Lustre filesystem for CMS storage element (SE)

Y Wu¹, B Kim¹, P Avery¹, Y Fu¹, D Bourilkov¹, C Taylor², C Prescott², J Rodriguez³
¹Dept. of Physics, Univ. of Florida, P. O. Box 118440, Gainesville, FL 32611, USA
²HPC Center, Univ. of Florida, P. O. Box 118440, Gainesville, FL 32611, USA
³Dept. of Physics, Florida International University, Miami, FL 33199, USA

E-mail: yujun@phys.ufl.edu

Abstract. This paper presents our effort to integrate the Lustre filesystem with BeStMan, GridFTP and Ganglia to make it a fully functional WLCG SE (Storage Element). We first describe the configuration of our Lustre filesystem at the University of Florida and our integration process. We then present benchmark performance figures and IO rates from the CMS analysis jobs and the WAN data transfer performance that are conducted on the Lustre SE.

1. Introduction

The amounts of scientific data generated by large-scale experiments have reached a level not easily handled by individual users, groups or even small institutions. Many of these experiments need national or even global collaborations. The CMS experiment, one of the Large Hadron Collider (LHC) experiments, has more than 3600 people from 183 institutions in 38 countries. CMS is expected to produce petabytes of data per year at the new world’s highest energy collider at Geneva, Switzerland. The data produced with the CMS experiment will be distributed globally [1]. How to manage these large data volume efficiently in distributed environments and replicate them to heterogeneous storage systems is a big challenge. In order to tackle the storage management and replication challenges in large-scale scientific computing, several Grid storage service implementations have been developed and a Storage Resource Manager Interface Specification v2.2 (SRM v2.2) has been created to define the common control interface for storage resource management systems [2]. Most of these implementations use the popular GridFTP protocol [3] as the underlying transfer protocol.

Lustre filesystem is an open source, distributed filesystem generally used for large scale HPC (High Performance Computing) cluster computing [4]. The University of Florida HPC Center (UF-HPC), and the CMS Tier-2 Center (UF-T2) in collaboration with the Florida International University (FIU) CMS Tier-3 Center have been testing the Lustre filesystem with CMS applications since 2008. According to the tests, the Lustre filesystem with CMS applications has shown good data access performance and remote access features [5] [6]. In addition, we explored the possibility of integrating the Lustre filesystem into dCache [7] and the system worked reasonably well although some functionality overlaps between the Lustre filesystem and the dCache were noticed [5]. Putting data directly into the Lustre filesystem without the dCache will avoid these functional overlaps, simplify the overall data storage architecture, and allow users to access data directly using Lustre’s POSIX compliant feature. In this paper, we explore the possibility to use the Lustre filesystem with BeStMan [8], one of the SRM v2.2 implementations, GridFTP [3] and Ganglia monitoring system [9] and to make the whole system as a fully functional WLCG SE (Storage Element). We will report a series of
measurements on our implemented Lustre SE, including IO rate and WAN data transfer performance, and discuss the interoperability with other storage technologies.

2. Architecture and Implementation

Our integrated Lustre SE architecture consists of the following components: Lustre filesystem [4], BeStMan storage resource manager [8], GridFTP servers [3], and Ganglia monitoring system [9]. We implemented the integrated Lustre SE system by fully utilizing the existing infrastructures at the UF-HPC and UF-T2. Here we first give a brief overview of the components in our Lustre SE and then describe further steps to bring the whole system together.

2.1. Components of Lustre SE

As just mentioned, major components we use to build the Lustre SE are Lustre, BeStMan, GridFTP and Ganglia. Lustre filesystem, developed by Sun Microsystems (now part of Oracle Corporation), generally consists of one MDS (Metadata server) and multiple OSSes (Object Storage Server) that contain various numbers of OSTs (Object Storage Target). The actual data are stored in OSTs and accessed through OSSes. Lustre is able to deliver high I/O performance by distributing data objects across OSTs and allowing clients to directly interact with OSSes without a single bottleneck [4].

BeStMan, or Berkeley Storage Manager, is a full implementation of SRM v2.2. It is developed in Java and works with disk based storage system and mass storage system. We have chosen the BeStMan SRM implementation because of its modular design feature that allows it to be able to integrate with different types of storage systems easily [8]. GridFTP is developed by Globus Alliance as one of the Grid software components for high performance reliable data transfer with Grid security for authentication and authorization [3]. BeStMan works with GridFTP as one of its file transfer services to transfer files between different sites. Ganglia, an open-source distributed monitoring system [9], is used to provide monitoring information on the GridFTP servers.

