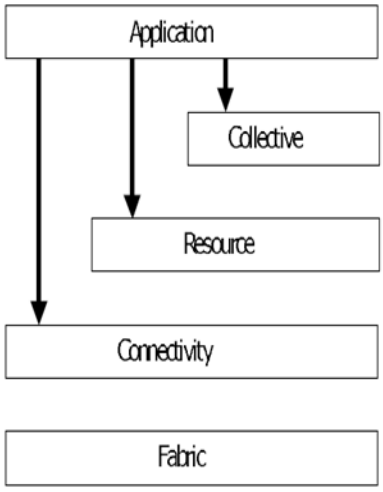
	Прикладной уровень (application) — включает любые пользовательские приложения, созданные поверх вышеупомянутых протоколов и API, и функционирует в средах виртуальной организации (ВО)
	Коллективный уровень (collective) — перехватывает взаимодействия между наборами ресурсов, службы каталогов, допускает контроль и открытие ресурсов ВО
	Уровень ресурса (resource) — определяет протоколы для публикации, открытия, согласования, контроля, учета и оплаты совместного использования операций на отдельных ресурсах
	Уровень связи (connectivity) — определяет базовую связь и протоколы аутентификации для простых и безопасных сетевых транзакций
	Структурный уровень (fabric) — обеспечивает доступ к различным типам ресурсов, таким как: вычисление, хранение, сетевой ресурс, репозиторий кода и т. д.

Таблица 3. Архитектура облачных вычислений

Application	Прикладной уровень (application) — содержит приложения, которые работали бы в облаке
Platform	Уровень платформы (platform) — прибавляет набор специализированных инструментов, промежуточного ПО и служб поверх объединенных ресурсов, чтобы обеспечить платформу разработки и/или развертывания (среда веб-хостинга, служба планирования)
Unified Resource	Объединенный уровень (unified resource) — содержит ресурсы, которые абстрагировались/инкапсулировались так, чтобы они могли быть представлены верхнему уровню и пользователям как интегрированные ресурсы (компьютер/кластер, логическая файловая система, система баз данных)
Fabric	Структурный уровень (fabric) — содержит необработанные аппаратные ресурсы, такие как вычислительные ресурсы, ресурсы хранения и сетевые ресурсы

Грид сосредоточен на интеграции существующих ресурсов с их аппаратным обеспечением, ОС, управлением локальными ресурсами и инфраструктурой безопасности. Грид определяет и обеспечивает набор стандартных протоколов, промежуточного ПО, инструментов и услуг, построенных на основе этих протоколов. Интероперабельность и безопасность — основа инфраструктуры грид. Облака обычно представляются как источник вычислительных ресурсов или ресурсов хранилища, к которым можно получить доступ через стандартные протоколы и абстрактные интерфейсы. Облака могут быть построены на многих существующих протоколах (Web Services, Web 2.0). Из сказанного следуют различия в представленных выше архитектурах.

Модели интероперабельности

Для грид

На рис. 2 представлена модель интероперабельности грид, зафиксированная в ГОСТ Р 55768-2013.

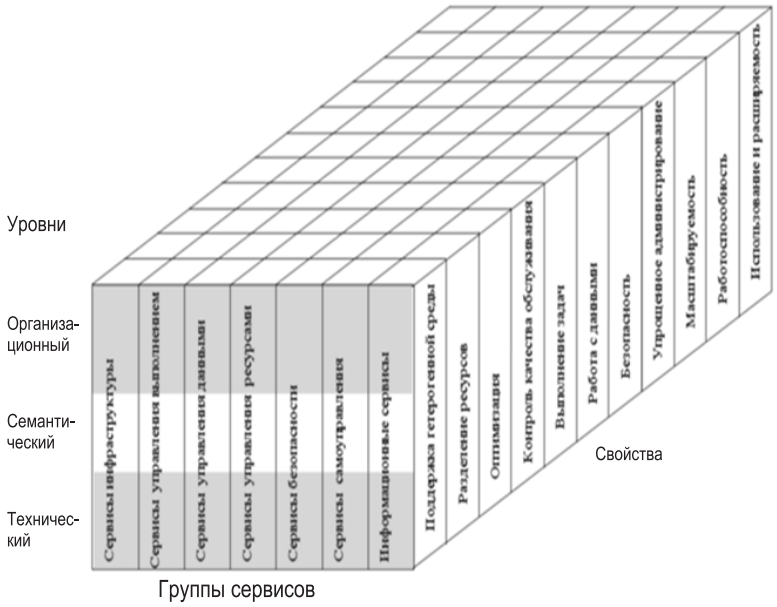


Рис. 2. Модель интероперабельности грид

На этом рисунке по оси абсцисс перечислены группы сервисов, обеспечивающих реализацию свойств грид-среды. По оси ординат представлены три уровня (технический, семантический и организационный) интероперабельности сервисов. По третьей оси перечислены свойства грид-системы [Журавлев и др., 2012].

Для облаков

На рисунке 3 представлена предлагаемая нами модель интероперабельности облаков [Журавлев, Иванов, Олейников, 2013].

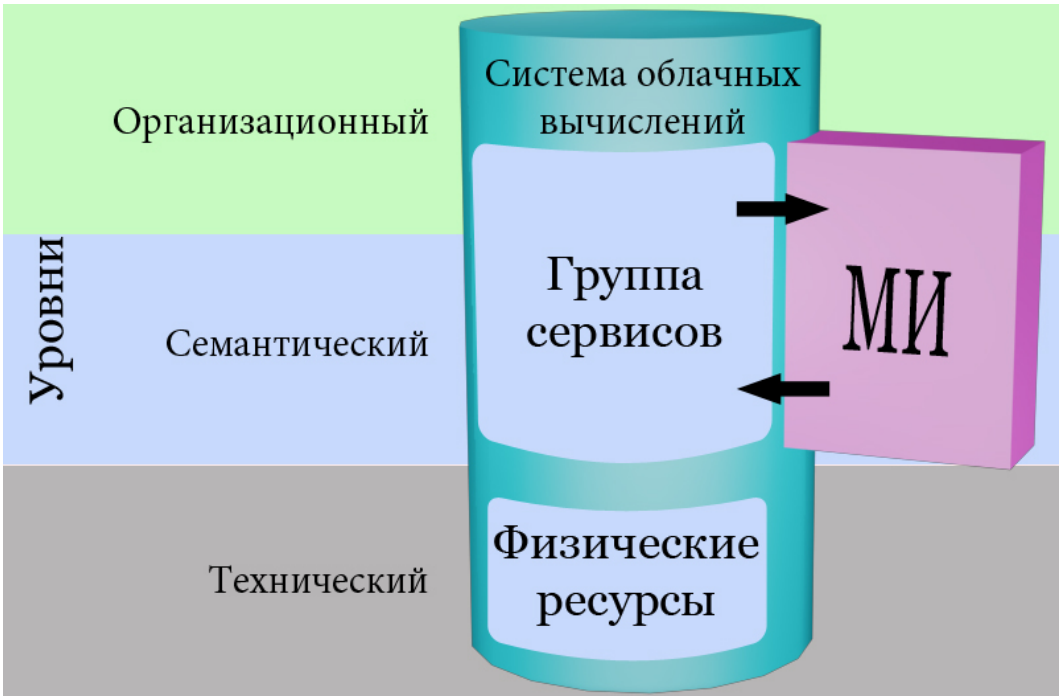


Рис. 3. Модель интероперабельности облаков

На рисунке изображена система облачных вычислений в виде цилиндра, которая содержит элемент «Группа сервисов», включающий сервисы, обеспечивающие реализацию свойств облаков. Также имеется элемент «Физические ресурсы», который представляет собой набор технических средств, используемых системой облачных вычислений. Представлены три уровня: технический, семантический и организационный интероперабельности сервисов. Ключевой особенностью модели является элемент «МИ» («Модуль интероперабельности»), способный управлять сервисами. Из модели видно, что МИ затрагивает два уровня — семантический и организационный.

Профиль интероперабельности грид и облаков

Как известно, профиль подразумевает согласованный набор стандартов, структурированный в терминах модели интероперабельности, который должен обновляться по мере актуализации входящих в него стандартов и может быть издан как отдельный нормативно-технический документ [ГОСТ Р 55062-2012, 2012]. Согласно модели интероперабельности в профиль интероперабельности должны войти стандарты технического, семантического и организационного уровня. Следует отметить, что согласно действующим правилам расположение стандартов на тех или иных уровнях интероперабельности требует коллективного обсуждения.

Данный этап в настоящее время проработан наиболее слабо. Авторы планируют использовать материалы организаций, приведенных в таблице 4.

Результаты работ этих организаций должны войти в профили интероперабельности.

Таблица 4. Организации и стандарты для грид и облаков

Грид	Облака
Open Grid Forum (OGF)	European Telecommunications Standards Institute (ETSI)
European Telecommunications Standards Institute (ETSI)	DMTF, Open Virtualization Format (OVF)
Distributed Management Task Force(DMTF)	OGF, Open Cloud Computing Interface (OCCI)
Internet Engineering Task Force (IETF)	SNIA, Cloud Data Management Interface (CDMI)
ITU Telecom munication Standardization Sector (ITU-T)	OASIS, Topology and Orchestration Specification for Cloud Applications (TOSCA); Cloud Application Management for Platforms (CAMP)
Organization for the Advancement of Structured Information Standards (OASIS)	IEEE P2301, Draft Guide for Cloud Portability and Interoperability Profiles
Storage Networking Industry Association (SNIA)	IEEE P2302, Draft Standard for Intercloud Interoperability and Federation
TeleManagement Forum (TMF)	ISO/IEC DIS 17226 «Information Technology» — Cloud Data Management Interface (CDMI) — SNIA
	ISO/IEC DIS 17963 «Web Services for Management» (WS — Management) DMTF
	ISO/IEC DIS 16680 information technology — The Open Group Service Integration Maturity Model(OSIMM)
	ISO/IEC 27001, ISO/IEC 27002
	ГОСТ Р 55022-2012, Информационная технология. Спецификация языка описания представления задач (JSDL). Версия 1.0

Дополнительные этапы

Для выполнения дополнительных этапов авторы ориентируются на следующие документы (разработки):

- ГОСТ Р (в стадии утверждения) «Информационные технологии. Архитектура служб открытой Грид-среды. Термины и определения» (см. п. 9 рис. 1), а также другие разработанные нами стандарты и, кроме того, зарубежные документы;
- SIENA European Roadmap on Grid and Cloud Standards for e-Science and Beyond (http://www.sienainitiative.eu/Repository/Filescaricati/8ee3587a-f255-4e5c-aed4-9c2dc7b626f6.pdf);
- NIST Cloud Computing Standards Roadmap Working Group (http://www.nist.gov/itl/cloud/upload/NIST_SP-500-291_Jul5A.pdf).

Заключение

Предложена методика обеспечения интероперабельности для грид и облаков. Методика использует принципы системной инженерии, базируется на едином подходе, разработанном авторами ранее и зафиксированном в ГОСТ Р 55062-2012. Существуют различия в содержании этапов методики для грид и облаков, которые должны найти отражение в составе профилей и реализации. Можно утверждать, что профиль грид и профиль облаков будут иметь общие стандарты на техническом уровне и отличающиеся — на более высоких уровнях. По завершению разработки методики целесообразно, чтобы она получила статус ГОСТ Р. Для этого обязательным условием должно служить активное коллективное обсуждение документа, и авторы приглашают всех заинтересованных лиц и организаций.

Список литературы

ГОСТ Р 1.2-2004 // ГОСТ Эксперт — база ГОСТов РФ. 2014. URL: http://gostexpert.ru/gost/gost-1.2-2004 (дата обращения: 20.09.2014).

ГОСТ Р 55062-2012. Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные полжения [электронный ресурс] // Центр открытых систем ИРЭ РАН. Создание и внедрение профилей на основе технологии открытых систем. — 2012. URL: http://opensys.info/files/data_20130514161145.pdf (дата обращения: 19.06.2013).

Журавлев Е. Е., Корниенко В. Н., Олейников А. Я. Вопросы стандартизации и обеспечения интероперабельности в GRID-системах // Распределенные вычисления и грид-технологии в науке и образовании: Труды IV Межд. конф. (Дубна, 28 июня – 3 июля, 2010 г.). — Дубна: ОИЯИ, 2010. — С. 364–372.

Журавлев Е. Е., Корниенко В. Н., Олейников А. Я. Исследование особенностей проблемы интероперабельности в GRID-технологии и технологии облачных вычислений // Распределенные вычисления и грид-технологии в науке и образовании: Труды V Межд. конф. (Дубна, 16–21 июля, 2012 г.). — Дубна: ОИЯИ, 2012. — С. 312–320.

Журавлев Е. Е., Корниенко В. Н., Олейников А. Я., Широбокова Т. Д. Модель открытой грид-системы [электронный ресурс] // Журнал радиоэлектроники (электронный журнал). — 2012. — № 12 // Сайт ИРЭ им. В. А. Котельникова РАН. URL: http://jre.cplire.ru, 2012, http://jre.cplire.ru/koi/dec12/3/text.html (дата обращения: 21.05.2014).

Журавлев Е. Е., Иванов С. В., Олейников А. Я. Модель интероперабельности облачных вычислений [электронный ресурс] // Журнал радиоэлектроники (электронный журнал). — 2013. — № 12 // Сайт ИРЭ им. В. А. Котельникова РАН. URL: http://jre.cplire.ru, 2013, http://jre.cplire.ru/jre/dec13/12/text.pdf (дата обращения: 21.05.2014).

Журавлев Е. Е., Иванов С. В., Каменщиков А. А., Олейников А. Я., Разинкин Е. И., Рубан К. А. Интероперабельность в облачных вычислениях [электронный ресурс] // Журнал радиоэлектроники (электронный журнал). — 2013. — № 9 // Сайт ИРЭ им. В. А. Котельникова РАН. URL: http://jre.cplire.ru, 2013. URL: http://jre.cplire.ru/jre/sep13/4/text.pdf (дата обращения: 21.05.2014).

Журавлев Е. Е., Корниенко В. Н. Тенденции в стандартизации интероперабельности в грид и облачных технологиях // Сборник трудов III Международной конференции «ИТ-Стандарт 2012». — Москва, МИРЭА. 16–17 октября 2012 г. С. 123–130.

Иванов С. В. Вопросы интероперабельности в облачных вычислениях // Распределенные вычисления и грид-технологии в науке и обрбазовании: Труды V Межд. конф. (Дубна, 16–21 июля 2012 г.). — Дубна: ОИЯИ, 2012. — С. 321–325.

Технология открытых систем / Под ред. А. Я. Олейникова. — М.: Янус-К, 2004.

Ian F., Yong Z., Ioan R., Shiyong L. Cloud Computing and Grid Computing 360-Degree Compared [электронный ресурс] // Microsoft Academic Search. URL: http://academic.research.microsoft.com/, 2008, http://academic.research.microsoft.com/Publication/50721241 (дата обращения: 27.06.2013).

Zhuravlev E. E., Olejnikov A. Y. The study of the interoperability problems in the grid-based technologies and cloud computing // Distributed Computing and Grid-technologies in Science and Education: Book of Abstr. of the 5th Intern. Conf. (Dubna, July 16–21 2012). — Dubna: JINR, 2012. — P. 173.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 683−689

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Использование облачных технологий CERN для дальнейшего развития по TDAQ ATLAS и его применения при обработке данных ДЗЗ в приложениях космического мониторинга

А. И. Казымов, В. М. Котов, М. А. Минеев, Н. А. Русакович, А. В. Яковлев

Лаборатория информационных технологий, Объединенный институт ядерных исследований,
Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6

E-mail: [a] mineev@jinr.ru

Получено 30 сентября 2014 г.

Облачные технологий CERN (проект CernVM) дают новые возможности разработчикам программного обеспечения. Участие группы TDAQ ATLAS ОИЯИ в разработке ПО распределенной системы сбора и обработке данных эксперимента ATLAS (CERN) связано с необходимостью работы в условиях динамично развивающейся системы и ее инфраструктуры. Использование облачных технологий, в частности виртуальных машин CernVM, предоставляет наиболее эффективные способы доступа как к собственно ПО TDAQ, так и к ПО, используемому в CERN: среда — Scientific Linux и software repository с CernVM-FS. Исследуется вопрос о возможности функционирования ПО промежуточного уровня (middleware) в среде CernVM. Использование CernVM будет проиллюстрировано на трех задачах: разработка пакетов Event Dump и Webemon, а также на адаптации системы автоматической проверки качества данных TDAQ ATLAS — Data Quality Monitoring Framework для задач оценки качества радиолокационных данных.

Ключевые слова: облачные технологий, виртуальные машины, обработка данных в области дистанционного зондирования Земли, ATLAS TDAQ, ПО промежуточного уровня

Using CERN Cloud Technologies for the Further ATLAS TDAQ Software Development and for its Application for the Remote Sensing Data Processing in the Space Monitoring Tasks

A. I. Kazymov, V. M. Kotov, M. A. Mineev, N. A. Russakovich, A. V. Yakovlev

Joint institute for nuclear researches, Laboratory of Information Technologies, 6 Joliot-Curie st., Moscow reg., Dubna, 141980, Russia

Abstract. — The CERN cloud technologies (the CernVM project) give a new possibility for the software developers. The participation of the JINR ATLAS TDAQ working group in the software development for distributed data acquisition and processing system (TDAQ) of the ATLAS experiment (CERN) involves the work in the condition of the dynamically developing system and its infrastructure. The CERN cloud technologies, especially CernVM, provide the most effective access as to the TDAQ software as to the third-part software used in ATLAS. The access to the Scientific Linux environment is provided by CernVM virtual machines and the access software repository — by CernVM-FS. The problem of the functioning of the TDAQ middleware in the CernVM environment was studied in this work. The CernVM usage is illustrated on three examples: the development of the packages Event Dump and Webemon, and the adaptation of the data quality auto checking system of the ATLAS TDAQ (Data Quality Monitoring Framework) for the radar data assessment.

Keywords: Cloud technologies, virtual machines, remote sensing data processing, ATLAS TDAQ, middleware

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 683–689 (Russian).

Введение

Появление в CERN облачных технологий (проект CernVM) [Segal et al., 2010] дало новые возможности разработчикам программного обеспечения. Участие группы TDAQ ATLAS ОИЯИ в разработке ПО распределенной системы сбора и обработке данных эксперимента ATLAS (CERN) [The ATLAS HLT …, 2002] связано с необходимостью работы в условиях динамично развивающейся системы и ее инфраструктуры. Использование облачных технологий, в частности виртуальных машин CernVM, предоставляет наиболее эффективные способы доступа как к собственно ПО TDAQ, так и к ПО, используемому в CERN: среда — Scientific Linux и software repository с CernVM-FS.

Использование CernVM будет проиллюстрировано на трех задачах: разработка пакетов Event Dump и Webemon, а также на адаптации системы автоматической проверки качества данных TDAQ ATLAS — Data Quality Monitoring Framework для задач оценки качества радиолокационных данных.

Для участия в процессе разработки и поддержки созданного ПО необходимо учитывать следующие ключевые моменты:

– наличие программной среды, аналогичной CERN (ОС Scientific Linux);

– доступ к репозиторию актуальных версий ПО, такого, какое имеется в CERN на AFS. В первую очередь это TDAQ Release и инструментальное ПО для разработчиков (Root, компиляторы и т. д.);

– для некоторых задач нужны администраторские права. Например, при разработке вебсервисов или при тестировании стороннего ПО в качестве кандидатов на использование в разрабатываемых проектах;

– производительность системы должна позволять вести отладку программ с графическим интерфейсом пользователя.

Для малых групп разработчиков узким местом является поддержка ПО, необходимого для работы. Также для работы в сети CERN нужен аккаунт, поэтому трудно подключить к работе, например, студентов.

1. Облачный сервис CERN

При создании стенда нами были широко использованы возможности, предоставляемые облачным сервисом CERN — CernVM [Segal et al., 2010]. Цель проекта CernVM — создание для пользователя универсальной, переносимой и легко конфигурируемой среды для обработки и анализа данных, а также для разработки ПО как локально, на компьютере пользователя или вычислительной ферме рабочей группы, так и в Grid (независимо от программной и аппаратной платформы пользователя, на которой запускается ПО CernVM). Основными компонентами, позволившими создать такую универсальную среду разработки, являются: виртуализация — возможность работы с виртуальной машиной, на которой проинсталлировано и сконфигурировано необходимое ПО, и доступ с помощью CernVM-FS [Segal et al., 2010] к репозиторию инструментального ПО, которое внутри CERN доступно пользователям AFS. Пользователь виртуальной машины CernVM получает ее уже настроенной под конкретные задачи (можно, например, при инициализации определять эксперимент, в котором пользователь участвует). На саму виртуальную машину устанавливается лишь необходимый минимум ПО. Доступ же к репозиторию ПО CERN, аналогичного по объему существующему на AFS серверах, реализуется с помощью CernVM-FS — файловой системы, которая позволяет ускорить, по сравнению с AFS, скорость доступа к файлам для удаленных пользователей за счет кэширования. CernVM-FS может использоваться независимо от виртуальных машин. На данный момент для скачивания доступны виртуальные машины CernVM (образы дисков) под управлением Scientific Linux для нескольких гипервизоров (VirtualBox, VMWare, Xen, KVM, Hyper-V Server) и в вариантах, оптимизированных для различных задач (Batch Node, Basic, Desktop, Head Node, BOINC) и различной

разрядности процессора. Объем скачиваемого образа VM невелик (x86_64 Desktop version 2.7.2 — 389 MB в архиве), но при кэшировании ПО он увеличивается (те машины, с которыми мы имели дело, после работы имели порядка 2 GB).

2. Сервисы информационного обмена TDAQ ATLAS

В качестве промежуточного ПО (middleware) в TDAQ ATLAS используются специально созданные для этой цели сервисы информационного обмена [Barczyk et al., 2002], разделенные по типу данных на Information server (простые параметры), OH server (гистограммы в формате ROOT), Event Monitoring (фрагменты событий и полные события) и Error Reporting server (предупреждения, сообщения об ошибках) (см. рис. 1).

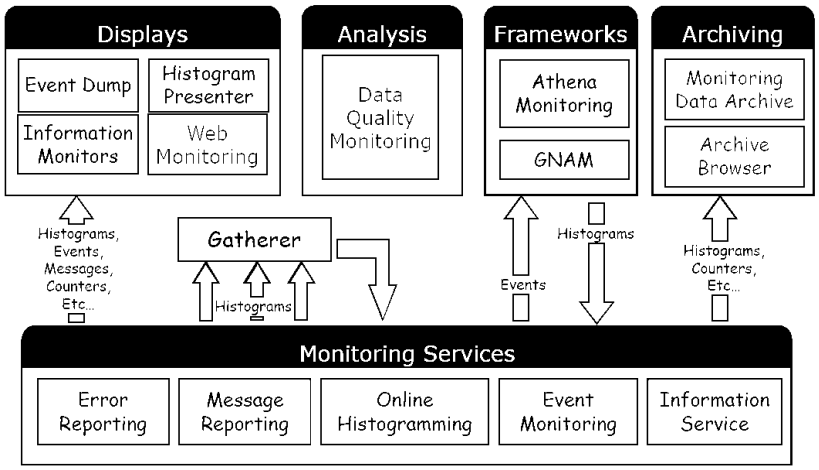


Рис. 1. Сервисы информационного обмена TDAQ ATLAS

Сервисы информационного обмена представляют отдельный интерес, так как могут быть использованы независимо друг от друга при создании, например, на их основе собственного ПО, не связанного с экспериментом ATLAS.

3. Описание архитектуры стендов

Первые тесты проводились на одной машине. Использовался гипервизор Virtual Box. Запускались две виртуальные машины CernVM, версия 2.6 Desktop. Для хранения файлов, содержащих настройки TDAQ Release (описание partitions в XML-формате, файл ipc.root.ref и другие), а также для хранения стороннего ПО (дистрибутивы некоторых программ, например NX-сервера) был создан общий диск (shared) средствами гипервизора.

Другой стенд (рис. 2) был создан на ферме Системы удаленного мониторинга реального времени (СУДРВ). Были использованы 3 реальные машины, на двух из которых стояли гипервизоры. На каждой запускалось до 3 виртуальных машин CernVM, общее дисковое пространство было создано с помощью Samba на третьем компьютере. На этом стенде тестировался случай, когда часть хостов виртуальны, а часть представлены реальными узлами фермы. В первую очередь на этих стендах нами проверялась работоспособность ПО промежуточного уровня (middleware), смогут ли компоненты TDAQ взаимодействовать друг с другом при запуске их на виртуальных машинах. Исследовались такие вопросы: можно ли с помощью IPC-пакета видеть запущенные partitions, серверы (IS, MRS, emon), доступна ли с других хостов опубликованная на этих серверах информация? Здесь проблем не возникло.

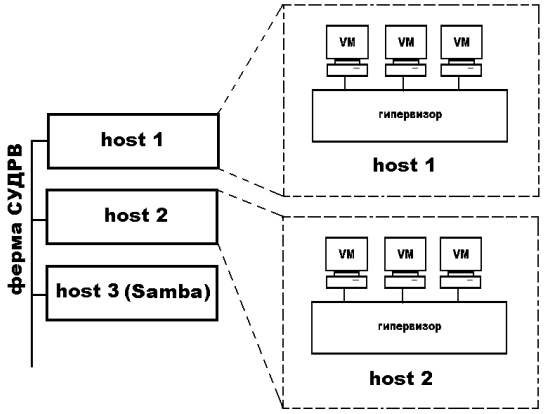


Рис. 2

4. Краткая характеристика разрабатываемого и поддерживаемого программного обеспечения

В этом разделе мы приведем краткое описание ПО, разработкой и поддержкой которого занимается наша группа.

4.1. Event Dump

Event Dump (ED) — программа мониторинга правильности форматирования необработанных физических данных (raw data) [Bee et al.]. Используя графический интерфейс пользователя, можно выбрать поставщика событий (уровень потока данных (dataflow) TDAQ ATLAS) и получить от него физическое событие. Далее имеется возможность просмотреть структуру события. Для того чтобы сделать запрос серверу мониторинга событий (Event Monitoring (emon) server) на получение события, необходимо задать параметры Sampling Address и Selection Criteria. Перед началом работы ED создается перечень доступных поставщиков событий, которые группируются по partition (программно-аппаратные комплексы, являющиеся частью детектора ATLAS, которые могут функционировать независимо друг от друга) в виде дерева.

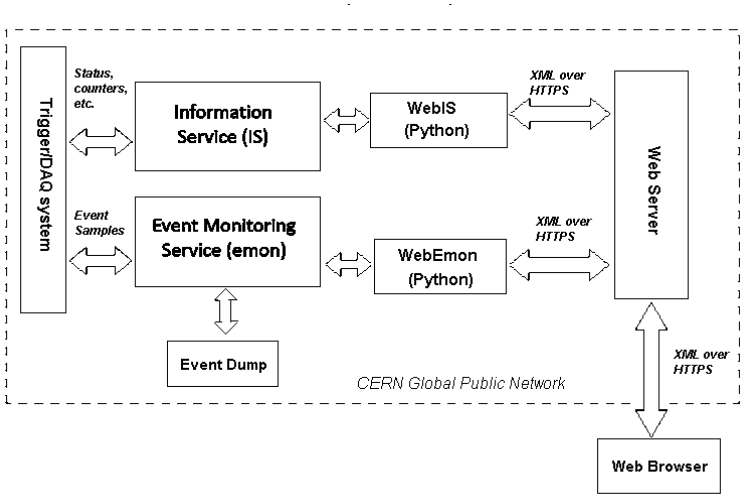


Рис. 3. Организация мониторинга событий (raw data)

Event Dump позволяет просматривать параметры заголовков и структуру событий. Для специфических параметров, характерных только для некоторых подсистем ATLAS, есть возможность написания кода обработки и добавления в ED в виде добавочных, созданных самими пользователями, панелей (User Panels).

4.2. Webemon

Event Dump имеет ряд ограничений в использовании, т.к. он сильно интегрирован в среду TDAQ ATLAS и для его работы необходимы сервисы информационного обмена — IPC и Event Monitoring. Не так давно в CERN был реализован веб-сервис — пакет WebIS [Gabriel, 2010], предоставляющий возможность пользователям, находящимся вне CERN и вне среды TDAQ, получать данные, опубликованные на IS- и OH-серверах. На его базе нами был написан сервер Webemon, который позволяет получать события (в формате raw-data [Bee et al.]) от сервера Event Monitoring, работающего в CERN. Само событие из формата raw-data при передаче конвертируется в XML-формат, а параметры (Sampling Address и Selection Criteria) передаются серверу Webemon в виде HTTP-запроса. Аналогично ED, есть возможность получения списка partitions и поставщиков событий. Эти списки также приходят в виде XML-документа. Сервер Webemon и конвертор формата raw-data в XML написаны на языке Python. Так как клиенты сервера Webemon уже не нуждаются при работе в среде TDAQ ATLAS, то это открывает широкие возможности для написания пользователями своих приложений, которые будут работать вне CERN.

4.3. Data Quality Monitoring Framework

Онлайн-мониторинг анализа качества физических данных в автоматическом режиме реализован в TDAQ ATLAS в виде системы Data Quality Monitoring Framework (DQMF) [Cuenca Almenar et al., 2010]. DQMF показал высокую эффективность при работе с физическими данными, сейчас нами проводятся исследования вопроса применимости данного ПО для анализа качества радиолокационных данных. В данной работе мы рассмотрим лишь аспект возможности работы с DQMF в облачной среде CERN. Работа DQMF (рис. 4) требует наличия сервисов информационного обмена TDAQ (IS, OH и т. д.), на которых публикуются данные (DQParameters), оценка которых производится системой. Описание конфигурации содержится в Cofiguration DB (структура DQRegions, описание используемых для оценки качества алгоритмов).

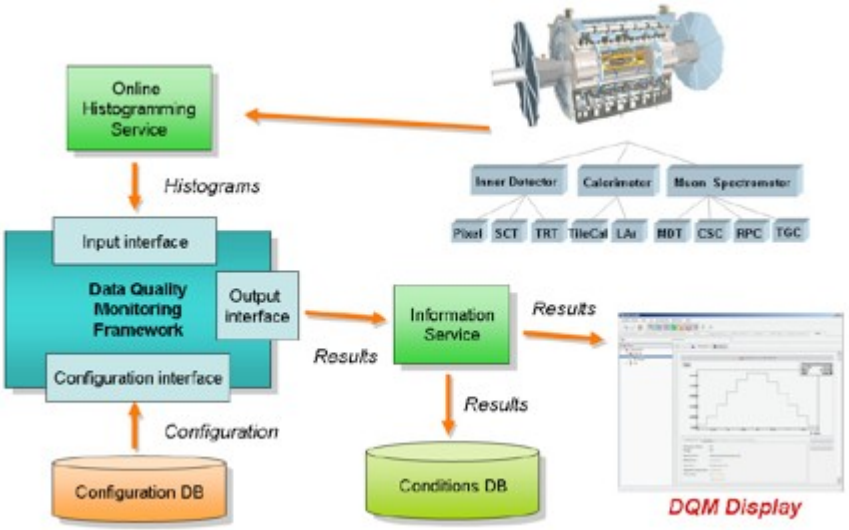


Рис. 4. Data Quality Monitoring Framework

На стендах была проверена работоспособность DQMF на тестах, которые содержит сам пакет (тестовая partition оценки качества данных, запускающая минимальный набор компонент инфраструктуры TDAQ, необходимый для функционирования DQMF).

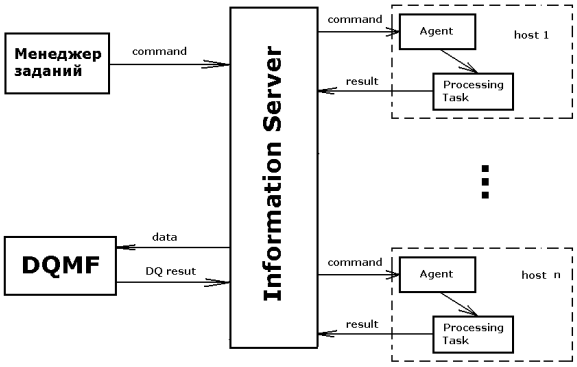


Рис. 5

Для использования DQMF в задачах ДЗЗ необходимо подготовить параметры (DQParameters) и опубликовать их на серверах информационного обмена, что можно сделать также с помощью сервисов TDAQ. Для этого предполагается создать систему, состоящую из менеджера задач (task manager) и агентов, которые будут запускаться на узлах вычислительной фермы (рис. 5). Менеджер задач по описанию конфигурационной базы будет высылать команды агентам через Information Service. Агенты после старта на узле подписываются на изменения своего статуса на IS. Как только статус обновляется, агент выполняет команду (команда представляет собой параметр статуса). В данный момент идет отладка агентов. При отладке важно иметь возможность запуска агентов, менеджера задач и IS на разных машинах. Такая задача решается с помощью виртуальных машин CernVM, которые работали на ферме удаленного мониторинга ATLAS СУДРВ (Системы удаленного мониторинга реального времени).

5. Заключение

Было проведено тестирование ПО TDAQ ATLAS на работоспособность в облачной среде CERN. Была показана возможность работы промежуточного программного обеспечения в этой среде. Работа с виртуальными машинами позволила смоделировать распределенную систему. Последним тестом был запуск DQM partition — независимая часть ПО TDAQ ATLAS, запускающая инфраструктуру DQMF. Замена AFS на CernVM-FS при удаленной работе не сказался на работоспособности DQMF и сервисов информационного обмена. Мы планируем использовать сервисы, предоставляемые CernVM в дальнейшем более широко, например, для исследования вопроса о том, как организовать работу DQMF для задач ДЗЗ с GRID. Пока наиболее вероятными путями является использование подхода, предложенного для High-Level Trigger ATLAS [Korcyl et al., 2008], или использование компонента Co-Pilot [Buncic, Harutyunyan, 2011]. Также мы планируем начать работать с виртуальными машинами в облачной инфраструктуре ЛИТ ОИЯИ [Облачная инфраструктура …, 2014], что даст больше возможностей как в аппаратном обеспечении, так и в организации самой работы с виртуальными машинами.

Список литературы

Облачная инфраструктура ЛИТ ОИЯИ. URL: https://cloud.jinr.ru/terms_rus.html

Gabriel A. Die Fernüberwachung des ATLAS Level-1 Kalorimeter Triggers // Diplomarbeit, Heidelberg, 2010. URL: http://www.kip.uni-heidelberg.de/Veroeffentlichungen/download.php/4875/ps/DiplomGAnders.pdf

Barczyk M. et al. Online Software Architecture. 2002. URL: http://atlas-onlsw.web.cern.ch/Atlas-onlsw/documents/doc/OnlSWArchitecture_03.pdf

Bee C. et al. The raw event format in the ATLAS Trigger & DAQ. URL: ATL-D-ES-0019, https://edms.cern.ch/document/445840/3

Buncic P., Harutyunyan A. Portable Analysis Environment using Virtualization Technology (WP9). Co-Pilot: The Distributed Job Execution Framework. 2011. URL: http://cernvm.cern.ch/portal/d/copilot/copilot-tech-rep-0.4.13.pdf

Cuenca Almenar C. et al. ATLAS Online Data Quality Monitoring // 17th IEEE NPSS Real Time Conference, Lisbon, Portugal, 24–28 May 2010, 5 p., ATL-DAQ-PROC-2010-015. URL: https://cds.cern.ch/record/1274856

Korcyl K. et al. The ATLAS Experiment On-line Monitoring and Filtering as an Example of Real-Time Application // Computer Science. — 2008. — Vol. 9. — P. 77–86. URL: http://journals.bg.agh.edu.pl/COMPUTER/2008/cs2008-07.pdf

Segal B. et al. LHC Cloud Computing with CernVM // Proceedings of the XIII. International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT10), Jaipur, 2010, PoS ACAT(2010)004. URL: http://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=93

The ATLAS HLT, DAQ & DCS Technical Design Report // ATLAS TDR-016, 12 November 2002. URL: http://atlas-proj-hltdaqdcs-tdr.web.cern.ch/atlas-proj-hltdaqdcs-tdr/

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 691−698**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.414.23, 519.876.5

Синтез процессов моделирования и мониторинга для развития систем хранения и обработки больших массивов данных в физических экспериментах

В. В Кореньков[a], А. В. Нечаевский, Г. А. Ососков, Д. И. Пряхина, В. В. Трофимов, А. В. Ужинский

Лаборатория информационных технологий, Объединенный институт ядерных исследований, Россия, 141980, Московская обл., г. Дубна, ул. Жолио-Кюри, д. 6

E-mail: [a]symsim@jinr.ru

Получено 10 февраля 2015 г.

Представлена новая система моделирования грид и облачных сервисов, ориентированная на повышение эффективности их развития путем учета качества работы уже функционирующей системы. Результаты достигаются за счет объединения программы моделирования с системой мониторинга реального (или модельного) грид-облачного сервиса через специальную базу данных. Приведен пример применения программы для моделирования достаточно общей облачной структуры, которая может быть также использована и вне рамок физического эксперимента.

