

Начальный этап исследования стенда передачи Больших Данных по параллельным каналам данных. SDN подход

С.Э. Хоружников,¹ В.А. Грудинин,¹ О.Л. Садов,¹

А.Е. Шевель,^{1,2} В.Б. Титов,^{1,3} А.Б. Каирканов¹

¹Факультет инфокоммуникационных технологий,
университет ИТМО, Санкт-Петербург, Россия

²Петербургский институт ядерной физики, Россия

³Санкт-Петербургский государственный университет, Санкт-Петербург, Россия

Передача Больших Данных по компьютерным сетям важная и неизбежная операция в прошлом, в настоящем и в обозримом будущем. Целый ряд разрабатываемых и уже работающих астрономических проектов связан с Большими Данными. Есть ряд методов передачи данных по глобальной сети (Интернет) с многообразными инструментами. В этой работе рассматривается передача фрагмента Больших Данных от точки Интернета к другой точке, вообще говоря на далекое расстояние: порядка тысяч километров. Анализируются несколько свободно распространяемых систем передачи больших данных. Отмечаются важнейшие архитектурные свойства и предлагается использовать методы протокола SDN Openflow для тонкой настройки передачи данных по параллельным каналам связи.

Ключевые слова: Астрономические приложения, передача данных, Linux, ПКС, Openflow, сети

1. ВВЕДЕНИЕ

Проблема «Больших Данных»¹ известна много лет. В разные периоды термин «Большие Данные» подразумевал разный объем и характеристики данных. Имея в виду «три V»: Скорость (Velocity), Объем (Volume), Разнообразие (Variety), мы можем обратить внимание на то, что все эти черты относятся к текущему состоянию технологии. Например, в 80-х объем в 1 ТБ рассматривался как гигантский объем. В 1824 году Чарльз Бэббидж получил Золотую медаль Королевского Астрономического общества «за свое изобретение машины для вычисления математических и астрономических таблиц» с беспрецедентной точностью и скорость этих вычислений соответствовала «Большим Данным» того времени. Разработка в 70-е–80-е

¹ http://en.wikipedia.org/wiki/Big_data

FITS-формата стандартизовала обмен изображениями между различными астрономическими учреждениями. В наши дни разрабатываются и используются системы, которые работают с Большими Данными в современном понимании. Стандарты доступа к большим массивам астрономических данных (стандарты для метаданных, форматов, языков запроса и т. д.), технологии работы, в том числе с Большими Данными, разрабатываются и поддерживаются международным альянсом IVOA (International Virtual Observatory Alliance), который был создан, чтобы «способствовать международной координации и сотрудничеству, необходимому для разработки и внедрения инструментов, систем и организационных структур, необходимых для обеспечения международного использования астрономических архивов, как единой и функционирующей виртуальной обсерватории». Все проекты, работающие с Большими Данными, руководствуются рекомендациями IVOA. Таковы проекты ESO/VLT, NOAO/СТЮ, NASA/Kepler, NASA/HMS и др.

Есть ряд сторон этой проблемы: хранение, анализ, передача и т. д. В этой статье мы обсуждаем один из важных аспектов Больших Данных: передачу по глобальной компьютерной сети.

2. ИСТОЧНИКИ БОЛЬШИХ ДАННЫХ

Известен длинный список видов человеческой деятельности (научной и деловой), которые являются генераторами значительного объема данных [1–3], см. проекты SKA², LSST³, FAIR⁴, ITER⁵, а также сайты CERN⁶ и CLDS⁷.

Согласно [1] полный объем деловой переписки в мире в 2012 году составил примерно 3000 ПБ ($3 \cdot 10^{18}$). Общепринятая оценка полного объема сохраняемых данных с 2000 года возрастает ежегодно в 1.5–2 раза. В этой работе (и для наших тестирований) мы будем полагать, что данные объемом около 100 ТБ (10^{14}) и более можно называть Большими Данными. Наверняка объем Больших Данных со временем будет расти.

