Exploration of Parallel Storage Architectures for a Blue Gene/L on the TeraGrid

Michael Oberg[2], Henry M. Tufo[1,2], and Matthew Woitaszek[2]
oberg@ucar.edu

[1] University of Colorado, Boulder, CO
[2] National Center for Atmospheric Research, Boulder, CO

Abstract. This paper examines the construction of a cost-effective storage cluster for an IBM Blue Gene/L supercomputer on the TeraGrid. The performance of storage clusters constructed using commodity Opteron servers and directly-attached fibre-channel RAID systems is analyzed utilizing the GPFS, Lustre, and PVFS file systems. In addition to traditional client scalability results, server scalability is also examined, identifying the number of servers required to saturate a 10 Gbps infrastructure network uplink. The results highlight test and system configurations that provide scalable throughput and identify configurations where the environment restricts scalability. Moreover, the Blue Gene/L's multiplexed I/O design introduces an additional point of contention that affects each file system to varying degrees.

1 Introduction

Joining the TeraGrid has led to new demands on NCAR's storage systems. Where users were formerly satisfied with independent storage systems attached to individual supercomputing systems, local capacity and performance are no longer sufficient. Groups of collaborators expect high-throughput access to shared storage space accessable natively on-site and through GridFTP from other institutions. These requirements have accelerated NCAR's need for and adoption of collaborative storage.

Our objective is to construct an inexpensive storage cluster for use with NCAR's TeraGrid resources. The primary computational resource is a single-rack IBM Blue Gene/L (BG/L) system called Frost, and additional resources include visualization systems and Web-based portals to NCAR's mass storage system. For our evaluation, cost includes factors such as hardware, software, and administrator time. Not only must the system components be affordable, but the file system deployment process must be sufficiently robust to avoid large time committments in response to common system administration activities.

To decouple our storage resources from individual computer system lifecycle planning, we are interested in multi-cluster architectures. In this design, the storage cluster is constructed as a resource onto itself, attached to a high-throughput network, and shared with multiple computational resources. This approach allows storage system growth to be addressed independently and eliminates data

migration between computational systems, but also introduces complexity as the file system software must be compatible with multiple client architectures and operating systems.

In this paper, we evaluate three parallel file systems (GPFS, Lustre, and PVFS) on two direct-attached fibre-channel storage platforms (Xyratex 5402 and DDN 9550) as well as the legacy IBM GPFS FAStT storage systems that are more tightly coupled with the BG/L. In addition to the common file system vs. storage system testing regime, we also evaluate *server scalability* and examine the number of servers and storage systems required to meet throughput requirements and minimize the resultant cost. Each file system is installed and configured with limited tuning as described in the product's installation guide, providing measurements that describe system performance for a standard out-of-box experience. The popular IOR benchmark is used to identify peak performance, and a custom NCAR application benchmark evaluates MPI-IO performance.

The remainder of this paper is organized as follows: Section 2 presents related storage system benchmarking efforts. Section 3 describes our deployment environment and corresponding storage system architectures. Section 4 provides results of our benchmarking activities, and the paper concludes with our recommendations for storage cluster deployment.

2 Related Work

Storage system performance benchmarking is a common component of any supercomputer system deployment activity. Most relevant to this work is characterication of the I/O capabilities of the Blue Gene/L with GPFS by IBM and Argonne National Laboratory [10]. Similarly, Sandia National Laboratories' investigation of storage system scalability for RedStorm demonstrated component-based throughput evaluation for one of the storage systems also examined here [8].

We have also previously worked with shared parallel file systems in heterogeneous multi-cluster environments. In 2005, we analyzed the functionality and performance of Lustre, PVFS, and GPFS in an environment with both x86 and PowerPC clients [7]. This environment was then used to provide a root file system for diskless booting directly from a parallel file system [6]. This work extends our prior experience with file systems in heterogeneous environments by examining current product offerings intended for a larger-scale production environment.

