High Throughput WAN Data Transfer with Hadoop-based Storage

2011 J. Phys.: Conf. Ser. 331 052016

(http://iopscience.iop.org/1742-6596/331/5/052016)

International Conference on Computing in High Energy and Nuclear Physics (CHEP 2010) IOP Publishing
Journal of Physics: Conference Series **331** (2011) 052016 doi:10.1088/1742-6596/331/5/052016

High Throughput WAN Data Transfer with Hadoop-based Storage

A Amin[2], B Bockelman[4], J Letts[1], T Levshina[3], T Martin[1], H Pi[1], I Sfiligoi[1], M Thomas[2], F Wüerthwein[1]

[1]University of California, San Diego, 9500 Gilman Dr, San Diego, CA 92093, USA
[2]California Institute of Technology, 12000 East California Blvd, Pasadena, CA 91125, USA
[3]Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510, USA
[4]University of Nebraska-Lincoln, 118 Schorr Center, Lincoln, NE 68588, USA

E-mail: hpi@physics.ucsd.edu

Abstract Hadoop distributed file system (HDFS) is becoming more popular in recent years as a key building block of integrated grid storage solution in the field of scientific computing. Wide Area Network (WAN) data transfer is one of the important data operations for large high energy physics experiments to manage, share and process datasets of PetaBytes scale in a highly distributed grid computing environment. In this paper, we present the experience of high throughput WAN data transfer with HDFS-based Storage Element. Two protocols, GridFTP and fast data transfer (FDT), are used to characterize the network performance of WAN data transfer.

1. Introduction

Hadoop[1] is a newly emerged Java-based framework including a distributed file system and tightly integrated applications that can quickly process a large volume of data stored in the system. This framework is extensively used in search engine and advertising businesses by Yahoo and other companies with computational scale up to thousands of servers and Petabytes of raw data. The Hadoop Distributed File System (HDFS) is a key component of Hadoop that is designed to store data in commodity hardware while still achieving high throughput for accessing data, which mainly benefits from the simple data coherency model used by HDFS, write-once-read-many. The reliable data replication and detection of failure implemented in Hadoop enable fast and automatic system recovery. The emphasis of high throughput instead of low latency makes HDFS appealing for batch processing. The portability of Hadoop makes integration easier with heterogeneous hardware and software platforms.

In 2009, a few computing centers from the Compact Muon Solenoid (CMS) at the Large Hadron Collider (LHC) experiment accepted HDFS as the major technique to establish a fully functional Storage Element (SE) [2]. This effort was endorsed by the Open Science Grid (OSG)[3]. Later a number of computing centers ranging from middle scale at universities to large scale at national laboratories deployed the resulting SE.

Wide Area Network (WAN) data transfer is one of the important data operations for large high energy physics experiments to manage, share and process datasets of PetaBytes scale in a highly distributed grid computing environment. At CMS, all the official data are shared among more than 50 sites according to a tiered architecture. How to effectively handle the WAN data transfer from multiple sources to multiple sinks at a massive scale from network point of view is a challenge. CMS developed a dedicated data management layer,

called PhEDEx[4] for Physics Experiment Data Export, which implements an agent at each site that initiates and controls a series of processes for the WAN data transfer. The WAN throughput of CMS Tier-1, Tier-2 and Tier-3 sites run by PhEDEx from November 2010 to January 2011 is shown in Fig. 1. Currently more than 2000 WAN links among CMS computing sites are serving data for thousands of users across the collaboration.

Efficient WAN data transfer speeds up the data flow and increases the utilization of idle resources (CPU and storage) awaiting for user jobs which in general have very specific demand for certain datasets to process. To quickly and reliably deliver the data to the user, it requires a high throughput WAN data transfer to be in place and seamlessly integrated with every major computing site and various grid middleware platforms to meet the needs of the whole collaboration.

In this paper, we focus on some important WAN data transfer issues with the HDFS-based storage, which is not only the continuity of efforts to enhance the HDFS-based SE technology, but also an attempt to study the WAN networking tools to maximize the utilization of network bandwidth. Eventually we believe this work will be integrated into the network-based solution for a more dynamic data distribution model for high energy experiment and other scientific projects.