Ключевые слова: имитационное моделирование, грид, облака, хранение данных, оптимизация, мониторинг

Synthesis of the simulation and monitoring processes for the development of big data storage and processing facilities in physical experiments

V. V. Korenkov, A. V. Nechaevskiy, G. A. Ososkov, D. I. Pryahina, V. V. Trofimov, A. V. Uzhinskiy

Joint institute for nuclear researches, Laboratory of Information Technologies, 6 Joliot-Curie st., Moscow reg., Dubna, 141980, Russia

Abstract. — A new grid and cloud services simulation system are described. This system is developed in LIT JINR Dubna and focused on improving the efficiency of the grid-cloud systems development by using work quality indicators of some real system to design and predict its evolution. For these purpose the simulation program are combined with real monitoring system of the grid-cloud service through a special database. An example of the program usage to simulate a sufficiently general cloud structure, which can be used for more common purposes, is given.

Keywords: grid computing, cloud computing, data storage, monitoring, optimization, simulation

Работа выполнена при поддержке Гранта РФФИ № 14-07-00215.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 691–698 (Russian).

Введение

В различных областях деятельности существует множество вычислительных систем различного масштаба для обработки информации. Наибольший интерес представляют системы, обрабатывающие сверхбольшие объемы данных. В качестве примера можно привести WLCG — грид-систему распределенной обработки данных Большого адронного коллайдера (БАК). По доступной статистике, объем информации, сохраненный и обработанный в четырех экспериментах БАК на протяжении первого прогона (RAN 1), составил сотни петабайт.

В 2015–2020 гг. на экспериментах БАК ожидается увеличение объема данных и неизбежный переход к грид-облачным комплексам. Это необходимо для потенциально новой физики, но сталкивается с новыми серьезными требованиями к компьютингу БАК, а именно:

- значительное увеличение вычислительных мощностей и сетевых ресурсов хранения данных;
- необходимость доступа к данным из грид и облаков;
- активное использование распределенных параллельных вычислений;
- совершенствование кодов программ анализа и моделирования.

Столь быстрое развитие распределенных вычислительных систем требует непрерывного моделирования всех процессов хранения, передачи и анализа данных.

В настоящее время при проектировании грид-систем используется подход, когда задача создания модели и формулировки рекомендаций по построению выполняется однократно при проектировании системы. В предыдущей работе авторов [Кореньков и др., 2013] описана программа моделирования, основанная на использовании языка GridSim [GridSim..., 2012] и алгоритмов планирования потока заданий ALEA [Klusacek et al., 2008]. Для запуска программы требуется задать состав и топологию центров обработки моделируемой грид-структуры, а также распределение ресурсов между заданиями. После этого программа выполняет имитационное моделирование процессов прохождения сгенерированного набора заданий через грид-структуру. В качестве результатов вычисляются временные оценки искомых параметров потока заданий.

Однако эксперименты продолжаются годами и десятилетиями, одновременно с эксплуатацией системы происходит ее развитие, не только качественное, но и количественное. При эволюции WLCG произошло качественное изменение систем хранения информации, а вместо планируемых трех уровней обработки данных появилось четыре. Таким образом, даже при значительных усилиях, вложенных на этапе проектирования в понимание конфигурации систем и их количественных характеристик, невозможно развивать систему без дополнительных исследований. Разработчики и эксплуатирующие организации сталкиваются с проблемой прогнозирования поведения системы после проведения планируемых модификаций.

Моделирование системы позволяет ответить на ряд вопросов. При создании распределенной системы требуется принять решения по архитектуре инфраструктуры, количеству ресурсных центров, объему требуемых ресурсов. Кроме того, необходимо обеспечить достаточную пропускную способность, решить проблемы сохранности данных (устойчивость к повреждениям и удалениям) на протяжении всего жизненного цикла проекта, обеспечить распределение ресурсов между различными группами пользователей, выбрать алгоритмы обработки и запуска задач и многое другое.

Таким образом, требуется создание методологии и программного окружения, позволяющего моделировать системы на постоянной основе, прогнозировать поведение системы при значительных изменениях.

Объединив моделирование и мониторинг в рамках одного программного пакета, можно добиться существенного снижения эксплуатационных затрат и вложений в увеличение мощности с целью сохранения скорости получения результата экспериментов при постоянном повышении потока данных.

Выбор средств моделирования

Говоря о том, какую технологию моделирования применить, следует учесть, что возможность применения аналитических моделей для рассматриваемых задач ограничена по следующим соображениям. Существует несколько подходов при аналитическом моделировании грид- и облачных систем, которые можно сгруппировать в два типа:

– система рассматривается как многоканальная система массового обслуживания с состояниями, управляемыми марковским процессом, с ограничениями на распределения входных потоков и на дисциплины обслуживания, вызванными теоретическими предпосылками;

– система рассматривается как динамическая стохастическая сеть, описываемая системами уравнений, позволяющими учитывать как маршрутизацию, так и распределение ресурсов в сети, причем изучению подлежат равновесные и неравновесные состояния сети [Попков, 2003].

Оба подхода выдают результат моделирования, как правило, в виде асимптотических распределений и в силу ограниченных теоретических предпосылок не могут быть применены для моделирования конкретных сложных компьютерных сетей многоуровневой архитектуры с реальными распределениями входных потоков заданий, сложной многоприоритетной дисциплиной их обслуживания и динамическим распределением ресурсов. Поэтому мы считаем правильным использовать имитационное моделирование.

На сегодняшний день существуют различные программные инструменты имитационного моделирования грид-систем и облаков [Nechaevskiy, Korenkov, 2009; Кореньков, Муравьев, Нечаевский, 2014]. Например, GridSim — библиотека классов, предназначенных для построения модели грид-системы. Она, в свою очередь, построена на стандартной библиотеке SimJava, с помощью которой можно моделировать поток дискретных событий во времени. Однако моделирование облачных вычислительных центров этой системой не предусмотрено.

Облачные вычислительные центры могут быть определены как тип параллельных и распределенных систем, состоящих из набора взаимосвязанных и виртуальных компьютеров, которые предоставлены динамически как один или несколько объединенных вычислительных ресурсов на основе соглашения об уровне обслуживания через договор между провайдером сервиса и потребителем. Для моделирования облачных инфраструктур существуют различные программные продукты, например CloudSim, iCanCloud, CReST (см. обзор в [Кореньков, Муравьев, Нечаевский, 2014]). Эти программные пакеты позволяют создавать модели облачных систем с определенной функциональностью и конфигурацией. Готовая модель запускается на прогон с модельным потоком заданий, в результате чего системы моделирования предоставляют статистическую информацию по наиболее важным характеристикам: время выполнения задач, жизненный цикл виртуальных машин, использование ресурсов. Эти системы моделирования ориентированы на моделирование определенного уровня облака. Функциональность CloudSim позволяет наиболее подробно моделировать уровни SaaS и IaaS. Для анализа работы уровней PaaS и SaaS облачной инфраструктуры можно использовать iCanCloud. Разработку дата-центра с минимальными затратами электроэнергии и эффективным охлаждением можно реализовать в CReST, который подробно моделирует PaaS-уровень. Однако представленные системы моделирования рассчитаны на решение своих узкоспециализированных задач и не обладают набором функций для полноценного моделирования облачных вычислительных центров для хранения и обработки данных физических экспериментов.

Предлагаемое нами программное решение основано на расширении классов GridSim и их объединении в программу, которая моделирует обработку потока заданий грид-облачной структурой, обладающей заданными ресурсами и дисциплиной их резервирования и использования. В последнее время выдвигается идея интеграции в грид-инфраструктуры центров, построенных по принципу облачных вычислений, а также реализации служб грид на оборудовании «облачных» центров. Поэтому методы и средства, которые разрабатываются в рамках про-

екта, допускают моделирование объединения в грид-инфраструктуру центров, имеющих облачную архитектуру.

Описание подхода к моделированию

Постоянное развитие современных грид-систем требует непрерывных корректировок большинства параметров моделирования. Это необходимо для прогнозирования поведения системы при значительных ее изменениях. Для корректировки параметров предлагается использовать статистику эксплуатации системы, получаемую на основе имеющихся программных средств ее мониторинга.

В связи с этим возникают две проблемы:

1) обеспечение совпадения исходных данных для модели с реальными;

2) проверка адекватности моделирования, т. е. доказательство того, что моделирование произведено корректно и поведение модели не отличается от поведения реальной системы.

Наш подход состоит в следующем.

1. Если речь идет о модернизации существующей установки обработки, то использовать подходящие накопленные данные. К примеру, в проекте WLCG имеются как глобальные, так и специализированные под конкретные эксперименты системы мониторинга и аккаунтинга. При этом результаты моделирования обработки потока заданий должны совпадать в пределах погрешности с результатами мониторинга прохождения того же потока заданий в системе.

2. Для новых установок эта проблема разрешается выдвижением гипотез о типах потоков входной информации, их параметрах, и процедурах их обработки с последующим моделированием как самих входных потоков, так и процессов их обработки. Такие гипотезы можно сформулировать на основании данных мониторинга подобных систем (оценивая интенсивность и основные характеристики потоков заданий и файлов). Обработка результатов моделирования заключается в анализе распределения времени событий, которые генерируются при обработке входного потока данных. Затем эти распределения сравниваются с результатами, полученными из мониторинга существующей системы.

Таким образом, модель должна рассматриваться как неотъемлемая часть системы обработки данных, а данные мониторинга — как входные для моделирования. Это позволит принимать более обоснованные проектные решения при развитии системы.

Предлагаемый нами подход состоит в интеграции средств мониторинга процессов прохождения задач и передачи данных с возможностями имитационного моделирования. В рамках этой концепции техническое решение проверяется на модели прежде, чем обсуждается его практическая реализация. В идеале процесс принятия решений по развитию вычислительной установки должен выглядеть следующим образом: данные мониторинга реальной грид-системы поступают в базу данных, далее на основе данных мониторинга пользователь задает входные параметры модели и потока заданий, модель обрабатывает задания и возвращает пользователю результат для дальнейшего анализа. Центральным компонентом такой процедуры принятия решения является имитационная модель вычислительной структуры, в которую в качестве входной поступает информация, накопленная в ходе мониторинга существующей установки и модифицированная в соответствии с представлениями о том, как она будет меняться.

Для реализации модели потребовались существенные изменения GridSim:

- введены классы, описывающие специфическое для облачных центров хранилище информации;
- входной поток заданий формируется через базу данных;
- принцип обмена данными изменен с симуляции пакетов на симуляцию передачи файла;
- обработка результатов моделирования вынесена за рамки программного пакета.

Для иллюстрации возможностей разработанной программы SyMSim (Synthesis of Monitorng and Simulation) ниже приведен пример ее применения для оптимизации простой облачной структуры.

Пример использования программы SyMSim

Объектами моделирования являются вычислительные установки, предназначенные для обработки информации объемом до десятков петабайтов в год, который производят ускорители заряженных частиц, например LHC-CMS [CMS detector, 2015], LHC-Atlas [ATLAS detector, 2015] и находящиеся в процессе создания или проектирования FAIR-PANDA [PANDA..., 2015], BES-III [BESIII..., 2015], NICA-MPD [Сисакян, Сорин, 2011]. Как показал многолетний опыт работы центров разных уровней для распределенных вычислений и хранения данных, объединенных в систему WLCG, единственный способ хранения объемов информации, производимых такими детекторами, является использование роботизированных библиотек. Данные затем обрабатываются на фермах, включающих тысячи процессоров. Предполагается, что моделируемая структура предназначена для обработки данных физического эксперимента, но другие структуры, связанные с хранением и обновлением больших массивов цифровой информации, также могут быть смоделированы.

Итак, рассматривается модель реализации облачной структуры, предназначенной для хранения данных в роботизированной библиотеке с тысячами кассет с магнитными лентами, из загрузчиков-драйвов которых робот автоматически достает требуемые ленты и устанавливает в одно или несколько устройств чтения–записи. Схема прохождения задания через систему моделирования SyMSim представлена на рис. 1. Задание начинает выполняться, если есть свободный слот-процессор и все файлы доступны на дисковом хранилище облака. Если файл хранится в роботизированной библиотеке, задание резервирует слот, но выполнение задерживается до момента его загрузки на диск. Процесс перемещения файла из библиотеки в дисковое хранилище включает в себя операцию помещения ленточного картриджа на драйв, которую выполняет рука робота, монтирования файловой системы картриджа на драйве и записи файла на диск.

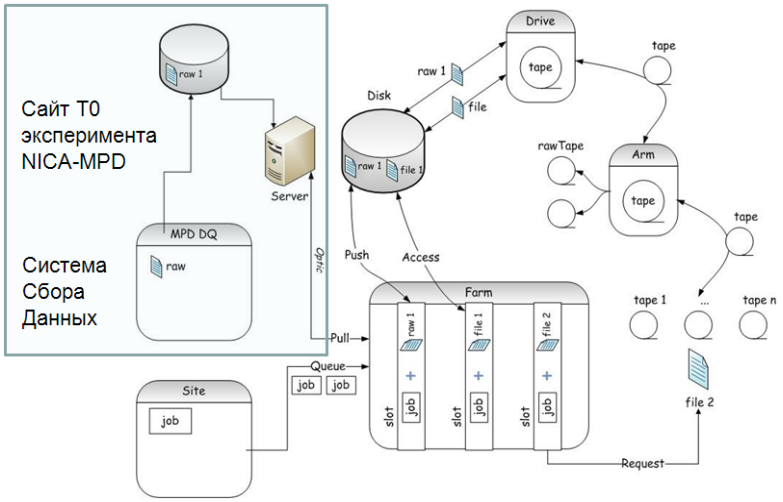


Рис. 1. Схема прохождения заданий через SyMSim

Проектируемая структура состоит из ленточного робота IBM 3500 [IBM..., 2015], массива ленточных картриджей (220 лент) и кластера из 100 абстрактных процессоров. Было взято 9 драйвов LTO6 для пула из 150 лент для работы с файлами и 2 драйва LTO6 для пула из 70 лент для записи файлов с «сырыми» данными (RAW), технические параметры которых соответствуют реальным. Дисковый пул T1 — 590 ГБ. Канал связи — 10 Гб/с. Имитация заключалась в моделировании прохождения 1000 заданий по этой структуре. Поток заданий генерируется на основе распределений, полученных при статистическом анализе данных, доступных для эксперимента Atlas.

Рассмотрим на этом примере возможности модели.

Моделирование нагрузки на процессоры показано на рис. 2, а ее равномерность во времени — на рис. 3. Равномерность загрузки зависит от равномерности поступления заданий, настроек очередей, системы вторичной памяти и т. д. Все астрономическое время выполнения пакета разбито на одинаковые интервалы.

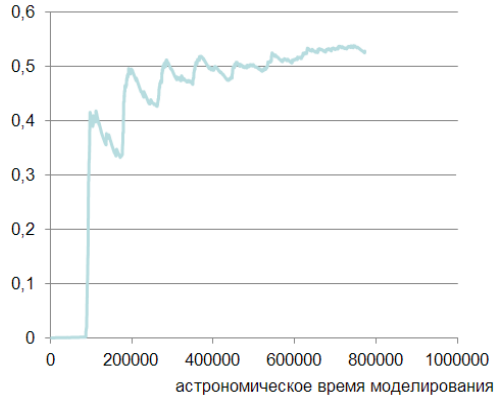


Рис. 2. Отношение затраченного процессорного времени к максимально возможному затраченному к этому моменту

Рис. 3. Количество заданий, завершившихся во временном интервале

Размер дискового буфера. Одно из ограничений модели: одно задание может требовать только один файл. Однако разные задания могут требовать один из файлов, которые уже подгружены. Алгоритм сборщика мусора заимствован из dCache [dCache..., 2015]. Возникает вопрос, хватит ли нам буфера. На рис. 4 мы видим, что буфер используется от 60 % до 80 %.

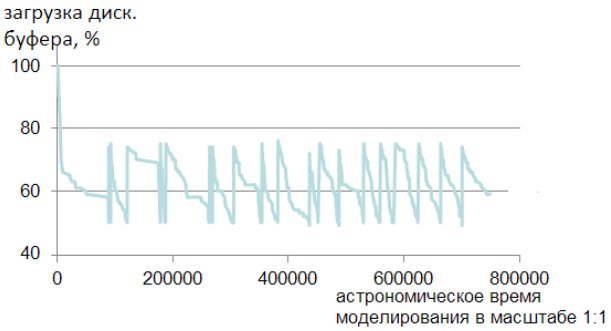


Рис. 4. Процент загрузки дискового буфера

Нагрузка на руку робота показана на рис. 5. Нагрузка определяется следующим образом: исходя из среднего времени движения руки 6 с, вычисляется максимальное количество движений за временной промежуток. Загрузка — отношение количества движений при моделировании к максимально возможному количеству движений. Оказалось, что рука робота будет загружена не более чем на 4 %. Причем вначале нагрузка на руку возрастает, потому что идет массовая загрузка файлов, которые требуются для выполнения задач, а потом нагрузка снижается, потому что часть файлов уже есть в буфере.

Таким образом, для данной интенсивности потока задач мощность вычислительных узлов достаточна, если SLA допускает прохождение данного пакета за $0.8*10^6$ с. Рука робота загружена слабо, т. е. можно использовать шкафы высокой плотности. Диски объемом 0,5 ТБ достаточны для поставленных задач.

Результаты моделирования по критерию минимального времени прохождения задания при достаточно высокой загрузке процессоров могут служить обоснованием выбора конфигурации облачного кластера и аргументом в пользу покупки или отклонения более дорогого оборудова-

ния, хотя не следует забывать, что на выбор конфигурации также влияют и другие соображения: надежность, перспективы развития, величина резерва и т. д.

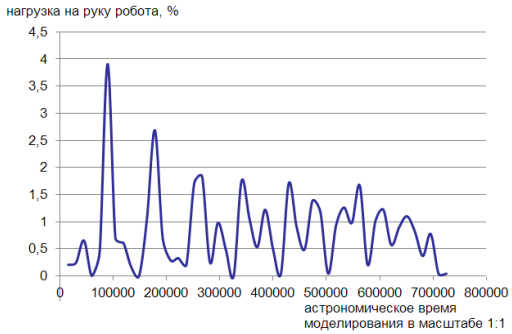


Рис. 5. Нагрузка на руку робота в процентах от максимальной

Заключение

Разработка грид-облачных систем сбора, передачи и распределенной обработки информации требует тщательного моделирования, эффективность которого зависит от наличия динамических данных о качестве работы уже функционирующей инфраструктуры. Авторами разработана система моделирования облачных вычислительных центров SyMSim. Предложенный подход к моделированию и анализу вычислительных грид-облачных структур основан на учете данных их мониторинга, используемых затем для динамической коррекции параметров моделирования. Новизна подхода в моделировании состоит в соединении моделирования и мониторинга в рамках одного проекта.

В силу общности своей реализации разработанная программа моделирования SyMSim может быть также применена для решения более широкого класса задач проектирования виртуальных центров обработки и хранения больших массивов данных, не ограниченных областью физического эксперимента.

Список литературы

Кореньков В. В., Муравьев А. Н., Нечаевский А. В. Пакеты моделирования облачных инфраструктур // Системный анализ в науке и образовании. — 2014. — Вып. 2. Дубна.

Кореньков В. В., Нечаевский А. В., Трофимов В. В. Разработка имитационной модели сбора и обработки данных экспериментов на ускорительном комплексе НИКА // Информационные технологии и вычислительные системы. — 2013. — No. 4. — С. 37–44.

Попков Ю. С. Макросистемы и grid-технологии: моделирование динамических стохастических сетей // Проблемы управления. — 2003. — № 3.

Сисакян А. Н., Сорин А. С. Многоцелевой Детектор – MPD для изучения столкновений тяжелых ионов на ускорителе NICA (Концептуальный дизайн-проект), версия 1.4. [электронный ресурс] — 2011. — URL: http://nica.jinr.ru/files/CDR_MPD/MPD_CDR_ru.pdf (дата обращения: 02.02.2015).

ATLAS detector [электронный ресурс] // CERN, Switzerland. — 2014. — URL: http://home.web.cern.ch/about/experiments/atlas (дата обращения: 19.01.2015).

BESIII — веб-портал проекта [электронный ресурс] // Beijing, China. — 2014. — URL: http://bes3.ihep.ac.cn/ (дата обращения: 12.01.2015).

CMS detector [электронный ресурс] // CERN, Switzerland. — 2014. — URL: http://home.web.cern.ch/about/experiments/cms (дата обращения: 17.01.2015).

dCache — веб-портал проекта [электронный ресурс] // URL: http://www.dcache.org (дата обращения: 26.01.2015).

GridSim: A Grid Simulation Toolkit For Resource Modelling And Application Scheduling For Parallel And Distributed Computing [электронный ресурс] // The University of Melbourne, Australia. — 2015. — URL: http://www.gridbus.org/gridsim (дата обращения: 06.01.2015).

IBM System Storage TS3500 Tape Library [электронный ресурс] // IBM. — 2014. — URL: http://www.ibm.com/ru/servers/storage/tape/ts3500 (дата обращения: 16.01.2015).

Klusacek D., Matyska L., and Rudova H. Alea — Grid scheduling simulation environment // In 7th International Conference on Parallel Processing and Applied Mathematics (PPAM 2007). Vol. 4967 of LNCS, pages 1029–1038. Springer, 2008.

Nechaevskiy A. V., *Korenkov V. V.* DataGrid simulation packages // System Analysis in Science and Education (Online), ISSN: 2071-9612, Issue 1, 2009.

PANDA — веб-портал проекта [электронный ресурс] // Darmstadt, Germany. — 2014. — URL: http://www-panda.gsi.de/ (дата обращения: 15.01.2015).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 699–705

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.021

Параллельное представление локального элиминационного алгоритма для ускорения решения разреженных задач дискретной оптимизации

Д. В. Лемтюжникова

Вычислительный центр имени А. А. Дородницына Российской академии наук,
Россия, 119333, г. Москва, ул. Вавилова, д. 40

E-mail: darabbt@gmail.com

Получено 19 марта 2015 г.

Алгоритмы декомпозиции являются методами решения NP-трудных задач дискретной оптимизации (ДО). В этой статье демонстрируется один из перспективных методов, использующих разреженность матриц, — локальной элиминационный алгоритм в параллельной интерпретации (ЛЭАП). Это алгоритм структурной из декомпозиции на основе графа, который позволяет найти решение поэтапно таким образом, что каждый последующих этапов использует результаты предыдущих этапов. В то же время ЛЭАП сильно зависит от порядка элиминации, который фактически является стадиями решения. Также в статье рассматриваются древовидный и блочный тип распараллеливания для ЛЭАП и необходимые процессы их реализации.

Ключевые слова: дискретная оптимизация, добровольные вычисления, локальный элиминационный алгоритм, параллельные вычисления, разреженные задачи, элиминационное дерево

Parallel representation of local elimination algorithm for accelerating the solving sparse discrete optimization problems

D. V. Lemtyuzhnikova

Dorodnicyn Computing Centre of RAS, 40 Vavilov st., Moscow, 119333, Russia

Abstract. — The decomposition algorithms provide approaches to deal with NP-hardness in solving discrete optimization problems (DOPs). In this article one of the promising ways to exploit sparse matrices — local elimination algorithm in parallel interpretation (LEAP) are demonstrated. That is a graph-based structural decomposition algorithm, which allows to compute a solution in stages such that each of them uses results from previous stages. At the same time LEAP heavily depends on elimination ordering which actually provides solving stages. Also paper considers tree- and block-parallel for LEAP and required realization process of it comparison of a several heuristics for obtaining a better elimination order and shows how is related graph structure, elimination ordering and solving time.

Keywords: discrete optimization, volunteer computing, local elimination algorithm, parallel computing, sparse problems, elimination tree

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 699–705 (Russian).

Модели многих задач, возникающих на практике, можно представить в виде задач дискретной оптимизации (1): задачи теории расписаний, задачи маршрутизации, задачи оптимизации производства и многие другие. Сложность таких задач заключается в том, что они зависят от большого числа дискретных переменных, поэтому для их решения естественно использовать алгоритмы, которые не обладают экспоненциальной сложностью, но при этом дают точное решение. В этом представляют интерес локальные элиминационные алгоритмы (ЛЭА), которые разбивают большую задачу на подзадачи и элиминируют переменные, понижая тем самым величину перебора [Щербина, 2008].

$$\begin{gathered} F(X) \to \max; \\ AX \sigma B; \\ X \in \{0;1\}; \\ \sigma = \{<, \le, =, >, \ge\}. \end{gathered} \tag{1}$$

Для работы ЛЭА кроме данных самой задачи — целевой функции, матрицы и вектора ограничений, — необходим порядок элиминации, который указывает, каким образом будут исключаться переменные в задаче. Для этого нужно построить элиминационное дерево (ЭД) задачи, которое строится из гиперграфа задачи при помощи некоторого критерия элиминации, связанного с характером задачи. Перспективным алгоритмом построения ЭД является древовидная декомпозиция [Щербина, 2007].

Для разветвленного ЭД имеет смысл применять технологию распараллеливания [Лемтюжникова, 2013], так как подзадачи, находящиеся на одной высоте дерева, являются независимыми. Такой вид распараллеливания будем называть *древовидным распараллеливанием*.

Рассмотрим подзадачу, соответствующую вершине ЭД. Она имеет блочную структуру, так как первичная задача была разреженной. Блоки такой подзадачи слабосвязные, поэтому имеет смысл разбивать такие блоки, перебирая переменные, которые являются сепараторами этих блоков. Такой вид распараллеливания будем называть *блочным*.

ЛЭА дважды проходит по ЭД. Прямой ход ЛЭА решает подзадачи и сохраняет промежуточные решения, а обратный ход ЛЭА анализирует и собирает решения подзадач.

Рассмотрим пример решения разреженной задачи при помощи параллельного локального элиминационного алгоритма блочного типа. Решим имеющую БД-структуру задачу ЦЛП, которой соответствует дерево инцидентности блоков (рис. 1):

$$z = 2x_1 + x_2 + 3x_3 + 2x_4 + x_5 + 3x_6 + 4x_7 + x_8 + 2x_9 + 3x_{10} + 5x_{11} + 6x_{12} + 2x_{13} + 3x_{14} \to \max$$

при ограничениях

$$\begin{gathered} B_1 : x_4 + x_5 + 4x_6 + 2x_7 \le 4; \\ B_2 : 3x_1 + 2x_2 + 2x_3 + 3x_4 \le 6; \\ B_3 : 3x_7 + 8x_8 + 3x_9 + x_{10} \le 8; \\ B_4 : x_9 + 2x_{11} + 3x_{12} \le 4; \\ B_5 : x_{10} + x_{13} + 2x_{14} \le 2; \\ x_j = 0,1,\ j = 1,2,...,14. \end{gathered} \tag{2}$$

Перейдем к решению задачи.

I. Строим дерево инцидентности блоков.

Дерево инцидентности блоков строится следующим образом. Вершины дерева — сами блоки, то есть каждой вершине соответствует некоторое ограничение задачи. Две вершины соединяются ребром в том случае, когда в соответствующих ограничениях существуют одни и те же переменные. Затем выбирается корневая вершина. Ее необходимо выбрать таким образом, чтобы дерево состояло из минимального количества слоев. Дерево инцидентности блоков соот-

ветствует ЭД. Так как подзадачи, соответствующие некоторым вершинам и находящиеся на одном слое являются независимыми (B_4 и B_5 на третьем слое, а также B_2 и B_3 на втором слое), имеет место *древовидное распараллеливание*.

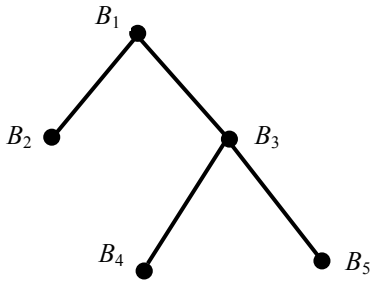


Рис. 1. Дерево инцидентности блоков

II. Решим параллельно подзадачи, соответствующие вершинам дерева инцидентности на последнем уровне.

1. Определимся с рассматриваемым блоком, множеством индексов переменных, принадлежащих одновременно текущему блоку и блоку, который является его предком и множеством потомков.

Начнем решение со слоя $v = L = 3,$ количество вершин на этом слое $l_3 = 2.$ Рассмотрим задачи, соответствующие блоку $r = 4,$ то есть четвертому ограничению, и блоку $r = 5$ — пятому ограничению. Они связаны ребром с вершиной $B_3,$ причем общей переменной ограничений B_3 и B_4 является $x_9,$ а для ограничений B_3 и B_5 — $x_{10}.$ Значит, сепараторами блоков $r = 4$ и $r = 5$ являются $S_{34} = \{9\}$ и $S_{35} = \{10\}$ соответственно. Множество потомков для обеих вершин пусто.

2. Параллельно построим функцию для задач, соответствующих данным блокам.

Построим функцию для задачи $D_4,$ соответствующей блоку $r = 4.$

Для этого смотрим, какие еще переменные помимо x_9 присутствуют в этом ограничении $B_4 : x_9 + 2x_{11} + 3x_{12} \le 4.$ Это переменные x_{11} и $x_{12}.$

Построим функцию для задачи $D_5,$ соответствующей блоку $r = 5.$

Для этого смотрим, какие еще переменные помимо x_{10} присутствуют в этом ограничении $B_5 : x_{10} + x_{13} + 2x_{14} \le 2.$ Это переменные x_{13} и x_{14}

Теперь посмотрим на целевую функцию нашей задачи

$$z = 2x_1 + x_2 + 3x_3 + 2x_4 + x_5 + 3x_6 + 4x_7 + x_8 + 2x_9 + 3x_{10} + 5x_{11} + 6x_{12} + 2x_{13} + 3x_{14} \to \max.$$

Обратим внимание на коэффициенты при вышеназванных переменных x_{11} и $x_{12}.$

Отсюда выписываем целевую функцию для задачи $D_4 : f_{D_4}\left(X_{S_{34}}\right) = \max\{5x_{11} + 6x_{12}\}.$

Обратим внимание на коэффициенты при вышеназванных переменных x_{13} и $x_{14}.$

Отсюда выписываем целевую функцию для задачи $D_5 : f_{D_5}\left(X_{S_{35}}\right) = \max\{2x_{11} + 3x_{12}\}.$

3. Определим ограничения для этих функций.

Общая переменная x_9 переносится в правую сторону неравенства, ограничение задачи D_4: $2x_{11} + 3x_{12} \le 4 - x_9; x_{11}, x_{12} = \{0;1\}.$

Общая переменная x_{10} переносится в правую сторону неравенства, ограничение задачи D_5: $x_{13} + 2x_{14} \le 2 - x_{10}; x_{13}, x_{14} = \{0;1\}.$

4. Рассмотрим возможные значения общей переменной для каждой задачи. Обратим внимание, что блоки подзадач D_4 и D_5 слабосвязные, поэтому разобьем эти блоки, перебирая сепараторы, используем *блочное распараллеливание*.

Пусть $x_9 = 0$.	Пусть $x_9 = 1$.	Пусть $x_{10} = 0$.	Пусть $x_{10} = 1$.
Ограничение для данной задачи			
$2x_{11} + 3x_{12} \le 4 - x_9$ имеет вид:		$x_{13} + 2x_{14} \le 2 - x_{10}$ имеет вид	
$2x_{11} + 3x_{12} \le 4 - 0$, то есть $2x_{11} + 3x_{12} \le 3$.	$2x_{11} + 3x_{12} \le 4 - 1$, то есть $2x_{11} + 3x_{12} \le 3$.	$x_{13} + 2x_{14} \le 2 - 0$, то есть $x_{13} + 2x_{14} \le 2$.	$x_{13} + 2x_{14} \le 2 - 1$, то есть $x_{13} + 2x_{14} \le 1$.
Переменные x_{11}, x_{12} могут принимать значения		Переменные x_{13}, x_{14} могут принимать значения	
$\{x_{11} = 1; x_{12} = 0\}$, $\{x_{11} = 0; x_{12} = 1\}$, $\{x_{11} = 0; x_{12} = 0\}$.	$\{x_{11} = 1; x_{12} = 0\}$, $\{x_{11} = 0; x_{12} = 1\}$, $\{x_{11} = 0; x_{12} = 0\}$.	$\{x_{13} = 1; x_{14} = 0\}$, $\{x_{13} = 0; x_{14} = 1\}$, $\{x_{13} = 0; x_{14} = 0\}$.	$\{x_{13} = 1; x_{14} = 0\}$, $\{x_{13} = 0; x_{14} = 0\}$.
Подставляя эти варианты в задачу D_4, получим, что максимальное значение для этой задачи $f_{D_4}(x_9) = \max\{5x_{11} + 6x_{12}\}$		Подставляя эти варианты в задачу D_5, получим, что максимальное значение для этой задачи $f_{D_5}(x_{10}) = \max\{2x_{13} + 3x_{14}\}$	
$f_{D_4}(x_9) = f_{D_4}(0) = 6$ при переменных $X_{D_4}(0) = \{x_{11} = 0, x_{12} = 1\}$.	$f_{D_4}(x_9) = f_{D_4}(1) = 6$ при переменных $X_{D_4}(1) = \{x_{11} = 0, x_{12} = 1\}$.	$f_{D_5}(x_{10}) = f_{D_5}(0) = 3$ при переменных $X_{D_5}(0) = \{x_{13} = 0, x_{14} = 1\}$.	$f_{D_5}(x_{10}) = f_{D_5}(1) = 2$ при переменных $X_{D_5}(1) = \{x_{13} = 1, x_{14} = 0\}$.

III. Решим задачи, соответствующие другим слоям дерева инцидентности.

1. Перейдем на слой $v = 2$, количество вершин на этом слое $l_2 = 2$. Рассмотрим задачи, соответствующие блоку $r = 2$, то есть второму ограничению, и блоку $r = 3$ — третьему ограничению. Их вершина-предок $p_2 = 1$, то есть сепараторами являются $S_{12} = \{4\}$ и $S_{13} = \{7\}$ соответственно. Множество потомков для вершины B_2 пусто, а для B_3 : $J_3 = \{4,5\}$.

2. Параллельно построим функцию для задач и ограничений, соответствующих данным блокам.

Составим задачу D_2.

Так как вершина $r = 2$ не имеет потомков то кроме переменных, которые входят в целевую функцию, не нужно учитывать значения предыдущих функций. Соответствующая задача имеет вид $f_{D_2}\left(X_{S_{12}}\right) = \max\{2x_1 + x_2 + 3x_3\}$

при ограничениях

$3x_1 + 2x_2 + 2x_3 \le 6 - x_4, x_1, x_2, x_3 = \{0;1\}$.

Если $x_4 = 0$, то $f_{D_2}(0) = 5$, $X_{D_2}(0) = \{x_1 = 0, x_2 = 0, x_3 = 1\}$.	Если $x_4 = 0$, то $f_{D_2}(1) = 3$ $X_{D_2}(1) = \{x_1 = 1, x_2 = 0, x_3 = 1\}$.

Составим задачу D_3.

Так как вершина $r = 3$ имеет потомков $J_3 = \{4,5\}$, то кроме переменных, которые входят в целевую функцию, необходимо также учитывать значения функций, которые были получены при решении задач соответствующих дочерних блоков. Функция имеет вид $f_{D_3}\left(X_{S_{13}}\right) =$

$= \max\left\{x_8 + f_{D_4}(x_9) + f_{D_5}(x_{10}) + 2x_9 + 3x_{10}\right\}$.

В ограничении общие переменные с другими блоками x_7, x_9 и x_{10} переносятся в правую сторону неравенства:

$8x_8 \le 8 - 3x_9 - x_{10} - 3x_7, x_8 = \{0;1\}$.

Если $x_7 = 0$, то $8x_8 \le 8 - 3x_9 - x_{10}$, $x_8 = 0$ $\forall x_9, x_{10}$ или $x_8 = 1, x_9 = 0, x_{10} = 0$.	Если $x_7 = 1$, то $8x_8 \le 5 - 3x_9 - x_{10}$, $x_8 = 0$ $\forall x_9, x_{10}$

3. Рассмотрим параллельно блоки-потомки для задачи D_3.

