Wide Area Filesystem Performance using Lustre on the TeraGrid

Stephen C. Simms, Gregory G. Pike, and Doug Balog

Abstract— Today's scientific applications demand computational resources that can be provided only by parallel clusters of computers. Storage subsystems have responded to the increased demand for high-throughput disk access by moving to network attached storage. Emerging Cyber-infrastructure strategies are leading to geographically distributed computing resources such as the National Science Foundation's TeraGrid. One feature of the TeraGrid is a dedicated national network with WAN bandwidth on the same scale as machine room bandwidth. A natural next step for storage is to export file systems across wide area networks to be available on diverse resources. In this paper we detail our testing with the Lustre file system across the TeraGrid network. On a single 10 Gbps WAN link we achieved single host performance approaching 700 MB/s for single file writes and 1GB/s for two simultaneous file writes with minimal tuning.

Index Terms— WAN file systems, Lustre, Cyberinfrastructure, TeraGrid

1 Introduction

One aspect of the growth and success of Grid computing is the geographically distributed nature of computing, data, and visualization. Simulation output from one location may be archived to a second location, post-processed at a third and then sent to yet more locations for visualization. This creates special challenges for data access. Grid computing has given users the ability to select the best resource for their task without regard to geographical location. High bandwidth interconnects are a prerequisite for smooth user interactions with such cyberinfrastructures. Users who have been accustomed to accessing their data through standard file system semantics are now being asked to learn new methods to move their data between resources. If distributed cyberinfrastructure is to accelerate scientific discovery to the greatest extent possible, we must create mechanisms for scientists to access their data in familiar ways: as though it were on a local disk.

Through the use of a wide area filesystem, we can permit scientists to access data remotely, as if it were mounted locally. Recent research at the San Diego Supercomputing Center (SDSC) [1] has shown that wide area filesystems are a feasible solution for sharing data between remote sites connected by a high-speed network, and the effort has been well received by users.

In November 2005, four TeraGrid sites began exploring using Lustre as a wide area file system. Indiana University (IU), Pittsburgh Supercomputing Center (PSC), Oak Ridge National Laboratory (ORNL), and the National Center for Supercomputing Applications (NCSA) began exporting, cross-mounting, and testing Lustre [3] file systems between the sites. Performance testing between IU and PSC resulted in demonstrations of the technology at the TeraGrid'06 conference. There it was shown that a host could achieve 90% of maximum theoretical performance for reads and writes through a 1 Gbs Ethernet connection [2,6]. Most recently, a team led by Indiana University with participants from ORNL and PSC demonstrated performance of Lustre across a wide area network as part of the Bandwidth Challenge at SC06, in which competitors were asked to get the most utility possible from a single 10 Gbps connection back to their home institution [7,8]. Using IU's recently completed Data Capacitor facility, an NSF-funded 535 TB Lustre storage cluster designed to provide short to mid-term storage [10], the team received Honorable Mention for their efforts. Tests run during the competition begged the question of using a single host with a 10 Gb Ethernet card for data transfer. This paper is an examination of sustained data transfer to a Lustre filesystem mounted across the TeraGrid network using a single 10 Gb client, and the subsequent feasibility of using Lustre as a wide area filesystem.

2 System Configuration

To test single host performance across the WAN, the Data Capacitor at IU Bloomington was mounted on a Lustre client at ORNL equipped with a single Myricom Myri-10G card across the 10 Gbps TeraGrid network connection (see Figure 1).

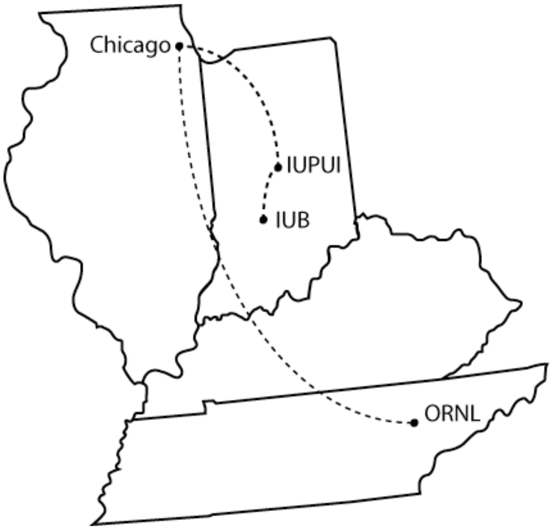

Fig. 1. The portion of the TeraGrid network used for testing

2.1 Client and Server Hardware

The Data Capacitor currently comprises 28 Dell 2950s with dual,

- *Stephen C. Simms, Indiana University, ssimms@indiana.edu*
- *Gregory G. Pike, Oak Ridge National Laboratory, pikeg@ornl.gov*
- *Doug Balog, Pittsburgh Supercomputing Center, balog@psc.edu*

