Environment for Data Transfer Measurement

Sergey Khoruzhnikov[1], Vladimir Grudinin[1], Oleg Sadov[1], Andrey Shevel[1,2], Stefanos Georgiou[1,3], and Arsen Kairkanov[1](✉)

[1] ITMO University, St. Petersburg, Russian Federation
abkairkanov@corp.ifmo.ru
[2] National Research Centre "Kurchatov Institute", B.P. Konstantinov, Petersburg Nuclear Physics Institute, Gatchina, Russian Federation
[3] Athens University of Economics and Business, Athens, Greece

Abstract. Measurement of the data transfer considered as often task when regular transfer over long distant network of large volume data is required. The data transfer over network depends on many conditions and parameters. Quite often measurements have to be repeated several times. The paper describes procedure how to implement appropriate measurements to store automatically all the parameters and messages during the data transfer. The saved history tracking permits to do detailed comparison of measurement results.

Keywords: Big data · Linux · Data transfer · Measurement · Internet

1 The Introduction

The large volume data is tightly connected to the big data [1]. The term "big data" have many aspects: store, analysis, transferring over data link, visualization, etc. In this paper authors are concentrating on procedure how to track data transfer. In some context the paper can be considered as part of research in SDN approach to the data transfer [2].

While performing a large data transfer many unexpected events may occur: data transfer rate could change, interruptions could appear, and further more. Usual desire in big data transfer over long distant network (Internet) is to increase the data transfer speed, increase the reliability, and so on. In order to achieve such a result, several quite reliable data transfer utilities [3] in Linux exist. However each specific case of data transfer might require specific parameter values, specific data transfer programs to achieve required results.

There are methods to decrease the time to transfer the data over network by compression the data just before transfer and uncompression immediately after. Such the approach requires additional assumptions: you have enough computing power to do compression and uncompression "on the fly" and you have precise knowledge that data might be compressed by significant factor. In reality many types of files are already heavily compressed and any attempts to compress them more will be resulted only in spending CPU time but no data volume decreasing. Anyway careful concrete analysis of balance between CPU time consumption and possible decrease data transfer time is very useful. Anyway to compare different data transfer possibilities it is very important to do many measurements and save all the detailed information about each measurement.

2 Related Work

Different data transfer tools have different features and impact for the transfer e.g., total volume size, data type, files sizes, and etc. Popular Linux tools are discussed below in Sect. 3. Authors [4, 5] provided interesting information in big scale data transfer in general: WAN data links architecture and capacity, data transfer monitoring, protocols, reliability, analysis, discussions, etc. At the same time a little information how to carefully compare their results with data transfer performed in another time. In most cases authors just show results to illustrate their ideas but not intended to compare with other concrete data transfer results, for example, [6].

3 Popular Programs for Data Transfer

The number of data transfer programs is huge, may be thousands. Several most popular free of charge tools in Linux is planned to be discussed here.

openssh family [6–8] — 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 openSSH flavors: patched SSH version [8] which can use increased size of buffers and SSH with Globus GSI authentication. No parallel data transfer streams.

bbcp [9] — utility for bulk data transfer. It is assumed that bbcp has been installed 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 [10] — utility for bulk data transfer. It implements its own transfer protocol, which is optimized for large files (significantly larger than 2 GB) 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 [11] — Java based utility used for efficient data transfer. Since is written in Java, theoretically it can be executed in any platform. It can run as a SCP or client/server applications. Also, it’s an asynchronous, flexible multithreaded system and is using the capabilities of Java NIO library. Features such as: support I/O buffer size tuning, restore files from buffers asynchronously, resume a file transfer session without any loss and when is necessary, and streams dataset continuously by using a managed pool of buffers through one or more TCP sockets.

gridFTP [12] is advanced transfer utility 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.

More sophisticated tool FTS3 [13] is advanced tool for data transfer of large volume of the data over the network. FTS3 uses utility gridFTP to perform data transfer. In this tool in addition to useful data transfer features mentioned above there are a number of others: good data transfer history tracking (log), ability to use http, restful, CLI interfaces to control the process of the data transfer, get the information about data transfer status and more.