$x_9 = 0,\ f_{D_4}(0) = 6,$	$x_9 = 1,\ f_{D_4}(1) = 6,$	$x_{10} = 0,\ f_{D_5}(0) = 3,$	$x_{10} = 1,\ f_{D_5}(1) = 5,$
$X_{D_4}(0) =$	$X_{D_4}(1) =$	$X_{D_5}(0) =$	$X_{D_5}(1) =$
$= \{x_{11} = 0, x_{12} = 1\}.$	$= \{x_{11} = 0, x_{12} = 1\}.$	$= \{x_{13} = 0, x_{14} = 1\}.$	$= \{x_{13} = 1, x_{14} = 0\}.$

Рассмотрим функцию f_{D_3} при $x_7 = 0$, а точнее их части, связанные с переменными x_9 и x_{10} :

$x_9 = 0,$	$x_9 = 1,$	$x_{10} = 0,$	$x_{10} = 1,$
$f_{D_4}(x_9) + 2x_9 + x_8 =$	$f_{D_4}(x_9) + 2x_9 + x_8 =$	$f_{D_5}(x_{10}) + 3x_{10} + x_8$	$f_{D_5}(x_{10}) + 3x_{10} + x_8$
$= 6 + 0 + 1 = 7,$	$= 6 + 2 + 0 = 8,$	$= 3 + 0 + 1 = 4,$	$= 2 + 3 + 0 = 5,$
$X_{D_4}(0) =$	$X_{D_4}(1) =$	$X_{D_5}(0) =$	$X_{D_5}(1) =$
$= \{x_{11} = 0, x_{12} = 1\}.$	$= \{x_{11} = 0, x_{12} = 1\}.$	$= \{x_{13} = 0, x_{14} = 1\}.$	$= \{x_{13} = 1, x_{14} = 0\}.$
Следовательно, выбираем значение $x_9 = 1$.		Следовательно, выбираем значение $x_{10} = 1$	

Для x_9 и x_{10} максимальные значения получаются при $x_8 = 0$. Отсюда, $f_{D_3}(1) = x_8 + f_{D_4}(x_9) +$ $+ f_{D_5}(x_{10}) + 2x_9 + 3x_{10} = 0 + 6 + 2 + 2 + 3 = 13,\ X_{D_3}(1) = \{x_{12} = 1, x_{11} = 0, x_9 = 1, x_{10} = 1, x_8 = 0, x_{13} = 1, x_{14} = 0\}$

Рассмотрим функцию f_{D_3} при $x_7 = 1$, а точнее их части, связанные с переменными x_9 и x_{10}:

$x_9 = 0,$	$x_9 = 1,$	$x_{10} = 0,$	$x_{10} = 1,$
$f_{D_4}(x_9) + 2x_9 + x_8 =$	$f_{D_4}(x_9) + 2x_9 + x_8 =$	$f_{D_5}(x_{10}) + 3x_{10} + x_8$	$f_{D_5}(x_{10}) + 3x_{10} + x_8$
$= 6 + 0 + 0 = 6$	$= 6 + 2 + 0 = 8$	$= 3 + 0 + 0 = 3$	$= 2 + 3 + 0 = 5$
$X_{D_4}(0) =$	$X_{D_4}(1) =$	$X_{D_5}(0) =$	$X_{D_5}(1) =$
$= \{x_{11} = 0, x_{12} = 1\}.$	$= \{x_{11} = 0, x_{12} = 1\}.$	$= \{x_{13} = 0, x_{14} = 1\}.$	$= \{x_{13} = 1, x_{14} = 0\}.$
Следовательно, выбираем значение $x_9 = 1$.		Следовательно, выбираем значение $x_{10} = 1$.	

Отсюда $f_{D_3}(1) = x_8 + f_{D_4}(x_9) + f_{D_5}(x_{10}) + 2x_9 + 3x_{10} = 0 + 6 + 2 + 2 + 3 = 13,$

$X_{D_3}(1) = \{x_{12} = 1, x_{11} = 0, x_9 = 1, x_{10} = 1, x_8 = 0, x_{13} = 1, x_{14} = 0\}$.

4. Перейдем к слою $\nu = 1$, здесь всего одна вершина, $r = 1$, причем $J_1 = \{2, 3\}$, $S_{13} = \{4\}$, $S_{12} = \{7\}$. Соответствующая задача Z_{D_r} имеет вид

$$f_{D_1} = \max\{x_5 + 3x_6 + f_{D_2}(x_4) + f_{D_3}(x_7) + 2x_4 + 4x_7\}$$

при ограничениях $x_5 + 4x_6 \le 4 - x_4 - 2x_7;\ x_5, x_6 = \{0;1\}$.

Если $\{x_4 = 0, x_7 = 0\}$,	Если $\{x_4 = 1, x_7 = 0\}$,	Если $\{x_4 = 0, x_7 = 1\}$,	Если $\{x_4 = 1, x_7 = 1\}$,
то $\{x_5 = 0, x_6 = 1\}$,	то $\{x_5 = 1, x_6 = 0\}$ или	то $\{x_5 = 1, x_6 = 0\}$ или	то $\{x_5 = 1, x_6 = 0\}$
$\{x_5 = 1, x_6 = 0\}$ или	$\{x_5 = 0, x_6 = 0\}$.	$\{x_5 = 0, x_6 = 0\}$.	или $\{x_5 = 0, x_6 = 0\}$.
$\{x_5 = 0, x_6 = 0\}$			

Рассмотрим блоки-потомки.

Если $x_4 = 0$, то	Если $x_4 = 1$, то	$x_7 = \{0,1\}\ f_{D_3} = 13,$
$f_{D_2}(0) = 5,\ X_{D_2}(0) =$	$f_{D_2}(1) = 3,\ X_{D_2}(1) =$	$X_{D_4} = \{x_9 = 1, x_{10} = 1, x_{11} = 0, x_{12} = 1, x_{13} = 1, x_{14} = 0\}.$
$= \{x_1 = 0, x_2 = 0, x_3 = 1\}.$	$= \{x_1 = 1, x_2 = 0, x_3 = 1\}.$	

Рассмотрим функцию f_{D_1}, а точнее ее часть, связанную с переменными x_4 и x_7:

$x_4 = 0$	$x_4 = 1$	$x_7 = 0$	$x_7 = 1$
$x_5 + 3x_6 + f_{D_2}(0) + 2x_4 =$	$x_5 + 3x_6 + f_{D_2}(1) + 2x_4 =$	$x_5 + 3x_6 + f_{D_3}(0) + 4x_7 =$	$x_5 + 3x_6 + f_{D_3}(1) + 4x_7 =$
$= 0 + 3 + 5 + 0 = 8.$	$= 1 + 0 + 3 + 2 = 6.$	$= 13 + 0 = 13.$	$= 1 + 0 + 13 + 4 = 18.$
Следовательно, выбираем значение $x_4 = 0.$		Следовательно, выбираем значение $x_7 = 1.$	

IV. Решим задачу, соответствующую корневой вершине:

$$f_{D_1} = \max\left\{x_5 + 3x_6 + f_{D_2}(x_4) + f_{D_3}(x_7) + 2x_4 + 4x_7\right\} = x_5 + 3x_6 + 3 + f_{D_3}(x_7) + 2 + 4.$$

Оптимальному решению соответствуют $x_4 = 0, x_7 = 1,$ причем $f_{D_1} = 23.$

V. Выпишем вектор значений исходя из решенных задач и произведем проверку:

$$X_{D_1} = \{x_{12} = 1, x_{11} = 0, x_{13} = 1, x_{14} = 0, x_8 = 0, x_9 = 1,$$
$$x_{10} = 1, x_4 = 0, x_7 = 1, x_5 = 1, x_6 = 0, x_1 = 1, x_2 = 0, x_3 = 1\}.$$

Упорядочивая переменные, приходим к тому, что решение исходной задачи (которое, как уже отмечено, совпадает с решением задачи Z_{D_1}) имеет вид $x_1 = 1,\ x_2 = 0,\ x_3 = 1,\ x_4 = 0,\ x_5 = 1,$ $x_6 = 0,\ x_7 = 1,\ x_8 = 0,\ x_9 = 1,\ x_{10} = 1,\ x_{11} = 0,\ x_{12} = 1,\ x_{13} = 1,\ x_{14} = 0,\ Z_{\max} = 23.$

При параллельной трактовке ЛЭА необходимо, чтобы реализовывались следующие процессы:

– процесс, который анализирует ЭД и выявляет подзадачи на данном уровне дерева;

– процесс, который распределяет подзадачи;

– процесс, который решает подзадачи;

– процесс, который анализирует подзадачи, записывает результат и создает задачи следующего уровня на основе полученной информации;

– процесс, который собирает информацию воедино;

– процесс, который анализирует полученное решение.

Для реализации этих процессов целесообразно использование парадигмы «Директор–мастер–рабочий». Директор анализирует ЭД, выделяет подзадачи на данном уровне и отправляет их мастеру. Мастер распределяет задачи среди рабочих, собирает полученные результаты и отправляет их обратно директору. Директор анализирует подзадачи, записывает результат в таблицу и, если корень ЭД не достигнут, создает задачи следующего уровня и отправляет мастеру. Если же корень ЭД был достигнут, директор просматривает таблицу промежуточных решений, выбирает оптимальное и проверяет целевую функцию. Если значение целевой функции в корне совпало со значением исходной целевой функции для полученного решения, то задача решена верно.

Для реализации парадигмы «директор–мастер–рабочий» хорошо подходит платформа распределенных вычислений BOINC (*Berkeley Open Infrastructure for Network Computing*) [Российские распределенные вычисления…], где директором назначается компьютер, на котором непосредственно решается задача, мастером назначается удаленный боинк-сервер, а рабочими процессами — удаленные компьютеры. Распределенные вычисления позволяют быстро решать объемные задачи, так как к процессу решения привлечено большое количество пользователей, а значит, и удаленных компьютеров. В данный момент ведется разработка параллельного ЛЭА для использования его на платформе BOINC.

Список литературы

Лемтюжникова Д. В., Щербина О. А. Локальный элиминационный алгоритм и параллельные вычисления // Интеллектуальные системы. МГУ. — 2013. — 17 (часть 5). — С. 490–494.

Российские распределенные вычисления на платформе BOINC [Электронный ресурс]. URL: http://www.boinc.ru/

Щербина О. А. Локальные элиминационные алгоритмы решения разреженных дискретных задач // Журнал вычислительной математики и математической физики. — 2008. — Т. 48, № 1. — С. 161–177.

Щербина О. А. Древовидная декомпозиция и задачи дискретной оптимизации (обзор) // Кибернетика и системный анализ. — 2007. — № 4. — С. 102–118.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 707–713

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.023

Размещение точек Штейнера в дереве Штейнера на плоскости средствами MatLab

Д. Т. Лотарев

Институт проблем передачи информации им. А. А. Харкевича РАН,
Россия, 127051, г. Москва, Большой Каретный переулок, д. 19, стр. 1

E-mail: dimlot@mail.ru

Получено 30 сентября 2014 г.

Рассматривается способ локализации точек Штейнера средствами MatLab в задаче Штейнера с потоком на евклидовой плоскости, когда соединяемые точки лежат в вершинах четырех-, пяти- или шестиугольника. Матрица смежности считается заданной. Метод использует способ решения трехточечной задачи Штейнера, в которой дерево Штейнера связывает три точки. Представлена визуализация найденных решений.

Ключевые слова: задача Штейнера, точка Штейнера, источник ресурса, потребитель ресурса, трехточечная задача Штейнера, задача для большего числа, понижение размерности

Allocation of steinerpoints in euclidean Steiner tree problem by means of MatLab package

D. T. Lotarev

A. A. Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, 19/1 Bolshoy Karetny per., Moscow 127051, Russia

Abstract. —The problem of allocation of Steiner points in Euclidean Steiner Tree is considered. The cost of network is sum of building costs and cost of the information transportation. Euclidean Steiner tree problem in the form of topological network design is a good model of this problem.
The package MatLab has the way to solve the second part of this problem — allocate Steiner points under condition that the adjacency matrix is set. The method to get solution has been worked out. The Steiner tree is formed by means of solving of the sequence of "three points" Steiner

Keywords: Steiner problem, Steiner point, source resource, consumer resource, “three points” Steiner problem, task for bigger number, decrease dimension

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 707–713 (Russian).

1. Введение

Компьютерная сеть представляет собой множество компьютеров, соединенных каналами связи, по которым передается информация. Несмотря на широкое распространение беспроводных технологий, многие компьютерные сети используют в качестве физической среды передачи информации медные или волоконно-оптические кабели. В этом случае при создании сети возникает необходимость прокладывать кабельные сети на земной территории. Земная территория существенно неоднородна по условиям прокладки кабеля. Следует учитывать, что каждый из элементов этой неоднородности определяет свои условия и стоимость прокладки. Стремясь минимизировать затраты на прокладку кабельной сети, приходится решать задачу размещения сети в условиях этой неоднородности.

Моделью задачи о минимизации затрат может быть либо задача о минимальном связывающем дереве (Minimal Spanning Tree — MST) [Прим, 1961], либо задача о минимальном дереве Штейнера — задача Штейнера (Steiner Minimal Tree — SMT) [Берн, Грэм, 1989; Курант, Роббинс, 2001]. В задаче об MST разветвления сети допускаются только в соединяемых точках, а в задаче о SMT, называемой просто задачей Штейнера, разветвления сети допускаются во всех точках области размещения кабелей. В данной работе именно эта задача принята за модель задачи размещения компьютерной сети на неоднородной территории.

Задача Штейнера представляет большой научный и практический интерес. Известно [Берн, Грэм, 1989; Курант, Роббинс, 2001], что на плоскости минимальное дерево Штейнера короче минимального связывающего дерева не более чем на 13.6 %. Но это NP-трудная задача и для ее решения применяются главным образом эвристические методы. Точное решение возможно только для задач малой размерности.

Задача минимизации длины дерева — это одна из старейших задач оптимизации. П. Ферма предложил задачу поиска точки $\boldsymbol{P}$, которая минимизирует суммарное расстояние от $\boldsymbol{P}$ до трех заданных точек плоскости. Исследованием этой задачи занимались математики того времени и нашли ее решение [Курант, Роббинс, 2001]. Если в треугольнике ABC, вершинами которого являются заданные точки, все углы меньше 120 градусов, то точка P лежит внутри треугольника, и каждый из трех углов $\angle APB$, $\angle APC$, $\angle BPC$ равен 120°. Если один из углов треугольника равен или больше 120°, то точка $\boldsymbol{P}$ лежит в вершине этого угла. В [Курант, Роббинс, 2001] описан способ решения задачи. Если на каждой стороне треугольника ABC построить дугу в 120°, то точка пересечения дуг — это искомая точка. В [Лотарев, 2008] представлено решение классической задачи Штейнера для трех исходных точек. Задача названа трехточечной задачей Штейнера и решается средствами MatLab.

Пакет прикладных программ MatLab предоставляет большие возможности в изучении этой задачи. В [Лотарев, 2008] представлены исследования по трехточечной задачи Штейнера без потока средствами MatLab. В данной работе представлено продолжение этих исследований.

Здесь рассматривается сеть коммуникаций, которая создается для транспортировки потока. Поэтому критерием оптимальности при оптимизации сети принята сумма затрат на строительство сети и на транспортировку по ней того потока, для перемещения которого сеть строится. Мы рассматриваем информационный поток, но он может быть также материальным или энергетическим.

2. Трехточечная задача Штейнера с потоком на евклидовой плоскости

В задаче размещения сети на земной территории требуется выполнение минимума некоторого критерия оптимальности. Каждому критерию отвечает некоторая своя оптимальная конфигурация сети.

Пусть на некотором равнинном участке земной территории задано размещение трех объектов, A_1, A_2, A_3, из которых объект A_3 является источником некоторого ресурса, а объекты A_1,

A_2 — потребителями. В соответствии с принятой моделью считаем, что они лежат на евклидовой плоскости, и сеть имеет вид, показанный на рис. 1. Используются следующие обозначения. A_3 — источник ресурса, A_1, A_2 — потребители, l_1, l_2, l_3 — ребра сети, p — точка разветвления, a_i, $i = 1, 2, 3$, — углы при точке разветвления.

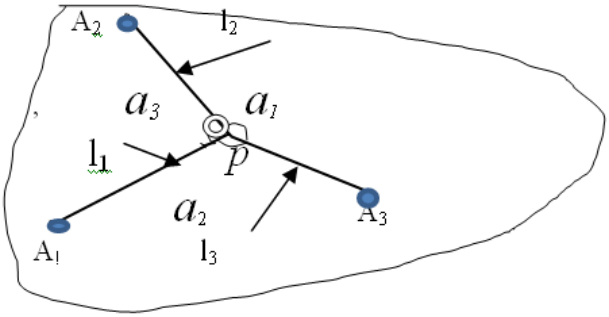


Рис. 1. Модель участка территории и сети на нем

Заданы значения следующих параметров:

$r_1 = \text{const}$ — значение удельных строительных затрат — затрат на строительство отрезка коммуникации единичной длины,

$r_2 = \text{const}$ — значение удельных транспортных затрат на транспортировку единичного потока по отрезку коммуникации единичной длины:

$q_1 = \text{const}$ — значение спроса стока A_1,

$q_2 = \text{const}$ — значение спроса стока A_2.

Значения r_1 определяются свойствами территории, r_2 — качеством самой коммуникации, а q_1, q_2— свойствами потребителей.

Требуется построить на этом участке сеть транспортных коммуникаций, связывающих источник A_3 с потребителями, A_1, A_2, которая обеспечивает протекание потока ресурса при минимальном значении критерия оптимальности.

Будем полагать, что искомая сеть имеет вид дерева Штейнера и что точка Штейнера является точкой разветвления сети. Задача состоит в том, чтобы найти локализацию точки Штейнера, выяснить, как изменяются ее локализация и значения углов между инцидентными ей ребрами при учете в критерии оптимальности затрат на транспортировку потока. Эти изменения зависят от значений параметров q_1, q_2, r_1, r_2. Значения этих параметров определяют и длину ребер l_1, l_2, l_3 (см. рис. 1). При некоторых значениях параметров q_1, q_2, r_1, r_2 сеть остается классическим деревом Штейнера, при других она становится кратчайшим связывающим деревом.

Затраты F на строительство сети и на транспортировку по ней потоков q_1 и q_2 можно записать в виде

$$F = (r_1 + r_2 q_1) l_1 + (r_1 + r_2 q_2) l_2 + (r_1 + r_2 (q_1 + q_2)) l_3. \tag{1}$$

Обозначим f_1, f_2, f_3 суммарные строительные и транспортные затраты на отрезке коммуникации единичной длины для ребер l_1, l_2, l_3 соответственно

$$f_1 = r_1 + r_2 q_1, \tag{2}$$

$$f_2 = r_1 + r_2 q_2, \tag{3}$$

$$f_3 = r_1 + r_2 (q_1 + q_2). \tag{4}$$

Тогда критерий (1) принимает вид

$$F = f_1 l_1 + f_2 l_2 + f_3 l_3. \tag{5}$$

Критерий оптимальности (5) можно рассматривать как работу сил f_1, f_2, f_3, приложенных к материальной точке p и направленных к точкам A_1, A_2, A_3, при возможном движении материальной точки p к точкам A_1, A_2, A_3. Эта работа равна потенциальной энергии материальной точки p относительно точек A_1, A_2, A_3 [Гантмахер, 1960].

Известно [Гантмахер, 1960], что потенциальная энергия механической системы имеет строгий минимум в положении устойчивого равновесия системы и что система находится в равновесии, если сумма сил, воздействующих на нее, равна нулю. Таким образом, необходимым условием минимума критерия (5) является равенство нулю векторной суммы $\vec{f}_1 + \vec{f}_2 + \vec{f}_3$:

$$\vec{f}_1 + \vec{f}_2 + \vec{f}_3 = 0. \tag{6}$$

Углы между векторами $\vec{f}_1, \vec{f}_2, \vec{f}_3$ в точке $\boldsymbol{p}$ можно определить из соотношения (6). Для этого все векторы $\vec{f}_1, \vec{f}_2, \vec{f}_3$, фигурирующие в (6), спроектируем поочередно на направление вектора f_1, направление вектора f_2, направление вектора f_3 (проекции на направление, противоположное направлению вектора f_3, показаны на рис. 2).

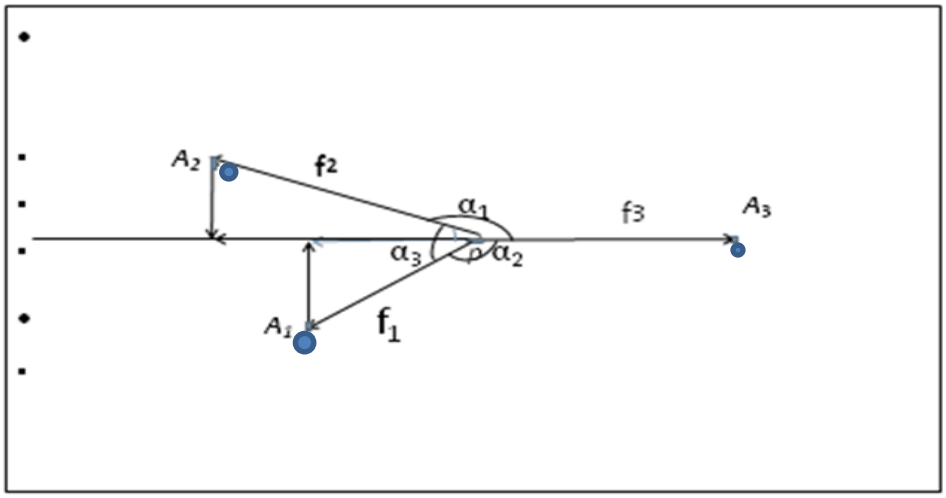


Рис. 2. Силы, действующие на точку Штейнера

Раскрывая в (6) суммы пар векторов $\vec{f}_1, \vec{f}_2$, $\vec{f}_1, \vec{f}_3$, $\vec{f}_2, \vec{f}_3$, получаем три уравнения:

$$(f_1^2 + 2f_1 f_2 \cos\alpha_3 + f_2^2) = f_3^2,$$

$$(f_1^2 + 2f_1 f_3 \cos\alpha_2 + f_3^2) = f_2^2,$$

$$(f_2^2 + 2f_2 f_3 \cos\alpha_1 + f_3^2) = f_1^2.$$

Из этих уравнений следуют соотношения

$$\cos\alpha_1 = \frac{f_1^2 - f_2^2 - f_3^2}{2f_2 f_3}, \tag{7}$$

$$\cos\alpha_2 = \frac{f_2^2 - f_1^2 - f_3^2}{2f_1 f_3}, \tag{8}$$

$$\cos\alpha_3 = \frac{f_3^2 - f_1^2 - f_2^2}{2f_1 f_2}. \tag{9}$$

Если удельные транспортные затраты малы по сравнению с удельными строительными затратами, то ими можно пренебречь и соотношения (2)–(4) принимают вид

$$f_1 = r_1,\ f_2 = r_1,\ f_3 = r_1. \tag{10}$$

Подставляя эти выражения для f_1, f_2, f_3 в соотношения (7), (8), (9), получаем, что в классическом дереве Штейнера на евклидовой плоскости каждый из углов a_1, a_2, a_3 между ребрами дерева, сходящимися в точке Штейнера, равен 120° и $\cos\alpha_1 = \cos\alpha_2 = \cos\alpha_3 = ½$.

Программа, код которой написан на М-языке системы MATLAB, позволяет определить координаты точки Штейнера и значения углов между ребрами, инцидентными этой точке, при различных значениях удельных строительных и транспортных затрат, а также получить визуальное представление результатов, показанное на рис. 3. На этом рисунке показаны положение дерева на плоскости (a, c, e), поверхность отклика и линии уровня (b, d, f).

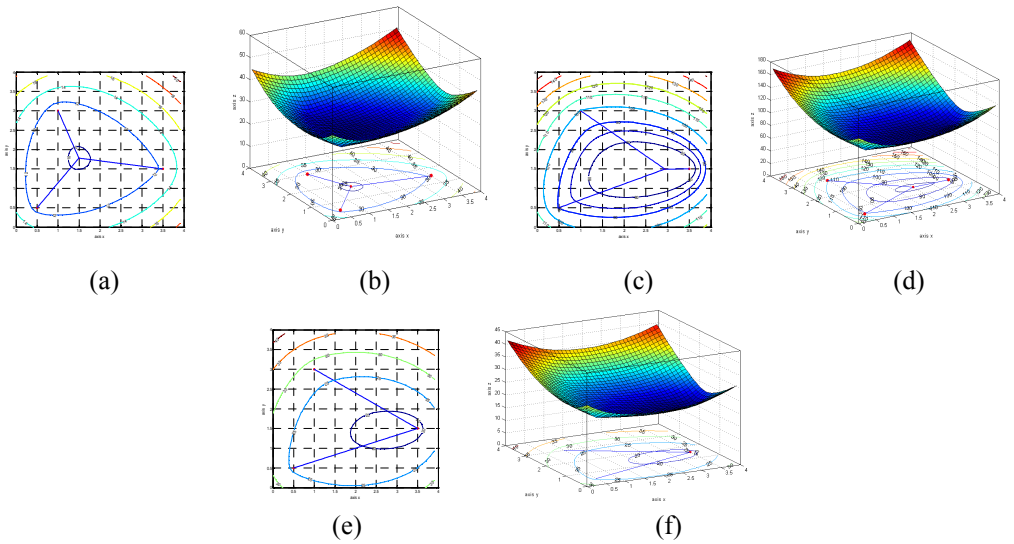


Рис. 3. Минимальное дерево Штейнера на плоскости при различных значениях удельных строительных (r_1) и удельных транспортных (r_2) затрат: (a, b) $r_2 = 0$; (c, d) $r_1 = 5$, $r_2 = 3$; (e, f) $r_1 = 0$

Из рисунка видно, что при изменении значений удельных строительных и удельных транспортных затрат меняются и положение точки Штейнера, и значения углов между инцидентными ей ребрами.

3. Понижение размерности

Умение решать трехточечную задачу позволяет понизить размерность исходной задачи, в которой число соединяемых точек более 3.

В дереве Штейнера для каждой пары вершин, смежных из одной точки Штейнера, существуют две замечательные точки: первая — это точка Штейнера, вторая — это эквивалентный сток.

Используя эквивалентный сток, можно решить задачу для точек, число которых больше трех. Для этого необходимо заменить пару стоков, для которых заданной матрицей смежности определена одна общая для них и смежная с ними точка Штейнера, одним эквивалентным стоком. Тем самым задача Штейнера для n точек сведется задаче с $n - 1$ точками. Нужно только найти локализацию этого эквивалентного стока. Для этого в трехточечной задаче на продолжении звена $A_3 p$ нужно найти такую точку A_4, что сток со спросом $(q_1 + q_2)$, размещенный в этой точке, эквивалентен двум стокам A_1 и A_2 по затратам на строительство коммуникации $[A_3, A_4]$

и на транспортировку по ней потока ($q_1 + q_2$) от источника A_3 до стока A_4. Этот сток и называется эквивалентным стоком.

Если для некоторой точки Штейнера еще не построенной сети известны и локализация точки Штейнера, и величина потока, который будет по ней протекать, то ее можно рассматривать источником для всех достижимых из нее стоков. Это означает, что процесс построения дерева Штейнера можно составить из двух групп шагов. Первая группа шагов — это шаги вперед. На этих шагах определяется локализация эквивалентных стоков. На последнем шаге этой группы определяется также окончательная локализация точки Штейнера, смежной с источником.

После этого начинаются шаги назад. На этих шагах определяется окончательная локализация остальных точек Штейнера. При этом на каждом шаге за источник принимается точка Штейнера, локализованная на предыдущем шаге.

Решение для 4-х исходных точек (рис. 4*a*) состоит из следующих шагов.

Далее всюду предполагается, что источник обозначен через A_1.

Шаг 1. Решаем трехточечную задачу для A_1, A_2, A_3 и находим локализацию эквивалентного стока ЭК1. Здесь же определяется локализация точки Штейнера TS_1. Но она верна только при отсутствии четвертого стока. Поэтому в дальнейшем она не используется.

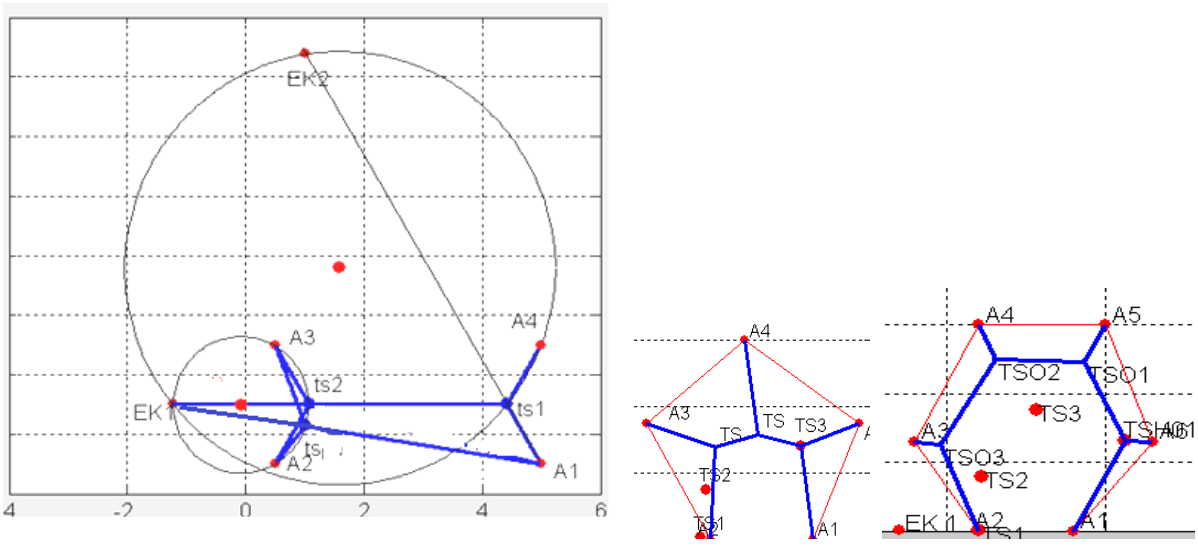
Шаг 2. Решаем трехточечную задачу для A_1, EK_1, A_4 и находим локализацию эквивалентного стока EK_2 и точки Штейнера TS_1, смежной с A_1 и A_4.

Шаг 3. Принимаем TS_1 за источник и, решая трехточечную задачу для TS_1, A_2, A_3, получаем окончательное решение задачи.

Шаг 1 и **шаг 2** считаются шагами вперед, а **шаг 3** — шагом назад.

Аналогичным образом ищутся деревья Штейнера для вершин правильного пятиугольника и вершин правильного шестиугольника (рис. 4*b*, 4*c*).

На рис. 4*a* можно видеть процесс пошагового решения задачи.



(a) (b) (c)

Рис. 4. Деревья Штейнера для вершин правильных многоугольников: (a) для четырехугольника, (b) для пятиугольника, (c) для шестиугольника

Заключение

Введение понятия эквивалентного стока позволяет быстро получить размещение точек Штейнера в задаче Штейнера небольшой размерности на эвклидовой плоскости при заданной матрице смежности.

Список литературы

Берн У., Грэм Л. Поиск кратчайших путей // Scientific American (издание на русском языке). — 1989. — № 3. — С. 64–70.

Гантмахер Ф. Р. Лекции по аналитической механике. — М.: Физматгиз, 1960.

Курант Р., Роббинс Г. Что такое математика. — М.: МЦНМО, 2001.

Лотарев Д. Т. Решение трехточечной задачи Штейнера на плоскости средствами MatLab // Труды ИСА РАН. — 2008. — Т. 32. — С. 159–165.

Прим Р. Кратчайшие связывающие сети и некоторые их обобщения // Кибернетический сборник. — 1961. — № 2. — С. 95–107.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 715–718

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Краудфандинг в организации построения распределенной grid-системы консолидации электронных библиотечных и интернет-ресурсов

Б. В. Олейников[a], А. И. Шалабай[b]

Сибирский федеральный университет,
Россия, 660041, г. Красноярск, пр. Свободный, д. 79/10

E-mail: [a] Oleynik48@mail.ru, [b] andrsh@gmail.com

Получено 30 сентября 2014 г.

При проектировании распределенной библиотечной системы кроме технических возникает ряд организационных проблем, идеям решения которых и посвящена данная статья. В качестве инструмента для привлечения участников и обеспечения финансовой безубыточности предлагается использовать подходы краудфандинговых платформ.

Ключевые слова: распределенная библиотечная система, краудфандинг, привлечение участников

Crowd funding in the construction of distributed grid-system of electronic library and internet resources

B. V. Oleynikov[a], A. I. Shalabay[b]

Siberian Federal University, 79 Svobodny pr., Krasnoyarsk, 660041, Russia

Abstract. — In the design of a distributed library system, in addition to technical, there are a lot of organizational problems, ideas, solutions which are the subject of this article. As a tool for attracting participants and financing project is using approaches crowd funding platforms.

Keywords: distributed library system, crowd funding, attracting participants

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 715–718 (Russian).

В данной статье рассматриваются организационные вопросы построения распределенной grid-системы консолидации библиотечных и интернет-ресурсов, концепция которой описана в [Олейников, Шалабай, 2012].

В дальнейшем участником grid-системы будем называть субъекта, размещающего цифровую литературу в системе «узел–поставщик», а пользователем — субъекта, осуществляющего поиск, просмотр и скачивание полных текстов книг (узел–потребитель).

При проектировании системы важной является не только техническая проработка деталей, но и решение организационных вопросов, в том числе:

– способы привлечения финансирования для обеспечения безубыточности функционирования системы; одним из важнейших источников пополнения фонда цифровых изданий является оцифровка бумажных носителей, что при больших масштабах влечет значительные финансовые затраты;

– необходимость заинтересовать потенциальных пользователей (как поставщиков, так и потребителей научной литературы) в использовании распределенной библиотечной системы. Использование современных маркетинговых приемов не является целесообразным в силу отсутствия рекламного бюджета. Причем такими участниками могут быть как юридические лица (государственные, частные библиотеки, научные учреждения, издательства), так и физические лица (ученые, сотрудники различных организаций, использующие в своей повседневной деятельности большое количество научной литературы и занимающиеся ее хранением на своих персональных компьютерах).

Итак, одним из направлений привлечения участников может стать краудфандинг оцифровки книг. Краудфандинг — это коллективное сотрудничество людей, которые добровольно объединяют свои финансовые ресурсы, чтобы поддержать усилия других людей или организаций. На практике это выглядит следующим образом: организация, имеющая какую-либо идею, но нуждающаяся в финансовых средствах для ее реализации, размещает ее описание и калькуляцию предполагаемых расходов на специальной интернет-платформе. Любой человек, заинтересованный в реализации проекта, может внести одну из указанных фиксированных сумм на счет этой платформы. В случае если набирается требуемая сумма, она перечисляется разработчику проекта, в противном случае денежные средства возвращаются всем участникам. Примером таких краудфандинговых платформ могут служить Kickstarter [Seven things…] и Boomstarter [Введение…]. В настоящее время реализовано множество проектов из совершенно различных сфер: разработка компьютерных игр, проектирование и изготовление сложных инженерных устройств, издательство книг, дизайнерские работы, благотворительность. Показательным является пример успешного сбора средств в России на издательство научно-популярных книг [Как мы собрали…].

Применительно к библиотечной grid-системе целью сбора средств может стать оцифровка больших объемов редкой литературы, доступ к бумажным копиям которой ограничен по причине малого количества сохранившихся экземпляров. Стимулом для внесения средств является преимущественный (ранний) доступ к оцифрованным изданиям. В публичный (бесплатный) доступ данная литература должна попадать по прошествии значительного периода времени после оцифровки.

Сами работы по оцифровке могут выполнять библиотеки (на договорной основе), фрилансеры либо коммерческие структуры. Так, в Японии компания Bookscan предоставляет подобные услуги. Стоимость оцифровки одной книги объемом 350 страниц составляет 100 иен, или около 35 рублей [Оцифровка книг…]. Клиенты компании — владельцы планшетных компьютеров и специализированных устройств для чтения.

Подобные услуги могут быть востребованы и в России, прежде всего среди школьников и студентов, которым приходится ежедневно носить значительное количество бумажной литературы. При этом книги возможно распространять в нераспознанном виде (в форматах djvu и pdf), так как значительная часть мобильных устройств оснащена экраном с диагональю 8–10 дюймов, что сопоставимо с размерами исходных, бумажных, книг.

Для изданий, пользующихся повышенным спросом, в процессе сбора средств возможно ставить цели «на вырост». Так, возможно не только оцифровывать, но и осуществлять распознавание текста с помощью OCR-систем (с «ручной» корректировкой ошибок), составлять интерактивные оглавления, улучшать качество иллюстраций и т. п. Очевидно, что в таком случае затраты возрастут многократно, но примеры краудфандинговых проектов показывают, что изначально недорогой проект может собрать сумму в десятки раз превышающую изначально требуемую, приобрести новые функции и превратиться в полноценный конкурентоспособный продукт. Примером может служить сбор средств на издание детской книги [Zachary], которая при изначально запрошенных 30000$ собрала более 380000$. В процессе сбора были достигнуты следующие цели: улучшение дизайна и качества бумаги, безвозмездная передача 800 экземпляров книги в публичные библиотеки, создание аудиоверсии на английском языке и другое.

Кроме того, размещение проекта на популярной краудфандинговой площадке способно привлечь к себе множество пользователей, заинтересованных в быстрой и качественной оцифровке книг.

Еще одним способом вовлечения участников может стать размещение рекламной информации при просмотре каталога организации. Средства, полученные от рекламодателей, могут пропорционально распределяться между участниками в зависимости от их популярности среди пользователей системы.

Удачным примером подобного подхода может служить популярный видеохостинг YouTube [О YouTube]: при просмотре видеороликов пользователь сначала видит рекламный ролик, и чем больше просмотров набирает конкретный видеоматериал, тем большую сумму зарабатывает его автор.