3 Design

For our depoloyment, we have chosen to use a multi-cluster architecture so that the storage system is an independent resource designed to be shared with multiple systems. The storage cluster is placed within NCAR's TeraGrid environment as a second-tier infrastructure system (see Figure 1). NCAR's central network provides one 10 Gbps connection to our Force10 infrastructure switch. The BG/L

system and its front-end nodes connect directly to this switch using aggregated 1 Gbps links, and the storage cluster is connected through a 10 Gbps uplink. This ensures that the system is close to the high-performance computational resources; i.e., local supercomputer file system traffic does not reach the routers controlling external network connectivity. In addition, NCAR's TeraGrid connection is at 10 Gbps, so this allows us to replicate situations where we could saturate the outgoing connection (but with the lowest-latency connection possible). Our results indicate that it is possible for the BG/L system (with a total of 32 x 1 Gbps Ethernet connections) to saturate the 10 Gbps uplink, but only for specific file system, hardware, and test combinations – the average throughput remains much less.

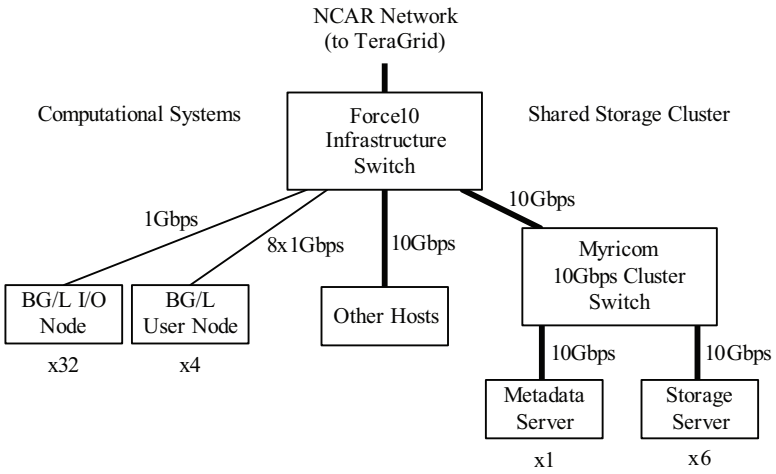


Fig. 1. Target system architecture

Common storage system design features include throughput, total capacity, incremental scalability, and fault tolerance. To minimize the cost of our system, we use a simple design strategy: identify the hardware required to meet throughput goals (in our case, saturating a 10 Gbps uplink) and then deploy additional disks as required to reach the desired capacity. The system features a modular design such that additional server and RAID pairs can be added at a later time to increase throughput, or additional disk expansion units can be attached to the current RAID controllers to increase capacity.

3.1 Target Storage Platforms

For the analysis of the selected parallel file systems, we chose a variety of underlying storage hardware:

– Xyratex 5402

  The Xyratex-based storage cluster is comprised of six storage servers. Each server is connected to two single-controller 24-disk Xyratex RS-1220-F4-5402E (5402) RAID enclosures. The 5402 storage enclosures house twelve 500GB SATA disk drives in a main chassis and connect to a 12-disk expansion unit through a four-lane SAS expansion port. These twelve 24-disk RAID enclosures have a single RAID controller with an 800MHz RAID processor and 1 GB of battery-backed cache. Each storage host connects to two of these RAID controller systems through a PCI-Express dual-port LSI Logic FC949X FC HBA.

  For the storage servers, each Xyratex 5402 provides four 6-disk RAID5 arrays using a chunk size of 64KB. A pair of high-availability metadata servers are both connected to a pair of 12-disk dual-controller Xyratex 5402 enclosures, and use a single 6-disk RAID10 array per file system with a chunk size of 128KB. In this configuration, each storage server controlls 8 LUNs and provides an aggregate raw write performance of up to 540MB/s. Thus, without file system overhead, two storage servers are capable of achieving our target throughput of saturating a single 10Gbps network link.

