Big Data transfer over long distant global computer networks

Sergey Khoruzhnikov, Vladimir Grudinin,
Oleg Sadov, Arsen Kairkanov
Faculty of Infocommunication Technologies
ITMO University
St.Petersburg, Russian Federation

Andrey Shevel
National Research Centre "Kurchatov Institute"
B.P.Konstantinov Petersburg Nuclear Physics Institute
Gatchina, Russian Federation

Abstract—**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 Big 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.

I. INTRODUCTION

The "Big Data" [1] 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 are 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 it 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 [2-7, 44,49-50].

In according [2] 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 aspect 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 were solved. Now more and more scientific and business sectors are dealing (or plan to) with the "Big data" [2-8].

III. DATA TRANSFER METHODS

In [9] authors considered the cost and constrains in data transfer. The system entitled Pandora helps to create plan for optimal data transfer with several possible ways over computer networks and traditional shipping chain like post company (for example, UPS, Pony Express, etc.) Let us discuss UPS (or PonyExpress or whatever) method and imagine we need to move 100TB of data from city A to city B. One of the methods is to buy around 25 disk drives (4 TB each), write the data to the disks at city A (to speed up the process you need 25 empty disk slots to insert all the disks to write in parallel all of them). After that you need to send all written disk drives by one of the post companies, and, finally, insert all the drives in city B into empty disk slots. The estimated time to transfer 100TB in such the way is around (taking into account that writing speed to the disk drive is not faster than 200MB/sec) 4TB/200MB/sec = 20000 secs = 5.56 hours + time to deliver by post company (quite probably about a day). So in total it will take 24+5.69 =29.56 hours = 106,416 secs. Average speed might be estimated as 100TB/106,416secs = 939.71 MB/sec.

More realistic estimation of data transfer time has to be done in different way: total volume of data to be transferred has to be divided by time between start point when decision of the data transfer is approved until the time when data is already transferred and available on remote computer. In above theoretical examples if we take into account all preparation time to move the data volume on both sides we will get average speed much less (depends on circumstances and can be thousand times) than in above estimation.

More conventional now method is to transfer the data over the computer network.

IV. 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 sec = 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 can be varied significantly. The data link might suffered of operation interruptions for different period: seconds, hours, days.

Now let us look at the Linux kernel network parameters. There are several hundreds of kernel network parameters. It can be seen in the directory /proc on Scientific Linux (clone of RedHat) version 6.5

-bash-4.1$ /sbin/sysctl -a | grep "^net\." | wc -l

484 (or something like that)

In other words there are about ½ thousand parameters describing the network link in the kernel. Not all of them are equally sensitive or influencing. Most important of them are TCP Window size, MTU, congestion control algorithm, etc. Of course quite important the number of independent data links which could be used in parallel. Also there are important network parameters like round trip delay time (RTT) and % of lost network packages. 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.

- Data compression on the fly – it is generally important feature especially in the case when compression methods can be applied by choice, because it might significantly decrease the data volume to be transferred. However this option might consume a lot of CPU time which in turn could introduce additional delay in data transfer process.
- 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.
- Data encryption on the fly.
- The method to bypass the network problems (errors, timeouts, etc.) In other words: in case of failure of the data transfer is it possible to continue the transfer after restart?

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):

1) One of low level protocols to transfer the data over the network is UDT [25]. 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.

2) The protocol RDMA over Converged Ethernet (RoCE) [8] has been studied and it was found that in many cases RoCE shows better results than UDP, UDT, conventional TCP.

3) MP TCP [29] is interesting protocol which permits to use several data links in parallel for one data transfer. The protocol is implemented as Linux kernel driver.

4) (Open) ssh family [52] - 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 (open) SSH flavors: patched SSH version [53] which can use increased size of buffers and SSH with Globus GSI authentication. No real restart after failure. No parallel data transfer streams.

5) BBCP [22] — 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:

a) TCP Window size;

b) number of TCP streams;

c) I/O buffer size;

d) compression on the fly;

e) multi-directory copy;

f) resuming failed copy;

g) authentication with ssh;

h) using pipes, where source or/and destination might be pipe;

i) special option to transfer small files;

j) and many other options dealing with many practical details.

6) BBFTP [21] 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

*i*n a file and encrypts the connection information. BBFTP main features are:

a) encoded user name and password at connection;

b) SSH and Grid Certificate authentication modules;

c) multi-stream transfer;

d) big windows as defined in RFC1323;

e) on-the-fly data compression;

f) automatic retry

g) customizable time-outs;

h) transfer simulation;

i) AFS authentication integration.

7) Xdd [46] – utility developed to optimize data transfer and I/O processes for storage systems.

8) FDP [23] – Java utility for multi-stream data transfer.