Для повышения качества размещаемых материалов целесообразно внедрение рейтинговой системы: каждому участнику, в зависимости от числа скачиваний размещенных им полных текстов и оценок пользователей, присваивается определенный рейтинг, в зависимости от которого участник получает дополнительные нефинансовые привилегии (более высокое место в поисковой выдаче, право изменения оформления своего каталога, увеличение минимального количества узлов — резервных держателей полных текстов книг и т. д.).

Для привлечения издательств к пополнению хранилища возможно введение платного скачивания части литературы. Таким образом, в системе будет функционировать полноценный каталогизированный интернет-магазин цифровых изданий с единой системой поиска книг. Следовательно, издательства при минимальных затратах получат еще один канал реализации цифровой литературы. При этом можно предположить, что для получения конкурентных преимуществ (нефинансовых привилегий) издательства будут стремиться постоянно пополнять каталог новинками книжного рынка.

При внедрении указанных мер поощрения участников grid-системы строго необходимыми являются гарантии соблюдения авторских прав. В противном случае к проблемам нарушения законодательства добавятся риски выхода из системы издательств, если у платных книг в этой же системе будут присутствовать бесплатные копии.

Поэтому важным аспектом становится регистрация и строгая фильтрация участников: с каждым необходимо заключать специальное соглашение, в котором, с одной стороны, участник обязуется не нарушать авторские права, а с другой — получает гарантии выполнения указанных выше финансовых мер привлечения. Каждая заявка на регистрацию должна рассматриваться индивидуально: доступ к пополнению каталога grid-системы должны иметь только организации определенных видов (библиотеки, научные и образовательные учреждения, издательства) и публичные физические лица (дорожащие своей репутацией, например известные ученые, деятели искусств и т. д.).

Конечно, изложенные в настоящей статье организационные вопросы находятся на стадии идей и нуждаются в глубокой проработке и финансовой оценке, которая невозможна без привлечения сторонних специалистов, прежде всего экономистов и юристов. Однако их изложение в общем виде позволяет представить возможные способы взаимодействия с участниками grid-системы.

Список литературы

Олейников Б. В., Шалабай А. И. О консолидации электронных библиотечных и интернет-ресурсов для образовательных и научных целей // Открытое и дистанционное образование. — 2012. — № 4 (48). — С. 40–46.

Seven things to know about kickstarter / Краудфандинговая платформа Kickstarter. URL: https://www.kickstarter.com/hello?ref=footer. 2014.

Введение в Boomstarter. / Краудфандинговая платформа Boomstarter. URL: https://boomstarter.ru/help/faq/введение_в_boomstarter/. 2015.

Как мы собрали 823 376 руб. на печать книги через краудфандинг. / Блог компании «Простая наука». URL: http://habrahabr.ru/company/gtv/blog/207366/. 2013.

О YouTube. URL: http://www.youtube.com/yt/about/ru/. 2015.

Оцифровка книг для букридеров / CoolIdea. Журнал идей для бизнеса (электронное издание). URL: http://coolidea.ru/2011/02/26/bookscan/. 2014.

Zachary W. Augie and the Green Knight: A Children's Adventure Book / Краудфандинговая платформа Kickstarter. URL: https://www.kickstarter.com/projects/weiner/augie-and-the-green-knight-a-childrens-adventure-b. 2015.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 719–725

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.023

Предварительная декомпозиция задач дискретной оптимизации для ускорения алгоритма ветвей и границ в распределенной вычислительной среде

С. А. Смирнов[a], В. В. Волошинов

Институт проблем передачи информации им. А. А. Харкевича РАН,
Россия, 127051, г. Москва, Большой Каретный переулок, д. 19, стр. 1

E-mail: [a] sasmir@gmail.com

Получено 30 сентября 2014 г.

В работе рассматриваются возможности реализации крупноблочных схем метода ветвей и границ для решения частично целочисленных задач линейного программирования. В качестве основы берется пакет оптимизации с открытым исходным кодом CBC. Анализируется возможность использования пакета для реализации крупноблочной схемы метода ветвей и границ. Система реализуется с использованием языка Erlang. Проводятся численные эксперименты на основе задачи о коммивояжере, показывающие заметное ускорение распределенной схемы решения задачи по сравнению с единичным однопоточным экземпляром пакета.

Ключевые слова: метод ветвей и границ, крупнозернистый параллелизм

Pre-decomposition of discrete optimization problems to speed up the branch and bound method in a distributed computing environment

S. A. Smirnov, V. V. Voloshinov

Institute for Information Transmission Problems of the Russian Academy of Science, Kharkevich Institute, 19/1 Bolshoy Karetny per., Moscow, 127051, Russia

Abstract. — The paper presents an implementation of branch and bound algorithm employing coarse grained parallelism. The system is based on CBC (COIN-OR branch and cut) open-source MIP solver and inter-process communication capabilities of Erlang. Numerical results show noticeable speedup in comparison to single-threaded CBC instance.

Keywords: branch and bound algorithm, coarse grained parallelism

Работа выполнена при финансовой поддержке программы Президиума РАН № 14 «Проблемы создания национальной научной распределенной информационно-вычислительной среды на основе grid-технологий, облачных вычислений и современных телекоммуникационных сетей».

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 719–725 (Russian).

Введение

Для решения трудоемких задач дискретной оптимизации широкое распространение получили комбинаторные методы, среди которых наиболее часто встречаются методы ветвей и границ (МВГ). МВГ можно представить в виде процесса последовательного разбиения множества допустимых решений на подмножества с последующим отсечением не содержащих оптимальное решение подмножеств. Его реализация требует обхода некоторого дерева поиска. Каждому узлу дерева соответствует вспомогательная (оценочная) задача, полученная ослаблением части ограничений исходной задачи. Работа алгоритма может существенно зависеть от порядка выбора узлов дерева (правил его обхода). Традиционно ускорение работы МВГ достигается за счет применения механизма параллельных вычислений, суть которого заключается в распределенной обработке оценочных задач некоторым пулом солверов (так называемом мелкозернистое распараллеливание). На нижнем уровне распараллеливанию подвергаются отдельные части алгоритма, однако взаимодействие их не меняется и параллельный алгоритм в целом ведет себя так же, как последовательный. Однако практическое применение такого подхода даже на базе специализированных программных инструментариев требует привлечения квалифицированных программистов, владеющих навыками работы с технологиями параллельных вычислений MPI и/или OpenMP.

Это обстоятельство требует исследовать возможности применения иной схемы декомпозиционного решения задачи дискретной оптимизации. Речь идет о гораздо реже применяемой, так называемой «крупноблочной» («крупнозернистой») схеме распараллеливания [Попов, 2007]. В ее основе — параллельное решение подзадач с обменом информацией о найденных (в ходе решения подзадач) значениях целевой функции на допустимых решениях (так называемой рекордов). При таком подходе полученные подзадачи могут решаться различными вариантами метода ветвей и границ (поиск «в ширину» или «в глубину» дерева ветвлений, различные эвристики выбора следующей вершины и т. п.). Распараллеливание на этом уровне меняет весь алгоритм в целом, и работа, производимая параллельной версией, порой существенно отличается от работы, производимой последовательной версией: возможно, некоторые ее части вообще не считаются одной из версий, но считаются другой, и наоборот. Несколько активных подзадач могут обрабатываться одновременно, каждая в своем, отдельном процессе. Если один из процессов находит допустимое решение, то соответствующее значение целевой функции (так называемой рекорд) может быть разослано остальным процессам, что позволит им, в принципе, существенно ускорить работу за счет отбрасывания заведомо «неоптимальных» частей исходной задачи. Подобный подход обсуждается в литературе [Попов, 2007; Valente and Mitra, 2008; Bussieck, Ferris, and Meeraus, 2009], но широкого применения пока не получил.

Роль исследователя при этом фактически сводится к поиску подходящего способа первоначальной декомпозиции исходной задачи (предпочтительно в форме программы на языке оптимизационного моделирования) и настройке параметров алгоритма МВГ для набора пакетов дискретной оптимизации, подключенных к распределенной системе. Программная реализация обмена значениями рекордов может оказаться значительно проще, чем для «мелкозернистой» схемы МВГ.

В настоящее время существует ряд пакетов оптимизации, применяющих МВГ для решения частично-целочисленных задач оптимизации: LPsolve, GLPK, CBC, SCIP, Cplex. Первые четыре из них имеют открытый исходный код. Таким образом, естественно желание реализовать на основе таких пакетов крупноблочную схему МВГ. В данной работе сделана попытка реализации крупноблочной схемы МВГ для частично целочисленных задач оптимизации на основе пакета CBC. Представленный в работе алгоритм обеспечивает распределенное решение подзадач на множестве хостов.

Пакет оптимизации CBC

Пакет CBC (COIN-OR branch and cut) [Forres, and Lougee-Heimer, 2005] имеет открытый исходный код на языке C++, разрабатывается в рамках проекта COIN-OR (Computational Infra-

structure for Operations Research) и предназначен для численного решения частично целочисленных задач линейного программирования методами отсечений и ветвей и границ. Использование CBC возможно как через автономное приложение, принимающее данные в форматах AMPL [Fourer, Gay, and Kernighan, 2002], MPS и др., так и в виде встраиваемой библиотеки. CBC может использовать многопоточность для ускорения вычислений и является одним из самых эффективных среди пакетов с открытым исходным кодом [Koch et al., 2011].

В своей документации пакет позиционируется в первую очередь как встраиваемый, хотя и обладающий рудиментарно реализованным автономным приложением. В документации CBC присутствует описание нескольких примеров его встраивания. Необходимый минимум действий состоит в создании экземпляра класса, ответственного за решение задач линейного программирования (обычно для этого применяют пакет CLP), и объекта класса CbcModel, с которым и работает пользователь.

При этом оказывается, что автономное приложение находит решения задач заметно эффективнее, чем встраиваемая версия пакета из примеров. Это объясняется использованием в автономном приложении множества дополнительных приемов и эвристик. Существуют и способы встраивания большей части автономного приложения. Для этого можно использовать функцию callCbc, объявленую в файле CbcModel.hpp исходных кодов CBC. Данная функция позволяет запустить процесс решения задачи, имея строку параметров в том же формате, что и у автономного CBC, а также объект модели CBC, нужным образом настроенный пользователем.

В CBC предусмотрены средства, позволяющие получить или установить текущее значение рекорда. У объектов CbcModel присутствует метод setBestSolution, позволяющий передать пакету оптимизации лучшее на данный момент допустимое решение или только значение целевой функции на этом решении. При этом CBC не будет проверять, является ли это решение на самом деле допустимым. Это свойство важно для обмена значениями рекордов, поступающих от решаемых одновременно подзадач, так как подзадачи часто имеют непересекающиеся множества допустимых решений. Контроль за процессом решения задачи возможен, если установить в CbcModel объект функций обратного вызова, реализующий интерфейс CbcCompare. С его помощью можно узнать об обнаружении нового рекорда, а также подбросить новое значение рекорда, не нарушив при этом работу CBC.

Схема работы алгоритма

Работу крупноблочной схемы распараллеливания, используемой в данной работе, можно разделить на два этапа.

Исполнимый модуль генерации подзадач создает набор AMPL-стабов подзадач на основе пользовательских параметров и AMPL-стаба исходной задачи.

Система на языке Erlang [Cesarini and Thompson, 2009] распределяет подзадачи по вычислительным узлам и обеспечивает обмен значениями рекордов между процессами адаптеров CBC, выполняемыми одновременно, а также аккумулирует файлы журналов и решений. После того как были обработаны все подзадачи, система сохраняет наилучшее из полученных от адаптеров допустимых решений, если оно есть.

Рассмотрим более подробно отдельные составляющие реализованной системы.

Модуль генерации подзадач

В частично целочисленной задаче линейного программирования часть переменных может принимать только целочисленные значения. Создание подзадачи происходит путем выделения подмножества множества допустимых решений исходной задачи. Для этого к исходной задаче можно добавить дополнительное ограничение на целочисленные переменные. Такое ограничение может делить диапазон значений одной переменной на две части и брать только одну из них, полностью фиксировать одну переменную, быть произвольным линейным ограничением

на целочисленные переменные. Добавляя несколько таких ограничений, можно разбить исходную задачу на множество взаимодополняющих подзадач.

Исполнимый модуль генерации подзадач nlmod принимает на вход AMPL-стаб исходной задачи, часть переменных в котором могут принимать только целочисленные значения, и список параметров разбиения, например, вызов

```
nlmod stub.nl split 20 split 21 split 22
```

создаст 8 подзадач из исходной задачи stub.nl, зафиксировав все значения каждой из трех переменных, если целочисленные переменные номер 20, 21 и 22 принимают по два значения (см. рис. 1).

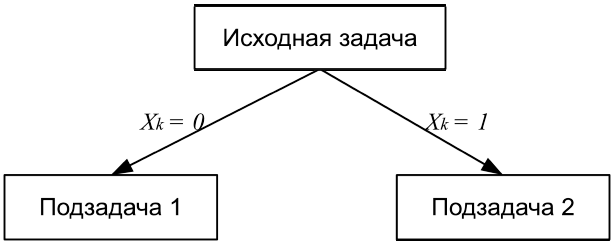


Рис. 1. Разбиение на подзадачи

Адаптер пакета оптимизации CBC

Адаптер пакета оптимизации CBC — это исполнимый модуль на языке C++, обеспечивающий обмен значениями рекордов между пакетом оптимизации и управляющей системой на языке Erlang. Адаптер и управляющая система представлены различными процессами в операционной системе, и поэтому взаимодействие между ними осуществляется с помощью передачи последовательностей байтов через каналы (pipes).

Управляющая система

Управление работой солверов, запускаемых на удаленных хостах, и обмен данными между ними осуществляет система на языке Эрланг. Программы на Эрланге состоят из множества «легких» процессов, взаимодействующих между собой при помощи передачи сообщений. Процессы в системе обмена данными могут быть нескольких типов, названных по имени модуля, содержащего основную часть кода, выполняемого ими.

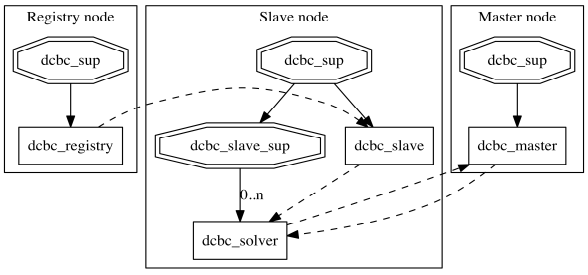


Рис. 2. Зависимости между процессами

Решением задачи управляет процесс dcbc_master, которому при запуске передаются список файлов подзадач и адрес узла, хранящего идентификаторы процессов, осуществляющих запуск пакета CBC на удаленных хостах. Управляющий процесс получает список доступных вычислительных узлов (dcbc_slave) и передает каждому из них один или несколько стабов. Далее вычислительные узлы запускают процессы, управляющие работой CBC (dcbc_solver) через адаптеры CBC. Ниже отдельные компоненты системы рассмотрены более подробно. На рис. 2 представлено дерево супервизоров распределенного приложения. Восьмиугольники — это

процессы-супервизоры, запускающие и контролирующие работу рабочих процессов (прямоугольники), например перезапуская дочерние процессы, завершившиеся аварийно. Стрелки с пунктирными линиями показывают, какие процессы отслеживают состояние других, чтобы выполнить те или иные действия, если отслеживаемый процесс завершился.

Процесс dcbc_solver

«Порты» — это один из способов взаимодействия Эрланга с внешним миром. С их помощью можно запустить внешнюю программу и затем взаимодействовать с ней посредством обмена последовательностями байтов. В свою очередь, внешняя программа может получать и отправлять последовательности байтов, пользуясь парой заданных заранее файловых дескрипторов.

Процессы dcbc_solver — это так называемые процессы-серверы, совершающие определенные действия для каждого полученного сообщения. Основное назначение процессов dcbc_solver состоит в запуске порта к адаптеру СВС с последующим преобразованием форматов между бинарными сообщениями, передаваемыми через порт, и сообщениями, которыми обмениваются процессы в Эрланге. Также данный процесс следит за состоянием адаптера СВС, обнаруживая завершение процесса.

Процесс dcbc_slave

Процессы dcbc_slave запускаются по одному на вычислительный узел. Они обеспечивают запуск процессов dcbc_solver по запросу от управляющего процесса. При этом контролируется число запускаемых процессов, чтобы не допустить перегрузки вычислительного узла.

Процесс dcbc_master

Управляющий процесс реализован модулем dcbc_master. При запуске он получает список имен файлов-стабов и сразу же отдает на выполнение максимально возможное число подзадач, а далее работает как сервер, принимая сообщения от процессов dcbc_solver, работающих на вычислительных узлах. Получив сообщение со значением рекорда, управляющий процесс сравнивает его с текущим и, если рекорд улучшился, рассылает новое значение всем вычислителям. Получив сообщение о завершении одного из вычислителей, управляющий процесс сохраняет решение и отправляет на тот же узел очередную подзадачу. После того как последняя подзадача была обработана, печатается значение текущего рекорда и сохраняется файл с соответствующим рекорду решением.

Обмен значениями рекордов

Улучшив значение рекорда, СВС сообщает об этом через обратный вызов адаптеру, который, в свою очередь, передает сообщение с новым значением рекорда через файловый дескриптор и порт Эрланга своему процессу dcbc_solver, а он получает сообщение с новым значением рекорда и оповещает об этом управляющий процесс (см. рис. 3). Если значение рекорда всей задачи было улучшено, описанная выше схема повторяется в обратном порядке, после чего всем адаптерам становится известно новое значение рекорда.

4. Вычислительный эксперимент

Вычислительные эксперименты проводились на двух вычислительных узлах: 8 потоков на 2 x Intel Xeon E5620 @ 2.40 ГГц, 16 Гб ОЗУ и 4 потока на Intel Core i7-2600K @ 3.40 ГГц, 8 Гб ОЗУ. Всего системе было доступно 12 потоков. Во всех перечисленных ниже запусках пакет

CBC работал в последовательном режиме (без использования встроенной многопоточности). Исходные коды разработанной системы: https://github.com/ssmir/dcbc

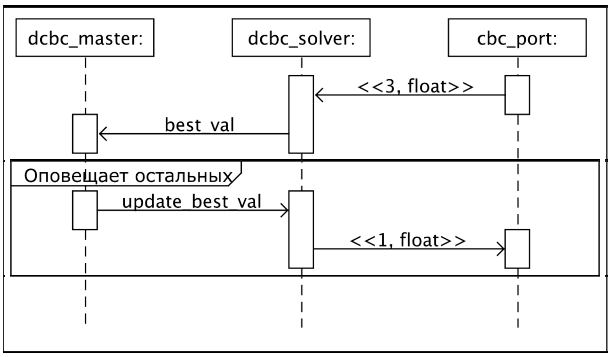


Рис. 3. Обмен рекордами

Тесты проводились на задаче о коммивояжере с числом городов N, равным 80, 90, 100 и 110. Расстояния между городами сгенерированы псевдо-случайным образом.

В качестве ориентира было произведено по одному запуску каждой из исходных задач с однопоточным CBC без какого-либо распараллеливания (время T в таблице 1), а также по одному запуску с пакетом оптимизации SCIP тоже без распараллеливания (время T_s в Таблице 1). Из таблицы видно, что на указанных задачах SCIP работает значительно быстрее CBC. Кроме того, во время тестов было замечено, что многопоточный вариант CBC ведет себя крайне нестабильно, аварийно завершаясь на задачах с N от 100.

Далее было проведено по одному «распределенному» запуску на указанных выше вычислительных ресурсах, результаты перечислены в таблице 2. Здесь M — число зафиксированных переменных, в скобках указано число подзадач, порожденных фиксацией переменных. Фиксировались переменные, соответствующие ребрам наименьшей длины в графе расстояний между городами. Время T_n получено при использовании модуля генерации подзадач, а время T_a — при делении на подзадачи средствами AMPL. Видно, что деление на подзадачи встроенными средствами языка AMPL менее эффективно, чем использование модуля генерации подзадач.

Таблица 1. Запуски исходной задачи

N	T, мин	T_s, мин
80	5,3	1,6
90	20,3	6,0
100	623,8	1,0
110	>10000	74,5

Таблица 2. Разбиение nlmod и AMPL

N	M	T_n, мин	T_a, мин
80	4 (16)	1,5	1,9
90	5 (32)	8,6	11,1
100	6 (64)	1627,0	228,9
110	7 (128)	?	1211,6

Также делались однократные «распределенные» запуски, где варьировалось число фиксируемых переменных (таблицы 3 и 4).

Таблица 3. Запуски для $N = 90$

M	T, мин
0 (1)	20,3
2 (4)	16,0
3 (8)	4,4
4 (16)	8,5
5 (32)	8,6

Таблица 4. Запуски для $N = 100$

M	T, мин
0 (1)	623,8
3 (8)	201,9
6 (64)	1627,0

Выводы

Для задачи коммивояжера представленная в работе схема крупноблочного метода ветвей и границ позволяет получить заметное ускорение по сравнению с однопоточным CBC.

Кроме того, представленная система позволяет решать задачи, на которых не работает многопоточный CBC, завершаясь аварийно.

Однако замечено, что даже в распределенном режиме задача коммивояжера решается не так эффективно, как однопоточным вариантом пакета SCIP. Это требует рассмотреть возможность подключения и SCIP в рассматриваемую в работе систему. Другим возможным направлением развития системы реализация других схем работы, хорошо ложащихся на текущую реализацию, являются, например, запуск пакетов оптимизации не только с различными подзадачами, но и с различными комбинациями настроек солверов, обрабатывающих подзадачи.

Список литературы

Попов Л. Д. Опыт многоуровневого распараллеливания метода ветвей и границ в задачах дискретной оптимизации // Автоматика и телемеханика. — 2007. — № 5. — С. 171–181.

Bussieck M. R., Ferris M.C., and Meeraus A. Grid Enabled Optimization with GAMS}, INFORMS // Journal on Computing. — 2009. — Vol. 21, No. 3. — P. 349–362.

Cesarini F. and Thompson S. Erlang Programming, O'Reilly Media, Inc., 2009.

Forrest J. and Lougee-Heimer R. CBC user guide, INFORMS Tutorials in Operations Research, 2005. — P. 257–277.

Fourer R., Gay D. M., and Kernighan B. W. AMPL: A Modeling Language for Mathematical Programming, second edition, Duxbury Press / Brooks/Cole Publishing Company, 2002.

Koch T. et al. MIPLIB 2010 // Mathematical Programming Computation 3.2. — 2011. — P. 103–163. URL: http://plato.asu.edu/ftp/milpc.html.

Valente P., and Mitra G. A grid-aware MIP solver: Implementation and case studies // Future Generation Computer Systems. — 2008. — Vol. 24, Iss. 2. — P. 133–141.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 727–733

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75

Российские участники добровольных распределенных вычислений на платформе BOINC. Статистика участия

В. И. Тищенко, А. Л. Прочко

Институт системного анализа РАН,
Россия, 117312, г. Москва, проспект 60-летия Октября, д. 9

E-mail: isa@isa.ru

Получено 6 октября 2014 г.

В статье проанализированы полученные авторами статистические показатели активности российских участников добровольных распределенных вычислений на платформе BOINC. Высказаны предположения о причинах недостаточной популярности концепции добровольных распределенных вычислений в среде российских пользователей Интернета.

Ключевые слова: добровольные распределенные вычисления, платформа BOINC, виртуальные сообщества

Russian participants in BOINC-based volunteer computing projects. The activity statistics

V. I. Tishchenko, A. L. Prochko

[1] *Institute of Systems Analysis, Russian Academy of Sciences,*
9, 60-letiya Oktyabrya pr., Moscow, 117312, Russia

Abstract. — The article analyzes the activity statistics obtained by the authors of the Russian participants of volunteer computing platform BOINC. The assumptions about the reasons for the lack of popularity of the concept of volunteer computing among Russian Internet users.

Keywords: volunteer computing, BOINC platform, virtual computing

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 727–733 (Russian).

Введение

На сегодняшний день BOINC (англ. — Berkeley Open Infrastructure for Network Computing) — открытая программная платформа Университета Беркли распределенных вычислений для проектов в области математики, молекулярной биологии, медицины, астрофизики и климатологии. BOINC предоставляет возможность исследователям научных проблем, требующих огромные вычислительные мощности, интегрировать свободные вычислительные ресурсы персональных компьютеров пользователей Интернета. Проект добровольных вычислений состоит из трех основных частей: сервера, веб-сайта и прикладного программного обеспечения (ПО). Процессом вычислений управляет сервер, именно на нем создаются задания для обработки. Для того чтобы подключиться к проекту, пользователь (доброволец) должен скачать и установить на свой ПК стандартный «BOINC-клиент». Подключение к проекту происходит путем указания URL сайта проекта, после чего BOINC-клиент автоматически скачивает с сервера прикладное ПО (ориентируясь на тип ОС и процессора ПК пользователя) и задания для обработки. Затем весь процесс вычислений на стороне пользователя происходит автоматически. В BOINC-клиенте существует гибкая система настроек, позволяющая эффективно задействовать свободные ресурсы ПК и распределять их между BOINC-проектами [Anderson, 2014].

Организаторы проекта, использующие ресурсы компьютеров участников добровольных распределенных вычислений (ДРВ), учитывают ряд факторов, которые, по их мнению, способствуют его активному развитию [Заикин и др., 2014]. Помимо технических задач это условия, мотивирующие пользователей предоставлять вычислительные ресурсы своих компьютеров на добровольной основе и, соответственно, к участию в проекте:

– ощущение причастности к важным научным исследованиям и, соответственно, получению значимых научных результатов;

– командный дух и атмосфера состязательности; участники ДРВ могут объединяться в команды по разным признакам (национальному, региональному и пр.); за выполненное задание пропорционально затраченным вычислительным ресурсам участникам проектов начисляются так называемые «кредиты». Количество «кредитов» является характеристикой, по которой команды и отдельные участники соревнуются между собой;

– информированность о командном и/или индивидуальном участии в проекте; при получении результатов обычно на сайте проекта выкладывается информация об участнике, на ПК которого был получен данный результат.

Все проекты при «запуске» проходят последовательно три статуса: «альфа», «бета» и «релиз». Статусы проектов, использующих платформу BOINC, отображаются на сайте Formula BOINC (http://formula-boinc.org/). На данный момент из 74 активных проектов 54 имеют статус «альфа», 8 — «бета» и 12 — «релиз». Статус «альфа» означает, что проект функционирует и находится на начальном этапе разработки.

На территории СНГ на данный момент активно действуют пять проектов добровольных вычислений на основе BOINC: четыре в России (Einstein@Home, OPTIMA@home и SAT@home, Gerasim@home) и один на Украине (SLinCA@home). Все эти проекты имеют статус «альфа».

Статистический анализ поведения российских участников добровольных распределенных вычислений

Для ознакомления со статистикой проектов, активностью индивидуальных участников ДРВ или командными очками используются сайты, на которых статистические показатели графически визуализируются при посредстве одного из приложения BOINC — API (англ. — application programming interface) [Программное обеспечение…, 2014].

С помощью API можно получить информацию о количестве суммарных очков участника ДРВ, его позиции в мировом рейтинге, его уникальном идентификаторе в системе BOINC,

а также статистику по каждому проекту (количество очков и команда, в составе которой он участвовал в этом проекте).

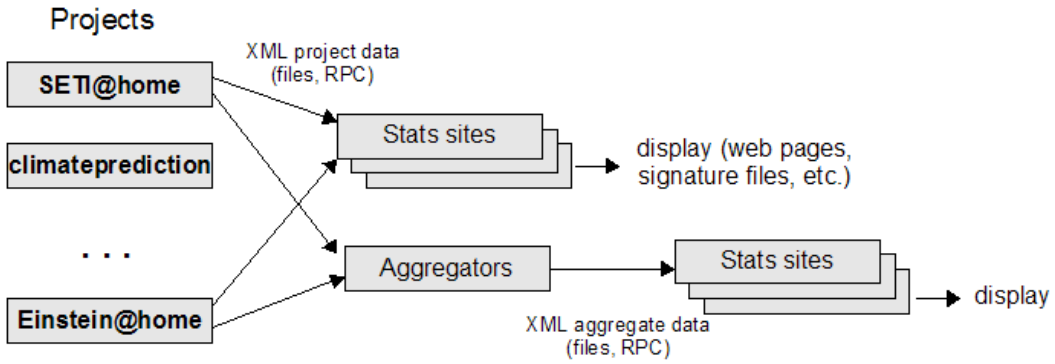


Рис. 1. Схема распространения информации через API системы BOINC

Для проведения статистического анализа активности российских участников ДРВ мы использовали данные, полученные при работе с API BOINC и с сайтом boincstats.com. Скрипт для получения данных и создания соответствующей базы данных с этого сайта был написан на PHP, для хранения данных использовались базы данных MySQL.

Таким образом, была организована база данных (см. рис. 2), содержащая данные по пользователям, которые указали в качестве своей «принадлежности» Россию. Таких пользователей мы рассматривали в качестве «российских пользователей». В базе мы аккумулировали показатели по всем проектам, включая архивные, в которых российские участники ДРВ принимали участие. Это позволило рассчитать показатели, характеризующие закономерности участия российских участников в проектах и командах BOINC.

Получив необходимые данные, можно сделать выводы о российских участниках в проектах на платформе BOINC. Всего на 1 июля 2014 года в системе BOINC находится 45 333 участника из России [Менеджер BOINC аккаунтов, 2014].

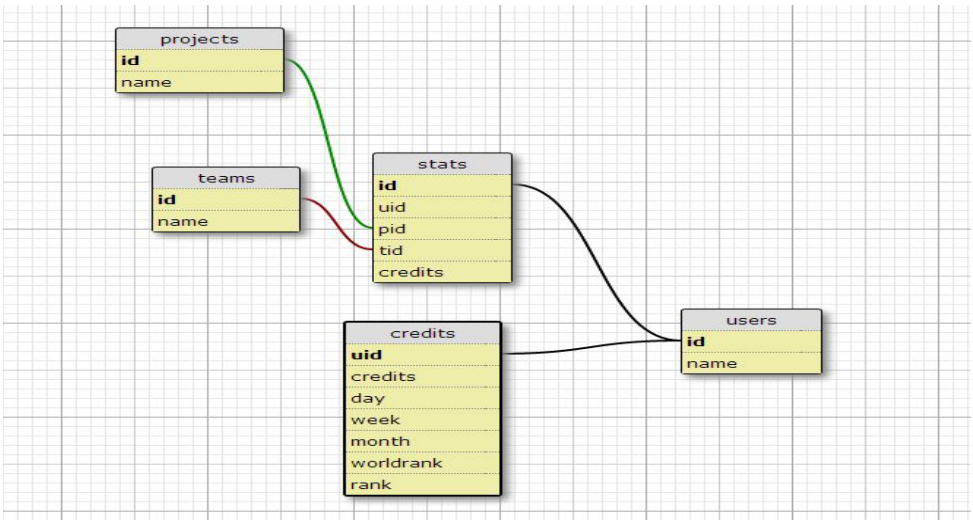


Рис. 2. Структура базы данных «Российские участники BOINC»

Рассмотрим полную статистику BOINC и выделим вклад и участие российского сегмента на 2014 год.

Как видно из таблицы, российские участники ДРВ составляют около 1.6 % от общего числа и привносят 2.5 % очков в систему. По очкам «за последний месяц» наблюдается похожая картина — 2.5 %. Если рассматривать позиции России в рейтинге стран, то она сохраняется уже

4 года. Россия (российские участники в совокупности) в рейтинге стран в настоящее время занимает 11 место по набранным очкам.

	Общая BOINC-статистика	Российский сегмент	%
Суммарные очки	2 117 324 186 166	53 691 650 954	2.5
Участники	2 816 118	45 333	1.6
Очков за последний месяц	56 540 311 021	1 434 955 070	2.5

На первый взгляд такое отставание от участников из США, Германии, Великобритании представляется естественным. Вместе с тем оно вызывает некоторое удивление, поскольку известно, что еще в сентябре 2011 года Россия вышла на первое место в Европе по числу уникальных пользователей, обогнав Германию [Тищенко, Жукова, Попков, 2014]. Еще более интересно, что Россия отстает по очкам в полтора раза от Польши, Франции, Австралии и Чехии. Представляется, что этот показатель, с одной стороны, свидетельствует об атмосфере закрытости и индивидуализма, царящей в среде российских интернет-пользователей, недостаточном распространении идей гражданской науки и краудсорсинга, а с другой, характеризует недостаточность распространенности концепции распределенных вычислений и, соответственно, добровольных распределенных вычислений в России. Еще более низкий показатель у России — так называемый показатель количества очков «на душу населения». По этому показателю Россия занимает одно из последних мест в общем рейтинге из-за низкой активности участников ДРВ.

Страна	Население	Всего участвовало	Активно участвует
США	293 млн чел.	674 000 чел.	>75 000 чел.
Германия	82 млн чел.	293 000 чел.	>18 000 чел.
Польша	38 млн чел.	46 000 чел.	3100 чел.
Чехия	10 млн чел.	33 000 чел.	3400 чел.
Япония	127 млн чел.	61 000 чел.	>9 000 чел.
Россия	140 млн чел.	>45 000 чел.	4500 чел.

Естественно, что существуют проекты, в которых российские участники внесли более 40 % очков. Это непосредственно российские проекты SAT@home и Gerasim@home. Проект SAT@home был запущен 29 сентября 2011 года двумя российскими институтами — ИДСТУ СО РАН (Иркутск) и ИСА РАН (Москва). Проект Gerasim@home стартовал в 2008 году. Сейчас в нем принимают участие 450 пользователей из России.

В то же время если же проанализировать данные за «последний месяц», то мы обнаружим, что акцент участия и, соответственно, предоставления вычислительных мощностей сделан не на российские проекты.

Название проекта	Очки
Collatz Conjecture	268 110 637
MilkyWay@home	257 472 010
GPUGRID	157 612 925
PrimeGrid	90 584 571
Einstein@Home	8 825 750
SETI@Home	59 671 842
World Community Grid	56 244 972
Asteroids@home	29 961 120
Moo! Wrapper	28 914 024
Rosetta@Home	20 087 165
theSkyNet POGS	7 993 382
POEM@HOME	7 028 420
LHC@Home Classic	6 935 433
SAT@home	6 274 717
Gerasim@Home	6 189 205

Очки российских участников в различных проектах (май 2014 года)

Российские проекты SAT@home и Gerasim@home по количеству очков и, естественно, по объему участию российских пользователей находятся на 14 и 15 месте соответственно.

Для того чтобы оценить вклад каждого участника, один из авторов работы провел так называемое включенное исследование и подключил свой компьютер к платформе BOINC. Дата присоединения — 3 марта 2014 года, проекты — «SAT@home» и «World Community Grid». Почти за 3 месяца было набрано 50 000 очков в системе BOINC. За это время в рейтинге российских пользователей удалось подняться с нуля на 30 000 мест, текущая позиция (май) ~12 000. За последний месяц профиль поднялся примерно на 3 000 позиций.

Исходя из того, что данный результат достигнут на компьютере, который использовался только в дневное время, можно сделать предположение, что в системе есть 12 000 участников, которые «позволяли» работать BOINC у себя на компьютере около трех месяцев и более. Ниже приводится статистика по российским пользователям и их суммарным очкам. Правая колонка — какое количество пользователей попадает под данный критерий.

Количество очков	Пользователей
<100 000	36000
>100 000	9300
>1 000 000	2700
>5 000 000	880
>20 000 000	300
>100 000 000	70
>500 000 000	18

Распределение российских пользователей по показателю «Суммарные очки»

Следующая характеристика поведения российских участников, которую мы анализировали, — количество участников в каждом из проектов (включая уже закрытые). Всего на сайте BOINC заявлено 135 проектов. Примерно в половине проектов принимало участие больше 100 человек. Больше тысячи участников приняло участие только в 16 проектах.

Российских участников	Количество проектов
<50	60
50–200	31
200–1000	28
1000–5000	11
>5000	5

Количество российских участников в различных проектах

Исходя из того, что многие участники включены в одни и те же проекты, можно предположить, что коммуникации среди российских участников развиты достаточно хорошо. Существует российский портал www.boinc.ru, который помогает объединить участников различных команд. На форуме имеются разделы (форумы) 12 российских команд, и при появлении «интересного» соревнования команд подключение членов команд происходит достаточно организованно. Помимо форумов коммуникации между участниками ДРВ проходят в социальных сетях. Так, например, в группе в социальной сети «ВКонтакте» [Группа…, 2014] состоят 6 000 пользователей. С учетом того, что всего российских пользователей немногим более 45 000, а активных — 4 500, с помощью соответствующей информационной политики по распространению материалов в сетях можно получить значительный прирост активности российских участников ДРВ.

В этой связи было интересно проанализировать зависимость количества проектов, в которых «работает» российский участник платформы BOINC, и количество его суммарных очков как характеристики активности на портале. Мы предположили, что с увеличением активности пользователя (количества очков) увеличивается и количество проектов, в которых он принимает участие (его «заинтересованность» в концепции добровольных распределенных вычислений).