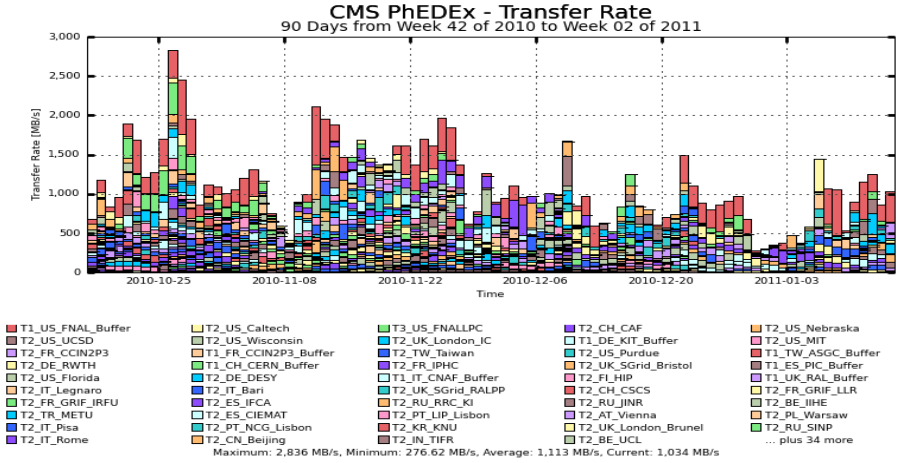

Figure 1. CMS WAN data transfer managed by PhEDEx

2. Architecture of HDFS-based Storage Element and WAN transfer interface

Following is a short review of the architecture of HDFS-based Storage Element (SE). HDFS namenode (NN) provides the namespace of the file system of the SE. HDFS datanode (DN) provides the distributed storage space to the SE. In the typical deployment strategy, the compute node can also be configured as DN. FUSE on HDFS provides the interface of HDFS for local data access of the SE from the applications. GridFTP on either the dedicated hosts or shared with data nodes or compute nodes provides WAN data transfer from or to the SE. BeStMan[5] server provides the SRM v2.2 interface of the SE and performs the load-balancing of the GridFTP servers.

The interface between GridFTP and storage system (e.g. a distributed file system) can be a serious bottleneck for high throughput data transfer. For the architecture using a distributed file system across the cluster, the WN I/O involves internal data replication in HDFS, external WAN-read or write of data, and local data access of user jobs. In the integration of HDFS-GridFTP, the GridFTP needs to use native HDFS client to access the file system and buffer the incoming data in memory because the HDFS doesn't support asynchronous writing. From current system integration and data access strategy, most of system dependent implementation takes place at GridFTP level. The overview of the typical SE architecture using BeStMan is shown in Fig. 2.

A set of services implemented by SE, from data transfer to data access, involve various components of the storage system and grid middleware as shown in Fig. 3. If there is no explicit bottleneck in the application level (file system, software, user jobs), the throughput for WAN transfer and local data access will be mainly limited by the network link bandwidth or the local disk I/O which is part of the nature of the hardware infrastructure. High throughput WAN transfer is one of important data operations that may put some challenge on HDFS:

- Various data access patterns and use cases may cause scalability problem for certain services. For example, a large volume of data access for small files in the SE introduces significant overhead in the

BeStMan and file system.

- To sustain WAN transfer rate reaching 10 Giga bit (Gb) per second (Gbps), a scalable distributed file system is needed to efficiently read or write data simultaneously across the whole cluster.
- The local file access shouldn't be affected by high throughput WAN transfer, which requires the HDFS to have enough room to ensure the serving of data won't be blocked by a weakest point, such as the slowest node or hot node that has the data or storage space needed by many applications. Data redundancy and replication play important roles in the stability and scalability of HDFS to sustain high throughput data transfer.
- The normal HDFS operation shouldn't be affected by high throughput data transfer which may last hours or days. For a functioning HDFS to reach its maximal efficiency, some internal routine data operation tasks must be able to run, for example the replication of data blocks, balancing and distributing files across all the data nodes.