Другой аспект Больших Данных — сохранение данных в течение долгого периода времени: несколько десятилетий и более. Многие стороны нашей личной, общественной, технической или деловой жизни хранятся теперь в цифровой форме. Большие объемы таких данных нуж-

² <http://skatelescope.org/>

³ <http://www.lsst.org/lsst/>

⁴ <http://www.fair-center.eu/>

⁵ <http://www.iter.org/>

⁶ <http://www.cern.ch/>

⁷ <http://clds.sdsc.edu/>

но запомнить и хранить. Например, результаты медицинских тестов, данные, порожденные разного сорта значимыми двигателями (авиадвигатели, генераторы электростанций и т. д.), и другие данные должны архивироваться на долгое время. Хранимые данные будут содержаться в распределенных (локально и глобально) хранилищах. Принимается, что точные копии (replicas) хранимых данных должны храниться в нескольких местах (континентах), чтобы исключить потерю данных в технических, природных или социальных бедствиях.

Исторически одной из первых областей, где появились Большие Данные, была Физика Высоких Энергий (HEP). В результате был исследован ряд аспектов передачи данных и был решен целый круг задач. В настоящее время все больше научных и деловых областей имеют дело (или планируют это сделать) с «Большими Данными» [4]. Вот список разрабатываемых или уже работающих астрофизических/физических проектов [5–9]:

- Hubble: 1989–1992 гг., общий объем данных 300 ГБ;
- ESO/VLT: с 1999 г., общий размер наблюдений 65 ТБ и растет на 15 ТБ/год;
- NASA⁸/KEPLER⁹: с 2009 г., 100 ГБ/мес;
- LOFAR: LOw Frequency ARray¹⁰, 2012 г., до 1 ПБ/сут;
- GAIA: Global Astrometric Interferometer for Astrophysics, 1 ПБ/год;
- ПРАО (Пушино): все проекты, 10-100 ГБ/сут;
- Радиоастрон: 1.28 ТБ/сут;
- CERN, все проекты, 1 ПБ/сут;
- LSST: Large Synoptic Survey Telescope, 2020 г., объем данных 10 ПБ/год;
- ITER: International Thermonuclear Experimental Reactor, 2020 г., 1 ПБ/сут;
- CTA: Cherenkov Telescope Array, 2015–2020 гг., 20 ПБ/год;
- SKA: Square Kilometer Array, 2019–2024 гг., 1500 ПБ/год.

⁸ <http://www.nas.nasa.gov/>

⁹ http://www.nasa.gov/mission_pages/kepler/

¹⁰ <http://www.lofar.org/>

Четыре из приведенных проектов еще не реализованы, и, чем дальше срок реализации, тем больше данных планируется получить. В конце декабря 2013 года вышел на планируемую работу GAIA, в настоящее время уже идет поток информации, который выйдет на 1 ПБ/год.

Часто наблюдатели не имеют возможности долго хранить данные, и хранится только избранная часть данных [10]. Для глубокого анализа данных требуется распределить полученные данные между участниками коллектива по всему миру. Это означает, что значительная часть экспериментальных данных должна передаваться по Интернету.

3. СВОБОДНО ДОСТУПНЫЕ УТИЛИТЫ/ИНСТРУМЕНТЫ ДЛЯ ПЕРЕДАЧИ ДАННЫХ ПО СЕТИ

Время передачи по глобальной компьютерной сети (Интернет) зависит от реальной пропускной способности канала связи и объема данных. Учитывая, что мы говорим об объемах в 100 ТБ и более, мы можно оценить минимальное требуемое время для копирования данных по сетевому каналу с пропускной способностью в 1 Гбит. Это дает нам около 100 МБ/с, следовательно, $100\text{ТБ}/100\text{МБ/с}=1\,000\,000\text{ секунд}=277.8\text{ часа}=11.6\text{ суток}$. В течение этого периода параметры сетевого канала могут измениться. Например, может значительно варьироваться процент потерянных сетевых пакетов. Канал данных может страдать от прерываний операций на различные периоды: секунды, часы, дни.

