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

Ки&М

ПЛЕНАРНЫЕ 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

Received December 1, 2014

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.

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.