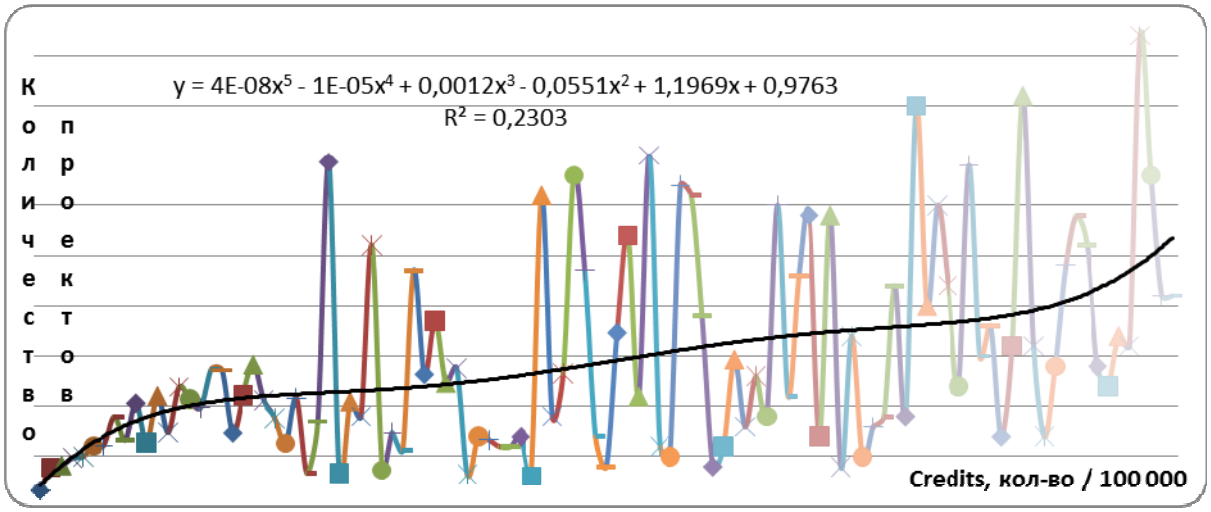


Рис. 3. Показатель среднего количества проектов для каждого диапазона очков с интервалом 100 000 (по оси абсцисс — количество проектов, по осиординат — очки (кредиты))

Действительно, чем активнее участник (чем больше у него общее число очков), тем большее количество проектов он поддерживает, предоставляя мощности своего компьютера. В целом заметное увеличение наблюдается начиная с 500 000–1 000 000 очков.

Как видно из приведенных статистически данных, добровольные распределенные вычисления в нашей стране не очень популярны, несмотря на то что высокоскоростной Интернет становится все более доступным и распространенным. И дело не в технологических проблемах. Как отмечают в своем исследовании среди участников проекта SETI@home Oded Nov с соавторами [Nov, Arazy, Anderson, 2014], важнейшим из мотивов, влияющих на участие в системе добровольных распределенных вычислений, оказывается информация о научных публикациях результатов исследовательских проектов. Те участники добровольных распределенных вычислений, которые были ориентированы на развитие проекта как способа решения фундаментальной научной проблемы и видели результаты своего труда, демонстрировали лучшие показатели в предоставлении вычислительных мощностей своих компьютеров — активность, постоянство, время подключения и т. п.

Что касается российских пользователей Интернета, то, очевидно, их в меньшей степени интересуют развитие фундаментального научного знания, сама атмосфера научного поиска. Конечно, все это требует дальнейшего эмпирического исследования, подтверждения или опровержения этого предположения и высказанных ранее замечаний.

Список литературы

Группа «Программа BOINC и проекты распределенных вычислений». URL: http://vk.com/boinc (дата обращения: 08.08.2014).

Заикин О. С., Посыпкин М. А., Семенов А. А., Храпов Н. П. Опыт организации добровольных вычислений на примере проектов OPTIMA@home и SAT@home. URL: http://omega.sp.susu.ac.ru/books/conference/PaVT2012/full/112.pdf доступен 08.08.2014.

Менеджер BOINC аккаунтов. URL: http://boincstats.com/ru/bam/, http://boincstats.com/ru (дата обращения: 08.08.2014).

Программное обеспечение с открытым исходным кодом для организации добровольных распределенных вычислений и распределенных вычислений в сети. URL: http://boinc.berkeley.edu/trac/wiki/GraphicsApi (дата обращения: 09.08.2014).

Тищенко В. И., Жукова Т. И., Попков Ю. С. Сетевые взаимодействия. Предмет исследования и объект моделирования. — М.: ЛЕНАНД, 2014.

Anderson D. P. Public Computing: Reconnecting People to Science // Presented at the Conference on Shared Knowledge and the Web, Residencia de Estudiantes, Madrid, Spain, Nov. 17–19 2003. URL: http://boinc.berkeley.edu/boinc2.pdf (дата обращения: 08.08.2014).

Nov O., Arazy O., Anderson D. Scientists@Home: What Drives the Quantity and Quality of Online Citizen Science Participation? PLOS One, April 1 2014. URL: http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0090375 (дата обращения: 08.08.2014).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 735–740

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.75, 004.45

Опыт использования puppet для управления вычислительным грид-кластером Tier-1 в НИЦ «Курчатовский институт»

И. А. Ткаченко

Национальный исследовательский центр «Курчатовский институт»,
Россия, 123182, г. Москва, пл. Академика Курчатова, д. 1
E-mail: tia@grid.kiae.ru

Получено 4 декабря 2014 г.

Доклад посвящен организации системы управления кластером при помощи puppet. Рассматриваются вопросы безопасности использования, организации совместной работы, автоматизации различных процессов. Сравниваются различные подходы к созданию puppet сценариев.

Ключевые слова: puppet, автоматизация настройки, совместное управление кластером, варианты использования puppet

Experience of puppet usage for managment of Tier-1 GRID cluster at NRC “Kurchatov Institute”

I. A. Tkachenko

National Research Center “Kurchatov Institute”, 1 Kurchatov Sq., Moscow, 123182, Russia

Abstract. — This report is about creation puppet-based cluster managment system. The report describes security, collaboration and process automatisation aspects of puppet usage. Report compares of different ways for creation puppet manifests.

Keywords: puppet, automation of configuration, joint cluster management, puppet use cases

Вычисления выполнялись на компьютерных ресурсах ЦКП «Комплекс моделирования и обработки данных исследовательских установок мегакласса», поддерживаемого соглашением с Минобрнауки России о предоставлении субсидии № 14.621.21.0006.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 735–740 (Russian).

Любой, кто занимается администрированием более двух одинаково настраиваемых серверов, рано или поздно начинает смотреть на системы, позволяющие автоматизировать процесс настройки. В НИЦ «Курчатовский институт» для этого используется puppet. В этой статье будет рассмотрен опыт его применения на большом количестве серверов (в данный момент управляемая инфраструктура — около 120 Linux серверов (Centos 5 и Centos 6), которые должны работать в режиме 24/7/365.

Автоматизации иногда бывает слишком много

Puppet [Puppetlabs, Web site] удобен тем, что все сценарии хранятся на центральном сервере и автоматически распространяются на клиентские машины. Это означает, что любая пропущенная ошибка за конечное время распространится на все серверы. Puppet предлагает в принципе готовое решение — environment, с помощью которого можно задать поведение клиента для разных случаев. Но это означает, что нужно отдельное хранилище для тестируемых правок — раз, отдельный puppet-master — два, дополнительный рестрат puppet-клиента (с новым environment) — три.

Также существенно усложняется работа системных администраторов при отладке сценариев и написании новых манифестов: каждый системный администратор должен работать в собственном environment и не забывать переключать клиентский puppet-сервер на него и обратно — на основной.

Чтобы избежать этих проблем, можно воспользоваться системой ручного запуска манифестов. Для этого манифесты должны находиться на локальной машине и вызов осуществляется через puppet apply. Это порождает новую проблему, которая раньше решалась puppet автоматически: синхронизация наборов сценариев на управляющем узле (его еще называют master) и клиентах.

Для решения проблемы синхронизации был разработан специальный манифест:

```
class sync inherits site_settings {
  File { owner => 'root', group => 'root', mode  => 0600 }
  $puppet_conf_dir = $site_settings::puppet
  $target_path = $environment ? {
    'production' => 'modules',
    default      => "environments/$environment/modules"
  }

  file { "$puppet_conf_dir/environments":
    ensure => 'directory',
  }

  file { "$puppet_conf_dir/environments/$environment":
    ensure => 'directory',
  }

  each($site_settings::admins) |$admin| {
    if (!defined(File["$puppet_conf_dir/environments/$admin"]))
{

      file { "$puppet_conf_dir/environments/$admin":
        ensure => 'directory',
      }
    }
  }
```

```
  file { "$puppet_conf_dir/$target_path":
      source => [        "pup-
pet://$site_settings::master/private/$site_settings::host_type/
puppet",        "puppet://$site_settings::master/files/",
"puppet://$site_settings::master/puppet/",
      ],
      ensure => 'directory',
      recurse => true,
      purge => true,
      force => true,
      ignore => ['.svn','.*.swp'],
      sourceselect => 'all',
  }

  file { "$puppet_conf_dir/puppet.conf":
    source => "pup-
pet://$site_settings::master/sysconf/puppet/puppet.conf",
    ensure => 'file',
  }
}
```

Поскольку не используется автоматическое распространение изменений на клиентские машины, правки можно делать прямо на мастере, синхронизировать один узел и тестироваться на нем. Затем, например через pdsh, синхронизировать остальные серверы и, опять же через pdsh, запускать нужные классы на всех серверах через `puppet apply -e "include some_class"` [Puppet apply…, Web site]. Также существенно упрощается одновременная работа: тестируемые правки не смешиваются с рабочей версией и окружения для разных администраторов полностью изолированы. Это позволяет избежать таких ошибок, как, например, пропущенная синхронизация для отмены изменений — puppet, при запуске без указания environment будет работать со стабильной рабочей версией.

Установка и перенастройка — одно и то же

Второе, с чем пришлось столкнуться, — первичная установка сервера.
При правильно написанных сценариях большую часть установки можно выполнить на этапе разливки узла, через `%post kikstart` файлов [Anaconda…, Web site].
Чтобы каждый раз не думать, какие классы должны исполняться для настройки узла, разумно вызывать всего один — `install`, который «подумает» за нас примерно таким образом:

```
class install inherits site_settings {
  if $site_settings::mlist {
    $scripts_dir=$site_settings::scripts_dir
    $list="$scripts_dir/install_list.sh"
    $mlist=$site_settings::mlist

    File {owner=> 'root', group => 'root', mode => 0700}
    file {"$scripts_dir":
      ensure => 'directory'
    } ->
    file {"$list":
      content => inline_template("#!/bin/sh -x\n<% mlist.each
do |val| -%>puppet apply -e 'include <%= val %>::install' &&
```

```
\\\n<% end -%>exit 0 || exit 1"),
    } ->
    exec {"$list":
      provider => 'shell',
      logoutput => true,
      timeout => 0,
    }
  }
}
```

Сам же mlist определяется на основе типа узла из двух частей, общей для всех узлов и специфичной для конкретного типа узла:

```
case $fqdn {
  ...
  /^io.*/ :                    { $group='eos-mgm' }
  ...
  /^edg[12].*/ :               { $group='eos-dsi' }
  'somehost':                 { $group='eos-fst' }
  /^sdns.*/ :                  { $group='dcache-namespace' }
  default:                     { $group='test' }
}

...

$mlist_default = ['hosts','dns', 'modules', 'sudo', 'ntp',
'mail','firstboot', 'firewall', 'ssh', 'yum', 'nagios',
'staff', 'logwatch']

...

$mlist_addons = $group ? {
  ...

  'nagios' => ['x509', 'nagios'],

  ...

  default     => ['']
}

$mlist = split(inline_template("<%=
(@mlist_default+@mlist_addons).join(',') %>"),',')
```

Таким образом, как только мы захотели получить настроенный узел, `puppet apply -e 'include install'` сделает это для нас в любой момент времени и из любого места. install-подкласс в каждом классе описывает, то в каком порядке какие подклассы нужно подключать, но об этом ниже. Таким образом, например, настройка ssh — это `puppet apply -e 'include ssh::install'`

Порядок прежде всего

Puppet устроен так, что порядок выполнения команд задается набором зависимостей, а не так, в каком порядке директивы следуют в тексте манифеста [Puppet resource…]. Это одновременно удобно и неудобно. С одной стороны, можно легко задавать сложные зависимости и условия, а с другой — проблемы возникают в самых неожиданных местах.

Например:

```
incldue class1
include class2
include class3
```

в большинстве случаев подключит все в правильном порядке. Но как только `class2` у нас сам делает `include`, например, вот так:

```
include class2.1
include class4
```

говорить о том, что `class4` отработает до `class3`, нельзя. Не спасает даже подключение классов как объектов:

```
Class{"calss1": } -> Class{"clas2":}-> Class{"class3":}
```

`class2` должен выполнится до `class3`, но условия причинности на все подключаемые внутри класса подклассы не распространяются.

Чтобы исключить подобные ситуации, проще всего от последовательного подключения классов перейти к созданию и запуску shell-скриптов.

Как было отмечено выше, подкласс `install` отвечает за правильный порядок выполнения манифеста.

На примере того же ssh:

```
class ssh inherits site_settings {
  File { owner => 'root', group => 'root' }
  $install_list=['sshd', 'keys::install']
  $users=[
    'root',
    't1monitor',
  ]
}

define apply::by_list($list, $sub_class='none') {
  if ( $sub_class == 'none' ) {
    $script="$site_settings::scripts_dir/
      ${caller_module_name}"
    $m_name=$caller_module_name
  } else {
    $script="$site_settings::scripts_dir/
      ${caller_module_name}_${sub_class}"
    $m_name="${caller_module_name}::${sub_class}"
  }
  File {mode => 0700, owner => 'root', group =>'root'}

  file {"$site_settings::scripts_dir":
    ensure => 'directory'
  } ->

  file {"$script":
    content => inline_template ("#!/bin/sh -x \n<% list.each do
|val| -%>puppet apply -e 'include <% if val == @m_name then
%><%= @m_name %><% else %><%= @m_name %>::<%= val %><% end %>'
&& \\\n<% end -%>exit 0 || exit 1")
  } ->
```

```
    exec {"$title":
      command => "$script",
      provider => 'shell',
      logoutput => true,
      timeout => 0,
    }
  }
```

Здесь вместо прямого подключения классов формируется скрипт, последовательно вызывающий `puppet apply` для всех нужных классов.

Такой подход позволяет игнорировать мнение puppet о том, в каком порядке подключать классы, и делать это в том порядке, в котором это нужно нам.

Кроме этого, это позволяет разбивать манифесты на смысловые части и, например, не обновлять rpm, когда нужно только обновить конфигурационные файлы.

Последние версии puppet начали поддерживать директиву `contain` [Relationships…], которая говорит puppet, что в этом классе содержатся вызовы других классов и к ним нужно применить все условия и зависимости, которые были определены для «родительского» класса.

Такой отказ от излишней, в нашем случае, автоматизации puppet позволяет без потери функционала существенно повысить удобство администрирования вычислительного комплекса и избегать многих ошибок.

Список литературы

Anaconda, Kickstart. Web site. URL: http://fedoraproject.org/wiki/Anaconda/Kickstart. 2015.

Puppetlabs. Web site. URL: http://puppetlabs.com. 2015.

Puppet apply man page. Web site. URL: https://docs.puppetlabs.com/references/3.7.0/man/apply.html. 2011.

Puppet resource ordering. Web site. URL: https://docs.puppetlabs.com/learning/ordering.html. 2015.

Relationships and Ordering. Web site.
URL: https://docs.puppetlabs.com/puppet/latest/reference/lang_relationships.html. 2015.

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 741–744**

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.62

Особенности управления данными в DIRAC

О. В. Устименко

Лаборатория информационных технологий, Объединенный институт ядерных исследований,
Россия, 141980, г. Дубна, ул. Жолио-Кюри, д. 6

E-mail: ustimenko@jinr.ru

Получено 4 декабря 2014 г.

Целью данной работы является ознакомление с технологиями хранения больших данных и перспективами развития технологий хранения для распределенных вычислений. Приведен анализ популярных технологий хранения и освещаются возможные ограничения использования.

Основными проблемами развития технологий хранения данных являются хранение сверхбольших объемов данных, отсутствие качества в обработке таких данных, масштабируемость, отсутствие быстрого доступа к данным и отсутствие реализации интеллектуального поиска данных.

В работе рассматриваются особенности организации системы управления данными (DMS) программного продукта DIRAC. Приводится описание устройства, функциональности и способов работы с сервисом передачи данных (Data transfer service) для экспериментов физики высоких энергий, которые требуют вычисления задач с широким спектром требований с точки зрения загрузки процессора, доступа к данным или памяти и непостоянной загрузкой использования ресурсов.

Ключевые слова: распределенное хранение данных, Big Data, программное обеспечение, DIRAC, сервис передачи данных, система управления данными

Features DIRAC data management

O. V. Ustimenko

Joint institute for nuclear researches, Laboratory of Information Technologies, 6 Joliot-Curie st., Dubna, Moscow reg., 141980, Russia

Abstract. — The report presents an analysis of Big Data storage solutions in different directions. The purpose of this paper is to introduce the technology of Big Data storage, prospects of storage technologies, for example, the software DIRAC. The DIRAC is a software framework for distributed computing.

The report considers popular storage technologies and lists their limitations. The main problems are the storage of large data, the lack of quality in the processing, scalability, the lack of rapid availability, the lack of implementation of intelligent data retrieval.

Experimental computing tasks demand a wide range of requirements in terms of CPU usage, data access or memory consumption and unstable profile of resource use for a certain period. The DIRAC Data Management System (DMS), together with the DIRAC Storage Management System (SMS) provides the necessary functionality to execute and control all the activities related with data.

Keywords: Distributed storage systems, Big Data, software framework, Data transfer service, Data Management System

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 741–744 (Russian).

1. Тенденции разработки распределенных систем хранения больших данных

Новые тенденции развития распределенных систем хранения включают системы, которые способны решать проблемы создания композитных приложений, интеллектуальную поддержку поиска и применения услуг, динамическое управление производительностью служб, гибкой интеграции с системами реального времени.

Формализованные знания необходимы для формирования интеллектуальных систем хранения для Big Data. Развитие интеллектуальных технологий аннотирования, поиска и обработки данных относятся к этому понятию. Управление производительностью приложений должно быть выполнено в исполнительной среде. Важно иметь возможность планировать выполнение, используя коммуникационную сеть общего назначения. В вышеописанных системах все функции должны работать в режиме реального времени (наличие центров принятия решений).

Грид-системы должны обеспечивать эффективные методы работы с огромным количеством данных. При этом сами данные могут представлять собой как файлы, так и распределенные базы данных.

При управлении данными следует руководствоваться следующими понятиями:

– целостностью данных; подразумеваются механизмы кэширования и репликации, не приводящие к получению устаревших (неактуальных) данных;

– интеграцией данных, т. е. наличием механизмов для обеспечения одновременной работы с несколькими источниками данных;

– поиск данных — механизмы эффективного поиска данных в грид-системе.

Прикладное задание должно быть в состоянии обратиться к своим данным независимо от фактического местоположения вычислительного ресурса и ресурса хранения [Барсегян и др., 2007].

Примером хорошо организованной распределенной архитектуры хранения большого объема данных является служба передачи файлов (File Transfer Service, FTS), разработанная при поддержке ОИЯИ, а также система контроля Tier3-центров анализа данных, глобальная система мониторинга передачи данных в инфраструктуру WLCG. Различные системы и услуги инфраструктуры используются для организации хранения данных. Наиболее широко используемыми системами хранения данных в грид-среде являются: Castor, DCache, DPM. Для совместимости таких систем был разработан Менеджер ресурсов хранения данных (Storage Resource Manager, SRM) [Кореньков, Ужинский, 2009].

2. Большие Данные в DIRAC

Основная роль DIRAC — интерфейс для интеграции и взаимодействия программного обеспечения (ППО) [Сайт проекта DIRAC, 2014]. DIRAC создает слой между пользователями и ресурсами, которые предлагают общий интерфейс для нескольких разнородных поставщиков (грид-сервисам, кластерам, облакам).

Для реализации качественного управления данными, исходя из основных понятий эффективной грид-системы, существует Подсистема управления данными (Data Management System, DMS), к базовым службам которой относятся службы файлового каталога File Catalog (FC) и каталога метаданных.

DMS предоставляет базовую функциональность для загрузки локального файла в Storage Element (SE) с возможностью регистрации соответствующей реплики в файловый каталог к массовым репликациям данных с использованием службы передачи файлов FTS и извлечением архивных данных для последующей обработки. Для достижения этой функции DMS и Storage Management System (SMS) требуют надлежащего описания привлеченных внешних серверов (SE, FTS и т. д.), а также ряд агентов и связанных серверов, обеспечивающих общий интерфейс.

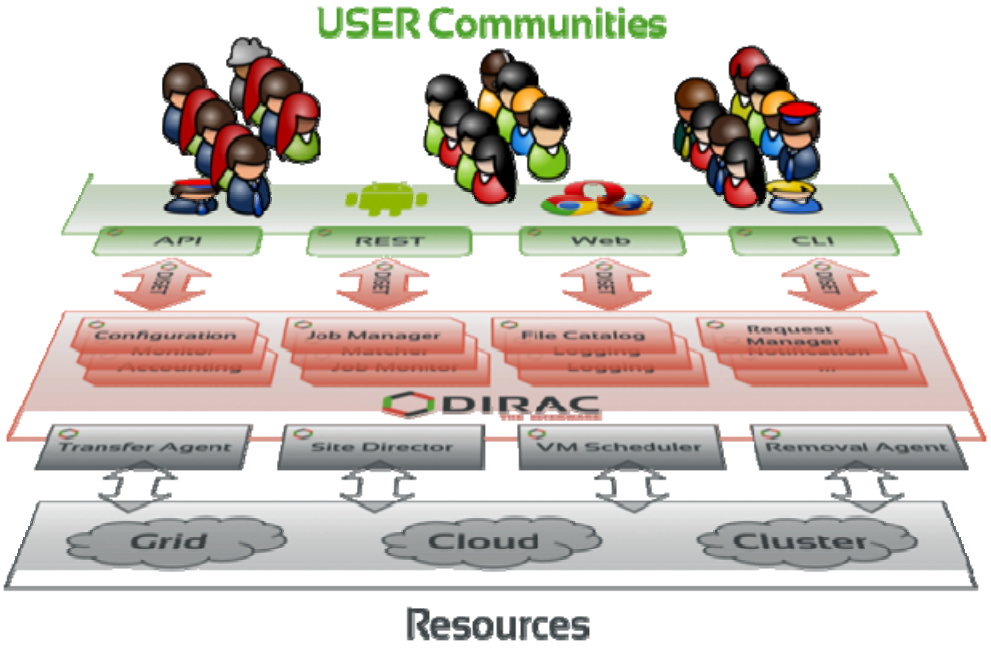


Рис. 1. Распределенная инфраструктура с дистанционным управлением агента — DIRAC

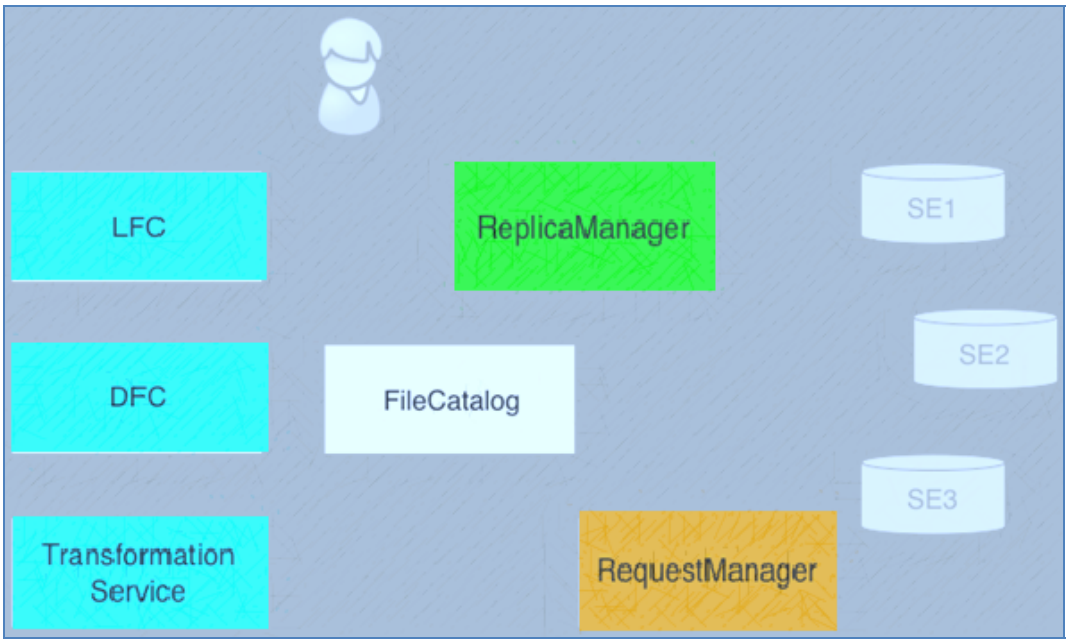


Рис. 2. Модульная структура DMS

DIRAC обеспечивает абстракцию интерфейса SE, который позволяет получить доступ к различного рода ресурсам с помощью единого интерфейса. Каркас ПО DIRAC базируется на модульной архитектуре, которая дает доступ к различным услугам с помощью специализированных плагинов или клиентов [Tsaregorodtsev et al., 2003].

Клиенты инкапсулированы от конкретных методов доступа, что позволяет работать одновременно с несколькими файловыми каталогами различных типов, обеспечивая необходимый уровень абстракции. Эта особенность позволила сторонним пользователям использовать одновременно и каталоги LFC, и каталоги DFC и в ходе доработки сервиса предоставила возможность пользователям легко добавлять и удалять наборы данных, выводить списки всех файлов

(не каталогов), получать данные, переносить свои инструменты и данные из одной службы в другую и оптимизировать модели вычислений.

Особенность DFC по сравнению с LFC в том, что логическое имя файла (LFN) должно быть уникальным для данного файла.

Файловые идентификаторы GUID поддерживаются для тех приложений, которые требуют их, а уникальность GUID может включаться и выключаться в настройках конфигурации.

Контроль доступа реализован в виде ряда плагинов, которые могут быть выбраны в соответствии с потребностями данного сообщества пользователей. При необходимости другие плагины могут быть реализованы для сообществ с особыми потребностями и включены в дистрибутив DFC.

Реплики могут сохраняться 2 способами, а именно:

1) физические имена файлов (PFN) хранятся в виде URL с полным доступом;

2) либо при применении условия, что PFN содержат соответствующее LFN как его конечную принадлежность.

3. Заключение

Одним из основных методов решения проблемы больших данных является использование облачных вычислений, которые предоставляют удаленный доступ к массивам информации и использованию распределенных вычислительных ресурсов, чтобы использовать их. Существует возможность увеличения производительности за счет высокой параллельности и множественности точек доступа в облаке вычислительных систем.

В то же время компьютерные системы для хранения и обработки массивов данных могут быть размещены в специализированных центрах обработки и хранения данных. Эти центры используются не только для удаленного хранения и резервного копирования большого количества научных данных, но и для удаленного доступа к приложениям, которые обеспечивают анализ этих данных.

В качестве альтернативы облачным технологиям возможно использование высокопроизводительных кластеров местных исследовательских центров и грид-технологии для решения задач обработки большого объема научных данных. Преимущество распределенных вычислений является то, что даже обычные основные компьютеры могут быть использованы в качестве отдельных единиц системы грид.

DIRAC является системой «легкого решения» благодаря своей модульности и возможности добавления плагинов. Может использоваться как система распределенного хранения и обработки аналитических и экспериментальных данных. Система управления данными (DMS), вместе с системой управления памятью (SMS), обеспечивает необходимую функциональность для выполнения и контроля всех мероприятий, связанных с данными пользователей научных экспериментов.

Список литературы

Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Технологии анализа данных. DataMining, VisualMining, TextMining, OLAP: Учеб. пособие. 2-е изд. — СПб.: БХВ-Петербург, 2007. — 384 с.

Кореньков В. В., Ужинский А. В. Система мониторинга сервиса передачи данных (FTS) проекта EGEE/WLCG // Вычислительные методы и программирование: новые вычислительные технологии. — 2009. — Т. 10. — С. 96–100.

Сайт проекта DIRAC [Электронный ресурс]. URL: http://diracgrid.org

Tsaregorodtsev A. et al. Computing in High-Energy Physics and Nuclear Physics. — 2003.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 745–752

Ки&М

СЕКЦИОННЫЕ ДОКЛАДЫ

УДК: 004.94

Реализация и применение параллельного алгоритма глобального поиска минимума к задаче оптимизации параметров молекулярно-динамического потенциала ReaxFF

К. С. Шефов[a], М. М. Степанова[b]

Санкт-Петербургский государственный университет,
Россия, 199034, г. Санкт-Петербург, Университетская наб., д. 7-9

E-mail: [a] k.s.shefov@gmail.com, [b] mstep@mms.nw.ru

Получено 30 сентября 2014 г.

Молекулярно-динамические методы, использующие силовое поле ReaxFF, позволяют получать достаточно хорошие результаты при моделировании больших многокомпонентных химически-реактивных систем. Здесь представлены алгоритм поиска оптимальных параметров силового поля ReaxFF для произвольных химических систем, а также его реализация. Метод основан на способе многомерного поиска глобального минимума, предложенном Р. Г. Стронгиным. Алгоритм хорошо масштабируемый и хорошо подходит для работы на параллельных вычислительных кластерах.

Ключевые слова: численное моделирование, молекулярная динамика, потенциал взаимодействия, химически-реактивные системы, реактивное силовое поле, оптимизация параметров, параллельный алгоритм, поиск глобального экстремума

An implementation of a parallel global minimum search algorithm with an application to the ReaxFF molecular dynamic force field parameters optimization

K. S. Shefov[a], M. M. Stepanova[b]

Saint Petersburg State University, 7-9 University nab., St. Petersburg, 199034, Russia

Abstract. — Molecular dynamic methods that use ReaxFF force field allow one to obtain sufficiently good results in simulating large multicomponent chemically reactive systems. Here is represented an algorithm of searching optimal parameters of molecular-dynamic force field ReaxFF for arbitrary chemical systems and its implementation. The method is based on the multidimensional technique of global minimum search suggested by R.G. Strongin. It has good scalability useful for running on distributed parallel computers.

Keywords: numerical simulation, molecular dynamics, reactive force field, chemically reactive systems, parameter optimization, parallel algorithm, absolute extremum search

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 745–752 (Russian).

1. Введение

ReaxFF (Reactive Force Field) [Duin et al., 2001; Nomura et al., 2008] является силовым полем, позволяющим моделировать химические реакции методом молекулярной динамики. Для того чтобы получать адекватные результаты при моделировании химически взаимодействующих систем при помощи ReaxFF, необходимо подобрать набор параметров, от которых зависит данное силовое поле и которые определяются конкретным классом моделируемых систем, например гидриды металлов или углеводороды. Этот процесс является достаточно трудоемким и состоит в предварительном выборе целевой функции, которая зависит от параметров ReaxFF и определяет их оптимальность с последующим поиском минимума этой функции. Целевая функция представляет собой сумму квадратов разностей неких характеристик соединений, входящих в конкретный класс, рассчитанных методами квантовой химии (или взятых из эксперимента) и вычисленных с помощью ReaxFF. Набор характеристик, полученных методами квантовой химии или взятых из эксперимента, называется оптимизирующим набором, а соединения, которым принадлежат характеристики, — моделями оптимизирующего набора. Авторами уже был реализован алгоритм оптимизации параметров на основе метода однопараметрического поиска [Шефов, Степанова, 2014]. Однако этот метод, наряду с достоинствами, такими, как отсутствие требования непрерывности целевой функции и возможность включения в оптимизирующий набор широкого спектра данных, обладает и рядом недостатков. К недостаткам можно отнести следующие факторы: сходимость метода однопараметрического поиска гарантируется лишь в локальный, а не в абсолютный минимум; сложность параллельной реализации метода; очень долгое ожидание сходимости алгоритма, если не известна примерная область расположения минимума. Кроме того, для получения характеристик требуется оптимизация геометрии моделей оптимизирующего набора, причем необходимо выбрать методы, которые гарантированно оптимизируют геометрию в локальный минимум, а не в седловую точку, что замедляет расчет. В данной работе предлагается альтернативный метод, основанный на алгоритме глобального поиска (АГП) Стронгина [Стронгин, 2009]. Он позволяет быстро локализовать область минимума функции, гарантирует сходимость в абсолютный минимум на заданной области определения целевой функции, и алгоритм прост в распараллеливании. АГП успешно применяется для решения многоэкстремальных задач в различных областях, однако для поиска параметров молекулярно-динамического силового поля данный алгоритм используется впервые.

2. Алгоритм глобального поиска

Алгоритм глобального поиска (АГП) позволяет находить абсолютный минимум функции на отрезке и основан на вероятностном подходе.

На основе набора известных значений функции в точках отрезка ищется интервал между соседними точками, на котором абсолютный минимум наиболее вероятен. На этом интервале берется точка, соответствующая математическому ожиданию положения минимума, вычисляется значение функции в ней. Точка добавляется в список известных значений, и происходит переход к следующей итерации. Алгоритм останавливается, когда расстояние между точками отрезка последовательных итераций становится меньше заданного критерия.

Единственным требованием, которому должна удовлетворять целевая функция $g(x)$, — это выполнение обобщенного условия Липшица на всем интервале поиска:

$$\left|g(x_1) - g(x_2)\right| \le K\rho(x_1, x_2), \tag{1}$$

где x_1 и x_2 — любые числа из интервала поиска, K — константа, а ρ — метрика, $\rho(x_1, x_2) = f(|x_1 - x_2|)$, и функция f имеет обратную себе f^{-1}.

Опишем шаги алгоритма [Стронгин, 2009]. Рассмотрим функцию $g(x)$ на отрезке $[a,b]$ вещественной оси. Обозначим $z_i = g(x_i)$ значения целевой функции в точках x_i отрезка $[a,b]$.

В начале алгоритма положим $x_0 = a$, $x_1 = b$ и вычислим значения функции $g(a)$ и $g(b)$ в этих точках. Процедура $k + 1$ итерации состоит в следующем. Пусть x_i и z_i для $i = 0,1,\ldots,k$ нам уже известны из предыдущих k итераций.

1. Перенумеровать точки x_i, $i = 0,1,\ldots,k$, в порядке возрастания значений $a = x_0 < \ldots < x_k = b$.

2. Оценить максимальное абсолютное значение относительной первой разности:

$$M = \max_{1 \le i \le k} \frac{|z_i - z_{i-1}|}{p_i},$$

где $\rho_i = \rho(x_{i-1}, x_i) = f(x_{i-1} - x_i), 0 \le i \le k$.

3. Положить

$$m = \begin{cases} 1, M = 0, \\ rM, M > 0, \end{cases} \tag{2}$$

где $r > 1$ есть заданный коэффициент, параметр алгоритма, который определяется из предположений о коэффициенте K в условии Липшица (1).

4. Для каждого интервала (x_{i-1}, x_i), $0 \le i \le k$, вычислить величину

$$R(i) = m\rho_i + \frac{(z_i - z_{i-1})^2}{m\rho_i} - 2(z_i + z_{i-1}),$$

которая называется характеристикой интервала и определяет вероятность нахождения глобального минимума на этом интервале. Чем она больше, тем больше вероятность.

5. Определить интервал (x_{t-1}, x_t), которому соответствует максимальная характеристика $R(t) = \max_{1 \le i \le k} R(i)$. Если максимальная характеристика соответствует нескольким интервалам, то в качестве t выбирается минимальное число.

6. Положить

$$x_{k+1} = \frac{x_{t-1} + x_t}{2} - \frac{\operatorname{sign}(z_t - z_{t-1})}{2r} f^{-1}\left\{\frac{r|z_t - z_{t-1}|}{m}\right\}. \tag{3}$$

Алгоритм останавливается, когда

$$\rho_t \le \varepsilon \text{ или } x_t - x_{t-1} \le f^{-1}(\varepsilon), \tag{4}$$

где ε — заданное число. Число ε определяет точность приближения к минимуму.