Теперь давайте посмотрим на сетевые параметры ядра Linux. В директории `/proc` в Scientific Linux (клон RedHat) версии 6.5 имеется около полутысячи параметров, описывающих сетевой канал в ядре. Не все они одинаково чувствительны или оказывают влияние на процесс передачи данных. Наиболее важные из них это размер окна TCP, MTU, алгоритм устранения перегрузки и т. д. Конечно, очень важно число независимых сетевых каналов, которые можно было бы использовать параллельно. Важны также такие сетевые параметры, как время прохождения сигнала туда и обратно (RTT) и процент потерянных сетевых пакетов. Наконец, понятно, что, чтобы достичь максимальной скорости передачи данных, в каждой передаче данных большого объема нам нужно иметь возможность в течение процесса передачи настраивать (устанавливать) разное число потоков (threads), разный размер окна TCP и т. д.

Рассмотрим теперь свободно доступные инструменты/утилиты передачи данных, которые можно использовать для передачи Больших Данных по сети.

А. Концепции сравнения утилит передачи данных

Прежде всего, краткий обзор характеристик для сравнения утилит передачи данных, которые могли бы помочь при передаче данных.

- Режим многопоточной передачи данных — способность использовать несколько потоков ТСР в параллель.
- Режим многоканальной передачи данных — способность использовать более, чем один канал в параллель; важное свойство, особенно, если есть возможность учитывать, что доступные сетевые каналы не одинаковы по производительности и условиям (надежность, цена, реальное состояние и т. д.).
- Возможность устанавливать параметры нижнего уровня, например, размер окна ТСР и т. д.
- Метод обхода сетевых проблем (ошибок, задержек и т. д.). Другими словами, возможно ли продолжить передачу данных после перезапуска в случае сбоя при передаче?

По сути передача данных состоит из многих шагов: чтение данных из хранилища, передача данных по сети, запись полученных данных в хранилище на удаленной компьютерной системе. В этой работе наше внимание концентрируется больше на сетевом процессе передачи.

В. Утилиты/инструменты передачи данных низшего уровня

Мы можем отметить несколько утилит для передачи данных по сети (по крайней мере, часть из них известны уже лет десять):

- один из протоколов низшего уровня для передачи данных по сети — UDT¹¹. UDT это и библиотека, которая реализует протокол передачи данных, позволяющий использовать *udt*, а не *tcp*. В некоторых случаях библиотека может помочь улучшить использование канала данных, т. е. уменьшить время передачи.
- протокол RDMA по конвергированному Ethernet'y (RDMA over Converged Ethernet, RoCE) изучался в [4]. Обнаружилось, что во многих случаях RoCE показывает лучшие результаты, чем UDP, UDT, стандартный ТСР.

¹¹ <http://udt.sourceforge.net/>

- МР TCP¹² — интересный протокол, который для одной передачи данных позволяет в параллель использовать несколько каналов данных. Протокол реализован как драйвер ядра Linux.
- семейство (open) ssh¹³ — хорошо известные утилиты передачи данных предоставляет строгую аутентификацию и ряд алгоритмов шифрования данных. Возможно также сжатие данных до шифрования для уменьшения объема передаваемых данных. Есть две известные (open) разновидности SSH: подкорректированная версия SSH¹⁴, которая может использовать увеличенный размер буферов и SSH с аутентификацией Globus GSI. Полноценного возобновления работы после сбоя нет. Нет и параллельных потоков передачи данных.
- bbcp¹⁵ — утилита для смешанной передачи данных. Предполагается, что bbcp выполняется на обеих сторонах, т. е. передатчик, как клиент, и получатель, как сервер.
- утилита bbftp¹⁶ для передачи массивов данных. Он реализует свой собственный протокол передачи, который оптимизирован и для больших файлов (больше чем 2 ГБ), и защищен, поскольку он не читает пароль из файла и шифрует информацию о соединении.
- Xdd [11] — утилита, разработанная для оптимизации передачи данных и процессов ввода/вывода для систем хранения.
- fdp¹⁷ — Java-утилита для многопоточной передачи данных.
- gridFTP¹⁸ усовершенствованная утилита передачи данных для среды инфраструктуры безопасности Globus (GSI).

