2014

Distributing CMS Data between the Florida T2 and T3 Centers using Lustre and Xrootd-fs

Gary Kaganas
Department of Physics, Florida International University, gkaganas@fiu.edu

Jorge Luis Rodriguez
Department of Physics, Florida International University, jrodrig@fiu.edu

Mengxing Chen
Florida International University

P. Avery
University of Florida

D. Bourilkov
University of Florida

See next page for additional authors

Recommended Citation
G Kaganas et al 2014 J. Phys.: Conf. Ser. 513 042037

Journal of Physics: Conference Series **513** (2014) 042037 doi:10.1088/1742-6596/513/4/042037

Distributing CMS Data between the Florida T2 and T3 Centers using Lustre and Xrootd-fs

G. Kaganas[1], J.L. Rodriguez[1], M. Cheng[1], P. Avery[2], D. Bourilkov[2], Y. Fu[2], J. Palencia[3]

[1]Florida International University, Miami FL, USA
[2]University of Florida, Gainesville FL, USA
[3]Pittsburgh Supercomputing Center, Pittsburgh PA, USA

jrodrig@fiu.edu, bourilkov@phys.ufl.edu

Abstract. We have developed remote data access for large volumes of data over the Wide Area Network based on the Lustre filesystem and Kerberos authentication for security. In this paper we explore a prototype for two-step data access from worker nodes at Florida Tier3 centers, located behind a firewall and using a private network, to data hosted on the Lustre filesystem at the University of Florida CMS Tier2 center. At the Tier3 center we use a client which mounts securely the Lustre filesystem and hosts an XrootD server. The worker nodes access the data from the Tier3 client using POSIX compliant tools via the XrootD-fs filesystem. We perform scalability tests with up to 200 jobs running in parallel on the Tier3 worker nodes.

1 Introduction

The LHC computing community is exploring alternatives to the traditional strategy of deploying storage and processing resources at the same facility. In this model inefficiencies arise when jobs wait for computing resources to free up at sites with particularly popular data sets or when the data sets are duplicated across many sites that all want to access these interesting data. In addition, facilities with modest computing resources would need to deploy sizeable disk or tape storage systems that greatly add to the complexity of the facility, making it more difficult to maintain and operate. This is especially problematic for small Tier3 centers with limited access to suitably trained manpower. Recently efforts within the CMS [1] and Atlas [2] experiments are underway to explore federated data stores which feature remote data access currently through XrootD services running on the grid. A computer can access data by simply specifying a globally defined logical filename which is then translated by the infrastructure, through a series of XrootD services, to find the location of actual files which are then served to the computer from anywhere at any time. So far, these efforts focus on using these federated data stores as a failover mechanism to access data already stored locally.

In this paper we measure the I/O performance of typical HEP work flows accessing data remotely via the Lustre cluster filesystem (FS) [3] distributed on a Wide Area Network (WAN). We utilized a portion of the ExTENCI [4] testbed between the CMS Tier2 center located in Gainesville, at the University of Florida (UF) and the Tier3 center at Florida International University (FIU) located in Miami. The testbed features clients that are separated by 560 km (14 ms Round Trip Time "RTT") from the storage devices. The storage is mounted over a 10 Gbps network with Kerberos [5] authentication between clients and server. We also report on the performance impact of using XrootD services to bridge the remote Lustre FS mounted to WorkerNodes (WN) when the WNs are connected only via a private VLAN.

