Transfer of Large Volume Data over Internet with Parallel Data Links and SDN

S.E. Khoruzhnikov[1], V.A. Grudinin[1], O.L. Sadov[1], A.Y. Shevel[1,2(✉)], and A.B. Kairkanov[1]

[1] ITMO University, St.Petersburg, Russian Federation
shevel_a_y@niuitmo.ru
[2] National Research Centre, "Kurchatov Institute" B.P. Konstantinov, Petersburg Nuclear Physics Institute, Moscow, Russian Federation

Abstract. The transfer of large volume data over computer network is important and unavoidable operation in the past, now and in any feasible future. There are a number of methods/tools to transfer the data over computer global network (Internet). In this paper the transfer of data over Internet is discussed. Several free of charge utilities to transfer the 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: Big Data · Linux · Transfer · SDN · Openflow · Network

1 Introduction

Instead of large volume data the people quite often use the term "Big Data" [1] which is known for many years. Keeping in mind Big Data description - "triple V": Velocity, Volume, Variety we can pay attention that all those features are relative to current state of the technology. For example in 1970-s the volume of 1 TB was considered as huge volume. There is a range of aspects of the problem: to store, to analyze, to transfer, to visualize. In this paper we discuss one of important aspects of the Big Data – the transfer over global computer network.

1.1 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 [2-10].

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.

Y. Tan et al. (Eds.): ICSI-CCI 2015, Part III, LNCS 9142, pp. 463–471, 2015.
DOI: 10.1007/978-3-319-20469-7_49

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" [2-7, 33]. The interest to data transfer of increasing volumes is growing [11, 12].

1.2 Big 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 data 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 data 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 periods: secs, hours, days. Also a number of Linux kernel network parameters might affect the data transfer speed. Among most important of them are TCP Window size, MTU, congestion control algorithm, etc. Finally in each data transfer of large volume we need to be able to tune the number of data transfer streams, the size of TCP Window, etc. Now it is time to observe popular freely available data transfer tools/utilities.

2 Freely Available Utilities/Tools for Data Transfer over the Network

2.1 Low Level Data Transfer Protocols

We could mention several tools:

- one of low level protocols to transfer the data over the network is UDT [13]. UDT is library which implements data transfer protocol which permit to use *udp*. 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) has been studied in [33] and it was found that in many cases RoCE shows better results than UDP and conventional TCP.

MPTCP [14] is interesting protocol which permits to use several data links in parallel for one data transfer. The protocol is implemented as Linux kernel driver.

2.2 Experimenting with MPTCP

Among other things we tested available version driver MP TCP. In our scheme we used our own server with Linux kernel 3.14.15-303.mptcp.el7.centos.x86_64 and two data links with two different routing tables as it explained in MP TCP docs. Opposite server was mutipath-tcp.org. To test the speed of data transfer we used the command: `wget -c ftp://multipath-tcp.org/500MB`.

First of all we used mode when MPTCP is switched OFF (`systcl net.mptcp.mptcp_enabled=0`). Here we got the speed 1.84 MB/sec. Next test when MPTCP is switched ON and the speed was 3.33 MB/sec – reasonably higher. Another test measurement was dealing with MPTCP is ON however one of the data link was set down (i.e. one data link became broken). In this test we got the speed 911 KB/sec – reasonably less then with two data links and even less than MPTCP is OFF (with one data link).

Above results were signs for us that for long time data transfer we need to watch regular the real status of data links when we plan to speed up the data transfer.

2.3 Popular Programs for Data Transfer

openssh family [15] – 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 [16] which can use increased size of buffers and SSH with Globus GSI authentication. No parallel data transfer streams.

bbcp [17] — 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;
- multi-stream transfer;
- I/O buffer size;
- resuming failed data transfer;
- authentication with ssh;
- other options dealing with many practical details.

bbftp [18] – 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;
- ability to tune I/O buffer size;

— restart failed data transfer;
— other useful practical features.

fdt [20] – Java utility for multi-stream data transfer and support I/O buffer size tuning.

gridFTP [21] is advanced redesign of well known utility *ftp* for globus security infrastructure (GSI) environment. The utility has many features:

— 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);
— multi-stream transfer;
— ability to tune I/O buffer size;
— restart failed data transfer;
— other useful practical features.