– DDN 9550

  A DDN 9550 installation comes in a "couplet" configuration with two active-active RAID controllers that provide a total of eight FC or Infiniband host ports and 20 FC back-end disk connections and up to 960 disks. Our system was configured with the eight 4 Gbps FC host ports, and five of the SAF4248 disk chassis for a total of 240 disk drives. The disks were configured in twenty-four 10-disk RAID6 arrays. The arrays were configured to use disks in stripes throughout the five disk chassis, which provides the most fault-tolerant and highest-performance configuration for the given number of disks.

– Terascala RTS 1000

  The Terascala RTS 1000 system is an integrated storage appliance that provides a drop-in Lustre solution. An 8U Terascala chassis contains 5 server blades and an integrated Linux-based management module. Each blade has 8 hot-swap disk drives with integrated hardware RAID and supports dual 1Gbps Ethernet interfaces as well as an optional 10Gbps Ethernet or Infiniband interface. Intended for the construction of storage clusters with minimal administrative effort, the Terascala solution comes preconfigured with a current version of Lustre.

  Our tested configuration consisted of 2 Terascala chassis running Lustre 1.6.4-2. One blade was allocated as the Lustre MGS/MDS, and the 9 OSS

blades provided a total of 30TB usable storage. The blades were connected to our network using a single 1Gbps Ethernet interface.

– IBM FAStT-500 SAN and Direct-Attached IBM FAStT-900

Our IBM Blue Gene/L system installation includes two IBM storage systems: a set of dual-controller FAStT-500 SAN-connected disk subsystems and a FC-AL connected pair of dual-controller FAStT-900 disk subsystems, both with 10,000 RPM 73GB SCSI disks. We include the performance results of these systems for comparison purposes.

The FAStT-500 system consists of nine dual-controller storage systems each with four 10-disk drawers. Each dual-controller chassis provides 7 RAID5 (4+p) LUNs and a hot-spare pool of 5 disks, for a total of 360 disks with 17.2 TB of usable space. The storage controllers connect to a pair of Brocade fibre channel switches, and a Linux multipath configuration provides balanced and high-availability I/O access between the two paths to any LUN. The four Blue Gene/L I/O servers connect to the FAStT-500 SAN switches via eight 2 Gbps fibre channel links.

The FAStT-900 system consists of two dual-controller storage system each with five 14-disk drawers. Configured with 8+p LUNs, the system contains a total of 140 disks and a 5.97 TB usable space. The controllers are wired using dual arbitrated loops providing a path fault-tolerant configuration and a total of 16 2Gbps fibre channel links between the four Blue Gene/L I/O servers and the storage.

For all benchmarks, the Xyratex and DDN arrays are tested using 2, 4, or 6 Opteron servers in a dedicated storage cluster. The FAStT-500 and FAStT-900 storage systems use the Blue Gene/L front-end nodes as GPFS servers, and the Terascala RTS 1000 system is directly accessed by the BG/L clients.

3.2 Parallel File Systems

The three parallel file systems that we selected for this analysis are GPFS [1], Lustre [3], and PVFS2 [4]. GPFS is a commercial product offered by IBM, is commonly run on IBM supercomputer and cluster systems, and is perceived as a production-level general-purpose file system. Lustre competes with GPFS as the file system of choice on some of the largest supercomputers in the world. As an open-source parallel file system, it supports a wider range of computer and network architectures than GPFS. PVFS is commonly used as high-performance scratch space on Linux clusters and features native MPI-IO support through the popular MPICH software package. These file systems were chosen for their wide use in HPC environments, our familiarity with their installation and configuration, and their support for the BG/L system.

- The GPFS file system was configured using the GPFS native client for BG/L provided by IBM using the multicluster capability of GPFS. The servers ran GPFS 3.1-13, and the BG/L I/O nodes used GPFS 3.1-8. All GPFS file systems used the following options: `socketSndBufferSize` and `socketRcvBufferSize` were set to 1048576, and `maxMBpS` was set to 500.
- Lustre 1.6.2 was used on the clients and the servers, with the `localflock` client mount option specified as required for MPI-IO functionality. Striping was configured on each target directories to stripe over all available storage targets using the default (1 MB) stripe size.
- PVFS 2.6.3 was used on the clients and the servers, with striping manually configured such that subsequent stripes accessed subsequent servers. The `TCPBufferSend` option was set to 524288, and `TCPBufferReceive` was set to 1048576.