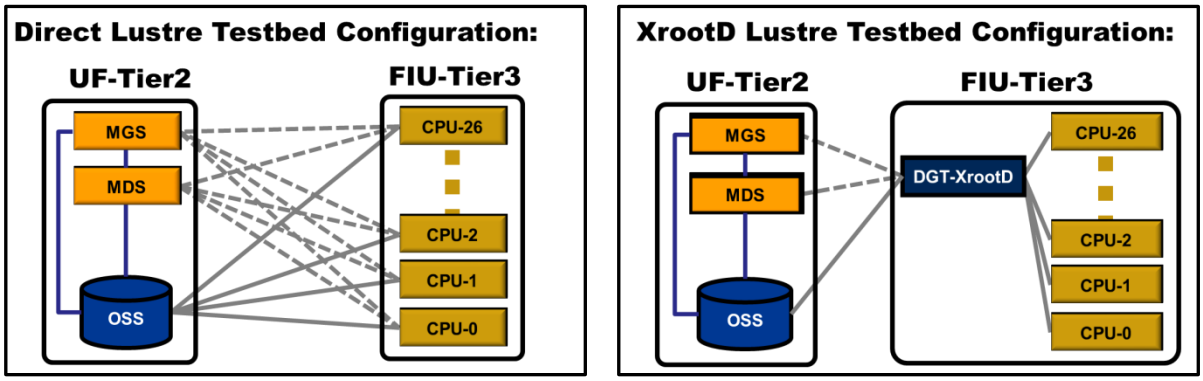

Figure 1. In the direct Lustre testbed configuration (left), each WN has its own public interface and FQDN. Kerberized access to Lustre storage is thus possible through Lustre clients running on the nodes. In the XrootD Lustre testbed configuration (right), all WNs have access only to a private network. Kerberized access to the Lustre storage is bridged through XrootD services running on a dedicate server with access to WAN over 10 Gbps network. The XrootD server also has 10 Gbps connectivity to the private network.

2 The ExTENCI Lustre testbed

The ExTENCI project established a distributed storage testbed that connected computing centers with storage resources located at Fermi National Laboratory and the University of Florida. The testbed linked Lustre clients located at sites across the US: Pittsburg Supercomputing Center, Florida State University and Florida International University. For this study only the segment linking the Tier3 to the UF was used over the research network established primarily over the Florida Lambda Rail. We configured the testbed in two distinct ways and compared the performance in both. In Figure 1, we show a schematic of the two configurations used in this study and briefly describe the configuration and hardware utilized in the sub-sections below.

2.1 UF ExTENCI Lustre Storage System

The Tier2 facility in Gainesville, FL housed the main components of the Lustre storage system used in this study. The system consisted of a series of servers and disk arrays that together formed the Lustre cluster filesystem which was configured to allow remote client mounts over Wide Area Network (WAN). The systems used at the Tier2 include:

- Main Lustre management systems were hosted on a single box which ran the MGS/MDS (ManaGement/Meta Data Server) services. The hardware consisted of a dual quad-core AMD Opteron 2378 with 8 GB RAM and 1 Gpbs connectivity provided via its onboard NIC.
- The storage servers which hosts the Lustre OSS (Object Storage Service), provide client I/O services and handles network request for the Lustre OST where configured with dual quad-core AMDs with 16 GB of RAM. Each server was connected to storage arrays via three QLogic QLE2462 dual-channel cards. Network connectivity to the WAN was provided by a Chelsio T310 10GbE NIC
- The Storage Arrays (where the block devices live) and referred to as the OSTs (Object Storage Targets) used are RAID Inc. Falcon III F16SF4Gs, three each configured with a total of 72 Seagate 1TB SATA HDD.

2.2 FIU Tier3 Linux Cluster

The Tier3 facility in Miami, FL is a USCMS Tier3 site with the usual assortment of hardware and software services needed to operate a CMS grid enabled site. In this study we only employed the Tier3's 216 core (27 WNs) HTCondor [6] Linux cluster and a server, referred to as "DGT" (see Figure 1) that housed the XrootD services needed to bridge Lustre mounts when the WNs were behind the private network. The DGT machine was configured with a dual port 10 Gbps NIC which provided 10 Gbps connectivity to both the WAN and private network. All of the servers, headnodes, WNs and XrootD server were configured with dual 2.4 GHz Xeon CPUs with 16 GBs of RAM. All servers except the XrootD server were connected via dual 1 Gbps NIC to a 10 Gbps capable Dell 6248 switch. The FIU Tier3 also connects at 10 Gbps to the campus research network.