Многие из них очень эффективны для передачи данных с точки зрения использования пропускной способности канала. Однако передача Больших Данных предполагает значительное

¹² <http://mptcp.info.ucl.ac.be/>,

<http://multipath-tcp.org/>

¹³ <http://www.openssh.org/>

¹⁴ <http://sourceforge.net/projects/hpnssh/>

¹⁵ <http://www.slac.stanford.edu/~abh/bbcp/>

¹⁶ <http://doc.in2p3.fr/bbftp/>

¹⁷ <http://monalisa.cern.ch/FDT/>

¹⁸ там же

время передачи (может, несколько часов, дней или еще больше). Для длительных промежутков трудно положиться на такие простые процедуры передачи. Как мы уже упоминали, может измениться пропускная способность и процент потерянных пакетов в сетевом канале, может израсходоваться квота дискового пространства и так далее.

С. Службы передачи файлов среднего уровня

FTS3¹⁹ — относительно новый и перспективный инструмент для передачи данных большого объема по сети. Он имеет множество свойств, в том числе уже упомянутых. Есть усовершенствованная возможность для отслеживания передачи данных (log), возможность использовать интерфейсы http, restful и CLI для контроля процесса передачи данных.

Другая интересная разработка — SHIFT²⁰, которая предназначена для выполнения надежной передачи данных в LAN и WAN. Уделено много внимания надежности, усовершенствованного отслеживания, производительности передачи данных и использования параллельной передачи данных между так называемыми эквивалентными хостами (между компьютерными кластерами).

Д. Служба управления данными высокого уровня: PhEDEx

PhEDEx^{21,22} — экспорт данных физических экспериментов (Physics Experiment Data Export) используется (и разрабатывается) в сотрудничестве по эксперименту Компактный Мюонный Соленоид (Compact Muon Solenoid, CMS) [12, 13] в ЦЕРНЕ. Эксперимент дает большое количество экспериментальных данных (в 2013 году было записано около 130 ПБ). Обработка данных требует копирования данных на ряд больших компьютерных кластеров (около 10 мест в различных странах и континентах) для анализа и архивирования данных. Позже части данных могут быть скопированы на меньшие вычислительные ресурсы (более, чем 60 мест). Полный объем передачи данных достигает 350 ТБ/сут [13]. В ближайшем будущем ежедневный объем, возможно, будет расти. Поскольку между несколькими сайтами может

¹⁹ http://www.eu-emi.eu/products/-/asset_publisher/1gkD/content/fts3;
<https://svnweb.cern.ch/trac/fts3>

²⁰ <http://fasterdata.es.net/data-transfer-tools/>

²¹ <https://cmsweb.cern.ch/phedex>,
<https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhedexAdminDocsInstallation> и <http://hep-t3.physics.umd.edu/HowToForAdmins/phedex.html>

²² <http://fasterdata.es.net/data-transfer-tools/>

быть более одного канала, в PhEDEx разработан метод маршрутизации, который позволяет проверить альтернативный маршрут, если умалчиваемый маршрут недоступен.

Наконец, система PhEDEx является очень сложной, и служба управления зависит от среды кооперации физических экспериментов. Маловероятно, что PhEDEx можно использовать в другой среде без модификации.

4. ОБСУЖДЕНИЕ

Отмеченные утилиты имеют несколько полезных для передачи данных свойств. Среди них:

- все утилиты имеют архитектуру клиент-сервер;
- способны устанавливать размер буфера, размер окна TCP и т. д.;
- имеют возможность выполнять различные операции до реальной передачи данных и после передачи данных, например, сжатие/восстановления, использовать ряд драйверов/методов для чтения/записи файлов на/с вспомогательную память;
- использовать ряд методов аутентификации;
- использовать для передачи данных более одного потока, более одного сетевого канала;
- использовать несколько алгоритмов аутентификации;
- использовать ряд методов, чтобы сделать передачу данных надежнее;
- утилиты не одинаковы по числу параметров и сфере решаемых задач. Часть из них хорошо подходит для использования в качестве независимых утилит передачи данных почти во всех средах. Другие, как PhEDEx (в CMS) и сравнимых системах в кооперации ATLAS²³ предназначены для использования, как часть более сложной и специфической компьютерной среды.