Mentioned utilities are quite effective for data transfer from point of view of data link capacity usage. The more or less typical data transfer over real shared (the traceroute consists of 9 hops) 1 Gbit long distance data link is shown on Fig. 1. The program bbcp was used for real measurements. In this measurement the directory with test data files located in main memory was transferred. Total volume of the directory was 25 GB. The content of files was specially randomly generated binaries. The sizes of files in test directory was randomly distributed with average 100MB and standard deviation was 50 MB. Horizontal axis shows TCP Window sizes in bytes. It can be seen on the picture that with small number (1-2) of TCP streams the maximum speed is achieved with the size of TCP Window around 2097KB and with numbers of streams 3 and 4 the maximum is around 524KB. In contrast with relatively large number (18 and more) of TCP streams the maximum is achieved even with TCP Window size around 131KB. Existing fluctuations in data transfer speed curves are explained by the fact that shared data link is used for those measurements. The transfer of large volume data assumes significant transmission time (may be many hours, days or more). The transmission speed might change very seriously over shared data links and over so long time.

2.4 Middle Level File Transfer Service

The FTS3 [22] 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. In particular FTS3 uses utility gridFTP to perform data transfer. 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 [23] 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).

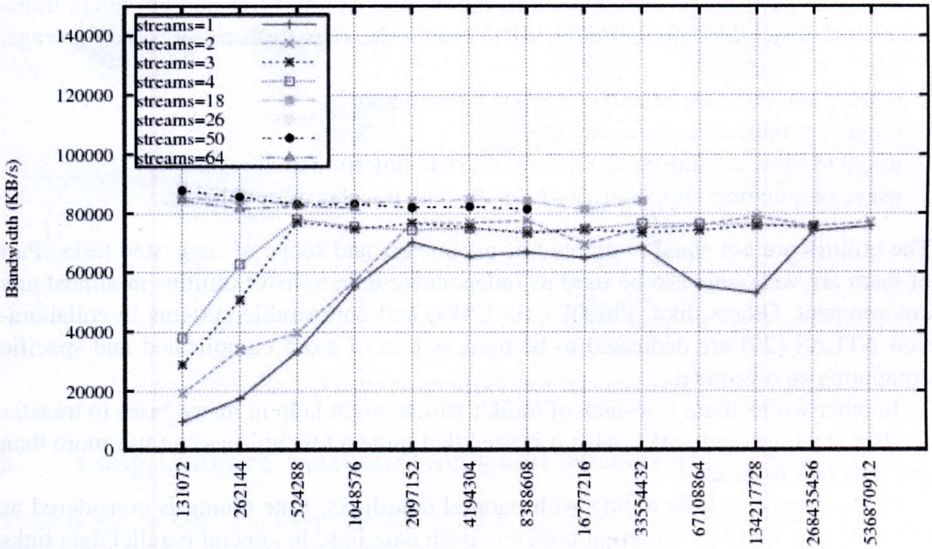

Fig. 1. Data transfer speed depends on the number of streams and size of TCP Window

2.5 High Level Data Management Service: PhEDEx

PhEDEx - Physics Experiment Data Export is used (and developed) in collaboration around Compact Muon Solenoid (CMS) experiment [24-27] 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 (to distribute) 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 [25]. It is possible that in nearest future the volume per day will be increased. Because in between several sites there are more than one data 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.

3 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, number of streams, 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;
usage more than one of authentication techniques;
usage a number of transfer streams;
usage in some conditions more than one data 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 [29] are dedicated to be used as part of more complicated and specific computing environment.