3 Testbed Configuration

Connectivity between the Tier2 and Tier3 sites was established over the dedicated 10 Gbps research network. On both sites, campus firewalls and filters are avoided through the research networks. We tested the connectivity first by running iperf tests between the sites showing a sustained throughput of approximately 8 Gbps. To minimize transfer performance peculiarities due to the implementation of Kerberos security measures, (like file integrity checks), all tests reported were performed using the null Kerberos flavor. Using the null flavor still preserves the most important aspects of Kerberos authentication, namely establishing a secure link between machines with proper credentials without impacting performance [7].

In this study the Lustre storage system at the Tier2 remained intact without alteration for all tests described below. The Tier3 cluster however, was configured in two distinct ways.

3.1 XrootD Lustre Configuration

In the XrootD Lustre configuration all WNs at Tier3 were connected only to a private network. The lack of direct access to the WAN prevented the establishment of secure connections to the Lustre storage through Kerberos. To remedy this we deployed XrootD services on a machine with WAN access to bridge the Lustre mounts to all 27 WNs. It is worth mentioning that we initially tried to bridge the Lustre FS through NFS v3 but found that unreliable and prone to crashes. On each of the WNs we accessed the Lustre storage either through XrootD-fs clients (required for POSIX access by UNIX dd) or via built in XrootD support in the root and CMSSW applications: essentially using the string

`root://<XrootDserver.local:1094//extenci/cms…`

to specify the path to the data files. Here the string "`/extenci`" is the top level directory or mount point for the remote Lustre FS. Naturally, the Kerberos authentication reaches as far as the mount point on the XrootD file server. Once mounted the files are outside the Kerberos realm and are free to be exported elsewhere. Since all WNs are behind a private network, each with 1 Gbps capability, the total bandwidth for a job is theoretically limited by the maximum network bandwidth capabilities of XrootD server which is configured with 10 Gbps connectivity to both the private and public networks. We found however, that network bandwidth was likely not an important factor in degraded I/O performance of the testbed since throughput ever exceeded more than 20 percent of the 10 Gbps link.

.

3.2 *Direct Lustre Configuration*

In the second configuration, which we label "direct Lustre" we eliminate the XrootD bridge server. Instead we assigned each worker node its own public IPv4 address and a fully qualified domain name as well as Kerberos principles for everyone. Public IPs are necessary to allow the establishment of Kerberized mounts to each node. Each WN was also configured with the Lustre patchless client that provides the necessary kernel modules to allow mounting of a Lustre FS. File addressing in job scripts was done in the usual way one addresses files mounted locally. We observed I/O rates that were a substantial large fraction of the available network bandwidth so concluded that limited bandwidth was not impacting performance.

3.3 Tests Performed

All trials were run in batches of parallel jobs varied from 32 to 192 simultaneous jobs. In general, HTCondor will attempt to fill all the available cores on each node. This scheme impacts I/O bound applications, which tend to saturate the bandwidth capacity on a node. However, since the core per node filling scheme is the default used at the Tier3 and typically common we decided not to change the behavior for these trials. Two metrics were used: the I/O rates and wall time to complete a batch of simultaneously running jobs (time-to-completion). The I/O rates reported are averaged from measurements taken at 5 second intervals over the entire run. These average I/O rates provide a general gauge of the I/O efficiency of the tested configuration, it may mask the effect of optimization features like file readaheads and read caching. A more comprehensive efficiency measure is the trial wall time. Wall times automatically assume all dominant effects of the chosen configuration, and

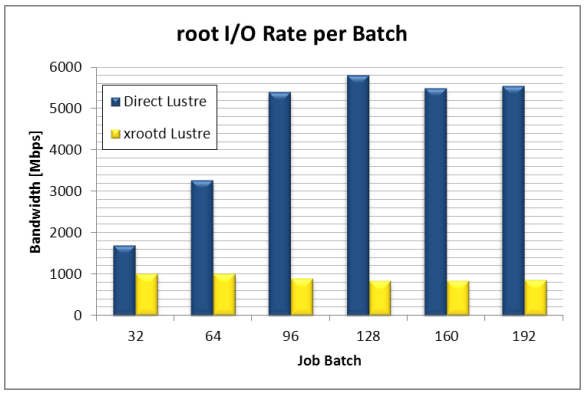

Figure 2. Bandwidth for job batches performing the I/O intensive root routine. We compare bandwidth consumption for jobs executed in the direct Lustre and the XrootD Lustre mode.