Each mentioned tool has specific features. It is not possible to tell which tool is most effective in concrete task of big data transfer before testing.

4 Measurement Procedures

4.1 Basic Requirements for Measurement

At first, the important measurement requirement is option to reproduce earlier obtained test results (so called history tracking data). In other words there is need to guarantee that concrete measurement can be repeated later on with exactly same parameters. Also for series of test runs is possible to compare different data transfer programs or equipment.

At second, one would need to remember that the measurement VM are running in Openstack cloud infrastructure. Among other things it is useful to remember that such parameters like TCP window must be set in VM and in host machine as well. Otherwise the changing parameters just in VM does not affect data transfer at all. A lot of advanced recommendations available elsewhere, for example, in [3] are describing cases where any Linux kernel parameters can be changed at any time. Such the reasons might prevent the achievement the maximum data transfer speed when you have no ability to change parameters in directory/proc (the root credentials are required) on host machines. Such the consideration might be applied to any cloud environment. Indeed no reason to expect that data transfer from one cloud environment to other cloud environment would be performed in most effective manner from the transfer speed point of view.

Each measurement run needs the measurement tracking: to write all the conditions during measurement. For example, start time, end time, all messages generated by the data transfer utility, system data. System data are quite important: used hardware (CPU, motherboard, network card, etc.), the values in directory/proc - about ½ thousand parameters might affect the data transfer over network, the kernel version, the list and versions of all Linux packages, etc. There is Linux sosreport [14] to do above job. Sosreport is a command in Linux flavor RHEL/CentOS which collects system configuration and diagnostic information of your Linux box like running kernel version, loaded modules, services and system configuration files. This command will normally complete within a few minutes. Once completed, sosreport will generate a compressed a file under/tmp folder. Different versions use different compression schemes (gz, bz2, or xz). The size of generated file might be around several MB.

Also important to save logs of CPU usage, swap usage, and other parameters during the data transfer for history tracking.

Another issue the state of the data link which is also needs to be watched and history saved during the transfer. Among useful tools to watch the state of the data link is probably most interesting pefrsonar. Perfsonar (cite from [15]) “provides a uniform interface that allows for the scheduling of measurements, storage of data in uniform formats, and scalable methods to retrieve data and generate visualizations. This extensible system can be modified to support new metrics, and there are endless possibilities for data presentation”. There is whole network of deployed perfsonar servers (many perfsonar installations) in many distributed hosts which are interested for some communities. When the community members transfer the data each other the perfsonar network is used to find out data transfer bottleneck. Most of deployed perfsonar are dedicated to monitor high speed networks 10 and 100 Gbit. On less powerful data lines the desire to use perfsonar in quite intensive way might be stressful for data link capacity. The perfsonar is just tool to observe data links, but it can’t replace own understanding what is going on in the data link.

All tracking data have to be written in special log directory. Presumably, this log directory needs to be saved for long time: often - years. Obviously the writing of all conditions must be done with special script or program to store all test conditions automatically.

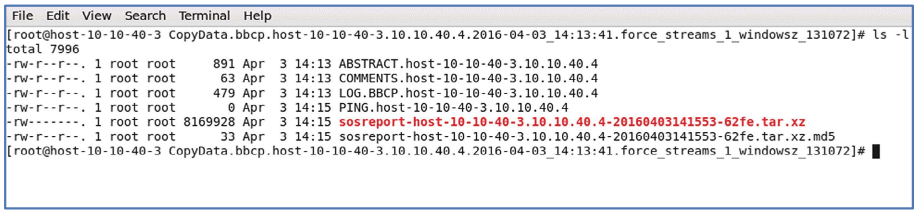
4.2 Testbed and Tracking Data Architecture

It is used testbed developed for research in Software Defined Network in the Network Laboratory [16]. Usually for each measurement run we prepared two Virtual Machines (VM) in framework of Openstack.org. One VM was data transmitter and another one VM was used as receiver.