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

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 data 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 [30-31]. The special tool PerfSonar [32] can be used to watch the data links status.

4 Proposal and Testing

Let us image two parallel data links between two points in Internet. Both data links might be different in its nature and independent of each other. The data transfer might be performed with use both data links. Usually any data transfer program creates several data streams. On both sides (independently on source and destination) a control program might know the port pairs on source and destination hosts. With such the knowledge the control program might build up the appropriate routing table for network switch with ability of SDN Openflow. To be synchronous both control program have to use exactly same algorithm to arrange data streams among the data links. If we have 20 data streams in our example for each data links 10 streams would be arranged. If the status of data links is changed, for example, one data link experienced interruption, control programs on both sides have to get information about the problem from perfsonar: each control program from its own perfsonar. The schematic picture can be seen at Fig.2.

In our local testing (on one laboratory table) everything is working as expected. Now we are preparing the larger series of long distance data transfer. To perform the testing the special long distant testbed has been developed and deployed.

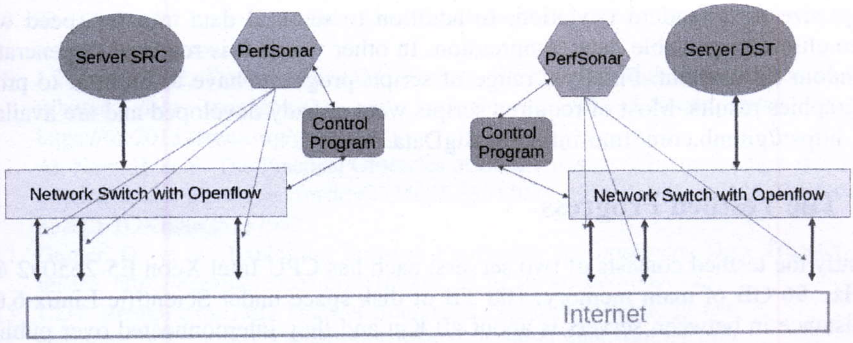

Fig. 2. Proposed scheme of the data transfer

5 Long Distance Testbed: Goals and Measurement Procedures

There is special aspect in the procedure of the comparison of the utilities to transfer the Big Data over real computer network. Not trivial scheme for data transfer 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 [28]. The need for testbed is becoming obvious for us thanks to measurement results performed by authors [11]. The authors did the comparative measurements for one data transfer stream and many data 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 in [11] there is no information about the Linux kernel parameters and other details of measurement.

In developed testbed it is planned to get precise answers on those questions. Also in the testbed we are taking into account the ideas described in [12].

The testbed is intended to be platform to compare a number of data transfer methods in particular with SDN Openflow approach.

As the first step it was planned to perform comparative measurements under variety of the measurement conditions. To do that we need detailed measurement tracking: writing all available details of the measurements (hardware types, buffer sizes, kernel parameters, generated messages, program parameters, etc). Everything has to be written in special log directory with date stamps. Presumably this log directory needs to be kept available long time: may be years. Obviously the writing of all conditions must be done with special script or program to perform saving all test conditions automatically.

Another question is dealing with test data which intended to be transferred over data links. Here we assume the transfer of the test directory consisting of files. It is known that sizes of files might affect the data transfer speed. That fact requires the special procedure to generate test directory with files of random sizes with defined average size and standard deviation. In addition to see real data transfer speed we need to eliminate possible data compression. In other words it is required to generate the random file content. Finally a range of scripts/programs have to be used to produce graphics results. Most of required scripts were already developed and are available in https://github.com/itmo-infocom/BigData.