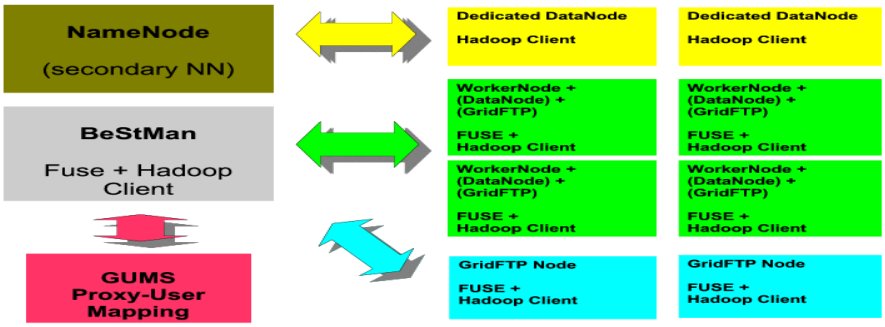

Figure 2. Architecture of HDFS-based Storage Element

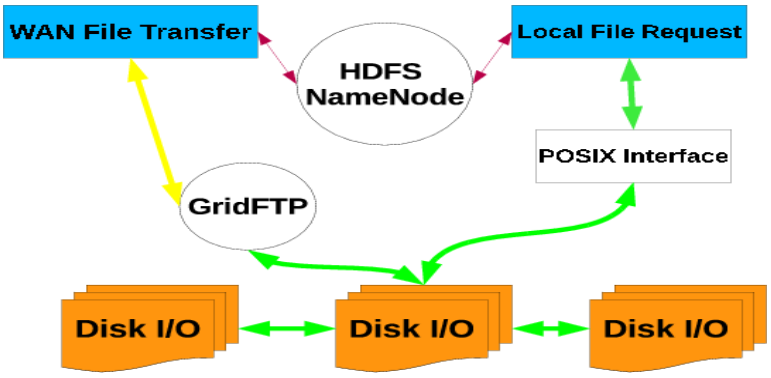

Figure 3. File transfer and data access between various components of the HDFS based SE

3. BeStMan Scalability and Testing Tool

The first version of BeStMan was based on the Globus container. Currently this flavor of BeStMan is widely deployed in production at many OSG Grid sites. BeStMan2 based on Jetty container in the test release now. It is expected to eventually replace the Globus container because of its lack of maintenance in recent years and observed limitation in its scalability and reliability.

GlideinWMS (Glide-in based Workload Management System) [6] was used to measure the BeStMan scalability. GlideinWMS is a Grid workload management system to automate condor glide-in configuration, submission and management. GlideinWMS is used widely in several large scientific Grids for handling users jobs for physics analysis and Monte Carlo production. In our work, we used two methods based on GlideinWMS to benchmark the BeStMan:

- Use GlideinWMS to send jobs to some typical Grid sites and use the normal job slots at the site. The test application will be shipped by glide-in and run at the WorkNode. In this way, the randomness of user job access pattern is repeated in the test environment, since the availability of job slots at the remote sites, type of WN and all the local hardware and WAN networking features are the same as normal Grid jobs.

The performance of BeStMan using GlideinWMS is shown in Fig. 4.

- Use Glidein Tester[7], a fully automated framework for scalability and reliability testing of centralized, network-facing services using Condor glide-ins. The main purpose of this testing system is to simplify the testing configuration and operation. The system can actively acquire and monitor the number of free CPU slots in the Condor pool and manage to send testing jobs to those slots when some conditions are met, for example we want to run X jobs simultaneously. The performance of BeStMan2 measured by Glidein Tester is shown in Fig. 5.

We measured the scalability of recent OSG released BeStMan and BeStMan2. There is significant difference in the scalability features between two types of BeStMan technology. The BeStMan2 shows better performance and reliability from the high throughput and high available service point of view.