9) GridFTP [23] 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:

a) two security flavors: Globus GSI and SSH;

b) the file with host aliases: each next data transfer stream will use next host aliases (useful for computer cluster);

c) pipes;

d) special debugging mode to find bottleneck in data transfer;

e) backend module name for source and destination *sites;*

f) *number of parallel data transfer streams;*

g) *buffer size;*

h) *restart failed operations and number of restarts.*

Many of them 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 rely on so simple transfer procedures. As we mention above the network link might change the capacity and percent of lost network packages, the disk space may run out of quota, and so on.

C. *Middle level File Transfer Service*

The FTS3 [24] 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 [30] which is dedicated to do reliable data transfer in LAN and WAN. There were 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 [10-13] at CERN [7]. 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 [11]. 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.

V. Consideration

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

- all utilities have client-server architecture;
- are able to set the buffer size, TCP Window size, etc.;
- have the ability to perform various operations before real data transfer and after data transfer, e.g. compression/decompression, use a range of drivers/methods to read/write files to/from secondary storage;
- use a number of authentication techniques;
- use more than one stream, more than one data link for data transfer;
- use several authentication algorithms;
- 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 [14] 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 data link.

No tool suggests fine tuning of more than one network data link usage. Fine tuning is considered as possibility to apply the policy to use more than one data link. In particular it is assumed QoS for each data link to be used in data transfer and ability to change the policy on the fly. It seems such the abilities are important for Big Data transfer.

Another aspect connected to the procedure of the comparison of the utilities to transfer of Big Data over the computer network. The networks are different from each other. Also even in one data link in different time someone could meet completely new situation which might require another parameters values for the data transfer to accomplish the transfer in minimum time. 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 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 [54]. To watch the data link status the tool PerfSonar [41] is started to be used.

The testbed is intended to be platform to compare different utilities in the same environment. In addition it is planned to use advanced techniques to use more than one data link with taking into account imposed QoS on each data link. QoS might be set with protocol Openflow. That is part of Software Defined Network (SDN) approach which was described elsewhere [17,18]. At first step it is planned to perform comparative measurements (comparative study) of 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.

VI. The testbed progress

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. It is planned to test several mentioned data transfer systems (initially: bbcp, bbftp, fts3, GridFTP) in virtual environment. In addition to routine test results like data transfer throughput and reliability, the special procedure to simulate different types of data link problems (percent of lost packages, operation interruption) is under development. For each tested utility two virtual machines will be used. One VM as transmitter and another VM as receiver. In other words we have around ten VMs. To organize those VMs the system Openstack.org (version Icehouse) has been deployed. 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 generated test directory, average size of the files, dispersion of the file sizes are defined by the parameter of the procedure. The data inside each file in test directory is intentionally prepared to eliminate possible affect of the data compression (if any) during data transfer.

In initial stage it is planned to compare above data transfer systems in local area network to be sure that everything (all scripts) is functioning properly. The distinct problem is to save 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 save 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". Obviously everything has to be performed by scripts dedicated to do measurements.

Developed procedures (scripts) and short descriptions are written in the site https://github.com/itmo-infocom/BigData.

VII. Acknowledgement

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

References

[1] Big Data – http://en.wikipedia.org/wiki/Big_data.

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

[3] Square Kilometer Array - http://skatelescope.org/.

[4] Large Synoptic Survey Telescope - http://www.lsst.org/lsst/.

[5] Facility for Antiproton and Ion Research –http://www.fair-center.eu/.

[6] International Thermonuclear Experimental Reactor – http://www.iter.org/.

[7] CERN - http://www.cern.ch/.

[8] Brian Tierney, Ezra Kissel, Martin Swany, Eric Pouyoul //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

[9] Brian Cho and Indranil Gupta // Budget-Constrained Bulk Data Transfer via Internet and Shipping Networks // ICAC'11, June 14–18, 2011, Karlsruhe, Germany. Copyright 2011 ACM 978-1-4503-0607-2/11/06 // 10 pages.

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

[11] 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 // 8 pages.

[12] PhEDEx – CMS Data Transfers https://cmsweb.cern.ch/phedex

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

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

[15] WLCG Transfers Dashboard http://dashb-wlcg-transfers.cern.ch/

[16] Working Group on Big Data - http://bigdatawg.nist.org/

[17] 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)

[18] Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti // 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

[19] Harry Mangalam // How to transfer large amount of data via network - http://moo.nac.uci.edu/~hjm/HOWTO_move_data.html

[20] Brian Tierney and Joe Metzger // High Performance Bulk Data Transfer – http://fasterdata.es.net/assets/fasterdata/JT-201010.pdf (presentation on Joint Tech, Columbus OH, July 2010)

[21] BBFTP - Utility for bulk data transfer - http://doc.in2p3.fr/bbftp/.

[22] BBCP – utility to trasnfer the data over network - http://www.slac.stanford.edu/~abh/bbcp/.

[23] Fast Data Transfer - http://monalisa.cern.ch/FDT/.

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

[25] UDT: Breaking the Data Transfer Bottleneck - http://udt.sourceforge.net/.