For all configurations, the underlying servers were configured to use larger TCP buffers (min=4096, default=87380, max=67108864), and for all devices the SCSI read_ahead_kb parameter was increased from 128 to 1024. Both Lustre and PVFS were installed using the ZeptoOS 1.4 I/O node kernel [5], while GPFS used the IBM native I/O node kernel.

3.3 Blue Gene/L I/O Environment

The Blue Gene/L system allocates computational resources in discrete partitions with the restriction of having at least one BG/L I/O node per partition. On our BG/L, each I/O node serves 32 computational nodes. When running on a BG/L machine, I/O from the compute nodes is sent over the tree network to the nearest I/O node. Each I/O node, which has the same underlying hardware as the compute nodes but provides a single-processor Linux environment, mounts the parallel file system using a native client. A compute node I/O daemon (CIOD) user process is started on each I/O node and runs as the user ID of the user who submitted the job. The CIOD maintains a pool of connections to the tree network and processes all of the I/O, proxying data requests between the tree and Ethernet networks. From the parallel file system's perspective, each I/O node and the associated pool of compute nodes appear as a single client, and all client activity appears to originate from CIOD on each I/O node. CIOD loops over the pool of tree connections to provide reasonable I/O latencies, but this design sacrifices throughput by polling all nodes, including those which may not have outstanding I/O requests.

4 Results

When performance comparisons are a metric for evaluating system designs, the appropriate selection of benchmark software becomes critical. With the large variety of available benchmarks (Rajeev Thakur maintains a list of 18 popular parallel I/O benchmarks [9]), it is important to choose appropriate tests that

define a baseline performance for comparison and also represent the anticipated application workload. A prior analysis of high-performance I/O for the Blue Gene/L supercomputer conducted by IBM and Argonne National Laboratory used IOR, ROMIO's collective I/O, NAS BT with checkpointing, the FLASH2 I/O benchmark with HDF5 and PnetCDF, and NCAR's HOMME model [10]. Their results demonstrated that among the benchmarks, LLNL IOR was capable of producing the highest throughput measurements, with the remainder of the application benchmarks describing a more representative user experience.

Since our testing matrix consists of three file systems on four hardware platforms, three additional separate file systems, and configuration variations examining the effect of BG/L I/O node multiplexing, selecting a minimal yet useful benchmarking suite was of utmost importance. We selected IOR [2] for aggregate peak throughput measurement because of its wide use and support for a variety of file access methods, and an in-house benchmark POPIO to measure MPI-IO application performance. The use of IOR allows easy comparisons to other published results, and POPIO is one of NCAR's most scalable I/O codes.