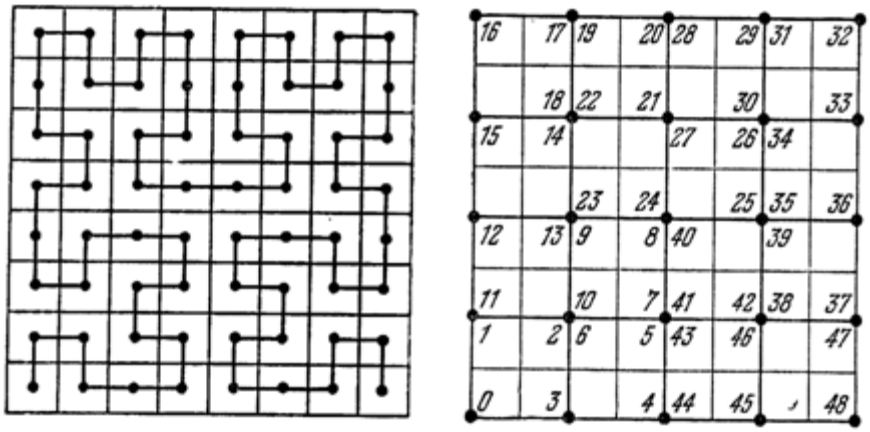
Алгоритм глобального поиска гарантированно сходится к абсолютному минимуму функции на отрезке, если на некоторой его итерации выполняется условие $m > 4K$, где K — константа из обобщенного условия Липшица (1), а m — число из формулы (2).

3. Обобщение АГП на многомерный случай

Функция многих переменных может быть приведена к функции одной переменной при помощи разверток типа кривой Пеано [Стронгин, 2009]. Многомерная область определения (гиперкуб) переводится в отрезок вещественной оси. Подобласти отображаются в одномерные интервалы. В зависимости от степени разбиения m число подобластей, на которые делится исходная область, различно, оно составляет $2^{N(m+1)}$, где N — размерность отображаемой области.

Таким образом, можно, выбрав параметр разбиения m, отобразить точку в центре каждой подобласти (сетку на гиперкубе) в точку в центре каждого интервала на отрезке (сетку на отрезке). Такое отображение будет взаимно однозначным. Вид пеаноподобной кривой для $m = 3$ приведен на рис. 1*а* [Стронгин, 2009]. Многомерная целевая функция, таким образом, будет

определена на одномерной сетке. Однако у такого отображения есть ряд недостатков, поскольку при потребности в увеличении параметра разбиения m (уменьшении шага сетки) придется начинать алгоритм глобального поиска сначала, поскольку новые узлы сетки уже не будут совпадать со старыми. Избежать этого можно, воспользовавшись так называемой неинъективной разверткой, когда в качестве узлов сетки на гиперкубе берутся не центры подобластей, а их вершины. Пример такой развертки (N = 2, m = 3) показан на рис. 1*б* [Стронгин, 2009]. Кривая пробегает гиперкуб по точкам с номерами по возрастанию. Это отображение не взаимно однозначное, как и идеальная кривая Пеано $(m = \infty)$, каждая точка гиперкуба может иметь до 2^N прообразов на отрезке. Также при увеличении m алгоритм глобального поиска можно будет продолжить с использованием уже имеющихся точек.



(а) (б)

Рис. 1. Вид пеаноподобных кривых для двумерия при m = 3

В данной работе используется многомерный вариант АГП с неинъективной разверткой. Метрика ρ в формуле (1) выглядит следующим образом: $\rho(x_1, x_2) = f(|x_1 - x_2|) = \sqrt[N]{|x_1 - x_2|}$. Поиск минимума проводится на конечной сетке на прямоугольном параллелепипеде размерности N (число оптимизируемых параметров ReaxFF), который отображается на отрезок [0, 1]. Число точек сетки определяется параметром разбиения m и равно $2^{(m+1)N} - 2^{mN}$. Поскольку неинъективная развертка не является взаимно однозначным отображением, точка параллелепипеда размерности N может иметь до 2^N прообразов на отрезке, у всех прообразов будет одинаковое значение целевой функции. Стандартный вариант АГП дополняется вычислением прообразов, и точка x_{k+1}, получаемая на каждой итерации по формуле (3), заменяется ближайшей к ней точкой сетки и другими прообразами соответствующей ей точки параллелепипеда. В данном случае минимум может быть найден лишь с точностью до шага сетки. Остановка вычислений происходит, когда точка отрезка, полученная на очередном шаге алгоритма, уже была получена на каком-либо из предыдущих шагов, либо, если ε больше шага сетки, остановка определяется из условия (4).

4. Параллельный вариант АГП

Для распараллеливания алгоритма глобального поиска используется метод вращающихся разверток [Стронгин, Гергель, Баркалов, 2009]. Каждый параллельный процесс оперирует со своей разверткой, повернутой относительно основной на углы $\pm\frac{\pi}{2}$ относительно какой-либо пары направлений. Всего таких поворотов можно сделать $N(N-1)$ для N-мерной области определения функции. Таким образом, всего программа использует $N(N-1)+1$ процессов, каждый из которых выполняет АГП и на каждой итерации сообщает свой результат всем остальным

процессам. Параллельный алгоритм ускоряет сходимость и компенсирует последствия потери информации о близости точек параллелепипеда при использовании развертки.

Опишем алгоритм по шагам. Точки области определения исходной многомерной функции (прямоугольного параллелепипеда) обозначим Y_i, значения целевой функции в них обозначим Z_i, а точки отрезка [0, 1], на который отображается параллелепипед, обозначим X_i.

Для инициализации алгоритма для каждого процесса вычислим $X_0 = 0$, $X_1 = 1$, X_2, — псевдослучайное число на отрезке [0, 1]. В качестве альтернативной инициализации предусмотрено чтение готового списка пар X_i, Z_i из файла. Теперь рассмотрим итерации. Пусть уже имеем точки X_i, $0 \le i \le k$, на [0, 1] и Z_i, $0 \le i \le k-1$, из предыдущих k итераций. Очередная итерация состоит из следующих шагов (для каждого процесса они одинаковые).

1. Отобразим точку X_k отрезка при помощи обратной неинъективной развертки на точку параллелепипеда Y_k. Поскольку для каждого процесса развертки будут разные (повернутые), то Y_k в большинстве случаев будут различными для разных процессов.

2. Проведем обмен Y_k со всеми остальными процессами. После этого каждый процесс будет знать точки Y_k всех остальных процессов.

3. При совпадении Y_k у разных процессов Y_k для процесса с наименьшим порядковым номером будет оставлен неизменным, а для остальных процессов с совпавшими Y_k новое значение Y_k получается из генератора псевдослучайных чисел. Это необходимо, чтобы не вычислять значения функции для одинаковых Y_k и рационально использовать ресурсы.

4. Для Y_k получаем значение целевой функции Z_k. Этот шаг самый трудоемкий.

5. Проведем обмен Y_k и Z_k со всеми остальными процессами. После этого каждый процесс будет знать точки Y_k всех остальных процессов и значения целевой функции Z_k в них.

6. Из Y_k получаем прообразы $X_{k,i}$. Прообразов может быть до 2^N, где N — размерность параллелепипеда. Всем прообразам соответствует одинаковое значение Z_k.

7. Добавим пары $X_{k,i}$ и Z_k в список. Список представляет собой одновременно связный список и красно-черное дерево, что делает быстрым как поиск в нем по значению $X_{k,i}$, так и пробег от начала до конца по всем элементам, причем узлы связного списка будут всегда упорядочены по возрастанию $X_{k,i}$. Каждые 100 итераций проводится сохранение списка в файл на случай непредвиденной остановки или же с целью продолжить расчет в более тонкой сеткой.

8. Используя АГП, описанный в разделе 2, находим X_{k+1} — очередную итерацию.

9. Проверяем, есть ли X_{k+1} уже в списке, если есть, ставим флаг завершения F_ρ, $1 \le \rho \le P$, в значение «истина», $P = (N-1)\,N + 1$ — число процессов в системе. Если X_{k+1} нет в списке, но $|X_{k+1} - X_k| < \varepsilon^N$, где ε — число из условия (4), то флаг завершения F_p также ставится в значение «истина». Обмениваемся флагами с остальными процессами. Если флаги завершения каждого из всех процессов находятся в значении «истина», то алгоритм останавливается, если нет, то происходит переход к следующей итерации.

Если алгоритм остановлен, каждый процесс ищет $X_{\min}$ с минимальным Z_i ($Z_{\min}$) и находит соответствующий ему $Y_{\min}$. Пара $Y_{\min}$ и $Z_{\min}$ будет результатом работы алгоритма. У всех процессов набор пар Y_k и Z_k будет одинаковым, а значит, минимум тоже будет одинаковым. Таким образом, результатом работы алгоритма будет абсолютный минимум целевой функции $Y_{\min}$ и $Z_{\min}$ с точностью до шага сетки либо с точностью до ε из условия (4).

5. Конкретная реализация параллельного многомерного АГП в случае поиска параметров МД силового поля ReaxFF

При оптимизации параметров ReaxFF методом однопараметрического поиска [Шефов, Степанова, 2014] в качестве целевой функции использовалась функция ошибки в виде:

$$\mathrm{Error} = \sum_{i=1}^{n} \left[\left(x_i^{QC/\mathrm{Lit}} - x_i^{\mathrm{ReaxFF}}(p_1, p_2, ..., p_N) \right) / \sigma_i \right]^2, \qquad (5)$$

Эта сумма представляет собой отклонение расчетов ReaxFF от данных оптимизирующего набора. $x_i^{QC/\mathrm{Lit}}$ и x_i^{ReaxFF} — микроскопические характеристики моделей оптимизирующего набора, расчитанные соответственно с помощью методов квантовой химии и методов молекулярной динамики с потенциалом ReaxFF, а именно длины химических связей, валентные углы, эффективные заряды атомов, энергии атомизации и теплоты образования моделей, зависимости энергии кристаллов на ячейку от объема ячейки. σ_i — весовые коэффициенты, имеющие размерность x_i, которые выбираются исходя из ожидаемой разности квантовых расчетов и ReaxFF в конце оптимизации. x_i^{ReaxFF} зависят от параметров потенциала $p_1, p_2, ..., p_N$.

Однако функцию (5) нельзя использовать как целевую для АГП, поскольку она не удовлетворяет условию Липшица (1). Характеристики x_i^{ReaxFF} являются результатами процедуры оптимизации поля ReaxFF по координатам атомов при фиксированных параметрах $p_1, p_2, ..., p_N$, следовательно, возможны скачки. Поэтому для АГП в качестве целевой мы используем другую функцию:

$$\mathrm{Error} = \sum_{k=1}^{L} \sigma_k \left| U_k^{QC} - U_k^{\mathrm{ReaxF}F} \right| + \sum_{k=1}^{L} \sigma_{L+k} \sqrt{\sum_{a=1}^{A} \sum_{i=1}^{3} (F_{kai}^{QC} - F_{kai}^{\mathrm{ReaxFF}})^2} \qquad (6)$$

Здесь U_k — потенциальные энергии моделей оптимизирующего набора, F_{kai} — компоненты сил, действующих на атомы каждой модели, L — число моделей оптимизирующего набора, A — число атомов в моделях, σ — весовые множители. Индексы QC и ReaxFF означают, что характеристики были получены соответственно методами квантовой химии и молекулярной динамики с ReaxFF. U_k^{ReaxFF} и $F_{kai}^{\mathrm{ReaxFF}}$ зависят от параметров ReaxFF, причем, как следует из явных формул ReaxFF [Nomura et al., 2008], эта зависимость имеет непрерывную первую производную. На этом основании можно утверждать, что функция (6) удовлетворяет условию Липшица (1).

Для вычисления целевой функции вида (6) оптимизация геометрии моделей не проводится, в отличие от функции (5) модели берутся статические, положения атомов в них всегда фиксированы. Это избавляет от необходимости проводить оптимизацию потенциала по координатам атомов, однако взамен добавляет необходимость в наборе статических конфигураций каждой оптимизирующей модели около ее положения равновесия, полученного методами квантовой химии (деформированные в различных направлениях молекулы).

Достоинствами метода глобального поиска являются гарантированная сходимость в абсолютный минимум на заданной сетке, простота параллельной реализации, отсутствие необходимости оптимизации геометрии моделей на каждой итерации. К недостаткам АГП можно отнести то, что ресурсоемкость растет экспоненциально с ростом числа одновременно оптимизируемых параметров ReaxFF.

6. Выбор параметров, наиболее сильно влияющих на целевую функцию

Экспоненциальный рост сложности алгоритма с ростом числа параметров потенциала вынуждает отказаться от одновременного поиска сразу по всем возможным параметрам. Возникает необходимость выбора тех из них, которые наиболее сильно влияют на целевую функцию. Выполнив поиск сначала по более важным параметрам, можно потом провести его по менее важным и т. д. В качестве критерия важности берется величина коэффициентов корреляции изменения слагаемых целевой функции (6) и изменения параметров ReaxFF при случайной выборке вектора параметров в заданных пределах.

В данной работе была создана и используется программа, которая выстраивает параметры по убыванию максимальной величины корреляции изменения конкретного параметра и изменения конкретного слагаемого в формуле целевой функции. В дальнейшем поиск методом АГП

выполняется по полученным группам, при необходимости можно сделать пересортировку и повторить проход по списку.

7. Реализация АГП для поиска параметров ReaxFF. Проверка работоспособности

В этом разделе приводятся результаты тестирования программы оптимизации параметров в последовательной и параллельной реализации при разном числе параметров и сравнение времени выполнения. Параллельный алгоритм глобального поиска для параметров ReaxFF реализован на языке C++ под ОС Linux. Для обмена сообщениями используется стандарт MPI (Message Passing Interface), в качестве реализации MPI используется библиотека MPICH2 версии 1.4.1p1. В качестве реализации потенциала ReaxFF используется молекулярно-динамический пакет LAMMPS [Пакет LAMMPS], собранный в качестве библиотеки. Программа выполнялась на узлах кластера с конфигурацией Intel Xeon 3 GHz 64 bit, 4 ядра, HTT, LAN 1 Gb/s, Scientific Linux 5.

В таблице 1 представлено сравнение времени работы программы при последовательном и параллельном запусках, при $N = 2$ и $N = 6$ оптимизируемых параметрах. $P = N(N - 1) + 1$ — число процессов в параллельном варианте. Под последовательным запуском программы здесь понимается запуск многомерного АГП на одном процессоре с одной разверткой (см. раздел 3). Под параллельным запуском понимается запуск параллельной программы на нескольких процессорах, для $N = 2$ это будут 3 процесса и один узел кластера, для $N = 6$ это будут 31 процесс и 4 узла кластера. В качестве оптимизируемых параметров были взяты параметры, наиболее сильно влияющие на целевую функцию, полученные программой, описанной в разделе 6. Параметр алгоритма r во всех случаях был положен равным 4. Параметр разбиения для разверток m был взят равным 3 (9 точек сетки на параметр).

Таблица 1. Сравнение времени работы программы при последовательном и параллельном запусках

	Послед. АГП 2 параметра	Парал. АГП 2 параметра $P = 3$ процесса 1 узел кластера	Послед. АГП 6 параметров	Парал. АГП 6 параметров $P = 31$ процесс 4 узла кластера
Среднее время одной итерации (T)	10,2 с	11,1 с	12,5 с	40 с
Среднее время вычисления целевой функции в одной точке (T/P)	10,2 с	3,6 с	12,5 с	1,3 с
Время сходимости алгоритма	4 ч	1,6 ч	240 ч	27 ч
Число итераций для сходимости	1413	528	69600	2400

Как видно из таблицы 1, при оптимизации двух параметров ускорение параллельной версии программы в сравнении с последовательной составило 2,5 раза, а при оптимизации шести параметров — почти 9 раз. Более того, параллельный вариант сходится быстрее и по числу итераций, поскольку на каждой итерации в параллельном варианте получается в среднем в P раз больше пар точка отрезка — значение функции, чем в последовательном.

Была также проведена проверка правильности выдаваемого алгоритмом результата для случая двух параметров ($N = 2$) путем расчета значения функции во всех 65536 точках сетки.

Результат алгоритма совпал с минимальным значением функции на сетке. Для $N = 6$ ввиду большого числа узлов сетки (около 16 миллионов) такая проверка не делалась.

Еще одной характеристикой производительности алгоритма является выигрыш во времени по сравнению с вычислением функции во всех точках сетки. Параллельный алгоритм при $N = 2$ выдал результат, вычислив значение функции только в 4 тысячах узлах сетки из 65 тысяч, что означает выигрыш в 16 раз. Параллельный алгоритм при $N = 6$ выдал результат по 3,5 млн точек из 16 млн, то есть выигрыш составил 4,6 раза.

Заключение

В данной работе реализован параллельный многомерный алгоритм глобального поиска Стронгина применительно к оптимизации параметров МД потенциала ReaxFF. Также разработан и реализован алгоритм упорядочивания параметров потенциала по степени влияния на целевую функцию на основе корреляции, который позволяет выбирать для оптимизации наиболее критичные параметры. Параллельная версия программы дает заметное ускорение в сравнении с последовательной версией, что было проверено запуском алгоритма на узлах вычислительного кластера. АГП позволяет за приемлемое время локализовать абсолютный минимум сложной многопараметрической целевой функции, после чего можно продолжить оптимизацию либо другим алгоритмом, либо тем же АГП. Таким образом, применение АГП значительно улучшает эффективность поиска параметров потенциала по сравнению с однопараметрическим поиском.

Список литературы

Пакет LAMMPS. URL: http://lammps.sandia.gov (дата обращения: 20.09.2014).

Стронгин Р.Г., Гергель В. П., Баркалов К. А. Изв. вузов. Приборостроение. — 2009. — Т. 52, № 10.

Стронгин Р. Г. Численные методы в многоэкстремальных задачах, — М.: Наука, 1978.

Шефов К. С., Степанова М. М. Реализация алгоритма оптимизации параметров молекулярно-динамического потенциала ReaxFF // Программные продукты и системы. — 2014, № 2(106). — С. 141–145.

Duin Van A. C. T., Dasgupta S., Lorant F., Goddard W. A. III ReaxFF: a reactive force field for hydrocarbons // Journ. of Physical Chemistry A. — 2001. — Vol. 105. — P. 9396–9409.

Nomura K., Kalia R. K., Nakano A., Vashishta P. A scala-ble parallel algorithm for large-scale reactive force-field molecular dynamics simulations // Computer Physics Communications. — 2008. — Vol. 178. — P. 73–87.

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 753–758**

Ки&М

СТЕНДОВЫЕ ДОКЛАДЫ

УДК: 004.75

Cloud Computing for Virtual Testbed

A. B. Degtyarev[1,a], Myo Min Swe[2,b], Wunna Kyaw[2,c]

[1] St.Petersburg State University, Saint-Petersburg, Russia
[2] St.Petersburg State Marine Technical University, 3 Lotsmanskaya Str., St.Petersburg, 190008 Russia

E-mail: adeg@csa.ru, beltson@gmail.com, cwna.ru.pol@gmail.com

Получено 1 октября 2014 г.

Now days cloud computing is an important and hot topic in arena of information technology and computer system. Several companies and educational institutes have been deployed cloud infrastructures to overcome their problems such as easy data access, software updates with minimal cost, large or unlimited storage, efficient cost factor, backup storage and disaster recovery and several other benefits compare with the traditional network infrastructures. There we study cloud computing technology for marine environmental data and processing. A cloud computing of marine environment information is proposed for the integration and sharing of marine information resources that its aim to perform empirical requiring numerous interactions with web servers and transfers of very large archival data files without affecting operational information system infrastructure is highly desirable. In this paper, we consider about the cloud computing for virtual testbed to minimize the cost. That it related with real time infrastructure and the take advantages of the cloud computing technology.

Облачные вычисления для виртуального полигона

А. Б. Дегтярев[1], Мьё Мин Све[2], Вунна Киав[2]

[1] Санкт-Петербургский государственный университет, Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35
[2] Санкт-Петербургский государственный морской технический университет, Россия, 190008, г. Санкт-Петербург, ул. Лоцманская, д. 3

В настоящее время облачные вычисления являются важной и актуальной темой в ИТ. Многие компании и учебные заведения развертывают облачные инфраструктуры, чтобы преодолеть свои проблемы, такие как легкость доступа к данным, обновление программного обеспечения с минимальными затратами, возможности неограниченного хранения данных и ряд других преимуществ по сравнению с традиционными сетевыми инфраструктурами. В работе рассматривается применение технологий облачных вычислений при моделировании морской среды и обработке данных. В данном случае облачные вычисления предлагается для интеграции и совместного использования морских информационных ресурсов. В статье облачные вычисления рассматриваются как средство снижения затрат при организации виртуального полигона в морских исследованиях.

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 753–758 (Russian).

1. Introduction

Cloud computing is very new concept of computing. It has been thought and realized as a future generation of computing with the promise of revolutionizing the ICT services by transforming computing into a ubiquitous utility. The rapid change in technology, frequent updates in users' demand and associative high cost in managing users' demand and available technology have been a big challenges in the history of computing. The disparity in the cost of hardware and software has made the big obstacles in computing. Buying the required infrastructures, operating it and upgrading it are the great issues in computing field. The users' community was worried and expecting something miracle to overcome these challenges. They were expecting the computing facilities and required resources should be available on demand like other utility facilities. Their expectations have been addressed by cloud computing where computing is treated as a utility and computing facilities would be provided on demand. The cloud computing is the computing in cloud where as the cloud refers to virtual servers; distributed hosting, and shared resources available over Internet. The users can access the cloud for available service through the web browser and Internet connectivity. In cloud computing, software, hardware and network are the main actors. The collective effort of these actors make the cloud computing. We can also visualize the cloud as a cluster of computers which are based upon distributed system that provide services in real time over a network and these characteristics make the possibility of using cloud computing in ICT based system. ICT based system needs to provide required services to required users on the required time. There are many ICT based systems using in different organizations.

2. Basic characteristics

Cloud computing is the collection of scalable, virtualized resources, which is capable of hosting application and providing required services to the users and can charge as per the uses like utility. A could computing has also been defined by Ian Foster et al, as a distributed computing paradigm that is driven by economies of scale, in which pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms and services are delivered on demand to external customers over the Internet [Grossman, 2009]. It is a model for enabling convenient, on- demand network access to a shared pool of configurable computing resources such as networks, servers, storage, application that can be rapidly provisioned and released with minimal management effort or service provider interaction [ISACA …, 2014].

The main goal of cloud computing is to provide ICT services with shared infrastructure and the collection of many systems. In cloud computing every facility is provided in term of service. It provides infrastructure as a service, software as a service, platform as a service, network as a service, and data storage as a service. The main philosophy of cloud computing is to provide every required things as a service. In order to be clearer, the services in the cloud can be shown in term of layer structure.

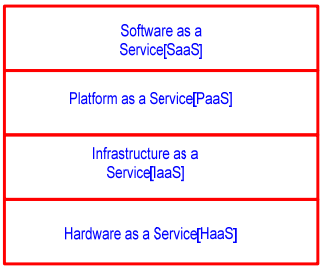


Fig. 1. Layers in cloud computing

The Figure 1 depicts the services in terms of the layers. The foundation or lowest layer is HaaS(Hardware as a service) that provides every required hardware. The layer just above this is IaaS (Infrastructure as a service) that takes the care of required infrastructure. Similarly, we have two top

most layer i.e. PaaS(Platform as a service) and SaaS(Software as a service) which provide the required platform and required software to the users. In general, the bottom three layers are used by developer whereas the top layer is used by user communities. The use of such services by different users is given in Section 2.1 with UML use case diagram.

3. Cloud Computing Technology

Cloud computing has the set of various supporting technologies. A single technology is not sufficient to provide smooth functioning of cloud computing. In cloud computing, it is assumed that software and hardware services are essentially stored in the web servers, the cloud rather than spread over the single computers connected in Internet [Aymerich et al., 2008]. The advent and maturity of virtualization technology enables compute clouds which make demand for cloud that can store, we call this as storage cloud. The storage cloud and compute cloud together make cloud infrastructure as given in layer 2 of above Figure 1. This cloud infrastructure makes the possibility of platforms and finally it supports for the application in the top layer of above figure. More than one cloud makes the cloud aggregators. It needs the cooperation of many available technologies. They are given in Figure 2.

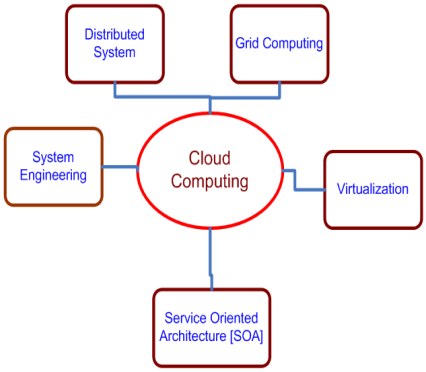


Fig. 2. Technology in Cloud Computing

In Figure 2, it has shown the required technologies in cloud computing. *Distributed system* is one of the technologies to provide the support in cloud computing because the nature of cloud is distributed and it needs a special kind of collaboration and cooperation among the computing nodes. *Grid computing* is another one which is required to solve a common problem with the help of clustering computing nodes in clouds. There are still many similarities between grid and cloud computing. *Virtualization* can be used in cloud to virtualize all the physical resources. Virtualization is a technology that combines or divides computing resources to give the environment for one or many operating systems using hardware and software partitioning. *Service oriented architecture (SOA),* which is a kind of software architectures, is required to provide all the facilities in terms of services. The main philosophy of cloud computing is based upon the Service Request and Service Provide. The components in SOA communicate each other with standard SOAP (Simple Object Access Protocol) which is mostly written in XML code that makes the possibility of interacting among the heterogeneous environment. This is the reason of using SOA in cloud computing. The last but not the least is system engineering. System engineering and its approach is the basic foundation for cloud computing. Cloud has to be thought in terms of system and it has to go through system development life cycle. All phases in system development life cycle is crucial in cloud computing.

4. Types of Cloud and Features

In general, cloud is classified into two types. They are as: *private cloud* and *public cloud.* The classification is based upon the degree of authentication and authorization in cloud resources. The re-

sources in public cloud can be access by everyone where as the resources in private cloud cannot be accessed by everyone. There is a restriction. Mostly, the big organization makes the cloud private for their internal use. Sometimes there is a need of controlling the access of the resources in cloud and at the same time it is required to give access to limited users outside the organization, which is possible with another type of cloud i.e. *hybrid cloud*. It has the features of both private as well as public cloud. It is a composition of two or more clouds such as private cloud, public cloud and sometimes community cloud that remain unique entities but are bound together by standards technology that enables data and application portability. Amazon's EC2, S3 are the examples of public or hosted cloud and Google uses GFS, MapReduce and BigTable as private cloud.

The popularity of cloud computing is because of the available features in cloud computing. There are many identified features but we reveal only prime features that are pertinent to all system.

Scalability: It is one of the features in cloud computing in which the use of multiple resources can be expanded whenever it is needed and can be reduced if it is not needed. Both expansion and reduction of resources can be done without paying extra money or concerning of extra infrastructure. Considering the constrained capacity, cloud computing offers increased flexibility on scalability for evolving ICT needs.

Low Risk: In cloud computing, there is nothing to purchase except the required services. It is also possible to cancel the services if it is not needed. All the services are kept in the cloud with proper testing mechanism. These features reduce the risk of buying the product and unreliability of any product.

Reliability and Survivability: Virtualization is one of the technologies that are used in cloud computing. As we know, virtualization increases the reliability, increases the survivability and also reduces the downtime.

Availability: Cloud computing provides maximum availability on the system because of multiple cloud. This means if one cloud goes down then another takes the charge immediately. Fault tolerance capability is very high in cloud computing. Cloud service providers have enough infrastructure and bandwidth to accommodate business needs for high speed access, storage, computing and applications. Since the service provider use the features such as load balancing the systems are not overloaded and not service delayed.

Low Cost: "Pay as Work" philosophy of cloud computing makes the cost very low. If servers are not requires, we can turn it off and if it is required then turn it on again. There is no need to buy any infrastructure or pay for it in advance and also no capital expenses are required.

Ubiquitous Network Cloud can provide the capabilities over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms such as cell phone, tablet, notebook, and I-Phone etc.

5. Prospective area and future challenges

Despite of huge welcome by many communities, there are still many doubts to be removed, questions to be solved and issues to be addressed in cloud computing. It has given the opportunity for students, researchers and research-based organizations to conduct respective researches and find the concrete solutions. We identify the following prospective areas in which researches are required.

Cloud security: Security is one of the prime issues in cloud computing. The cloud users are still reluctant to share or give their critical data to cloud provider. Still there is no convincing security mechanism in cloud computing. Perimeter security and contain (data) security could be the area for the research.

Cloud computing in ubiquitous computing: There are so many ubiquitous devices now days. People use their cell phone, notebook, I-phone and other devices for their daily activities. Still there are so many complain such as low latency rate, unreliability, unavailability of services, bandwidth congestion etc. A research can be done to solve these issues. Along with these, the issues of pervasive software can also be solved using cloud computing.

Interoperability among cloud provider (Data Lock-In): At present, there are many cloud service providers and they have standard APIs to interact each other but tomorrow there may be many cloud service providers. One user may use more than one cloud service providers or one cloud service provider may have many different types of users. Still there are much works required for the standardization in cloud's APIs. Hence there is a need of details interoperability among them.

Cloud computing in traffic management: Current GPS system of traffic management can be replaced by cloud computing based traffic management. A good comparison can be made between existing GPS based traffic management and proposed cloud-based traffic management.

Cloud computing in e-business: Electronic business has blanketed entire world. People around the world use maximum e-business for purchasing and their demand of using it is keep on changing. The service provider has a very tough time to update their system to meet the people's demand in regular basis. A significant research can be done in using cloud computing in electronic business to address all these problems.

Automated Efficient Backup and Recovery in Cloud computing: Backup the critical data to the cloud and recovery from it whenever it needs are one of the activities of cloud. Efficient recovery plan in case of disaster is also one of the areas in which researches can be done.

Cloud based Software Testing: Software testing has been very sensitive task in software development process especially in embedded software testing. Rapid change in business process, increasing complexity, shorter time to market have made software testing more complex , time consuming and expensive. The cost of finding and removing bugs has become exponential and there is a need to invest huge amount in infrastructure to set up machines for analyzing source code, developing test cases, running the tests etc. Hence, we can analyze and compare between traditional-based software testing and cloud-based software testing with the resources provided by the cloud service providers such as simple storage solution(S3), elastic computing cloud(EC2) etc.

Quick response in resource scaling up and down: Scalability is one of the features of cloud computing. Sometimes the system in the cloud service provider does not response quickly. A research for quick response as per the load balance can be done without affecting service level agreement (SLA).

In cloud computing there are three main models, i.e. computation, storage and network model. These models do require significant research to make them more effective than today [Armbrust et al., 2009].

Along with above mentioned areas, a special thought can be given in each service of cloud computing. There are some unidentified problems in each service. These hidden problems can be identified and proper solution can be suggested.

Conclusion and future work

Virtual testbed can be deployed within cloud environment to minimize the implementation cost and achieve more benefits during communication. The cloud computing platform is a most reliable and cost less solution for real time infrastructures, including applications oriented to consumers, applications supporting the enterprise, and applications for large scale science. This trend is expected to continue for the foreseeable future. However, in many ways research in cloud computing is still immature. There is limited understanding of basic issues, such as exploiting data locality, load balancing, and identifying nodes and collections of nodes that are damaging the overall performance of an application. We also exposed potential areas of cloud computing in which interested researcher can contribute their work to make cloud computing more better in coming days.

References

Aymerich Maria Francesco, Fenu Gianni, Surcis Simone. An Approach to Cloud Computing Network // IEEE-2008.

Armbrust Michael et al. Above the Clouds: A Berkeley View of Cloud Computing // Technical Report No. UCB/EECS-2009-28. — http://www. eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html, Feb, 2009.

Aronson Jesse S., "Making IT a Positive Force in Environment Change", IEEE-2008.

Bianco, Phil., Kotermanski, Rick., Merson, Paulo, "Evaluating a Service-Oriented Architectures", Technical Report, SEI, Carnegie Mellon University, September-2007.

Estevez Elsa, Janowski Tomasz, "Building a Dependable Messaging Infrastructure for Electronic Government". UNU/IIST Report No 368. April 2007.

Foster Ian, Zhao Yong, Raicu Ioan, Lu Shiyong, "Cloud Computing and Grid Computing 360-Degree Compared", IEEE.

Greenberg, Albert, Lahiri, Parantap, Maltz, David A., Patel, Parveen, Sengupta, Sudipa, "Towards A Next Generation Data Center Architecture: Scalability and Commodigitization", ACM-2008.

Grossman Robert L. The case of cloud computing // IEEE-2009.

Heddaya, Abdelsalam, Helal Abdelsalam Helal, "Reliability, Availability, Dependability and Performability: A User Centered View", Boston University, Dec 04,1996.

Heek Richard, "Implementing and Managing E-Government", Vistaar Publication- 2006.

ISACA, "Cloud Computing: Business Benefits with Security, Governance and Assurance Perspectives," www.isaca.org

Janssen Marijn, Cressworld Anthony, "Enterprise Architecture Integration in E-Government", 38th Hawaii International Conference on System Sciences- IEEE- 2005.

Kencl, Lucas, Schwarzer, Christian, "Traffic Adaptive Packet Filtering of Denial of Service Attack", IEEE- 2006.

Leavitt Neil, "Is Cloud Computing Really Ready for Prime Time?" IEEE-2009.

Lin, Mei, Yongsen, Xu, "An Adaptive Dependability Model of Component-Based Software", ACM SIGSOFT, Software Engineering Notes Volume 28 No 2, March 2003.

Murugesan San, "Harnessing Green IT: Principles and Practice.", IEEE-2008.

Youseff Lamia, Butrico Maria, Da Silva Dilma, "Towards a Unified Ontology of Cloud Computing"Grid Computing Environments Workshop 2008, GCE08'.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 759−766

Ки&М

СТЕНДОВЫЕ ДОКЛАДЫ

УДК: 004.94

Моделирование поведения опционов. Формулировка проблемы

А. В. Богданов[a], В. В. Мареев[b], Э. А. Степанов[c], М. В. Панченко[d]

Санкт-Петербургский государственный университет,
Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

E-mail: [a] bogdanov@csa.ru, [b] map@csa.ru, [c] e.an.stepanov@gmail.com, [d] stpank@mail.ru

Получено 27 октября 2014 г.

Объектом исследований является создание алгоритма для расчета цен большого числа опционов с целью формирования безрискового портфеля. Метод базируется на обобщении подхода Блэка–Шоулза. Задача состоит в моделировании поведения всех опционов, а также инструментов их страхования. Для данной задачи характерен большой объем параллельных вычислений, которые требуется производить в режиме реального времени. Проблематика исследования: в зависимости от исходных данных используются разные подходы к решению. Существует три метода, которые могут использоваться при разных условиях: конечно-разностный метод, метод функционального интегрирования и метод, который связан с остановкой торгов на рынке. Распределенные вычисления в каждом из этих случаев организуются поразному и требуют использования различных подходов. Сложность задачи также связана с тем, что в литературе ее математическая постановка не является корректной. Отсутствует полное описание граничных и начальных условий, а также некоторые предположения, лежащие в основе модели, не соответствуют реальным условиям рынка. Необходимо дать математически корректную постановку задачи и убрать несоответствие между предположениями модели и реальным рынком. Для этих целей необходимо расширить стандартную постановку за счет дополнительных методов и улучшить методы реализации для каждого направления решения задачи.

Ключевые слова: финансовая математика, ценообразование опционов, азиатский опцион, корректная постановка, граничные условия

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 759–766

Ки&М

СТЕНДОВЫЕ ДОКЛАДЫ

Modeling of Behavior of the Option. The Formulation of the Problem

A. V. Bogdanov, V. V. Mareev, E. A. Stepanov, **M. V. Panchenko**

Saint Petersburg State University, 35 University ave., Peterhof, St. Petersburg, 198504, Russia

Abstract. — Object of research: The creation of algorithm for mass computations of options‘ price for formation of a riskless portfolio. The method is based on the generalization of the Black–Scholes method. The task is the modeling of behavior of all options and tools for their insurance. This task is characterized by large volume of real-time complex computations that should be executed concurrently The problem of the research: depending on conditions approaches to the solution should be various. There are three methods which can be used with different conditions: the finite difference method, the path-integral approach and methods which work in conditions of trade stop. Distributed computating in these three cases is organized differently and it is necessary to involve various approaches. In addition to complexity the mathematical formulation of the problem in literature is not quite correct. There is no complete description of boundary and initial conditions and also several hypotheses of the model do not correspond to real market. It is necessary to give mathematically correct formulation of the task, and to neutralize a difference between hypotheses of the model and their prototypes in the market. For this purpose it is necessary to expand standard formulation by additional methods and develop methods of realization for each of solution branches.

Keywords: financial mathematics, options pricing, asian option, wellposedness formulation, boundary conditions

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 759–766 (Russian).