Другими словами, есть набор инструментов, которые во многих случаях могут помочь передать Большие Данные по сети. В то же время ясно, что целый ряд утилит может использовать более одного сетевого канала.

²³ <http://rucio.cern.ch/>

Вместе с тем не предлагается никаких средств тонкой настройки параллельных каналов данных. Тонкая настройка рассматривается как возможность применить различные правила к различным каналам данных. Вообще говоря, параллельные каналы данных могли бы быть совершенно различными по природе, свойствам и условиям использования. В частности, при передаче данных предполагается использование QoS для каждого сетевого канала и способность оперативно, на лету изменять правила. Все это приводит к мысли, что необходимо специальное приложение, которое могло бы следить за состоянием каналов данных и изменять параметры передачи согласно реальной ситуации в каналах данных. Параметры сетевого канала планируется устанавливать, используя протокол OpenFlow²⁴ [14]. Чтобы следить за состоянием каналов данных, будет использовано специальное средство PerfSonar [15].

Очевидно, требуется специализированная установка для теста, чтобы исследовать процесс передачи данных с помощью приведенных утилит и описанного оборудования. Настраиваемый испытательный стенд должен иметь возможность моделировать, по крайней мере, основные сетевые проблемы, например, изменение RTT, задержки, процент утраченных пакетов и т. д. Разработка такого стенда начата в ИТМО, в лаборатории сетевых технологий в распределенных вычислительных системах²⁵. Такое направление работы привлекает много исследователей [16].

Ожидается, что стенд будет платформой для сравнения различных утилит в одной и той же среде. Как первый шаг планируется выполнить сравнительные измерения с набором утилит передачи данных, подробно записывая все условия измерений. Это позволит в будущем сравнить на стенде другие методы передачи данных точно в такой же среде.

5. РАБОТА СТЕНДА

Стенд состоит из двух серверов HP DL380p Gen8 E5-2609, Intel(R) Xeon(R) CPU E5-2640 @2.50GHz, 64 GB под Scientific Linux 6.5. Поскольку планируется все тестировать в виртуальной среде для каждой упомянутой системы передачи данных, используется две виртуальные машины. Одна VM, как передатчик, а другая — как приемник. Другими словами, у нас есть около десяти VM. Чтобы соединить эти VM, развернута и запущена платформа Openstack с OpenStack.org. Также развернута PerfSonar.

²⁴ <https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf>

²⁵ <http://sdn.ifmo.ru/>

Для изучения различных типов данных была разработана специальная процедура, генерирующая тестовую директорию с файлами произвольной длины, полный объем тестовой директории определяется параметром процедуры. При генерации тестовых данных возможно задать среднее значение и дисперсию размера файла. Данные в каждом файле тестовой директории намеренно готовятся так, чтобы исключить возможный эффект сжатия данных (если таковое есть) при передаче данных.

На начальной стадии планируется сравнить все приведенные системы передачи данных в локальной сети, чтобы убедиться, что все (все скрипты) функционирует должным образом. Отдельная проблема — записать в период измерений все журналы, параметры и т. д. В частности, это подразумевает требование автоматически записывать всю директорию /proc в некоторое место, скажем, «директорию log». Также требуется записать все параметры и сообщения движка/утилиты передачи данных. Наконец, состояние канала данных также предполагается записывать. Вся упомянутая информация должна сохраняться в «директории log». Очевидно, все должно осуществляться скриптами, предназначенными для выполнения измерений.

Разработанные процедуры (скрипты) и краткие описания записаны на сайте <https://github.com/itmo-infocom/BigData>.