dual core 3.0 GHz Xeon 5160 processors and 4GB of RAM. Two servers are used for cluster management and monitoring; two are used for Lustre metadata; and the remaining 24 are used for object storage, each equipped with a dual port Qlogic card and a Myri-10G card in Ethernet mode. The client used for testing had no Qlogic card, but was identical to the Data Capacitor object storage servers (OSSs) in every other respect.

2.2 Storage Hardware

Data Direct Networks (DDN) provides the storage backend for the Data Capacitor. Both metadata servers are directly attached to a DDN EF2800 Fibre Channel storage array. Six DDN S2A9550 storage couplets front 535 TB usable SATA disk and are attached to the OSSs via 4 Gb Fibre Channel (FC). The DDN controllers are configured to serve six 4TB logical units (LUNs) to each OSS, which function as Lustre object storage targets (OSTs). Each of the OSS's Qlogic ports function as a primary path for three OSTs (see Figure 2).

2.2 Operating System and Tuning

All servers were running a 2.6.9 RHEL 4 kernel patched for Lustre, which is freely available from Cluster File Systems [4]. The maximum values for tcp_rmem and tcp_wmem were set to 48MB on client and server, doubling the 24MB bandwidth delay product for a 10 Gb connection across the 19 ms round trip time (RTT) separating IU and ORNL. Version 1.1.0 of the myri10ge driver was installed; card performance was significantly improved with tcp segment offload (tso) off.

2.3 Lustre Configuration

Lustre 1.4.7.1 was running on all servers at the time of testing. Lustre debug (on by default) was disabled because of its adverse effect on read performance. Lustre’s ksocklnd module was configured to disable irq affinity in order to use each cpu socket’s 1333 MHz front side bus. Lustre's max_rpcs_in_flight parameter has a significant effect on performance across the WAN, though for these tests we chose not to change the default value of 8.

Tests used a 356 TB filesystem created on 16 OSSs for a total of 96 OSTs which could be considered stripes (see Figure 2). Lustre permits the user to specify striping attributes for files and directories. For a 10 Gb client writing to the Data Capacitor this is particularly important, because in the current configuration a single stripe file can be written no faster than the speed of a single 4 Gb FC port. In order for a 10 Gb client to realize its full potential, at least three stripes would be required. To maximize throughput, the Data Capacitor's striping order is across servers so that each additional stripe offers an additional FC port as seen in Figure 2.

3 METHODOLOGY

3.1 Testing

Specific testing goals were to measure sustained read and write transfer rates for a single file varying block size from 16KB to 2MB for stripe counts of 1, 2, 4, 8, and 16; and to measure sustained read and write transfer rates for two simultaneous file transfers varying block size from 16k to 2M for stripe counts of 1, 2, 4, 8, and 16. For the sake of consistency, each stripe configuration tested contained the same OSTs from trial to trial. For consistency between stripe configurations, larger stripe count tests contained the smaller stripe counts as subsets.