Формула Блэка–Шоулза [Black, 1973; Merton, 1973] была создана в 1973 году как инструмент оценки реальной стоимости деривативов и в рамках модели дала возможность составления безрискового портфеля с определенным уровнем прибыли через определенный промежуток времени. С ее появлением использование такого рода страховочных инструментов с каждым годом становилось все более популярным и количество деривативов резко увеличивалось. Моделирование поведения деривативов на некоторый базовый актив и выбор подходящих позволяют сделать покупку или продажу актива безрисковой. Таким образом, формула Блэка–Шоулза в рамках своих положений позволяет создавать портфели определенной доходности и лишенные риска. Основное уравнение такого подхода имеет вид

$$rS(t)\frac{\partial V}{\partial S}+\frac{\partial V}{\partial t}+\frac{1}{2}\sigma^2 S(t)^2\frac{\partial^2 V}{\partial^2 S}-rV(S,t)=0, \quad (1)$$

где r — значение безрисковой ставки, S — цена базового актива, V — теоретическая цена опциона в момент экспирации, σ — волатильность. Однако формула Блэка–Шоулза является инструментом, позволяющим только приближенно прогнозировать поведение деривативов. Еще большая точность достигается при различных подходах по уточнению формулы и использованию дополнительных деривативов в качестве хеджирования (страховки). Далее приводятся основные проблемы модели и пути из решения.

Моделирование поведения опционов при использовании формулы Блэка–Шоулза не соответствует поведению деривативов на реальных рынках в основном из-за трех проблем, не учитываемых в формуле: наличия стохастической волатильности (а не постоянной, как предполагает формула), отличия распределения цены от нормального, а также наличие толстых хвостов и улыбки волатильности. Рассмотрим эти пункты более подробно.

1. Улыбка волатильности и длинные хвосты

Согласно допущениям модели Блэка–Шоулза базисный актив характеризуется постоянным уровнем волатильности. Если это предположение верно, то опционы на данный актив должны обладать одинаковой внутренней волатильностью (прогнозируемая на момент экспирации) вне зависимости от цен исполнения и сроков истечения контрактов. Однако практика показывает, что опционы на один и тот же актив с одинаковой ценой исполнения, но разными сроками истечения контрактов имеют разные внутренние стандартные отклонения. Аналогично разной внутренней волатильностью характеризуются опционы с одинаковым сроком истечения, но разными ценами исполнения [Буренин, 2005]. Изгиб возникает из-за неопределенности стоимости базисного актива при наступлении срока, в то время как стоимость опциона известна.

Зависимость для опционов с одним сроком истечения контрактов между ценой исполнения и внутренней волатильностью является квадратичной или монотонной. График квадратичной зависимости и есть улыбка волатильности. До финансового кризиса 1987 года улыбка волатильности была не сильно выражена. После кризиса она приобрела более выпуклую книзу форму и для ряда рынков перестала быть симметричной [Taleb, 2012].

График улыбки волатильности для опционов на акции и индексы характеризуется нисходящей кривой. Улыбке волатильности соответствует вероятностное распределение цены базисного актива, которое имеет более толстый левый хвост, более тонкий правый хвост и которое более островершинно по сравнению с логнормальным распределением. Это говорит о том, что опционы с большим выигрышем и большим проигрышем стоят на рынке дороже, чем это предполагается формулами, основанными на логнормальном распределении [Hull, 2008]. Возможное объяснение такой формы графика улыбки волатильности состоит в том, что после финансового кризиса 1987 г. участники рынка в большей степени стали опасаться повторения подобной ситуации. Поэтому они дороже оценивают опционы с более низкой ценой исполнения.

Еще одно объяснение можно связать с эффектом финансового рычага, т. е. соотношением между заемным и собственным капиталом компании [Cox, 1979]. При росте курса акции фи-

нансовый рычаг уменьшается. В результате акция становится менее рискованной и стандартное отклонение ее доходности падает. В случае падения цены бумаги финансовый рычаг возрастает и увеличивается рискованность акции. Волатильность акции растет. Таким образом, в рамках данного подхода волатильность акции можно рассматривать как убывающую функцию ее цены [Чекулаев, 2002].

Логнормальное распределение придает большую значимость текущей цене акции и меньшую — будущим ценам. Допущение меньшей вероятности экстремумов в распределении существенно уменьшает шансы возникновения большой стоимости опциона по истечении его срока и влияет на уменьшение ожидаемой стоимости. Это, однако, допускает вероятность очень экстремальных движений [Connolly, 1997].

Модель Блэка–Шоулза не предполагает возможности описания длинных и толстых хвостов. Их наличие и невозможность учета при применении модели не гарантируют формирование безрискового портфеля. Поэтому, чтобы избежать возможности риска, улыбка волатильности должна быть обрезана.

2. Отличие распределения от нормального

Модель Блэка–Шоулза основана на предположении о том, что цены базовых активов следуют процессу геометрического броуновского движения, т. е. логарифмические доходности этих активов имеют в каждый момент времени нормальное распределение. Доказано, что такой рынок может существовать тогда и только тогда, когда система рынков базовых и производных активов является в совокупности полным рынком, т. е. таким, на котором существуют идеальные хеджи — безрисковые портфели базовых и производных активов с положительной ожидаемой дисконтированной доходностью [Taleb, 2004]. В реальности сведение риска портфеля ценных бумаг к нулю невозможно, что связано с проявляющимися в динамике цен базовых активов свойствами.

Как показывает практика, распределения с высокими пиками и толстыми хвостами обладают практически все финансовые активы. Данные распределения моделируются функциями, отличными от гауссовых [Джеффри, 2002].

Динамика логарифмической доходности базовых активов рассматривается как безгранично делимый процесс Леви, поскольку он являются естественным и весьма правдоподобным кандидатом при построении вероятностных моделей несимметричных и островершинных распределений. В динамике процесса выделяют две составляющие: во-первых, броуновское движение, называемое диффузией процесса и отвечающее за идеально хеджируемый риск изменения цены базового актива, и, во-вторых, чисто скачкообразную компоненту, риск изменения которой не хеджируется в портфеле, а в зависимости от характеристик распределения отдельных скачков итоговое распределение логарифмических доходностей демонстрирует характерные финансовым рядам свойства [Морозова, 2011]. В настоящее время зарубежными авторами предложено множество альтернативных безгранично делимых распределений в качестве вариантов распределения доходностей базовых активов. Вычисление на основе этих распределений риск-нейтральной меры рынка в основном проводится в предположении, что наблюдаемые рыночные цены являются справедливыми, и параметры риск-нейтральной меры оцениваются на основе регрессий теоретической цены опциона и его рыночной стоимости.

Таким образом, использование подходящих распределений, отличных от нормального, позволяет более точно моделировать и прогнозировать поведение деривативов на реальных рынках и дает возможность составить безрисковый портфель, что и является основной задачей.

3. Динамическая волатильность и возможность резких скачков

Еще одним ограничением модели является необходимость использования статической волатильности. Для модели Блэка–Шоулза нет описанного стандартного метода, позволяющего

учитывать изменения волатильности, а также нет способов для прогнозирования резких скачков. В модели используется постоянная волатильность, задаваемая исходя из средних значений предыдущих периодов.

Условно решение проблемы стохастической волатильности для уравнения Блэка–Шоулза можно разделить на 3 пункта: значение уровня волатильности невелико, имеет средние значения и — самый критичный случай — уровень волатильности достигает значения, при котором торговля на рынке приостанавливается.

Когда значение волатильности небольшое, то для решения лучшим образом подходит конечно-разностный метод. В литературе [Daniel, 2006] приводятся постановки граничных и начальных условий, однако ни в одном из источников нет формулировки непротиворечащих и полных условий. Далее приведен вывод корректных граничных условий для азиатского опциона.

Азиатский опцион — контракт, который дает право держателю купить актив, базирующийся на средней цене за некоторый определенный период времени.

Среднее арифметическое для стоимости базового актива за некоторый интервал времени:

$$I = I(t) = \int_0^t S(\tau)\, d\tau. \tag{2}$$

Другая непрерывная формулировка дает

$$A(t) = \frac{I(t)}{t} = \frac{1}{t}\int_0^t S(\tau)\, d\tau. \tag{3}$$

Дифференциальное уравнение в частных производных, которое моделирует поведение азиатских опционов:

$$rS(t)\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^V}{\partial S^2} + S\frac{\partial V}{\partial A} - rV(S,t) = 0. \tag{4}$$

Решение рассматривается в области $\Omega = (0, A_{\max}) \times (0, S_{\max}) \times (0, T)$, где r и σ — константы.

4. Граничные условия

Начальное условие $t = T$, $\tau = T - t$:

$$V(S, A, T) = \max\left(\frac{A}{T} - K, 0\right), \quad S \in [0, S_{\max}], \quad A \in [0, A_{\max}], \tag{5a}$$

$$V(0, A, \tau) = e^{-r\tau} \max\left(\frac{A}{T} - K, 0\right), \quad A \in [0, A_{\max}], \tag{5b}$$

$$V(S_{\max}, A, T) = 0, \quad 0 \le A \le KT, \tag{5c}$$

$$V(S_{\max}, A, \tau) = e^{-r\tau} \max\left(\frac{A}{T} - K, 0\right) + \frac{S_{\max}}{rT}\left(1 - e^{-r\tau}\right), \quad KT \le A \le A_{\max}, \tag{5d}$$

$$V(S, A_{\max}, \tau) = \max\left[\left(\frac{A_{\max}}{T} - K\right)e^{-r\tau} + \frac{S}{rT}\left(1 - e^{-r\tau}\right), 0\right], \tag{5e}$$

$$V(S, 0, \tau) = \frac{S}{rT}\left(1 - e^{-r\tau}\right), \quad S \in [0, S_{\max}], \tag{5f}$$

$$V(S_{\max}, A, \tau) \approx \max\left\{\left(\frac{A}{T} - K\right)e^{-r\tau} + \frac{S_{\max}}{rT}\left(1 - e^{-r\tau}\right), 0\right\} \tag{5g}$$

для $S_{\max} \to \infty$, $A \in [0, A_{\max}]$, $t \in [0, T]$.

В статье Хаггера [Hugger, 2006] сказано, что с финансовой точки зрения не может быть никаких выполнимых граничных условий на нижней $(t=0)$ и передней $(A=0)$ сторонах области. Для правой стороны $(S=S_{\max})$ области нет точно установленных граничных условий, но в особом случае существует приближенное граничное условие.

Для вывода приближенных граничных условий при $S=S_{\max}$ требуется, чтобы выполнялось не только условие $A \geq KT$, но и чтобы доход (payoff) был гарантированно положительным. Хаггер отмечает, что при $\sigma > 0$ доход будет положительным, только если $A > KT$, но если $\sigma = 0$, то тогда должна быть задана гораздо большая расчетная область. Для такого случая функции *S*, *A*, *V* являются неубывающими функциями от времени *t*.

В данной постановке не предложено непротиворечивых граничных условий, которые можно использовать при численном моделировании рассматриваемого случая.

При $\sigma = 0$ тип исходного уравнения меняется с параболического на гиперболический, что ведет за собой изменения в граничных условиях и оценке размеров вычислительной области.

Приведенный случай рассматривается в пределе для $S_{\max} \to \infty$. Как только σ становится больше нуля даже на малую величину, значение $S=S_{\max}$ может не находиться в области, где гарантируется положительность дохода. Тем не менее при стремлении $S_{\max} \to \infty$ вероятность того, что при $S=S_{\max}$ в некоторый момент времени доход станет нулевым, в конечном итоге стремиться к нулю. Следовательно, стоимость опциона также должна быть в пределе такой же, как если бы рисковой ставки там не было вообще. В этом случае мы можем использовать (5g) в качестве граничного условия также для $\sigma > 0$. Фиксируя $S_{\max} < \infty$, можно обосновать приближенное граничное условие (5g).

Так как для $K \geq S \geq S_{\max}$ кривая, где Γ рассекает плоскость $S=S_{\max}$, представляет собой приблизительно прямую от $A=KT$ и $t=T$ до $A=0$ и $t=T-KT/S_{\max}$, лежащую полностью внутри правой стороны, необходимо вывести граничное условие выше этой кривой. Для этого необходимо выполнение следующего непрерывного граничного условия:

$$V\left(S_{\max},A,T\right) \approx \max\left\{\left(\frac{A}{T}-K\right)e^{-\int_t^T r(u)du} + \frac{S_{\max}}{T}\int_t^T e^{-\left[\int_t^T \gamma(u)du+\int_t^T r(u)du\right]}d\tau, 0\right\} \tag{6}$$

для $S_{\max} < \infty$, $A \in [0, A_{\max}]$, $t \in [0,T]$.

Тогда получается ситуация, когда некоторые значения при расчете в угловых точках расчетной зоны не совпадают.

Поэтому мы выбираем граничные условия с учетом непрерывности значений на углах границы:

$$V\left(S,A,T\right) = \max\left(\frac{A}{T}-K,0\right), \quad S \in [0,S_{\max}], \tag{7a}$$

$$V\left(S_{\max},A,\tau\right) = e^{-r\tau}\max\left(\frac{A}{T}-K,0\right) + \frac{S_{\max}}{rT}\left(1-e^{-r\tau}\right), \quad KT \leq A \leq A_{\max}, \tag{7b}$$

$$V\left(S,0,\tau\right) = \frac{S}{rT}\left(1-e^{-r\tau}\right), \quad S \in [0,S_{\max}], \tag{7c}$$

$$V\left(S,A_{\max},\tau\right) = \left(\frac{A_{\max}}{T}-K\right)e^{-r\tau} + \frac{S}{rT}\left(1-e^{-r\tau}\right). \tag{7d}$$

Для случая волатильности, при которой моделирование конечно-разностным методом не дает корректных результатов, но и торги еще не останавливаются, на финансовом рынке используются методы механики Бома. Этот подход является развитием классического формализма Гамильтона на фазовом пространстве в переменных «цена–волатильность» для описания

классической эволюции цен. Такая динамика цен определяется производственными финансовыми факторами. Эти факторы можно описать математически с помощью классического финансового потенциала.

На реальном финансовом рынке производственные факторы не являются единственным источником изменения цен. Информация и психология рынка играют важную роль в ценовой динамике. Предлагается описать эти информационные финансовые факторы, используя модель квантовой механики с ведущей волной. Применение теории финансовых ментальных (или психологических) волн вызвано потребностью учета психологии рынка.

Реальные траектории цен определяются с помощью финансового аналога второго закона с Ньютона двумя финансовыми потенциалами: классическим (производственные факторы) и квантовым (информационные факторы рынка). Дж. Сорос верно заметил [Soros, 1987], что «нементальный» рынок развивается благодаря классическим случайным флуктуациям. Но эти флуктуации не дают адекватного описания ментального рынка. Он предложил использовать аналогию с квантовой теорией. Элементарные частицы ведут себя стохастически в силу эффектов возмущения, порожденных измерениями [Heisenberg, 1930]. Согласно Соросу финансовые агенты на финансовом рынке ведут себя стохастически в силу свободы выбора для каждою из них. Сочетание огромного количества таких свобод выбора для финансовых агентов порождает дополнительную стохастику на финансовом рынке, которая не может быть сведена к классическим случайным флуктуациям (порожденным нементальными факторами). Здесь Дж. Сорос использовал стандартную точку зрения на квантовую стохастику. Такой подход позволяет применить квантовый формализм к финансовому рынку.

Постановку задачи для решения квантовым методом можно найти в [Baaquie, 2004; Харитонов, 2007]. Данная постановка является гибким инструментом для возможности включения различных уточнений в формулировку задачи. Так, в данном подходе легко решается проблема тяжелых хвостов и улыбки волатильности за счет ограничения области. Также можно легко перейти от нормального распределения к более подходящим, к примеру распределению Леви.

Для третьего варианта, когда уровень волатильности становится критическим, на биржах были разработаны механизмы, призванные решить проблемы относительно резких скачков цен, в том числе и остановка торговли [Investor Bulletin: New Measures..., 2014].

Первый — Limit Up Limit Down — останавливает торги по отдельной ценной бумаге в случае, если ее цена превосходит некоторые установленные границы. Вычисляется средняя цена исходя из колебаний цены ценной бумаги за последние 5 минут торгов. Устанавливаются ценовые границы в виде допустимого отклонения относительно средней цены (5 %, 10 %, 20 % либо минимум от 75 % и 15 центов в зависимости от значения средней цены). Если цена ценной бумаги в течение 15 секунд находится вне этих границ, то торги по ней прекращаются на 5 минут.

Второй механизм — Market Wide Circuit Breakers — касается рынка в целом. Данный механизм останавливает межрыночные торги при значительном снижении рыночных показателей. Устанавливается 3 уровня (7 %, 13 % и 20 %), при достижении которых торги прекращаются. Эти проценты вычисляются от цены на индекс S&P500 при закрытии биржи.

Польза от механизма прерывания торгов состоит в том, чтобы дилеры и брокеры смогли связаться со своими клиентами для получения инструкций в связи с большими ценовыми движениями и обсуждения дополнительной маржи. С помощью остановок ограничиваются кредитные риски и возможности потери доверия. Этот период бездействия также дает возможность инвесторам сделать паузу, оценить ситуацию и избежать паники. Наконец, прерывание торговли развеивает иллюзию рыночной ликвидности, демонстрируя простой жизненный факт, что рынки ограничены в своей способности поглотить массивные несбалансированные объемы предложения и спроса. Они, таким образом, вынуждают больших инвесторов, типа портфельных менеджеров пенсионных и взаимных фондов, принимать во внимание воздействие «размеров их ордеров» и, таким образом, возможно, амортизируют большие рыночные движения. С другой стороны, остановка торговли может увеличивать риск благодаря стимуляции торговли в ожидании остановки. Неудобство этой ситуации состоит в том, что остановки препятствуют некоторым трейдерам ликвидировать свои позиции и, таким образом, создают рыночные искажения, мешая реализовывать активы по предварительно выбранной цене.

Нами разработан программный инструментарий для оценки стоимости опционов в разных ситуациях на основе сформулированных подходов. Первый и третий случай реализованы в рамках конечно-разностного подхода, второй — путем расчета континуального интеграла методом Монте-Карло.

Таким образом, можно сказать, что подход Блэка–Шоулза позволяет формировать безрисковый портфель при условии, что будут использоваться некоторые его уточнения в разных ситуациях в зависимости от условий на рынке и будут накладываться дополнительные условия на область, в которой меняются показатели. Кроме того, важно использовать дополнительные методы хеджирования для страховки от рисков.

Список литературы

Буренин А. Н. Форварды, фьючерсы, опционы, экзотические и погодные производные / А. Н. Буренин. — М.: Научно-техническое общество академика С. И. Вавилова, 2005. — 534 с.

Джеффри О. К. Энциклопедия торговых стратегий / Пер. с англ. / О. К. Джеффри, Д. Л. МакКормик — М.: Альпина Паблишер, 2002.

Морозова М. М. Алгоритм расчета справедливой цены на неполных рынках с арбитражными возможностями // Исследования молодых ученых: отраслевая и региональная экономика, инновации, финансы и социология / Под ред. С. А. Суспицына [и др.]; ИЭОПП СО РАН. — Новосибирск, 2011. — С. 412–420.

Харитонов В. В. Эконофизика / В. В. Харитонов, А. А. Ежов. — М.: МИФИ, 2007 — 624 с.

Чекулаев М. В. Риск-менеджмент. Управление финансовыми рисками на основе анализа волатильности / М. В. Чекулаев. — М.: Альпина, 2002 — 344 с.

Black F. The Pricing of Options and Corporate Liabilities / F. Black, M. Scholes // Journal Political Economy. — 1973. — Vol. 81. — P. 637–659.

Baaquie B. E. Quantum finance. — Cambridge: Cambridge University Press, 2004

Connolly K. B. Buying and Selling Volatility / K. B. Connolly. — Hoboken, NJ: Wiley, 1997 — 230 с.

Cox J. Option Pricing: A Simplified Approach / J. Cox, S. Ross, M. Rubinstein // Journal of Financial Economics. — 1979. — No. 7.

Daniel J. Finite Difference Methods in Financial Engineering / J. Daniel, J. Duffy. — Hoboken, NJ: Wiley, 2006. — P. 268.

Heisenberg W. Physical principles of quantum theory. — Chicago: Chicago Univ. Press, 1930.

Hugger J. Wellposedness of the boundary value formulation of a fixed strike Asian option / J. Hugger // Journal of Computational and Applied Mathematics. — 2006. — Vol. 185. — P. 460–481.

Hull J. C. Options, Futures, and Other Derivatives / J. C. Hull. — Toronto: Pearson Prentice Hall, 2008. — 836 с.

Investor Bulletin: New Measures to Address Market Volatility [Электронный ресурс]. URL: http://www.sec.gov/investor/alerts/circuitbreakersbulletin.htm (дата обращения: 22.09.2014).

Merton R. C. Theory of Rational Option Pricing / R.C. Merton // Bell Journal of Economics and Management Science. — 1973. — Vol. 4.

Soros O. The alchemy of finance. Reading of mind of the market. — NY: Wiley, 1987.

Taleb N. N. The Illusion of Thin-Tails Under Aggregation / N. N. Taleb, G. Martin // Journal of Investment Management. — 2012.

Taleb N. N. On the Unfortunate Problem of the Nonobservability of the Probability Distributions / N. N. Taleb, A. Pilpel, 2004.

**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 767–774**

Ки&М

СТЕНДОВЫЕ ДОКЛАДЫ

УДК: 004.94

Алгоритмическое построение явных численных схем и визуализация объектов и процессов в вычислительном эксперименте в гидромеханике

(Пространственные числовые объекты тензорной геометрии для аппроксимации элементарных деформируемых частиц и моделирования вычислительных операций физической теории поля)

А. Б. Дегтярев, Т. Р. Ежакова[a], В. Н. Храмушин

Санкт-Петербургский государственный университет,
Россия, 198504, г. Санкт-Петербург, Петергоф, Университетский просп., д. 35

E-mail: [a]t.r.ezhakova@gmail.com

Получено 28 октября 2014 г.

Проектные и поверочные этапы в разработке сложных вычислительных алгоритмов для создания прямых вычислительных экспериментов в гидромеханике при моделировании нестационарных процессов и физических полей механики сплошных сред должны опираться на строгие правила конструирования числовых объектов и направленного синтеза методов их использования в строгом соответствии с прикладной задачей гидромеханики. Возможность использования троичной логики для разрешения противоречий функционального и декларативного программирования одновременно приводит к новым схемам тензорной математики, которые позволяют оптимизировать эффективность и обосновывать корректность результатов моделирования, в том числе с использованием интерактивных графических методов для визуализации промежуточных результатов и управляемого воздействия на ход вычислительного эксперимента под управлением инженеров-аэрогидромехаников — исследователей.

Ключевые слова: тензорная математика, метод крупных частиц, гидромеханика, вычислительный эксперимент, проектное решение, поверочная задача

Исследования выполняются при поддержке грантов РФФИ (№ 13-07-00747), СПбГУ (№ 9.38.674.2013, № 0.37.155.2014) и Комплексной программы ДВО РАН «Дальний Восток» (№ 15.3312-III-СО-08-023) на базе оборудования «Вычислительный центр СПбГУ».

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 767–774

Ки&М

СТЕНДОВЫЕ ДОКЛАДЫ

Algorithmic construction of explicit numerical schemes and visualization of objects and processes in the computational experiment in fluid mechanics (Spatial geometry of the objects of the tensor for the approximation of the elementary paticles and deformable modeling computational operations physical field theory)

A. B. Degtyarev, T. R. Yezhakova,V. N. Khramushin

Saint Petersburg State University, 35 University ave., Peterhof, St. Petersburg, 198504, Russia

Abstract. — Design and verifying steps in the software development of complex computational algorithms to create a direct computational experiments in fluid mechanics simulation of unsteady processes and physical fields of continuum mechanics, must be based on strict rules of constructing numerical objects, and directed synthesis methods for their use in strict accordance with the applied problems in fluid mechanics. The ability to use ternary logic to resolve differences of functional and declarative programming at the same time leads to new schemes tensor mathematics to help optimize the efficiency and correctness of the simulation, including the use of interactive graphics techniques to visualize the intermediate results and managed to influence the course of computational experiment running by aerohydrodynamics — researchers.

Keywords: tensor mathematics, large particles, fluidmechanics, computational experiment, design, verifying

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 767–774 (Russian).

Современное становление вычислительной архитектуры предъявляет повышенные требования к теоретической стройности, оптимальности и обоснованности выбора разнородных функциональных средств и графических инструментов для создания специализированных прикладных программ; требующих согласованности в логическом синтезе фундаментальных законов физики с их алгоритмическим представлением при унифицированном построении прямых вычислительных экспериментов, востребованных в соответствии со сценариями практического применения инженерных вычислительных систем — их интерактивных и динамически формируемых графических отображений.

1. Проектные и поверочные этапы

Проектные и поверочные этапы в разработке прикладных вычислительных комплексов для моделирования физических полей механики сплошных сред должны быть основаны на логической независимости методов представления элементарных числовых объектов-структур и операций с ними в составе больших числовых массивов, вовлекаемых в явные численные алгоритмы и контекстно-зависимые — рекурсивные функции для построения исходных физических полей, собственно моделирования и последующей визуализации результатов прямого численного моделирования нестационарных гидромеханических процессов и явлений.

В процессе проектирования и построения сложных моделирующих систем средствами формализованных языков программирования, в отличие от нечетких деклараций естественных языков, обязательно следование строго означенным алгоритмическим процессам с соблюдением функциональных зависимостей для моделируемых физических законов и явлений, формализуемых сложными структурами числовых данных и контекстно-зависимыми операциями над ними. Любой разлад формальной логики или последовательности алгоритмических операций, приведет исходные инженерные разработки к невыносимо трудной отладке программных комплексов либо к приведению разрозненных пакетов процедур в состояние неустранимых несоответствий между вычислительными объектами и связанными с ними операциями в целом. Рассмотрим особенности проектных построений для реализации новых алгоритмов вычислительной гидромеханики.

Непротиворечивое авторское изложение какой-либо идеи может стать вполне состоятельным, если для внешне разнородных понятий в описании природных явлений удается ввести специальные определения или особые сущности, достаточные для взаимно однозначного позиционирования или связывания всех смысловых или содержательных противоречий. В терминах троичной логики (трилектики[1]) предложения естественного языка оперируют диалектически разделенными сущностями, такими как подлежащее и сказуемое, которые связываются определениями, дополнениями и контекстно независимыми обстоятельствами.

В среде гидромеханического вычислительного эксперимента аналогичные понятия представляются числовым «объектом–явлением» и «операцией–действием», с которыми однозначно связываются геометрические «трансформации–алгоритмы» и физические «законы–функции», по аналогии существующие в обстоятельствах контекстно-зависимой среды исполнительных «алгоритмов» и множества «функциональных зависимостей».

На рис. 1 приведена схема построения вычислительного эксперимента в виде трех уровней аппаратной и языковой поддержки различных методов программирования, создаваемых по принципу «от множества расчетных алгоритмов к унифицированной функциональной среде» [Храмушин, 2005], ориентированной на внутренний контроль и согласование реологических свойств жидкости, и автоматизированную численную реализацию законов механики сплошной среды. Главные направления проектных исследований: «Проект» — согласование исходной задачи и ожидаемых результатов в избранной языковой среде; «Развитие» — этапы трансфор-

[1] «…Мудрый подход — китайская трилектика срединного пути, когда в логике научного поиска существуют как оппозиции, так и третьи позиции, оценивающие обстоятельства искомого выбора…»

мации моделируемых процессов аэрогидромеханики как при реализации поисковой, так и поверочной задачи; по взаимно ортогональной оси ① — «Явление» — детальное описание физических законов и реологии сплошной среды.

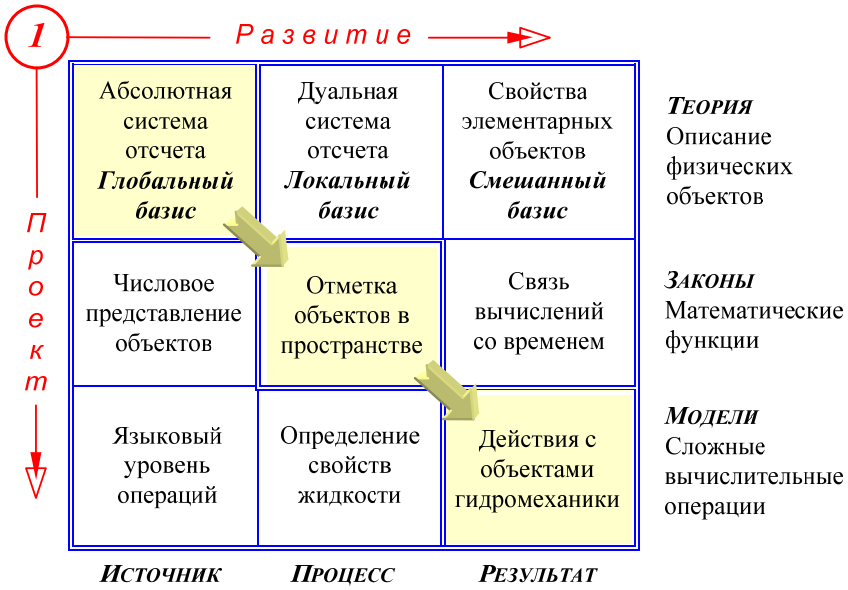


Рис. 1. Структура троичной матрицы, определяющей взаимосвязь проектных элементов для построения алгоритмов вычислительной гидромеханики с указанием условно применяемых к ним математических операций и логики пространственно-временных преобразований

2. Логическое построение математической модели

В предметной области ① — «Проект»–«Развитие» — конструируется компьютерное представление «Законов механики» на взаимно ортогональных основаниях «Математических моделей» и «Языков программирования», что в зависимости от типа проектных поисков в анализе «сверху вниз» по аналогии с частями речи будут выступать в качестве моделируемых объектов (существительных) и операций на ними (глаголов) и наоборот — как предопределенные действия (подлежащие) над изменяемыми данными (сказуемыми), при проектном синтезе «снизу вверх». Аналогичное встречное (виртуальное) проектирование алгоритмов допускается в компиляторах машинно-зависимых декларативных языков и отображается функциональными дополнениями (искусственным интеллектом) для интерактивного управления вычислительным экспериментом с автоматическим выбором гибридных численных схем или асимптотических решений в критических зонах.

По аналогии в определениях троичной логики выстраиваются числовые объекты — структуры данных и методы их обработки — алгоритмы и функции для реализации прямых вычислительных экспериментов в гидромеханике с использованием трехмерной тензорной математики. Ближайшим аналогом по постановке вычислительного эксперимента является метод крупных частиц [Белоцерковский, Давыдов, 1982], в котором замыкание проектных решений образуется этапами разделения задачи по физическим процессам континуально-корпускулярного подхода, что выражается в последовательном использовании математических моделей течения жидкости в эйлеровом представлении на неподвижной сетке и в лагранжевом смещении свободных деформируемых частиц жидкости.

В тензорной математике определяется строгий и однозначный метод записи состояния крупной частицы сплошной среды [Храмушин, 2005], обладающей свойствами сжимаемости, вязкости и упругости, с учетом ее динамической деформации во времени в соответствии с разностными схемами разложения для систем дифференциальных уравнений в частных производ-

ных, и в первую очередь с линейной интерполяцией между узлами (гранями) и центрами масс дискретных ячеек в числовых массивах для представления неразрывных физических полей.

Построение алгоритмических операций и функциональных зависимостей для реализации прямых вычислительных экспериментов в гидромеханике ведется с учетом привязки координатных базисов; при явном задействовании физических размерностей в контроле состояния моделируемой сплошной среды; при построении преимущественно явных численных схем с разделением этапов решения по естественным физическим процессам, допускающим наглядную визуализацию моделируемых свойств и потоков жидкости, и, как следствие, возможность задействования сложных функциональных зависимостей или инженерных решений в подобластях с проблемными или особыми условиями существования моделируемой сплошной среды.

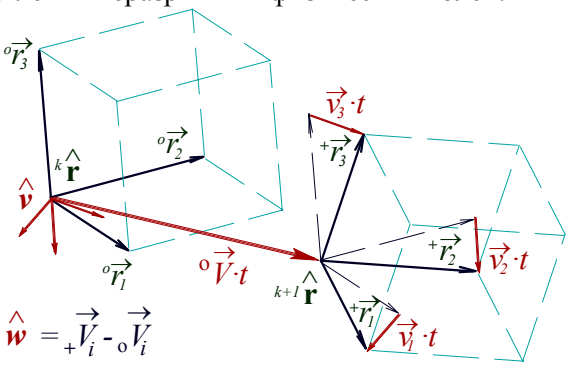


Рис. 2. Тензор локальных скоростей образуется поступательными и деформационными смещениями базисных векторов крупной частицы жидкости за расчетный интервал времени

В формализованной теории это сводится к правилам алгоритмического конструирования геометрических и естественно-физических операций с предопределенным физическим содержанием, что существенно ограничивает допустимое множество арифметических действий в трехмерной тензорной математике в отличие от многовариантности операций в базовых разделах тензорного исчисления с элементами аналитической геометрии и линейной алгебры.

3. Определение физических объектов и базовых вычислительных операций

Формальное построение физических объектов и операций в тензорной математике [Храмушин, 2005] приводит к строгим определениям для своеобразной модели мира вычислительной гидромеханики: 1) континуально-корпускулярная вычислительная модель метода крупных частиц в тензорной записи сводится к двойной линейной разностной интерполяции физических полей (*вместо интегрирования уравнений движения второго порядка*); 2) пространственное движение и взаимодействие крупных частиц жидкости описывается операциями произведения, что более соответствует физике процессов гидромеханики (*нет ограничений по малости дифференциальных приближений*); 3) возможность использования явных численных схем и дискретных полей с заданными физическими свойствами служит повышению эффективности прямых вычислительных экспериментов и дает возможность контроля корректности, и, по необходимости, задействования гибридных схем для достижения адекватных инженерных результатов моделирования.

Закон движения для частицы сплошной среды: $\vec{F} = \overset{>}{M} \cdot \vec{W} = \hat{r} \cdot \overset{\vee}{\rho} \cdot \vec{W}$; [Н]

Тензор вязких напряжений: $\overset{>}{f}_{н} = \hat{v}_{н} \cdot \overset{\vee}{\eta} / \Lambda = \overset{>}{v}_{н} \cdot \overset{>}{\eta} \cdot / \Lambda$; [Н/м]

Тензор упругих напряжений: $\overset{>}{f}_{\Gamma} = (\hat{r} + \hat{v}_{\Gamma} \cdot t) \cdot \overset{\vee}{c} / \Lambda = (\overset{\times}{1} + v_{\Gamma} \cdot t) \cdot \overset{>}{c} / \Lambda$, [Н/м]

где тензор локальных скоростей: $\hat{v} = {}_{+}\vec{V}_i - {}_{o}\vec{V}_i$ (рис. 2); реологические тензоры: $\overset{>}{M} = \hat{r} \cdot \overset{\vee}{\rho}$ [кг] — тензор массы — инерции; $\hat{r}$ [м^3] — тензор формы; $\overset{\vee}{\rho}$ [кг/м^3] — «условная плотность» для сохранения предыстории девиаций — внутренних «живых сил» крупных частиц жидкости; $\overset{>}{\eta}, \overset{>}{c}$ — тензоры динамической вязкости [кг/с] и жесткости [кг] реальной жидкости, Λ — дистанция ближнего взаимодействия смежных частиц.

Вычислительная модель содержит все три реологических свойства жидкости: сжимаемость, вязкость, упругость, при этом соотношение интенсивностей указанных напряжений может привести к критическому режиму с образованием струй, вихревых слоев и кавитационных разрывов. Под действием тензора внутренних напряжений частица жидкости получает приращение скорости внутреннего (замкнутого) движения:

$$\overset{\vee}{f} = \overset{\vee}{\varepsilon}\cdot \overset{\vee}{v}_0 \cdot t + \overset{\vee}{\mu}\cdot \overset{\vee}{v}_H + \overset{\vee}{c}\cdot \overset{\vee}{v}_\Gamma \cdot t = \overset{\vee}{f}_0 + \overset{\vee}{f}_H + \overset{\vee}{f}_\Gamma ,$$

где тензоры $\overset{\vee}{f}_0$ — давление, $\overset{\vee}{\varepsilon}$ — коэффициент сжатия. Динамические коэффициенты $\overset{\vee}{\mu}, \overset{\vee}{c}, \overset{\vee}{\varepsilon}$ отличаются от кинематических исключением величины скалярной плотности ρ.

Тензор внутренних напряжений в локальной системе отсчета представляется характеристическим полиномом для оценки внутреннего состояния крупной частицы жидкости, где реологические параметры проявляются в виде главных инвариантов тензора конвективных скоростей:

$\overset{>}{v}_0$: **I** ≠ 0 — сжимаемость → *кавитационный разрыв плотности*;

$\overset{>}{v}_H$: **II** ≠ 0 — поворот → *образование свободной струи или турбулентного вихря*;

$\overset{>}{v}_\Gamma$: **III** ≠ 0 — деформация, чистая — если другие инварианты тензоров равны нулю.

К примеру, если расчетная ячейка с присоединенным вихрем обращает в нуль детерминант внутреннего поля конвективных скоростей в смежной точке, в точке центра масс свободной сопряженной частицы жидкости, то можно воспользоваться алгоритмом переноса вихря с исходного эйлерова этапа вычислений в тензор «массы» сопутствующего лагранжевого этапа, что предопределит зарождение свободного турбулентного вихря внутри крупной частицы жидкости. Такой гибридный алгоритм можно использовать в том числе за пределами аппроксимационного разрешения на относительно грубых сетках, а для получения уточненного решения сеточная область в зоне вихреобразования может быть сгущена.