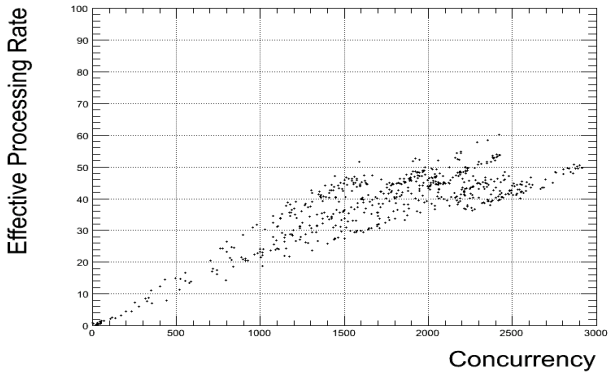

Figure 4. Correlation between Effective Processing Rate and Client Concurrency for small directories

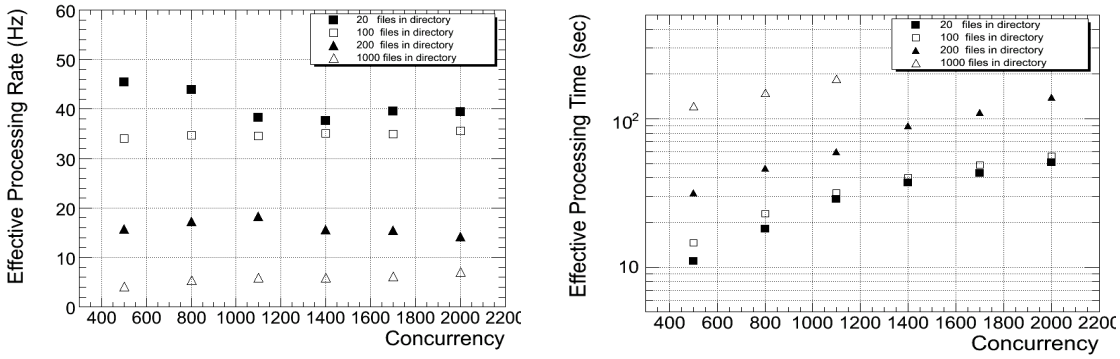

Figure 5. Correlation between Effective Processing Rate and Client Concurrency for large directories (left) and Correlation between Effective Processing Time and Client Concurrency for large directories (right)

4. WAN Data Transfer with HDFS-based SE

Fig. 6 shows the WAN transfer of CMS Tier-2 of University of California, San Diego (UCSD) that move data from or to other computing sites with two 10 Gpbs links with ~15 Gpbs of the maximal data transfer rate we are able to achieve. GridFTP is the major WAN transfer tool managed by BeStMan and PhEDEx for this operation.

It is important to understand how to efficiently fill the network pipe. Most of the efforts concentrate on optimizing the number of streams for each GridFTP for the data transfer and total number of GridFTP that can simultaneously transfer the data. Due to the nature of HDFS that it can't support asynchronous write, it is widely confirmed that for the case of sustained high throughput WAN transfer, single stream per GridFTP transfer is the best solution.

The GridFTP implements HDFS interface which use a local buffer to keep all the incoming data in memory or local file system before data being sequentially written to HDFS storage. If the GridFTP is not fast enough to sink incoming data into HDFS-SE, the buffer will be full quickly, so the existence of buffer won't increase the overall WAN transfer throughput. A second issue is that multiple streams per transfer requires large buffer size to keep all the out-of-order packets, which is limited by the amount of system memory. The third issue is that the use of local buffer will introduce significant overhead in the system I/O before the data written to HDFS, which has negative impact on the sustainable throughput.

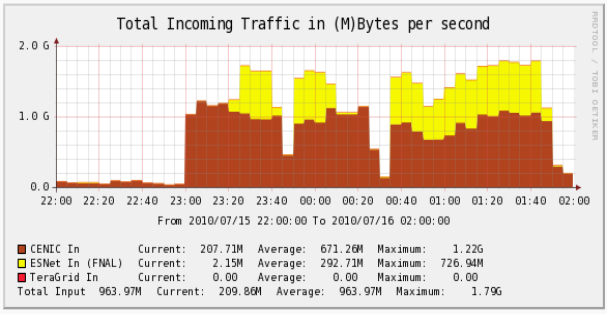

Figure 6. WAN transfer throughput using GridFTP with two 10 Gbps links measured at UCSD HDFS-based SE