2.2. Integration of Lustre SE and Implementation of GridFTP Load Balancing

We have brought the four components described in 2.1 together by configuring BeStMan with the Lustre filesystem as the underlying storage for the GridFTP servers and developed a Java plug-in to BeStMan that utilizes Ganglia as the information system for the GridFTP load balancing. The implementation of GridFTP load balancing has two parts. The first part is the system monitoring: it utilizes Ganglia to monitor the number of pending read/write transactions on the GridFTP servers and the Lustre filesystem. Then the information on the number of pending read/write transactions using Ganglia’s gmetric utility is published via Ganglia. The second part is the GridFTP server selection. A BeStMan Java plugin was developed to read a cost table to decide which GridFTP server to use for transfers. Since we are currently using only three GridFTP servers, we use a static cost table based on the GridFTP server configurations. In the future, we plan to update this cost table dynamically using the published monitoring info from Ganglia. Figure 1 shows the implementation architecture.

Figure 1. Implementation architecture
3. Experimental Results and Discussions

3.1. Hardware Setup
UF-HPC and UF-T2 have jointly deployed more than 330TB of a RAID storage using the Lustre filesystem. The latest deployment is 215TB of Lustre filesystem using five OSS servers with an underlying storage system using the Areca ARC-1680ix-16 PCIe SAS RAID controller and 2TB enterprise-grade SATA drives (configured as RAID 4+1). Specifically, a SUPERMICRO SYS-6026T-NTR+ 2U E5520 2.3GHz quad-core processor with 24 GB memory is used as the MDS and the Intel X25-M x 2 MLC solid-sate drive as the MDT. Lustre filesystem is mounted on about 3000 batch cluster nodes with InfiniBand connection. Lustre filesystem is also mounted directly on the BeStMan server and the GridFTP servers. We use three GridFTP servers with a 10Gb NIC card for the file transfers.

3.2. Local SE IO Performance
A Storage Element (SE) should provide applications with access to the data stored on the SE [10]. As a POSIX compliant filesystem, Lustre filesystem can be accessed by user applications without any special protocol. In our first experiment, we test the performance of local data access to the Lustre SE. The test includes the performance of the metadata and the data IO access.

The metadata performance is useful in measuring how fast an application can query the metadata info and/or create new entries in the SE. We used a well-known metadata benchmark tool called Mdtest [11] to perform the test. Mdtest is an MPI-coordinated metadata benchmark tool. We performed the test using more than 20 nodes to check the basic open/stat/close operations on files and directories. Figure 2 shows the benchmark results using the Mdtest. As we can see, the Lustre filesystem has a reasonably good metadata performance and scales well with the number of parallel processes as the number of processes is varied from 200 to 800 on the 20 nodes. If we assume each batch system worker node runs 3 CMS production/analysis jobs, 800 process benchmark result proximally reflects our Lustre filesystem metadata performance when 270 nodes are accessing the filesystem at the same time. Since the operations-per-second rate is almost flat when number of processes reaches 200, it is reasonable to infer that the metadata performance will scale well when more CMS jobs are accessing the data on the Lustre filesystem simultaneously.

Figure 2. Metadata performance on 20 UF-HPC nodes
We then try to run some CMS user analysis jobs to measure the job IO performance. We find it is a little difficult to run many user jobs through a single user at the same time on the UF-HPC cluster due to the requirements of UF-HPC batch system. As a result, we decide to observe the ongoing user activity generated by analysis jobs that are submitted to the UF-HPC cluster by the CMS Remote Analysis Builder (CRAB) [12] application. This resulted in a series of monitoring analysis. Figure 3 and Figure 4 show the network activities of the CMS Lustre nodes in a 24-hour period of time and the corresponding number of user jobs running on the UF-HPC cluster. Figure 3 shows the Lustre filesystem was able to provide 1.4GB/s of output read rate while there were some write processes in parallel. Figure 4 shows the number of CMS user jobs running on the UF-HPC cluster during the same period of time. Although the small-byte-read hit the underlying storage system very heavily, the Lustre SE can deliver more than 1MB/s/slot for CMS applications.

![Figure 3. CMS Lustre IO Nodes Network Activities during 24 hours period (Bytes/Sec vs. Time)](image1.png)

![Figure 4. The number of running CMS jobs at UF-HPC Cluster during the same period](image2.png)

After finishing the monitoring analysis, we conduct a detailed study by running a limited number of CMS analysis jobs using our Lustre SE. We use CRAB to submit the analysis jobs to the UF-HPC cluster or the UF-T2 cluster for CMS MC data and CMS real data. Figure 5 shows the read rate per slot for the real data and Figure 6 shows the read rate per slot for the MC data. The read rate is measured by submitting jobs that read files on the Lustre filesystem (blue). During the test, background jobs submitted by other users are running. As shown in the plots, the average rate for MC data is lower than that for real data due to a significant amount of time being spent during opening and closing files. We have set up a small scale Hadoop filesystem to make a simple comparison with the Lustre filesystem. The identical jobs that were submitted to the Lustre filesystem were submitted to analyze the identical files on the Hadoop filesystem. The results using the Hadoop filesystem (the red line in Figure 6) are generally a little worse than the results using the Lustre SE.
3.3. WAN Data Transfer Performance