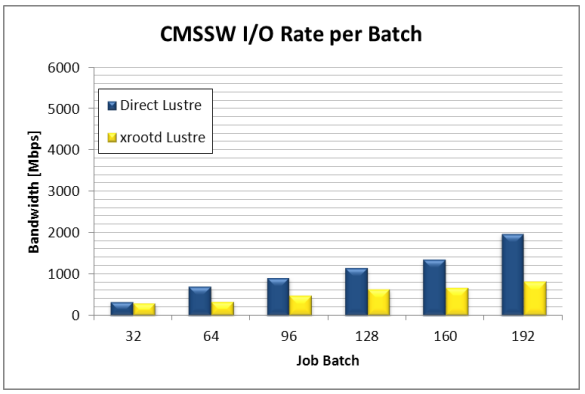

Figure 3. The I/O bandwidth for job batches performing the Higgs 4-lepton channel reconstruction. We compare bandwidth consumption for jobs executed in the direct Lustre and the XrootD Lustre mode.

afford a more relevant basis for comparison. With regards to measurement, the I/O bandwidth was probed by parsing the /proc/net/dev file on the server running the OSS. The total number of bytes transferred out of the public-facing OSS interface was measured at periodic intervals of 5 seconds. The difference between the late and early value divided by the sampling rate is what we define as the I/O bandwidth. Finally, we examined the performance of three different applications: UNIX dd, a lightweight root application which does very little processing but lots of I/O and a typical CPU bound CMSSW Higgs analysis. The tests were run once on the XrootD Lustre configuration and again on the Direct Lustre configuration. In all, six different tests were performed.

3.3.1 UNIX dd runs

To establish a performance checkmark, a UNIX dd read transfer was performed. The read was done from the remote mounted Lustre FS to the workernode's memory cache. Parallel transfers of up to 192 files were recorded where each file was unique to mimic real-world analyses that read in files once. The files used were CMS root files, ranging from 2 to 5 GB in size.

3.3.2 root I/O application

The root application utilized 192 unique files each with trees and branches filled with random numbers. The files sizes were all approximately 5.1 GB in size and again located on the remote Lustre FS. The root I/O application used simply reads in an event in its entirety, does some minimal processing, i.e. adds up entries, and moves on to the next event. Like the UNIX dd application, the root routine does not store data to the local storage system.

3.3.3 CMSSW Higgs reconstruction

A Higgs to 4-lepton channel analysis was chosen to model the behavior of a typical CMSSW application. The application spends some time processing events so is expected to be more CPU intensive than our root I/O application or dd runs. While the analysis itself only processed 32 distinct files; those 32 were duplicated 5 additional times, with unique file names, to recreate the conditions for a job execution on 192 cores.

4 Results

We report our results on the I/O performance and wall time measurements (time-to-completion) in Figure 2 through 5. The plots highlight the differences between the two testbed configurations: direct Lustre and XrootD Lustre as described above. The measurements were performed by running simultaneous jobs in batches of 32 jobs through 192 jobs. The I/O bandwidth measurements fluctuate over the run by a significant amount, especially for the trials involving XrootD. The variations likely indicate network congestion or resource throttling particularly evident at the XrootD server.