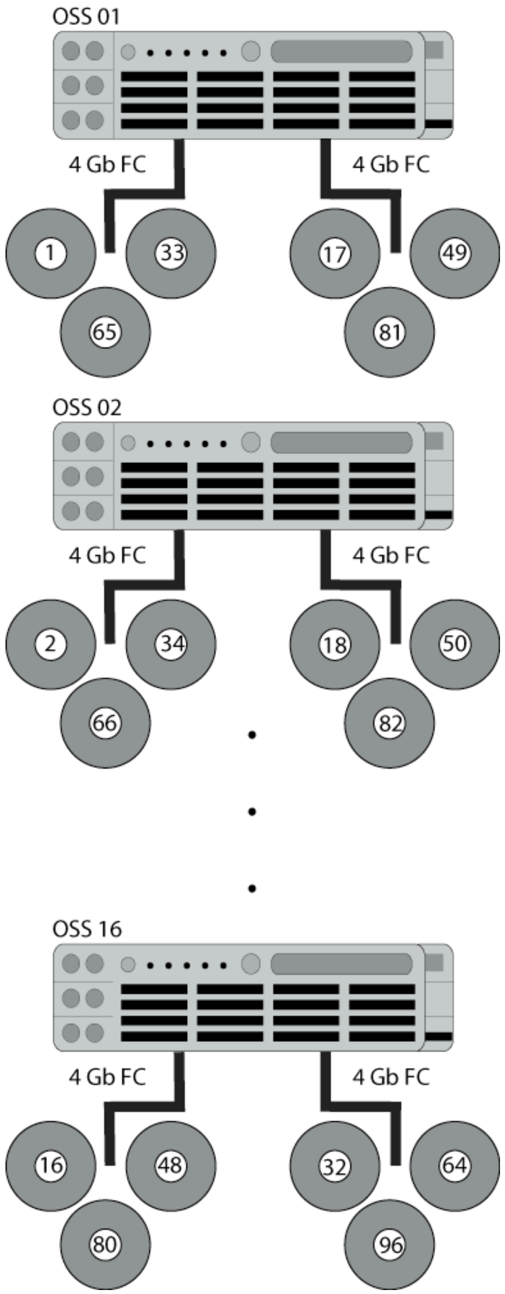

Fig. 2. Diagram of OST layout across the 16 OSSs used for testing

3.2 Tools

Measurements were made with the mib test program, developed at Lawrence Livermore National Laboratory (LLNL) to measure and analyze the performance of large Lustre clusters [9]. Mib utilizes MPI to synchronize multiple serial tasks reading and writing. For testing, LAM/MPI [5] was run on a single client machine with the host file recording a cpu count of 4 (one for each processor core).

An mib user has to specify a count of system calls, a block size for each call, and a time limit. The program stops either when it has completed the system calls or when the time limit has been reached. In the event that there are timing variations in multi-task runs, mib uses the start time of the task that began first and the end time of the task that finished last providing a "worst case scenario" result.

To measure sustained performance, mib was run in "stonewall" mode where time is the limiting factor and the number of system calls specified is very high; for our measurements, 9,000,000. Because of Lustre's aggressive client side caching, obtaining an accurate measure of sustained performance requires writing and reading a file at least twice the size of the client's RAM. For this reason, a time value of 60 seconds was chosen for all runs so that even the slowest measured transfer rates would exceed the necessary 8GB.

4 Results

4.1 Single File Writes and Reads

Figures 3 and 4 show performance of single file writes and reads. Mib was run for 60 seconds on a single processor core of the client node. Transfer rates were measured for block sizes ranging from 16KB to 2MB for 5 different striping patterns. Three trials of each measurement were taken and the mean aggregate transfer rates were then plotted to visualize the results.

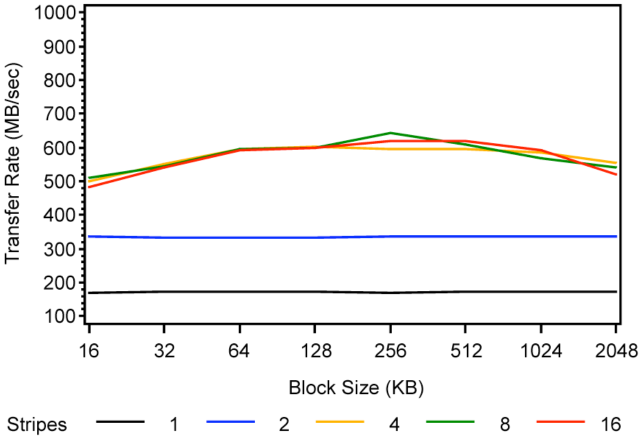

Fig. 3. Average rates for single file writes from ORNL.

The fastest measured single file write was 682MB/s using 8 stripes with a block size of 256KB while the fastest single file read was 455 MB/s using 8 stripes and a block size of 128KB.

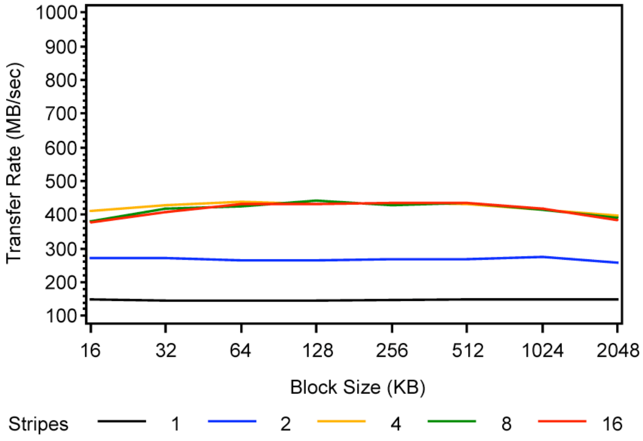

Fig. 4. Average rates for single file reads from ORNL.