As we mention earlier, the Lustre SE has been able to serve as an SRM endpoint and sends/receives files across the WAN through the BeStMan SRMv2 interface. To test the data transfer performance with our Lustre SE, we configure PhEDEx, the CMS official data transfer management system [13], as the driving tool to call BeStMan SRMv2 interface. We have been able to achieve very good and stable transfer performance using this setup. Figure 7 shows the transfer performance measured from FNAL CMS Tier-1 site to the UF Tier-2 site. The transfers lasted for a week with a stable rate of 200-300MB/s. The rate was limited only by the serving rate set at the source site. In another transfer performance measurement from Florida Tier-2 to Caltech Tier-2, the transfer rate reached above 600MB/s for 6 days as shown in Figure 8. We also observe that the PhEDEx transfers have been able to achieve 300-600MB/s from time to time. It should be noted that when we were conducting these tests, the parallel transfers to the Lustre filesystem were on-going as well. The maximum total IO output (local +WAN) we were able to observe at that time was about 2GB/s.

We also tested the transfer performance between local GridFTP servers within Florida Tier-2 by putting files into the Lustre filesystem. We were able to achieve about 1015MB/s through two GridFTP servers. We believe we can get much higher WAN transfer rates if the WAN path and remote site(s) have the capability.

3.4. Interoperability with Other Storage Technologies

Using our Lustre SE setup, we have been able to transfer files to and from more than 25 CMS sites, which include all national level CMS Tier-1 sites (8 sites) and many Tier-2 and Tier-3 sites spanning across Europe, Asia, North and South America. This can be observed live at the PhEDEx CMS Data Transfer web site [20]. Among the CMS sites, the storage management systems in use include: dCache [6], Enstore [14], Castor [15], Hadoop [16], REDDnet [17], INFN StoRM [18], etc. We can see the transfers using our setup work pretty well with all of these storage technologies as BeStMan is a full implementation of standard SRM v2.2.
4. Conclusions and Future Direction

We describe our implementation to integrate the Lustre filesystem with the BeStMan storage resource manager, the GridFTP, and the Ganglia monitoring system as the SE and our deployment of the Lustre filesystem at the University of Florida. CMS analysis jobs running against the data stored on the Lustre filesystem have demonstrated high IO data access rate. The Lustre filesystem metadata operations scale well when large number of processes access the system simultaneously. With the standardization of the SRM v2.2 interface, our Lustre SE setup interoperates well with other storage implementations used by CMS sites worldwide and very good transfer performance.

Further study is needed on how to select the best GridFTP server efficiently for transfers based on the monitoring data. Additional studies may also be needed on how to improve the transfer performance across the WAN. One point worth mentioning is the remote mounting feature with the Lustre filesystem. It is possible to mount the Lustre filesystem remotely via WAN. This may allow CMS Tier-3 sites to access data stored at Tier-2 or even Tier-1 sites directly. An effort is going on to study how to enable the Lustre WAN access securely and efficiently with the ExTENCI [19] distributed filesystem project.

5. References

[1] “The CMS computing model”, CMS NOTE/2004-31, CERN, Geneva, 2004.
[2] A. Sim, A. Shoshoni, F. Donno and J. Jensen, “Storage Resource Manager Interface Specification V2.2 Implementations Experience Report”, Grid Storage Resource Management, 2009: http://www.ogf.org/documents/GFD.154.pdf.
[3] Globus GridFTP web portal: http://www.globus.org/toolkit/data/gridftp.
[4] Lustre web site: http://www.lustre.org.
[5] Y. Wu, B. Kim, J. Rodriguez, Y. Fu, D. Bourilkov and P. Avery, “Utilizing Lustre file system with dCache for CMS analysis”, CHEP09, Prague, Czech Republic, 2009.
[6] J. Rodriguez, T. Brody, C. Prescott, Y. Wu, B. Kim, Y. Fu, D. Bourilkov and P. Avery, “Wide area network access to CMS data using the Lustre file system”, CHEP09, Prague, Czech Republic, 2009.
[7] dCache web site: http://www.dcache.org.
[8] Berkeley storage manager (BeStMan) web site: https://sdm.lbl.gov/bestman.
Acknowledgments
The authors want to thank the CMS colleagues Daniele Bonacorsi and Wolf Behrenhoff for their help with the presentation.