6. ЗАКЛЮЧЕНИЕ

При планировании проекта, который имеет дело с большим объемом экспериментальных данных, важно учесть затраты на передачу данных по сети. Можно указать несколько пунктов в наблюдательном цикле, где передача Больших Данных по сети является реальной необходимостью:

- сбор данных;
- быстрый контроль данных (и/или фильтрация);
 - *возможная передача данных (может быть локальной или удаленной);*
- сохранение данных во внешней памяти;
 - *возможная передача данных на удаленный вычислительный центр (может быть в несколько пунктов) для дальнейшего анализа;*
- анализ данных.

В этой работе описывается только техника передачи данных, которая является неотъемлемой частью наблюдательного цикла. Ясно, что в предстоящих экспериментах, где ожидается громадный объем данных, чем эффективнее передача данных, тем продуктивнее научные исследования.

БЛАГОДАРНОСТИ

Работа поддержана Санкт-Петербургским национальным исследовательским университетом информационных технологий, механики и оптики (www.ifmo.ru).

-
1. J. Pearlstein, Information Revolution: Big Data Has Arrived at an Almost Unimaginable Scale // <http://www.wired.com/magazine/2013/04/bigdata/>.
 2. L. Borovick, R.L. Villars. White paper. The critical Role of the Network in Big Data Applications // http://unleashingit.com/docs/B13/Cisco\%20UCS/critical_big_data_applications.pdf
 3. W.E. Johnston, E. Dart, M. Ernst, B. Tierney // Enabling high throughput in widely distributed data and analysis systems: Lessons from the LHC // <https://tnc2013.terena.org/getfile/402>, <https://tnc2013.terena.org/getfile/716>
 4. B. Tierney, E. Kissel, M. Swany, E. Pouyoul // Efficient Data Transfer Protocol for Big Data, http://www.es.net/assets/pubs_presos/eScience-networks.pdf // Lawrence Berkley National Laboratory // School of Informatics and Computing, Indiana University.
 5. M. Juric, J. Kantor, T.S. Axelrod, et al. 2013, American Astronomical Society Meeting Abstracts No 211, 221, N247.01.
 6. P. Dewdney, SKA1 System Baseline Design, 12.03.2013 07:09, SKA-TEL-SKO-DD-001. Revision: 1.
 7. B.S. Acharya, et al. 2013, Astroparticle Physics, v. 43, pp. 3–18, <http://dx.doi.org/10.1016/j.astropartphys.2013.01.007>.
 8. P. De Teodoro et al. Data Management at GAIA Data Processing Centers // Astrostatistics and Data Mining, ed. by L.M.Sarro et al., Springer Series in Astrostatistic 2, DOI 10.1007/978-1-4614-3323-1_10, 2012
 9. Е.А. Исаев, В.В. Корнилов, П.А. Тарасов, В.А. Самодуров, М.В. Шацкая, Препринт ФИАН, No. 8 (2014).
 10. S. Karpov, et al. Acta Polytechnica, **53**, 1, 38–43 (2013).
 11. S.W. Hodson, S.W. Poole, T.M. Ruwart, B.W. Settlemeyer // 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>.

12. The CMS Collaboration 2008. The CMS experimental at the CERN LHC JINST 3 S08004.
13. R. Kaselis, S. Piperov, N. Magini, J. Flix, O. Gutsche, P. Kreuzer, M. Yang, S. Liu, N. Ratnikova, A. Sartirana, D. Bonacorsi, J. Letts // 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
14. B.A.A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turlatti // A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks, http://hal.inria.fr/index.php?halsid=ig58511e1q1ekqq75uud43dn66&view_this_doc=hal-00825087&version=5.
15. J. Zurawski, S. Balasubramanian, A. Brown, E. Kissel, A. Lake, M. Swany, B. Tierney, and M. Zekauskas // perfSONAR: On-board Diagnostics for Big Data // http://www.es.net/assets/pubs_presos/20130910-IEEE-BigData-perfSONAR2.pdf
16. D. Gunter, et al // Exploiting Network Parallelism for Improving Data Transfer Performance // <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6496123> // High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion, ISBN 978-1-4673-6218-4, DOI: 10.1109/SC.Companion.2012.337.