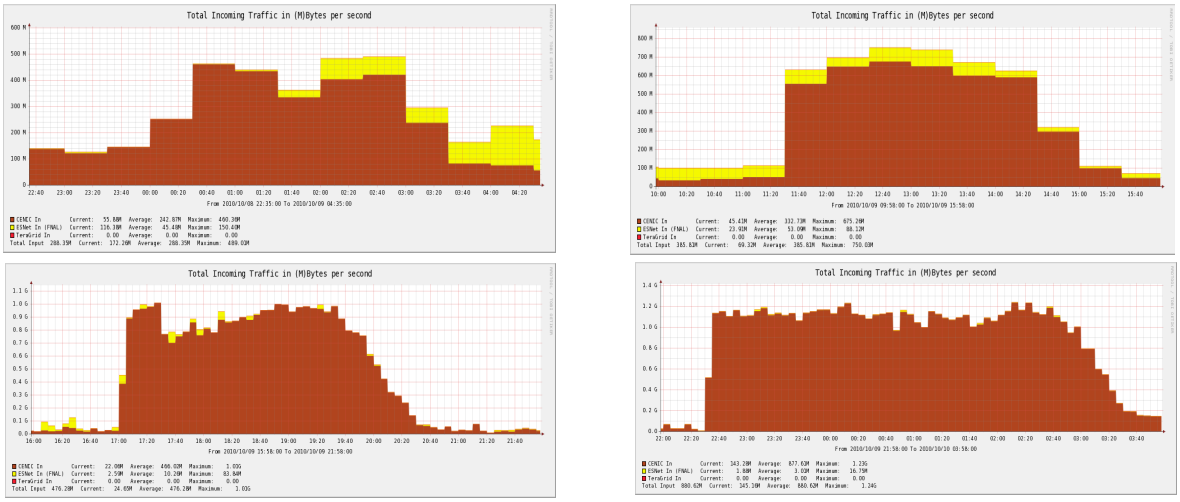

Figure 7. WAN transfer throughput in 10 Gbps link with 10 GridFTP clients (top left), 20 GridFTP clients (top right), 30 GridFTP clients (bottom left) and 60 GridFTP clients (bottom right)

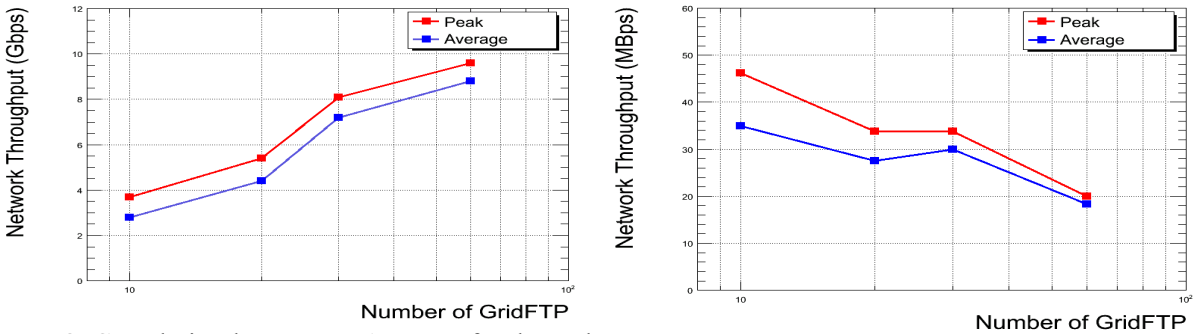

Figure 8. Correlation between WAN transfer throughput and number of GridFTP clients: Total throughput vs number of GridFTP clients (left) and average individual throughput vs number of GridFTP clients (right)

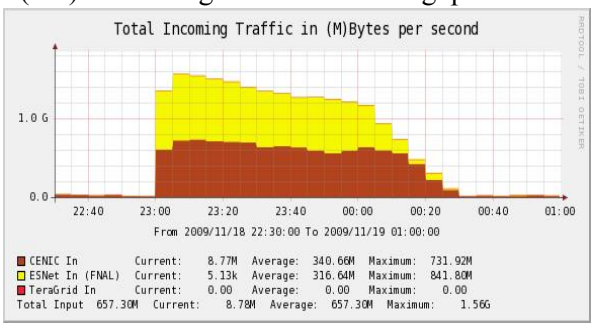

Figure 9. WAN transfer throughput using FTD with two 10 Gbps links measured at UCSD HDFS-based SE