4.2 Simultaneous Two File Writes and Reads

Figures 5 and 6 show performance of two simultaneous file writes and reads. Mib was run for 60 seconds on two processor cores (one from each physical processor) of the client node. Transfer rates were measured for block sizes ranging from 16KB to 2MB for 5 different striping patterns. Three trials of each measurement were taken and the mean aggregate transfer rates were then plotted to visualize the results.

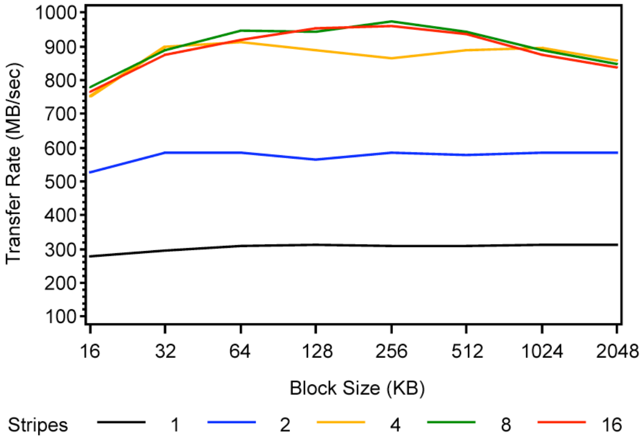

Fig. 5. Average rates for two simultaneous file writes from ORNL.

The fastest measured two file write rate was 977 MB/s using 8 stripes and a block size of 256 KB, the fastest read rate was 596 MB/s with 16 stripes and a block size of 256 MB.

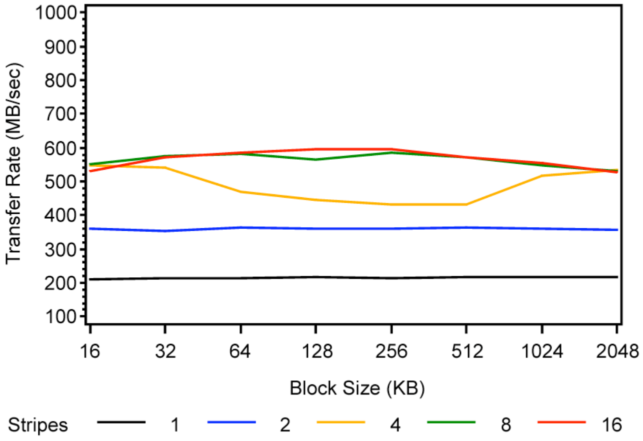

Fig. 6. Average rates for two simultaneous file reads from ORNL.

4.3 Local vs. Remote Performance

Tests described in sections 4.1 and 4.2 were performed on an identical Lustre client local to the Data Capacitor (connected to the same network switch). The best-performing configurations (local and remote) were compared and the results are shown in figures 7 and 8.

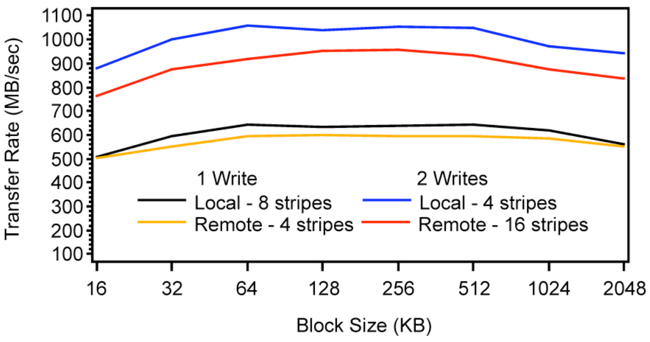

Fig. 7. Transfer rates of the best-performing stripe configurations for one and two file writes.

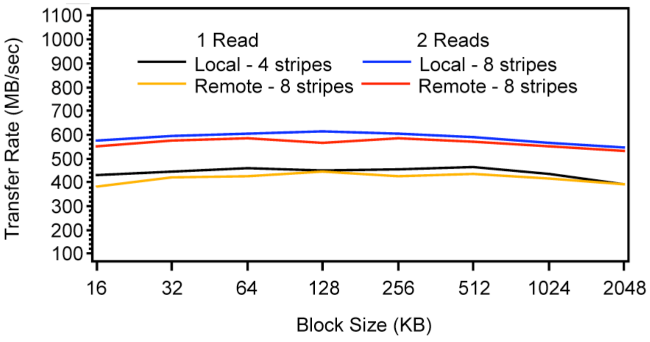

Fig. 8. Transfer rates of the best-performing stripe configurations for one and two file reads.