6 The Testbed Progress

Currently the testbed consists of two servers: each has CPU Intel Xeon E5-2650v2 @ 2.6GHz, 96 GB of main memory, 100 TB of disk space under Scientific Linux 6.6. The distance in between servers is about 40 Km and they interconnected over public Internet (1 Gbit is maximum bandwidth). Because it is planned to test everything in virtual environment for each mentioned data transfer systems two virtual machines are used. One VM as transmitter and another VM as receiver. In other words we have around ten Vms for different tests. The cloud infrastructure Openstack (version Icehouse) has been deployed to organize VMs. PerfSonar has been deployed as well on both sides.

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 standard deviation 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.

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 "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 links status is also to be written as well. All those features have been implemented in the scripts dedicated to do measurements. Total volume of the log data for one measurement might exceed 5 MB.

Acknowledgements. The research has been carried out with the financial support of the Ministry of Education and Science of the Russian Federation under grant agreement #14.575.21.0058.

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. Borovick, L., Villars, R.L.: 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

9. The Center for Large-scale Data Systems Research at the San Diego Supercomputer Center. http://clds.sdsc.edu/
10. Johnston, W. E., Dart, E., Ernst, M., Tierney, B.: Enabling high throughput in widely distributed data management and analysis systems: Lessons from the LHC. https://tnc2013.terena.org/getfile/402
11. Ah Nam, H. et al: The Practical Obstacles of Data Transfer: Why researchers still love scp. http://dl.acm.org/citation.cfm?id=2534695.2534703&coll=DL&dl=ACM&CFID=563485433&CFTOKEN=25267057
12. Gunter, D., et al.: Exploiting Network Parallelism for Improving Data Transfer Performance. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6496123
13. UDT: Breaking the Data Transfer Bottleneck. http://udt.sourceforge.net/
14. MutiPath TCP – Linux Kernel Implementation. http://multipath-tcp.org/
15. OpenSSH. http://openssh.org/
16. Patched OpenSSH. http://sourceforge.net/projects/hpnssh/
17. BBCP – utility to trasnfer the data over network. http://www.slac.stanford.edu/~abh/bbcp/
18. BBFTP – utility for bulk data transfer. http://doc.in2p3.fr/bbftp/
19. Hodson, S.W., Poole, S.W., Ruwart, T. M., Settlemyer, B. W.: Moving Large Data Sets Over High-Performance Long Distance Networks. http://info.ornl.gov/sites/publications/files/Pub28508.pdf
20. Fast Data Transfer. http://monalisa.cern.ch/FDT/
21. Grid/Globus data transfer tool. Client part is known as globus-url-copy. http://toolkit.globus.org/toolkit/data/gridftp/
22. File Transfer Service. http://www.eu-emi.eu/products/-/asset_publisher/1gkD/content/fts3
23. Data Transfer Tools. http://fasterdata.es.net/data-transfer-tools/
24. The CMS Collaboration: The CMS experiment at the CERN LHC. http://iopscience.iop.org/1748-0221/3/08/S08004/
25. 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. In: International Conference on Computing in High Energy and Nuclear Physics 2012. IOP Publishing Journal of Physics: Conference Series, vol. 396 (2012)
26. PhEDEx, CMS Data Transfers. https://cmsweb.cern.ch/phedex
27. PHEDEX data transfer system. https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhedexAdminDocsInstallation
28. Laboratory of the Network Technology. http://sdn.ifmo.ru/
29. The Rucio project is the new version of ATLAS Distributed Data Management (DDM) system services. http://rucio.cern.ch/
30. 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
31. Nunes, B. A., Mendonca, M., Nguyen, X., Obraczka, K., Turletti, T.: 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
32. Zurawski, J., Balasubramanian, S., Brown, A., Kissel, E., Lake, A., Swany, M., Tierney, B., Zekauskas, M.: perfSONAR: On-board Diagnostics for Big Data. https://www.es.net/assets/pubs_presos/20130910-IEEE-BigData-perfSONAR2.pdf
33. Tierney, B., Kissel, E., Swany, M., Pouyoul, E.: Efficient Data Transfer Protocol for Big-Data. http://www.es.net/assets/pubs_presos/eScience-networks.pdf