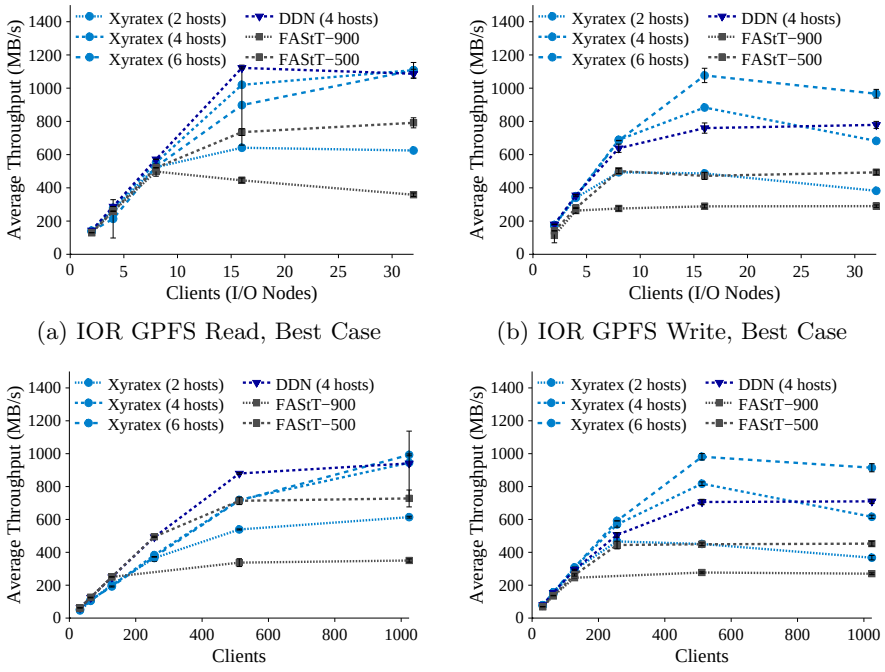
4.1 IOR Benchmark Results

We used the IOR benchmark to evaluate each configuration's maximum throughput. For each configuration, we ran specific tests to examine throughput from two perspectives: maximum BG/L parallelism with all 32 tasks per I/O node to a single shared file, and maximum client parallelism with 1 task per I/O node to independent files. The first test set demonstrates a worst-case scenario: it uses the maximum number of tasks per I/O node, is subject to contention at the I/O node, and exercises locking (if available) to mediate writes to a single shared file. The second test set illustrates the best performance possible, as it eliminates I/O node contention and accesses independent files with no byte-range locking requirements.

Each file system is tested using storage clusters consisting of 2, 4, and 6 hosts with Xyratex storage and 4 servers using DDN 9550 storage. The legacy FAStT-500 and FAStT-900 storage systems are tested using GPFS, and the Terascale is tested using Lustre.

GPFS: The GPFS file system provided the best performance with an out-of-box install for the IOR benchmark, under both test cases (see Figure 2). For reads, the larger configurations (4 and 6 servers) scale linearly and rapidly through 16 BG/L partitions, and then peak at the 10 Gbps network throughput. In terms of server scalability, clear distinctions are visible between the 2, 4 and 6 node configurations. Moreover, the best and worst case results are similar. The performance of GPFS under I/O node contention and single-file write access is significantly better than the other file systems, and shows much less performance degradation from the best-case single task per I/O node and multiple file case than the other file systems.

Lustre: The results from the Lustre file system (see Figure 3) show reasonable performance for the larger storage cluster sizes for the best-case scenario, with near linear scalability under reads. However, with I/O node contention