We use various number of GridFTP clients to measure the WAN throughput as shown in Fig. 7. In order to effectively fill up the 10 Gbps link, more than 30 clients are necessary. There is a very strong correlation between the total throughput and the number of GridFTP clients. While as the number of GridFTP clients increases, the data transfer rate per GridFTP actually decreases as shown in Fig. 8. We also used Fast Data Transfer (FDT) for the WAN transfer with 30 clients to fill up two 10 Gbps links as shown in Fig. 9 and observed similar performance as that of GridFTP based transfer. In the following, we list several important factors of the system architecture at UCSD that have non-trivial impact on the performance of the data transfer:

- The physicscal limit of each HDFS datanode. The throughput for a single node to write to HDFS is ~60-80 MB/s. To support up to 15 Gbps of WAN throughput, at least 30 nodes need to run simultaneously to sink the data. This is consistent with the observed WAN throughput vs number of GridFTP clients.
- Use of HDFS datanode for WAN transfer server and client. Ideally the WAN transfer server mainly deal with network issue, while the software needs to buffer the data in the local memory first, which introduces a lot of system I/O and CPU utilization, and has negative impact on the throughput and the performance of HDFS. The optimization of transfer software configuration remains a critical task, which has to be tuned based on each site's hardware condition, expected use case and data access pattern etc.

5. Summary and Outlook

In this paper, we present the experience of high throughput WAN data transfer with HDFS-based Storage Element. Two protocols, GridFTP and fast data transfer (FDT), are used to characterize the network performance of WAN data transfer. The optimization of WAN transfer performance is conducted to maximize the throughput. We see very high scalability at the middleware level, HDFS and BeStMan, for WAN transfer. Various issues related to understanding the results and possible future improvement are discussed.

For the future under the envision of 100 Gbps network being available for the scientific community, we believe the study of WAN transfer for HDFS-based SE will remain an important area, because:

- The continuously growing size of storage system will make HDFS a more attractive solution for SE. Its highly distributed and scalable nature provide plenty of room for its integration with existing and new grid middleware to conduct the high throughput data transfer.
- The availability of 100 Gbps network requires efficient way to move data between the grid middleware and HDFS. We expect a scale-out strategy by adding more parallel clients or implementing new technology to use smaller number of hosts but more powerful in handling the disk I/O.
- The WAN transfer tool must be able to fill the network pipe more efficiently and control possible network congestion due to running too many simultaneous and independent WAN transfer streams.
- Finally the multi-core system and increasing size of hard drive might make the overall throughput of disk I/O a potential bottleneck for high throughput WAN transfer. The growth rate of number of cores is much faster than that of I/O per disk. This could lead to a mismatch between the applications and available I/O from disks.

Reference

[1] Hadoop, http://hadoop.apache.org
[2] Pi H et al. “Hadoop Distributed File System for the Grid”, *Proceedings of IEEE Nuclear Science Symposium and Medical Imaging Conference 2009*
[3] The Open Science Grid Executive Board, “A Science Driven Production Cyberinfrastructure – the Open Science Grid”, OSG Document 976, http://osg-db.opensciencegrid.osg/cgi-bin/ShowDocument?docid=976
[4] Rehn et al. “PhEDEx high-throughput data transfer management system”, *Proceedings of Computing High Energy Physics 2006*
[5] Berkeley Storage Manager (BeStMan), https://sdm.lbl.gov/bestman
[6] Sfiligoi I et al. “The Pilot Way to Grid Resources Using glideinWMS”, *Proceedings of Computer Science and Information Engineering, 2009 WRI World Congress on, pp. 428-432, March 2009*
[7] Sfiligoi I et al. “Using Condor Glideins for Distributed Testing of Network Facing Services”, *Proceedings of Computational Science and Optimization, 2010 Third International Joint Conference on, vol. 2, pp. 327-331, May 2010*