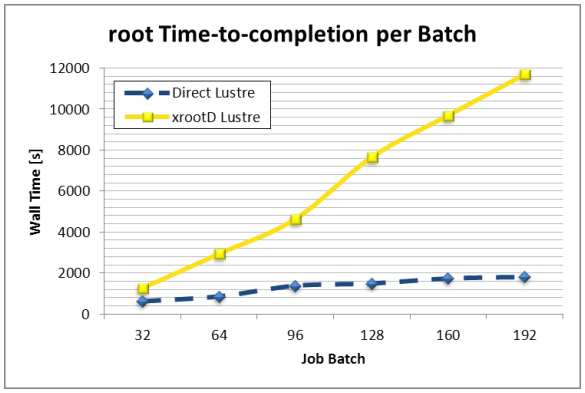

Figure 4. The wall time to complete the I/O intensive root application per job batch for the direct Lustre configuration compared against the XrootD Lustre configuration.

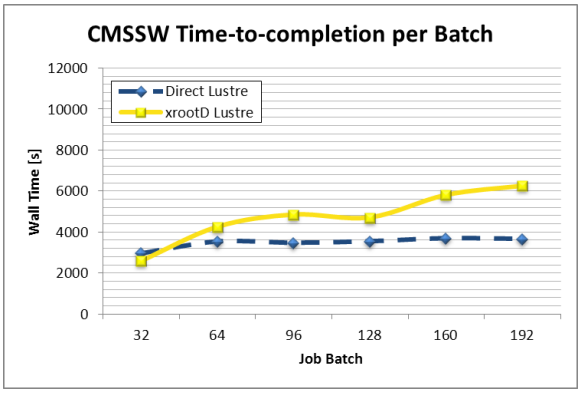

Figure 5. The wall time to complete the CPU bound CMSSW application per job batch for the direct Lustre configuration compared against the XrootD Lustre configuration.

4.1 Bandwidth Results

Bandwidth tests were conducted with the testbed configured with and without the XrootD server bridging the remote Lustre FS to the WNs. As shown in Figure 2, we first compared the aggregate I/O bandwidth resulting from a batch of parallel runs of root jobs with the testbed configured in direct Lustre mode against the testbed configured in the XrootD Lustre mode. In the direct Lustre mode we found aggregate I/O rates for root jobs to scale linearly up to a maximum of approximately 5.5 Gbps at 96 simultaneous jobs. This rate is near the maximum 8.0 Gbps I/O rate that the Tuer2 has measured on their Lustre storage locally [8]. In contrast, the maximum aggregate bandwidth for trials performed using the same root application on the testbed configured in the XrootD Lustre mode was less than 1.0 Gbps. In these trials there is also little appreciable difference with respect to the number of simultaneous jobs with regards to performance. This clearly illustrates the performance bottleneck resulting from using XrootD services running on a single server. Since the XrootD services were not tuned beyond that found at a CMS Tier2 and the service is run on a single server it is unclear whether the performance degradation is something that can be improved significantly beyond what we see in this study.

We also measured the aggregate I/O bandwidth using our reference CMSSW application that represents a typical HEP analysis job. Figure 3 shows that in the direct Lustre mode the aggregate I/O bandwidth increases steadily from 32 simultaneous jobs to 192. We expect the scaling to continue given that at 2.2 Gbps we are only 25% into the intrinsic capabilities of our Lustre storage system and well below the capabilities of the network. In the XrootD Lustre mode we see similar behavior as seen with the root application. But the presence of the XrootD services do not impacts the I/O performance as much since the application here is CPU bound.

4.2 Time to Completion Results

Our Time-to-Completion Figure 4 and 5 results qualitatively reflect the same conclusions that we draw from the aggregate I/O bandwidth results. For example, with the testbed in the direct Lustre configuration the I/O bound trials (root application) have low and flat time-to-completion profiles, reflecting the fact that the I/O bandwidth of the testbed is capable of scaling with the number of jobs slots available. We see that in this configuration more real work is done per unit time regardless of the I/O profile of the jobs because there is sufficient I/O bandwidth in the system. In the XrootD Lustre configuration we see again the effect of the XrootD bottleneck as reflected in the longer completion times when the number of jobs increases. This is particularly prevalent for the root jobs which are I/O intensive. There the wall time increases dramatically as the number of batch jobs increases from 32 to 192. Comparing the direct to XrootD Lustre mode at the 192 job batch trial, the direct Lustre batch completed 6.5 times faster than the XrootD. In fact, in XrootD mode the total wall time at 192 jobs is about 9 times slower than the 32 job runs thus the I/O bottleneck with 32 parallel jobs implies that throughput would not increase regardless of how many additional jobs are run in parallel. We