5 Discussion

In the test results we observe write rates approaching 80% of the theoretical maximum network bandwidth when writing two files simultaneously. We attribute this superior aggregate performance to the client's dual front side busses. Simultaneous reads also improve the transfer rate compared to single file reads, but not as significantly. The remote rates observed are comparable to those achieved from a local Lustre mount, demonstrating the minimal impact of the distance on performance.

Figures 3-6 show that maximum performance can be achieved between approximately 4 and 8 stripes. Additional stripes appear to be of little additional benefit except possibly in the two file write case. The performance achieved for one and two stripe transfers show that for these cases the limiting factor is the striping pattern.

Transfer rate is approximately independent of block size between 64KB and 512KB, with the exception of the anomalous four stripe result for two file transfers. The four stripe tests were repeated over time for this case and the anomaly was reproduced. The mechanism generating this anomaly is not understood at this time and bears further investigation.

Writing to stripes from a client is a "one to many" operation – like dealing playing cards – while reading from stripes is like collecting those cards and ordering them properly. Diminished read performance (compared to write performance) could be caused by this coalescing overhead on the client, as we see a large change in cpu usage between 1 and 2 stripe reads. The disparity between reads and writes may also be associated with the number of internal memory copies required for the respective operations – reads require at least one more memory copy than writes because reads must move coalesced data from kernel space into user space. At present, we are working with Cluster File Systems to determine the exact causes of the disparity and find a way to improve read performance.

6 Conclusions

Using Lustre as a wide area filesystem has the potential to provide users with outstanding transfer rates across significant distance while providing a familiar interface. With a single client and only two small changes to Lustre (debug off, irq affinity off), we have demonstrated sustained aggregate write performance that approaches 80% of the theoretical maximum. Aggregate read rates were somewhat smaller but rates of nearly 600 MB/s were achieved. The performance achieved using Lustre across the WAN was comparable to performance of a locally mounted Lustre file system and significantly faster than local disk performance. Increasing max_rpcs_in_flight could further increase performance, and an examination of the variable's effect would make an excellent topic for further study.

ACKNOWLEDGMENTS

The authors wish to thank John Cobb, Larry L. Simms, Scott Teige, Bret Hammond, Malinda Lingwall, Yu Ma, Randall Bramley, Kit Westneat, -Matt Link, Craig A. Stewart, and Ana Maguitman.

This material is based upon work supported by the National Science Foundation under the following programs: Terascale Extensions and TeraGrid Operations (Extensible Terascale Facility).

Support for the Data Capacitor project provided by the National Science Foundation is gratefully acknowledged and appreciated (CNS-0521433).

REFERENCES

[1] P. Andrews, P. Kovatch, C. Jordan, “Massive High-Performance Global File Systems for Grid Computing” in Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference.

[2] D. Balog, J. Huffman, G. Pike, S. C. Simms “Lustre Wan Demo” (demonstration given at TeraGrid ’06 conference, Indianapolis, IN, June 12-15, 2006).

[3] Peter J. Braam, Lustre: A Scalable, High Performance File System, 2002. http://www.lustre.org/docs/whitepaper.pdf.

[4] Download Lustre. http://www.clusterfs.com/download.html

[5] LAM/MPI Parallel Computing. http://www.lam-mpi.org/

[6] S. C. Simms, B. Hammond, M. Link, C. Stewart “The Data Capacitor Project” (demonstration given at TeraGrid ’06 conference, Indianapolis, IN, June 12-15, 2006).

[7] S. C. Simms, M. Davy, B. Hammond, M. Link, C. Stewart, R. Bramley, B. Plale, D. Gannon, M. Baik, S. Teige, J. Huffman, R. McMullen, D. Balog, G. G. Pike, “Bandwidth Challenge---All in a Day's Work: Advancing Data-intensive Research with the Data Capacitor” in Supercomputing, 2006. Proceedings of the 2006 ACM/IEEE conference on Supercomputing.

[8] SC06 – Conference – HPC Bandwidth Challenge. http://sc06.supercomputing.org/conference/hpc_bandwidth.php

[9] SourceForge.net: mib. http://sourceforge.net/projects/mibtest

[10] C. Stewart, R. Bramley, C. Pilachowski, B. Plale, T. J. Hacker “MRI: Acquisition of a High-Speed, High Capacity Storage System to Support Scientific Computing: The Data Capacitor". http://www.nsf.gov/awardsearch/showAward.do?AwardNumber=0521433