4. Реализация континуально-корпускулярных алгоритмов гидромеханики

Интерполяционные сеточные пространства с раскрепощенными для движения вычислительными объектами обусловливают суть вычислительных алгоритмов — этапов моделирования.

Прямые вычислительные эксперименты в гидромеханике на основе метода крупных частиц традиционно строятся исключительно с использованием прямоугольных ортогональных сеток. Исходя из аппроксимационных критериев может определяться минимальный шаг расчетной сетки в каждой локальной подобласти, что усложняет лишь рекурсивные функции автоматической переадресации пространственных координат расчетных ячеек с переинтерполяцией физических полей по дискретным узлам нерегуляризованных матричных массивов, и особо не перегружает вычислительные ресурсы на лагранжевых этапах со свободными частицами жидкости:

$$\{R\} = {}^{n}_{ijk} R,$$

как определение индексируемой регулярной сеточный области $\{i, j, k\}$, где нижние левые индексы задают пространственное местоположение: $\{X, Y, Z\} = \{i\cdot x, j\cdot y, k\cdot z\}$; а верхний левый индекс соответствует текущему циклу вычислительного эксперимента в отсчетах физического (моделируемого) времени $T = n\cdot t$.

В объектно ориентированном языке программирования C++ на уровне синтаксического разбора операторов возможно применение виртуальной перегрузки операций доступа к кон-

кретным числовым объектам в сложной сеточной области, что может использоваться для адаптивнных нерегуляризованных сеток (рис. 3) с пропусками узлов $_{ij}R$ и цельными ячейками — частицами: $_{ij}M$. Отчасти такие алгоритмы усложнят выбор и увеличат время доступа к конкретным числовым объектам, в то же время создается возможность оптимизации и значительного ускорения общего цикла вычислений и, что не менее важно, сохранения унификации вычислительных моделей механики сплошных сред, вплоть до простого алгоритмического перестроения внутренних и внешних граничных условий с помощью сглаживающих или экстраполяционных алгоритмов, в зависимости от аппроксимационных возможностей и доступности смежных числовых объектов.

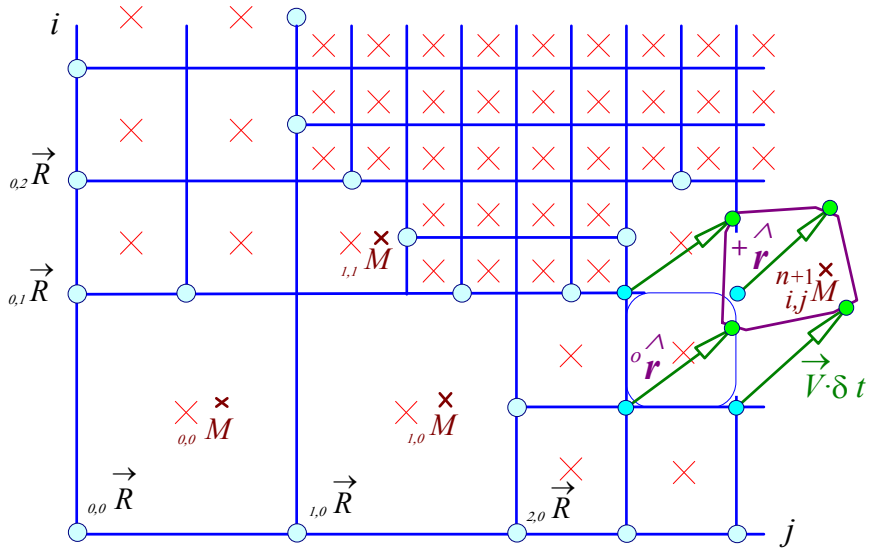


Рис. 3. Прямоугольная ортогональная нерегуляризованная сеточная область

При такой организации расчетной области местоположение крупных частиц жидкости $_{ij}M$ не обременяется смежными ортогональными связями для непосредственного применения разностного дифференцирования, что формально предопределяет возможность задействования этапа моделирования по Лагранжу, где формально независимые частицы участвуют в свободном и ускоренном движении по криволинейным траекториям в зависимости от их внутренней энергии (живых сил), аккумулированной в форме условной тензорной массы $_{ij}M$ (плотности).

5. Элементарные числовые объекты

Элементарные числовые объекты [Программа построения числовых объектов…, Роспатент № 2013619727] конструируются для выполнения строго определенного и ограниченного количества операций над числовыми объектами типа

```
typedef struct { token s; int d } integer; // — индексы и дискретные отсчеты,
typedef struct { token s; double d } real; // — скалярные физические величины,
```

где: `s` — формализованный элемент, использующийся для автоматического применения конвертирующих или адаптирующих операций по ходу вычислительного эксперимента. Скалярные числовые объекты служат для индикаторов времени и других инвариантных величин:

```
real T; // отсчет времени от начала вычислительного эксперимента,
real t; // шаг во времени для моделирования нестационарных процессов.
```

Векторные величины определяют точку как свободный вектор в пространстве:

```
typedef struct {Real X,Y,Z;} Point; // в абсолютной системе отсчета,
typedef struct { real x,y,z; } point; // внутри частицы жидкости
```

Крупная частица жидкости определяется с помощью числовой матрицы — тензора:

```
typedef struct { point x,y,z; } tensor; // свободный локальный базис,
typedef struct { Point R,X,Y,Z; } cell; // его видимость извне как ячейки.
```

Производные числовые структуры тензорной математики:

```
typedef struct { Point A; real x,y,z; } Vector; // с привязкой,
typedef struct { Point A; point x,y,z; } Basis; // местоположения.
```

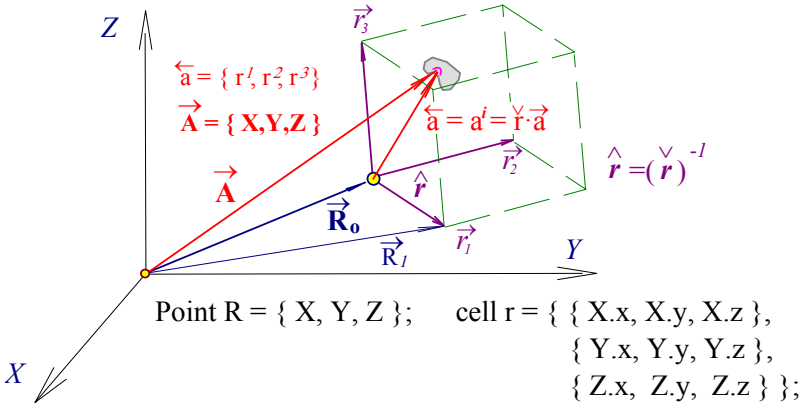


Рис. 4. Программное и пространственное представление элементарной частицы жидкости

Тензорные вычислительные операции относятся к тетраэдру базисных векторов (рис. 4), в которых все кинематические характеристики и свойства жидкости изменяются линейно. Изнутри деформированные параллелепипеды представляются ортонормированными базисами.

Заключение

Оптимальным по сложности инструментарием для проектирования вычислительных экспериментов можно признать алгоритмы трехмерной тензорной математики, где все числовые объекты идентичны представлениям в однородных координатах для визуализации вычислительных процессов в типовой графической среде типа OpenGL, задействующей визуализацию на параллельно работающих графических станциях. Создаются условия для контролируемого моделирования сложнейших явлений и процессов в гидроаэромеханике на основе интерактивного управления вычислительными процессами и автоматической адаптации численных схем.

Список литературы

Белоцерковский О. М., Давыдов Ю. М. Метод крупных частиц в газовой динамике. — М.: Наука, 1982. — 370 с.

Программа построения числовых объектов и функций трехмерной тензорной математики для вычислительных экспериментов в гидромеханике (Tensor). — СПбГУ, Роспатент № 2013619727.

Храмушин В. Н. Трехмерная тензорная математика вычислительных экспериментов в гидромеханике. — Владивосток: ДВО РАН, 2005.— 212 с.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 775–780

Ки&М

СТЕНДОВЫЕ ДОКЛАДЫ

УДК: 004.021

Неоднородные клеточные генетические алгоритмы

Н. М. Ершов

Факультет вычислительной математики и кибернетики,
Московский государственный университет им. М. В. Ломоносова,
Россия, 119991, г. Москва, ГСП-1, Ленинские горы, 1-52

E-mail: ershovnm@gmail.com

Получено 17 сентября 2014 г.

В работе вводится в рассмотрение понятие неоднородного клеточного генетического алгоритма, в котором ряд параметров, влияющих на работу генетических операторов, оказывается зависимым от местоположения клеток заданного клеточного пространства. Приводятся результаты численного сравнения неоднородных клеточных генетических алгоритмов со стандартными вариантами генетических алгоритмов, показывающие преимущества предложенного подхода при минимизации мультимодальных функций с большим числом локальных экстремумов. Рассматривается крупноблочная параллельная реализация неоднородных клеточных алгоритмов с использованием технологии MPI.

Ключевые слова: эволюционные алгоритмы, генетические алгоритмы, параллельные вычисления

Non-uniform cellular genetic algorithms

N. M. Ershov

Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University,
1-2 Leninskiye Gory, GSP-1, Moscow, 119991, Russia

Abstract. — In this paper, we introduce the concept of non-uniform cellular genetic algorithm, in which a number of parameters that affect the operation of genetic operators is dependent on the location of the cells of a given cellular space. The results of numerical comparison of non-uniform cellular genetic algorithms with the standard genetic algorithms, showing the advantages of the proposed approach while minimizing multimodal functions with a large number of local extrema, are presented. The coarse-grained parallel implementation of the non-uniform algorithms using the technology of MPI is considered.

Keywords: evolutionary algorithms, genetic algorithms, parallel computing

Работа выполнена при финансовой поддержке РФФИ (грант №14-07-00628 А).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 775–780 (Russian).

1. Введение

Клеточные генетические алгоритмы [Alba and Dorronsoro, 2008] обладают рядом преимуществ по сравнению с обычными генетическими алгоритмами. Во-первых, за счет локальности взаимодействия между особями популяции удается более долгое время поддерживать разнообразие в популяции, что потенциально ведет к получению более качественного решения. Во-вторых, благодаря регулярности расположения особей в клеточном пространстве и отсутствию глобальных операций клеточные генетические алгоритмы хорошо и масштабируемо распараллеливаются [Whitley, 1994]. Однако, как и в обычных генетических алгоритмах, в клеточном варианте остается актуальной проблема попадания алгоритма в локальные экстремумы. В настоящей работе предлагается подход к решению этой проблемы, основанный на введении зависимости работы операторов генетического алгоритма (прежде всего мутации) от положения особи в клеточном пространстве.

2. Оптимизационная задача

Принципы работы неоднородных клеточных генетических алгоритмов рассмотрим на примере решения задачи многомерной *непрерывной* оптимизации функции $F(x)$ в области $x_k \in [-100, 100]$, $k \in \{1\ldots n\}$. Рассматривались две функции — бимодальная и мультимодальная. Первая функция

$$F(\theta, x, y) = G(1, x + 5, y + 5) + 2\, G(\theta, x - 5, y - 5),$$

является двумерной и представляет собой сумму (с отрицательным знаком) двух гауссианов:

$$G(\theta, x, y) = 1 - e^{\frac{x^2 + y^2}{2o\theta}},$$

первый из которых имеет фиксированную ширину, а ширина второго определяется параметром θ. Функция имеет два минимума. Минимум в точке (–5, –5) является локальным и имеет фиксированную ширину. Минимум в точке (5, 5) является глобальным, значение функции в нем равно примерно 1. Параметр θ в численных экспериментах был меньше 1, поэтому область «притяжения» глобального минимума является более узкой по сравнению с локальным минимумом.

Вторая функция, с которой проводилось исследование, — это функция Растригина [Ke Tang et al., 2010]:

$$R(x) = \frac{1}{100} \sum_{k=1}^{n} \left[(x_k - 20)^2 + A\left(\frac{\cos(\pi x_k)}{5} \right) \right].$$

Данная функция является мультимодальной с единственным глобальным минимумом в точке $x_k = 20$, $k \in \{1\ldots n\}$. В двумерном случае в заданной области функция Растригина имеет 400 локальных минимумов, а 10-мерная функция — уже порядка 10^{13} минимумов. В численных экспериментах использовались функции с параметром $A = 2 \cdot 10^5$. Коэффициенты функции Растригина подобраны так, что локальные минимумы располагаются в точках вида $x_k = 10l$, $l \in Z$, а значения функции в этих точках является суммой n квадратов целых чисел.

3. Однородные генетические алгоритмы

Сравнение эффективности неоднородного клеточного генетического алгоритма (NCGA) производилось с генетическим алгоритмом (GA) и клеточным генетическим алгоритмом (CGA). Во всех рассматриваемых вариациях алгоритмов использовались одинаковые генетиче-

ские операторы (отбор, скрещивание, мутация и миграция). Для отбора применялась *турнирная* схема, в которой пара выбранных особей решает вопрос о выживании одной из них. Особь с лучшим значением целевой функции выживает с вероятностью p_{win}. Победитель замещает своей копией место побежденного. Для скрещивания использовалась равномерная схема, когда с заданной вероятностью p_{swap} меняются местами два соответствующих гена двух заданных особей. В силу специфики решаемой задачи (непрерывная оптимизация) оператор мутации выполняет случайное изменение каждого гена заданной особи с достаточно большой вероятностью p_{mut}:

$$x_k \leftarrow x_k + \rho d_{\text{mut}},$$

где ρ — случайное действительное число из диапазона [–1, 1], d_{mut} — величина мутации. Чтобы обеспечить разнообразие популяции на начальном этапе работы алгоритма и сохранить его сходимость, применялась технология *имитации отжига* по параметру d_{mut}. Оператор миграции использовался только в клеточных вариантах генетического алгоритма. Этим оператором две выбранные особи меняются местами.

Для однородного генетического алгоритма использовалась следующая схема работы. Выполнялось заданное число итераций. На каждой итерации выполнялись последовательно три оператора: отбор, скрещивание и мутация. После каждого оператора производилось полное перемешивание популяции. Для отбора и скрещивания брались пары соседних (по номерам) особей с номерами $2i$ и $2i + 1$, $i \in 0\ldots n/2$.

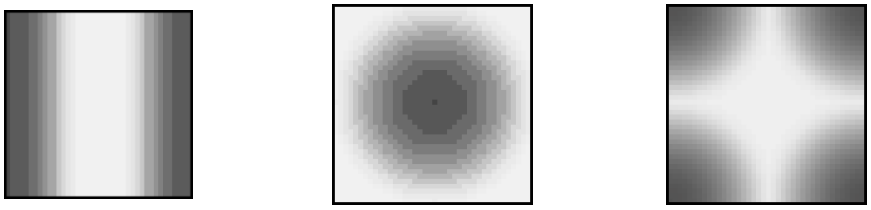
Однородный клеточный генетический алгоритм работал по следующей схеме. Все особи размещались по одной в клетках прямоугольной области. На каждой итерации алгоритма сначала выполнялись операторы отбора, скрещивания и миграции. Для этого клетки случайным образом делились на пары, так чтобы две клетки в одной паре всегда были соседними (т. е. имели бы общую сторону). Последним шагом на итерации выполнялась мутация всех особей.

4. Неоднородные клеточные генетические алгоритмы

Как уже упоминалось выше, проблемой однородных (клеточных или нет) генетических алгоритмов является то, что в итоге они сходятся к однородной (гомогенной) популяции. В такой популяции практически все особи имеют одинаковые геномы, поэтому наиболее мощный генетический оператор скрещивания перестает работать. Если алгоритм попал в локальный экстремум, то выйти из него можно только за счет мутации. Если мутация локальна, а более хорошее решение значительно удалено от найденного, то вероятность выхода из данного локального минимума оказывается очень низкой. Таким образом, проблема заключается в потере генетического разнообразия.

Используя клеточные генетические алгоритмы, можно поддерживать разнообразие популяции сколь угодно долгое время, не теряя при этом сходимости. Суть идеи заключается в том, что некоторые параметры генетического алгоритма делаются зависимыми от положения особи в клеточном пространстве. В простейшем варианте такой модификации подвергается величина мутации d_{mut}. В одних областях мы делаем значение этого параметра высоким, в других — низким. Поэтому первые области будут *все время* генерировать достаточно случайные решения, поддерживая таким образом необходимое разнообразие всей популяции. Области второго типа будут использоваться по своему основному назначению — селекции и скрещиванию лучших решений.

Для экспериментов было выбрано три варианта зависимости параметра d_{mut} от координат i и j клеток. В первом варианте (NCGAs, рис. 1*а*) параметр d_{mut} зависит только от горизонтальной координаты, так что по краям поля значение этого параметра является высоким, в середине — низким. Во втором варианте мутация в центре поля является высокой, а по его краям — низкой (NCGAm, рис. 1*б*). Наконец, в третьем случае, области с высокой величиной мутации, располагались в четырех углах поля (NCGAc, рис. 1*в*).



(а) NCGAs (б) NCGAm (в) NCGAc

Рис. 1. Градиент величины мутации

5. Минимизация бимодальной двумерной функции

В задаче минимизации бимодальной целевой функции $F(\theta, x, y)$ измерялось частота (в процентах) обнаружения генетическим алгоритмом глобального минимума функции. Для этого выполнялось 100 запусков каждой версии генетического алгоритма. Полученные результаты показаны в таблице 1.

Таблица 1

θ	GA	CGA	NCGAs	NCGAm	NCGAc
0.001	0	0	4	4	4
0.002	0	0	5	5	5
0.005	0	0	6	8	9
0.010	0	0	10	15	14
0.020	0	0	28	24	25
0.050	3	15	61	47	54
0.100	30	76	87	72	80
0.200	74	100	100	95	97
0.500	100	100	100	100	100

Все алгоритмы в данном случае выполняли одинаковое число (500) итераций. Видно, что частота обнаружения глобального минимума неоднородными алгоритмами существенно выше по сравнению с однородными версиями. При этом заметим, что показатели неоднородного алгоритма (последние три столбца таблицы) могут быть улучшены за счет увеличения числа итераций, в то время как показатели однородных алгоритмов таким образом уже не улучшаются.

6. Минимизация мультимодальной двумерной функции

Минимизация двумерной функции Растригина $R(x, y)$ выполнялась с помощью однородных генетических алгоритмов (GA и CGA) и неоднородного алгоритма (NCGAs). Вычислялась частота обнаружения алгоритмом того или иного локального минимума R. Результаты численного эксперимента приведены в таблице 2. В первом столбце таблицы показаны первые пять локальных минимумов (0 — глобальный минимум), заметим, что в данной задаче нет локального минимума со значением 3. В остальных столбцах таблицы показаны частоты попадания указанных алгоритмов в заданные локальные минимумы. Для неоднородного алгоритма показано, как изменяются частоты ответов в зависимости от числа итераций. Таким образом, видно, что уже при 3000 итераций неоднородный алгоритм практически гарантированно находит глобальный экстремум (из имеющихся 400 локальных экстремумов).

7. Минимизация мультимодальной многомерной функции

В последнем эксперименте исследовалась частота попадания генетического алгоритма в локальные минимумы 10-мерной функции Растригина $R(x)$. В этом случае имеется порядка

Таблица 2

R	GA 500	CGA 500	NCGA 500	NCGA 1000	NCGA 2000	NCGA 3000
0	51	44	65	82	90	98
1	41	46	29	18	10	2
2	8	6	5	0	0	0
4	0	3	1	0	0	0
5	0	1	0	0	0	0

10^{13} локальных минимумов, из которых всего один является глобальным. Все минимумы имеют целые неотрицательные значения. Первые расчеты показали, что даже неоднородные алгоритмы не способны обнаружить глобальный минимум (а также близкие к нему локальные минимумы). Поэтому в алгоритм NCGAs была внесена еще одна неоднородность: вероятность выполнения оператора отбора для двух особей из соседних клеток была сделана зависящей от вертикальной координаты i:

$$p_{\text{sel}} = 1 - | i / w - 1|.$$

Таким образом, сверху и снизу клеточного поля отбор практически не работает, что дает образующимся за счет высокой мутации в угловых областях особям больше времени на улучшение целевой функции.

Численные результаты показаны в таблице 3. В первом столбце таблицы перечислены первые 8 минимумов функции, последняя строка соответствует минимумам с большими значениями функции. В однородных алгоритмах выполнялось по 500 итераций, в неоднородном – 1000, 3000 и 5000 итераций. Видно, что с увеличением числа итераций неоднородный алгоритм все чаще обнаруживает глобальный экстремум, в то время как однородные алгоритмы глобальный минимум практически не находят.

Таблица 3

R	GA 500	CGA 500	NCGA 1000	NCGA 3000	NCGA 5000
0	0	2	0	17	64
1	5	7	0	36	31
2	15	16	3	29	4
3	26	19	7	12	1
4	9	20	3	2	0
5	16	13	13	4	0
6	10	5	13	0	0
7	3	10	11	0	0
>7	16	8	50	0	0

8. Параллельная реализация

Была выполнена MPI-реализация предложенного варианта клеточных генетических алгоритмов, в которой клеточное поле делилось на горизонтальные полосы (рис. 2).

Такая схема подходит для систем с небольшим числом процессоров. Например, на рис. 3 показана зависимость ускорения от числа используемых процессоров. Если число доступных процессоров является большим, то имеет смысл рассмотреть аналогичную, но чуть более сложную реализацию, в которой поле делится на квадратные блоки.

9. Заключение

В результате выполненной работы были получены следующие результаты: введено понятие неоднородного клеточного генетического алгоритма; проведено численное сравнение рабо-

ты однородного генетического алгоритма, однородного клеточного генетического алгоритма и неоднородного генетического алгоритма на трех задачах непрерывной многомерной оптимизации; показаны преимущества предложенного подхода; предложена схема параллельной MPI-реализации клеточных генетических алгоритмов.

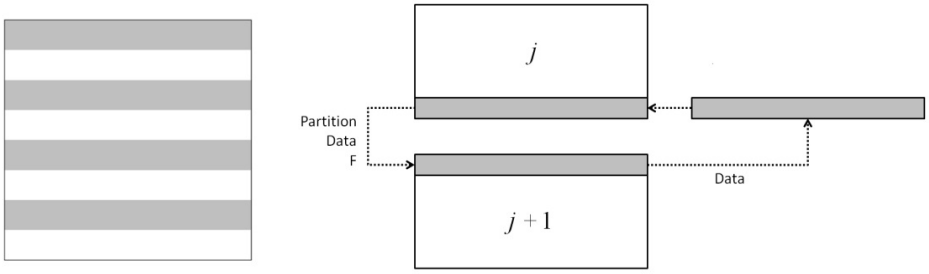


Рис. 2. Схема распараллеливания клеточного генетического алгоритма

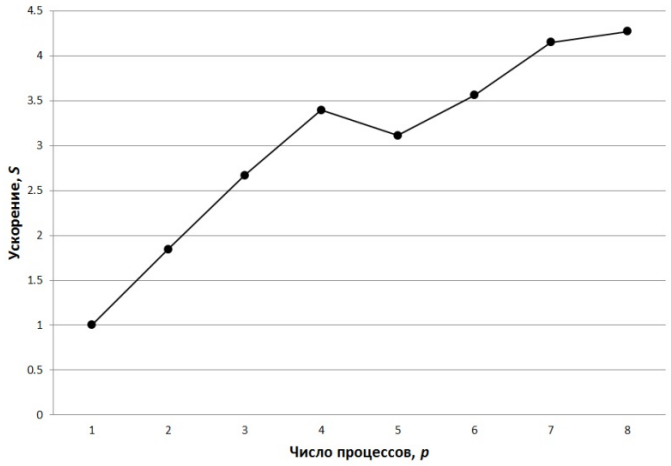


Рис. 3. Зависимость ускорения S от числа процессоров p при параллельной реализации клеточных генетических алгоритмов

Список литературы

Alba E. and Dorronsoro B. Cellular Genetic Algorithms. — Springer, 2008.

Ke Tang, Xiaodong Li, Suganthan P.N., Zhenyu Yang, Weise T. Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization, Technical report, University of Science and Technology of China (USTC). — 2010.

Whitley D. A genetic algorithm tutorial // Statistics and Computing. — 1994. — Vol. 4, No. 2. — P. 65–85.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 781−785

Ки&М

СТЕНДОВЫЕ ДОКЛАДЫ

УДК: 004.94, 378.1

Естественные модели параллельных вычислений

Н. М. Ершов[а], Н. Н. Попова[б]

Факультет вычислительной математики и кибернетики,
Московский государственный университет им. М. В. Ломоносова,
Россия, 119991, г. Москва, ГСП-1, Ленинские горы, 1-52

E-mail: [а]ershovnm@gmail.com, [б]popova@cs.msu.su

Получено 30 сентября 2014 г.

Курс «Естественные модели параллельных вычислений», читаемый студентам старших курсов факультета ВМК МГУ, посвящен рассмотрению вопросов суперкомпьютерной реализации естественных вычислительных моделей и является, по сути, введением в теорию естественных вычислений (natural computing) относительно нового раздела науки, образовавшегося на стыке математики, информатики и естественных наук (прежде всего биологии). Тематика естественных вычислений включает в себя как классические разделы, например клеточные автоматы, так и относительно новые, появившиеся в последние 10–20 лет, например методы роевого интеллекта. Несмотря на свое биологическое «происхождение», все эти модели находят широчайшее применение в областях, связанных с компьютерной обработкой данных. Исследования в области естественных вычислений также тесно связаны с вопросами и технологиями параллельных вычислений. Изложение теоретического материала курса сопровождается рассмотрением возможных схем распараллеливания вычислений, а в практической части курса предполагается выполнение студентами программной реализации рассматриваемых моделей с использованием технологии MPI и проведение численных экспериментов по исследованию эффективности выбранных схем распараллеливания вычислений.

Ключевые слова: естественные вычисления, эволюционные алгоритмы, искусственные биологические системы

Работа выполнена при финансовой поддержке РФФИ (грант №14-07-00628 А).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2015 Т. 7 № 3 С. 781−785

Ки&М

СТЕНДОВЫЕ ДОКЛАДЫ

Natural models of parallel computations

N. M. Ershov, N. N. Popova

Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, GSP-1, 1-2 Leninskiye Gory, Moscow, 119991, Russia

Abstract. — Course "Natural models of parallel computing", read for senior students of the Faculty of Computational Mathematics and Cybernetics, Moscow State University, is devoted to the issues of supercomputer implementation of natural computational models and is, in fact, an introduction to the theory of natural computing, a relatively new branch of science, formed at the intersection of mathematics, computer science and natural sciences (especially biology). Topics of the natural computing include both already classic subjects such as cellular automata, and relatively new, introduced in the last 10–20 years, such as swarm intelligence. Despite its biological origin, all these models are widely applied in the fields related to computer data processing. Research in the field of natural computing is closely related to issues and technology of parallel computing. Presentation of theoretical material of the course is accompanied by a consideration of the possible schemes for parallel computing, in the practical part of the course it is supposed to perform by the students a software implementation using MPI technology and numerical experiments to investigate the effectiveness of the chosen schemes of parallel computing.

Keywords: natural computing, evolutionary algorithms, artificial life

Работа выполнена при финансовой поддержке РФФИ (грант №14-07-00628 А).

Citation: *Computer Research and Modeling*, 2015, vol. 7, no. 3, pp. 781–785 (Russian).

1. Введение

Исследования в области естественных вычислительных моделей связаны с построением, анализом и применением методов и моделей, инспирированных разнообразными природными, прежде всего биологическими, системами. Эта тематика, включающая в себя такие понятия, как клеточные автоматы, искусственные нейронные сети, генетические алгоритмы, тесно переплетается с теорией искусственных биологических систем (artificial life).

Несмотря на биологическое «происхождение» большинства таких моделей, они находят широкое применение практически во всех областях, связанных с компьютерной обработкой данных: при моделировании в физике, химии, биологии, экономике и т. п.; в интеллектуальном анализе данных (data mining); при распознавании образов; для управления различными сложными системами и т. д.

История естественных вычислительных моделей началась практически одновременно с появлением первых электронных компьютеров. Одной из первых таких моделей стали клеточные автоматы [фон Нейман, 1971], придуманные Джоном фон Нейманов и Станиславом Уламом в 1940-х годах с целью моделирования процессов самовоспроизведения в живой природе. В середине 1950-х годов появляются модели искусственных нейронных сетей — персептронов [Розенблатт, 1965], разработанные Фрэнком Розенблаттом. Карл Петри в 1964 году в своей диссертации вводит понятие «сети Петри» [Котов, 1984], одним из предназначений которых было моделирование химических процессов. В конце 1960-х Аристид Линдемайер в рамках своих исследований по математическому моделированию процессов роста и формообразования растений строит формальную модель L-систем [Prusinkiewicz, Lindenmayer, 1996]. В 1970-х годах в работах Джона Холланда и его учеников формируется понятие генетических алгоритмов [Holland, 1975], являющихся в настоящее время одним из наиболее мощных методов решения сложных оптимизационных задач.

С 1990-х годов в данной области происходит мощный всплеск исследований, посвященных прежде всего разработке новых методов и основанных на моделировании тех или иных биологических систем, предназначенных для решения сложных (многомерных, многокритериальных, дискретных) задач оптимизации. Создаются муравьиные алгоритмы [Dorigo, Birattari, Stutzle, 2006], выполняются первые опыты в области ДНК-вычислений [Adleman, 1994], разрабатываются метод роя частиц [Kennedy, Eberhart, 1995], метод метод бактериального поиска [Passino, 2002], алгоритм пчелиного поиска [Pham et al., 2006]. Появляются и новые методы моделирования: искусственные иммунные системы, мембранные системы, программируемая материя и т. д. Этот всплеск был обусловлен во многом успехами в биологических исследованиях. Еще одной причиной появления большого количества новых методов и моделей явилась необходимость в решении все более сложных и более масштабных задач, прежде всего оптимизации, к которым оказались практически неприменимы существующие на то время классические алгоритмы и методы.

2. Параллельная структура естественных вычислительных моделей

Особенностью тематики естественных вычислительных моделей является то, что проведение исследований в этой области практически невозможно без использования компьютерной техники — основным методом исследований здесь является вычислительный эксперимент. Это связано во многом с дискретным характером используемых моделей и методов, что существенно затрудняет (или даже делает невозможным) какое-либо их аналитическое исследование. Поэтому, с учетом широчайшего применения таких методов, оказывается весьма актуальной задача построения их эффективных компьютерных и программных реализаций.

Общей чертой практически всех моделей в рассматриваемой области является то, что любая из них представляет собой систему взаимодействующих простых объектов. В клеточных автоматах такими объектами являются клетки, в нейронных сетях — нейроны, в генетических

алгоритмах — хромосомы и т. д. Эти объекты функционируют (развиваются, эволюционируют) параллельно друг с другом. Это значит, что любая искусственная биологическая модель обладает внутренним параллелизмом высокой степени, причем этот параллелизм является масштабируемым — увеличение размера системы приводит к пропорциональному увеличению степени ее параллелизма. Из вышеперечисленных фактов (широкое применение моделей рассматриваемого типа, необходимость в их эффективной компьютерной реализации, встроенный масштабируемый параллелизм) с очевидностью следует, что все такие модели являются весьма перспективными с точки зрения их реализации на современных массивно-параллельных вычислительных системах.

Вопросы, связанные с параллельной реализацией естественных вычислительных моделей, помимо их практической важности имеют и существенное методологическое и образовательное значение. Это связано с тем, что такие модели устроены, как правило, просто и поэтому достаточно легко могут быть реализованы на обычных последовательных компьютерах. В силу того, что различные модели отличаются друг от друга прежде всего способами взаимодействия (очень разнообразными) входящих в них объектов, проблема эффективной параллельной реализации фактически сводится к задаче оптимального отображения структуры коммуникации внутри модели на систему коммуникации параллельной вычислительной системы. Все это позволяет рассматривать данные модели в качестве весьма удобного средства для обучения технологиям параллельного программирования на всех типах современных параллельных вычислительных систем (с общей памятью, многопроцессорных кластеров, GPGPU).

Еще один немаловажный аспект, связанный с изучением естественных вычислительных моделей, заключается в том, что для большинства из них доказана алгоритмическая универсальность, что позволяет рассматривать такие модели в качестве теоретических моделей параллельных вычислений. С одной стороны, это способствует более глубокому пониманию параллельных вычислительных процессов, с другой — в будущем может привести к созданию новых вычислительных технологий, как это, например, происходит в настоящее время с ДНК-вычислениями.

3. Содержание курса

Программа учебного курса содержит два раздела. Первый раздел посвящен классическим моделям искусственных биологических систем. Второй раздел посвящен описанию современных моделей и методов в области естественных вычислительных моделей. Примерный список тем, освещаемых в данном курсе, выглядит следующим образом.

- Клеточные автоматы: понятие клеточного автомата, клеточные автоматы фон Неймана; клеточные автоматы Конвея, игра «Жизнь», алгоритмическая полнота автоматов Конвея, саморепликация в игре «Жизнь»; одномерные клеточные автоматы, типы поведения, способы определения, вопросы реализации; моделирование физических, химических и биологических процессов с помощью клеточных автоматов.
- Системы Линденмайера: понятие L-системы, классификация L-системы; система подстановок, эволюция, примеры построения фрактальных структур; моделирование процессов роста и формообразования с помощью L-систем; вариации L-систем — стохастические системы, контекстно зависимые системы, параметрические системы.
- Марковские автоматы: понятие марковского автомата, система подстановок, алгоритм применения; одномерные марковские автоматы; алгоритмическая универсальность; моделирование физических, химических и биологических систем с помощью марковских автоматов; двумерные марковские автоматы; алгоритмы параллельных подстановок.
- Сети Петри; понятие «сети Петри», места, переходы, метки; функционирование сетей Петри; классификация традиционных сетей Петри; моделирование с помощью сетей Петри; временные сети Петри, сети Петри с ингибиторными дугами, алгоритмическая универсальность; цветные сети Петри.

- Нейронные сети: понятие естественной нейронной сети, нейроны, синапсы, обработка информации в нервной системе; искусственный нейрон, искусственные нейронные сети; персептрон Розенблатта; многослойные персептроны, алгоритм обучения Error Back Propagation; рекуррентные нейронные сети, сети Хопфилда; применение искусственных нейронных сетей.
- ДНК-вычисления: понятие ДНК, операции над ДНК, синтез, анализ, секвенирование; применение ДНК для решения вычислительных задач; опыт Адлемана, кодирование графа, алгоритм отбора; применение ДНК для решения задачи SAT3, схема кодирования Липтона, алгоритм решения; стикерная модель.
- Мембранные системы: понятие P-систем; мультимножества и операции над мультимножествами; мембраны, способы их взаимодействия; алгоритмическая универсальность мембранных систем; решение с помощью мембранных систем сложных задач комбинаторной оптимизации.
- Генетические алгоритмы: основные понятия генетического кодирования; обобщенная схема генетического алгоритма, функция приспособленности, операторы отбора, мутации и скрещивания; функционирование генетического алгоритма; области применения, решение сложных комбинаторных задач с помощью генетических алгоритмов.
- Муравьиные алгоритмы: понятие муравьиного алгоритма; теоретические результаты; метаэвристика муравьиной колонии; вариации муравьиных алгоритмов; параллельная реализация; применение муравьиных алгоритмов для решения сложных оптимизационных задач, роевая робототехника.
- Алгоритмы роевой оптимизации: модель Рейнолдса коллективного поведения стаи птиц; метод роя частиц, вариации метода, параллельная реализация; метод бактериального поиска; пчелиные алгоритмы.

Изложение теоретического материала курса сопровождается рассмотрением возможных схем распараллеливания вычислений, а в практической части курса предполагается выполнение студентами программной реализации рассматриваемых моделей с использованием технологии MPI и проведение численных экспериментов по исследованию эффективности выбранных схем распараллеливания вычислений.

Список литературы

Котов В. Сети Петри. — М.: Наука, 1984.

фон. Нейман Дж. Теория самовоспроизводящихся автоматов. — М.: Мир, 1971.

Розенблатт Ф. Принципы нейродинамики: перцептроны и теория механизмов мозга. — М.: Мир, 1965.

Adleman L. M. Molecular computation of solutions to combinatorial problems // Science. — 1994. — 266, 11. — P. 1021–1024.

Dorigo M., Birattari M., Stutzle T. Ant colony optimization, technical report No. TR/IRIDIA/2006-023, September 2006.

Holland J. H. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.

Kennedy J., Eberhart R. C. Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, P. 1942–1948, 1995.

Passino K. M. Biomimicry of bacterial foraging for distributed optimization and control // IEEE Control Systems Magazine. — 2002. — 22. — P. 52–67.

Pham D. T., Ghanbarzadeh A., Koc E., Otr S. I., Rahim S., Zaidi M. The Bees Algorithm. — A Novel Tool for Complex Optimisation Problems // Proceedings of IPROMS 2006 Conference. — 2006. — P. 454–461.

Prusinkiewicz P., Lindenmayer A. The Algorithmic Beauty of Plants. — Springer-Verlag, 1996.