conclude that the use of XrootD for I/O intensive jobs or for many parallel CPU bound jobs without further work on tuning XrootD services or the underlying root I/O code is ill advised when operated over the Wide Area Network.

5 Summary and Conclusions

We describe our use of a Kerberized wide area Lustre filesystem to access CMS data remotely in an environment with realistic work flows over a 10 Gbps network. We present results that compare the I/O performance and time-to-completion of jobs running on a testbed deployed in two distinct ways: one where access to the remote storage is direct through public interfaces on each WN and another where access to the remote storage is bridged through XrootD services running on a single server with public and private 10 Gbps connectivity. We find excellent I/O performance of the testbed when configured in the direct Lustre mode even if the application is I/O intensive; seeing I/O rates comparable to the intrinsic capabilities of the Lustre storage system. I/O rates as high as 5.5 Gbps were easily obtained when running 192 simultaneous jobs.

Since we focused on testing the performance of a Lustre based solution to access data remotely we did not compare our results with the current XrootD only solutions being deployed by CMS and ATLAS. This is a simple extension of this work we will be exploring in the immediate future. The testbed was optimized to work over the 10 Gbps WAN. Further tuning of the XrootD Lustre configuration to improve the performance is likely possible.

From our results it is clear that from a performance, reliability and simplicity perspective it is reasonable to configure multiple sites to share storage and computing resources with the Lustre filesystem even if the sites are distributed geographically with sites separated by a few hundred kilometers; a potentially viable solution for regional distributed infrastructure.

References

[1] Bloom K *et al*. CMS Use of a Data Federation *Proc. of the CHEP 2013 Conference*, Amsterdam, NL October 2013, to be published in *J. Phys.:Conf. Ser.* available at https://indico.cern.ch/getFile.py/access?contribId=94&sessionId=6&resId=0&materialId=slides&confId=214784

[2] Vukotic I *et al*. Data Federation Strategies for ATLAS Using XrootD *Proc. of the CHEP 2013 Conference* (Amsterdam, NL October 2013) to be published in *J. Phys. Conf. Ser.* available at https://cds.cern.ch/record/1609593/files/ATL-SOFT-SLIDE-2013-836.pdf

[3] The Lustre 2 filesystem, available at http://wiki.whamcloud.com/display/PUB/documentation

[4] Bourilkov D *et al*. Secure wide area network access to CMS analysis data using the Lustre filesystem *Proc. of the CHEP 2012 Conference* (New York, NY May 2012) *J. Phys.:Conf. Ser*. **396** 032014, doi:10.1088/1742-6596/396/3/032014

[5] MIT Kerberos, available at http://web.mit.edu/kerberos.

[6] Thain D *et al.* Distributed Computing in Practice: The Condor Experience *Concurrency and Computation: Practice and Experience*, **17**, No. 2-4, pages 323-356, February-April, 2005, doi:10.1002/cpe.938

[7] Palencia J *et al.* Using Kerberized Lustre over the WAN for High Energy Physics Data *Proc. of the Lustre Users Group 2012 Meeting* (Austin, TX, April 2012) available at http://www.opensfs.org/wp-content/uploads/2011/11/lug2012-v20.pdf.

[8] Rodriguez J L *et al.* Wide area network access to CMS data using the Lustre filesystem *Proc. of the CHEP 2009 Conference* (Prague, Czech Republic March 2010), *J. Phys.: Conf. Ser.* **219** 072049, doi:10.1088/1742-6596/219/7/072049