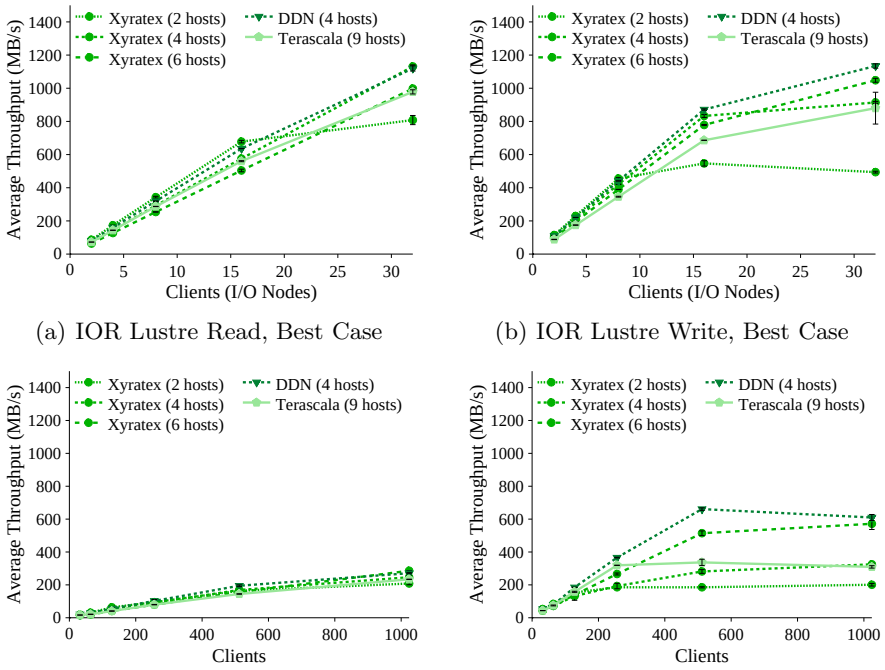
(a) IOR GPFS Read, Best Case

(b) IOR GPFS Write, Best Case

(c) IOR GPFS Read, Worst Case

(d) IOR GPFS Write, Worst Case

Fig. 2. IOR read and write throughput for GPFS: Best case plots 2(a) and 2(b) show maximum I/O client parallelism with 1 task per I/O node writing to independent files, and worst case plots 2(c) and 2(d) show maximum BG/L parallelism with 32 tasks per I/O node writing to a shared file.



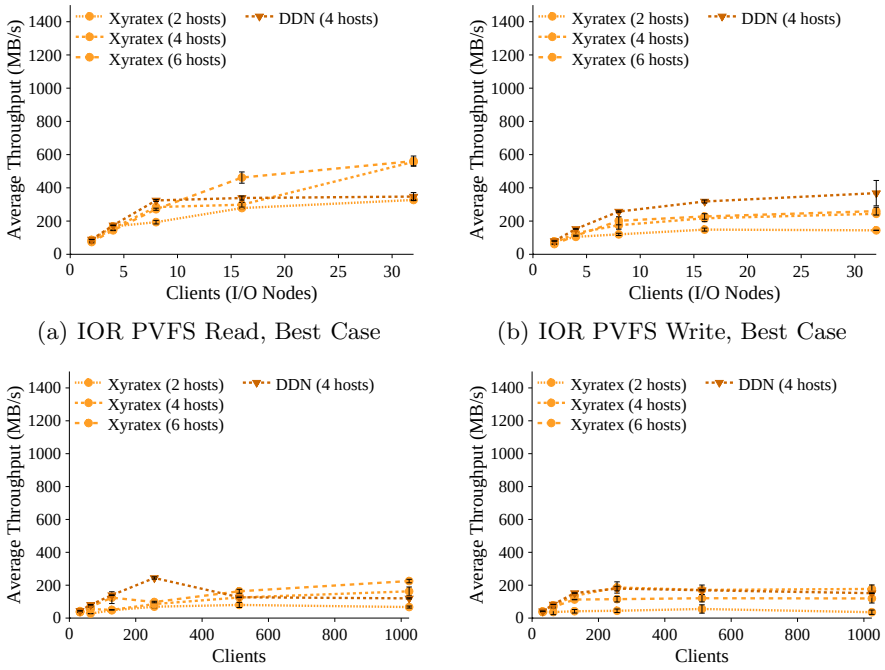
(a) IOR Lustre Read, Best Case

(b) IOR Lustre Write, Best Case

(c) IOR Lustre Read, Worst Case

(d) IOR Lustre Write, Worst Case

Fig. 3. IOR read and write throughput for Lustre: Best case plots 3(a) and 3(b) show maximum I/O client parallelism with 1 task per I/O node writing to independent files, and worst case plots 3(c) and 3(d) show maximum BG/L parallelism with 32 tasks per I/O node writing to a shared file.



(a) IOR PVFS Read, Best Case

(b) IOR PVFS Write, Best Case

(c) IOR PVFS Read, Worst Case

(d) IOR PVFS Write, Worst Case

Fig. 4. IOR read and write throughput for PVFS: Best case plots 4(a) and 4(b) show maximum I/O client parallelism with 1 task per I/O node writing to independent files, and worst-case plots 4(c) and 4(d) show maximum BG/L parallelism with 32 tasks per I/O node writing to a shared file.

and single-file access, the performance is reduced dramatically. Under the latter scenario, peak read performance drops from 1137 MB/s to 273 MB/s.