[26] Esnet Fasterdata knowledge database - http://fasterdata.es.net/

[27] Berkeley Storage Manager (BeStMan) - https://sdm.lbl.gov/bestman/

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

[29] MutiPath TCP – Linux Kernel Implementation – http://mptcp.info.ucl.ac.be/, http://multipath-tcp.org/

[30] Data Transfer Tools http://fasterdata.es.net/data-transfer-tools/

[31] Self-Healing Independent File Transfer (Shift) http://people.nas.nasa.gov/~kolano/projects/shift.html, http://sourceforge.net/projects/shiftc/

[32] Autonomic computing http://www.ibm.com/developerworks/autonomic/r3/probdet1.html

[33] Mehmet Balman, Surendra Byna // Open Problems in Network-aware Data Management in Exa-scale Computing and Terabit Networking Era // NDM'11, November 14, 2011, Seattle, Washington, USA. Copyright 2011 ACM 978-1-4503-1132-8/11/11 // 5 pages.

[34] Ezra Kissel, Martin Swany, Brian Tierney, Eric Pouyoul // Efficient Wide Area Data Transfer Protocols for 100 Gbps Networks and Beyond // NDM'13 November 17, 2013, Denver CO, USA. Copyright 2013 ACM 978-1-4503-2522-6/13/11 // 10 pages.

[35] Eun-Sung Jung, Rajkumar Kettimuthu, and Venkatram Vishwanath //Toward optimizing disk-to-disk transfer on 100G networks // http://www.mcs.anl.gov/publication/toward-optimizing-disk-disk-transfer-100g-networks

[36] Zhenzhen Yan, Malathi Veeraraghavan, Chris Tracy, Chin Guok // On How to Provision Quality of Service (QoS) for Large Dataset Transfers // http://es.net/assets/pubs_presos/HNTES-CTRQ-no-color.pdf [last read: 7.11.2013]

[37] Lavanya Ramakrishnan, Chin Guok, Keith Jackson, Ezra Kissel, D. Martin Swany, Deborah Agarwal // On-demand Overlay Networks for Large Scientific Data Transfers // 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing // dl.acm.org/ft_gateway.cfm?id=1845152&type=pdf [last read: 7.11.2013] // 9 pages.

[38] Chaitanya Baru, Milind Bhandarkar, Raghunath Nambiar, Meikel Poess, and Tilmann Rabl // Benchmarking big data systems and the bigdata top100 list // Big data March 2013 DOI: 10.1089/big.2013.1509 // 5 pages.

[39] Ricky Egeland Tony Wildish, and Simon Metson // Data transfer infrastructure for CMS data taking // XII Advanced Computing and Analysis Techniques in Physics Research November 3-7 2008 Erice, Italy // 11 pages.

[40] Eli Dart, Lauren Rotman, Brian Tierney, Mary Hester, Jason Zurawski // The Science DMZ: A Network Design Pattern for Data-Intensive Science // SC13 November 17-21, 2013, Denver, CO, USA Copyright 2013 ACM 978-1-4503-2378-9/13/11 // 10 pages.

[41] 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 [last reading date: 1.11.2013] // 6 pages.

[42] Nikolaos Laoutaris, Michael Sirivianos, Xiaoyuan Yang, and Pablo Rodriguez // Inter-Datacenter Bulk Transfers with NetStitcher // SIGCOMM'11, August 15–19, 2011, Toronto, Ontario, Canada. Copyright 2011 ACM 978-1-4503-0797-0/11/08 // 12 pages.

[43] Michael Ashenfelder // 21st Century Shipping. Network Data Transfer to the Library of Congress // D-Lib Magazine July/August 2009 Volume 15 Number 7/8 ISSN 1082-9873 // http://www.dlib.org/dlib/july09/ashenfelder/07ashenfelder.html [last read 1.12.2013]

[44] 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]

[45] Wantao Liu, Brian Tieman, Rajkumar Kettimuthu, Ian Foster // A Data Transfer Framework for Large-Scale Science Experiments // DIDC'10, June 22, 2010, Chicago, IL, USA. http://www.cse.buffalo.edu/faculty/tkosar/didc10/papers/didc02.pdf [last read 1.12.2013]

[46] Stephen W. Hodson, Stephen W. Poole, Thomas M. Ruwart, Bradley W. Settlemyer // 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]

[47] Bulk Data Movement // https://sdm.lbl.gov/bdm/ [accessed date: 1.12.2013]

[48] Alex Sim, Mehmet Balman, Dean Williams, Arie Shoshani, Vijaya Natarajan // Adaptive transfer adjustment in efficient bulk data transfer management for climate dataset // http://crd.lbl.gov/assets/pubs_presos/PDCS2010-balman.pdf [last read 1.12.2013]

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

[50] William E. Johnston, Eli Dart, Michael Ernst, Brian Tierney // 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)

[51] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers // http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation // Report| McKinsey Global Institute // May 2011

[52] OpenSSH http://openssh.org/

[53] Patched OpenSSH http://sourceforge.net/projects/hpnssh/

[54] Laboratory of the Network Technology – http://sdn.ifmo.ru/