Each start of any measurement script is creating special log directory which has the name in form copydata<utility_name><host-where-utility-started><host-where-to-transfer><date-time>. The directory contains following files:

- result of command sosreport;
- output of command ping <to remote host>;
- result of traceroute <to remote host>;
- all messages of tested utility generated during measurement;
- any comments to the current measurement;
- special abstract for the test measurement consisting of following lines:
  - total volume of data transfer;
  - number of files;
  - the directory name where files to be transferred are;
  - average of file size;
  - standard deviation of file size;
  - transmitter VM hostname;
  - receiver VM hostname.

Such the detailed logs permit to find out what was going on in the measurement even any time in future. Example for the log directory is shown in Fig. 1.



```
[root@host-10-10-40-3 CopyData.bbcp.host-10-10-40-3.10.10.40.4.2016-04-03_14:13:41.force_streams_1_windowsz_131072]# ls -l
total 7996
-rw-r--r--. 1 root root     891 Apr  3 14:13 ABSTRACT.host-10-10-40-3.10.10.40.4
-rw-r--r--. 1 root root      63 Apr  3 14:13 COMMENTS.host-10-10-40-3.10.10.40.4
-rw-r--r--. 1 root root     479 Apr  3 14:13 LOG.BBCP.host-10-10-40-3.10.10.40.4
-rw-r--r--. 1 root root       0 Apr  3 14:15 PING.host-10-10-40-3.10.10.40.4
-rw-------. 1 root root 8169928 Apr  3 14:15 sosreport-host-10-10-40-3.10.10.40.4-20160403141553-62fe.tar.xz
-rw-r--r--. 1 root root      33 Apr  3 14:15 sosreport-host-10-10-40-3.10.10.40.4-20160403141553-62fe.tar.xz.md5
[root@host-10-10-40-3 CopyData.bbcp.host-10-10-40-3.10.10.40.4.2016-04-03_14:13:41.force_streams_1_windowsz_131072]# 
```

Fig. 1. Example of the log directory.

Also a range of scripts were used to produce graphics results of the measurements. Most of required scripts have been developed and available in https://github.com/itmo-infocom/BigData.

5 Conclusion

Authors developed the procedure to do measurements of data transfer speed and the set of scripts to store all tracking information. Specific feature of the procedure is ability to store all values in defined format. Detailed tracking information gives ability to compare the results of measurements which have been performed in different days and environments.

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. ISO/IEC JTC 1 - Big Data. Preliminary report (2014). http://www.iso.org/iso/home/store/publication_item.htm?pid=PUB100361
2. Khoruzhnikov, S.E., Grudinin, V.A., Sadov, O.L., Shevel, A.Y., Kairkanov, A.B.: Transfer of large volume data over internet with parallel data links and SDN. In: Tan, Y., Shi, Y., Buarque, F., Gelbukh, A., Das, S., Engelbrecht, A. (eds.) ICSI-CCI 2015. LNCS, vol. 9142, pp. 463–471. Springer, Heidelberg (2015)
3. Data Transfer Tools. http://fasterdata.es.net/data-transfer-tools
4. NIST Big Data public working group. http://bigdatawg.nist.gov
5. Johnston, W.E., Dart, E., Ernst, M., Tierney, B.: Enabling high throughput in widely distributed data management and analysis systems: lessons from the LHC (2013). https://tnc2013.terena.org/getfile/402
6. OpenSSH. http://openssh.org
7. Ah Nam, H. et al: The practical obstacles of data transfer: why researchers still love scp. In: NDM 2013 Proceedings of the Third International Workshop on Network-Aware Data Management, Article No. 7 (2013). doi:10.1145/2534695.2534703
8. Patched OpenSSH. http://sourceforge.net/projects/hpnssh
9. BBCP – utility to transfer the data over network. http://www.slac.stanford.edu/~abh/bbcp

10. BBFTP – utility for bulk data transfer. http://doc.in2p3.fr/bbftp
11. Fast Data Transfer. http://monalisa.cern.ch/FDT
12. Grid/Globus data transfer tool. Client part is known as globus-url-copy. http://toolkit.globus.org/toolkit/data/gridftp
13. File Transfer Service. http://www.eu-emi.eu/products/-/asset_publisher/1gkD/content/fts3
14. Generate debug information for the system. http://linux.die.net/man/1/sosreport
15. Perfsonar. http://www.perfsonar.net
16. Laboratory of the Network Technology. http://sdn.ifmo.ru