PVFS: PVFS achieved the lowest aggregate throughput and least client scalability for the IOR benchmark (see Figure 4). The maximum throughput achieved was 424 MB/s (write) and 593 MB/s (read), which is less than half the peak values seen for the other two file systems. Although the aggregate performance is less than the other file systems, PVFS provides much more consistent performance between the two test cases.

4.2 Blue Gene/L I/O Node Contention

The selection of a BG/L system as a file system client introduces an additional level of complexity over simple Linux clusters. Whereas a typical Linux client has its own dedicated network connection and client scalability can be measured by the number of clients, the BG/L system multiplexes file system operations through I/O nodes. These I/O nodes are a point of possible contention.

The IOR measurements in the previous section (4.1) examine both extremes of operation, where I/O nodes are multiplexed with 32 BG/L clients and also where I/O nodes serve only a single BG/L client. To determine the effect of the BG/L I/O multiplexing on performance, we analyzed the aggregate throughput of a single I/O node with an increasing number of clients.

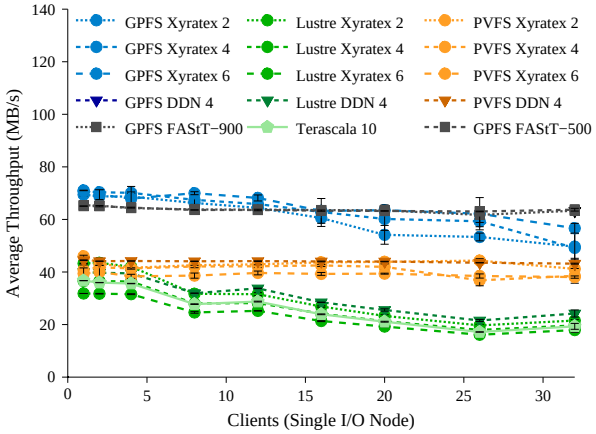
In general, aggregate throughput decreases with an increasing number of clients on a single I/O node (see Figure 5). This effect is clearly visible with reads under GPFS and Lustre, and to a lesser degree for all file systems for writes. For example, GPFS reads decrease from 66 to 46 MB/s, Lustre reads decrease from 40 to 20 MB/s, and PVFS reads remain constant around 43 MB/s.

It is particularly interesting that the file systems provide clearly stratified performance regimes and that these regimes are not consistent. GPFS provides the best performance for both reads and writes, whereas PVFS provides higher read throughput than Lustre, and Lustre provides higher write throughput than PVFS. This data also shows the maximum performance available from any file system for a single 32-processor partition. GPFS provides the best aggregate performance because it has almost double the single-partition performance of Lustre and PVFS, as the base partition performance defines the aggregate throughput to the system.

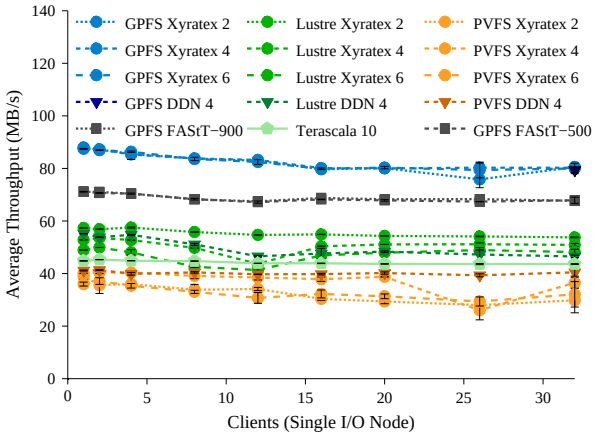
One substantial difference between the GPFS configuration and the PVFS and Lustre configurations is the I/O node kernel: GPFS uses IBM's I/O node kernel, and PVFS2 and Lustre were built using ZeptoOS, although both I/O node distributions are based on the Linux 2.6.5 kernel and use the same site-customized sysctl settings.

4.3 POPIO Benchmark Results

POPIO is a benchmark that estimates the I/O requirements of a high resolution ocean model. In particular, it measures the rate at which multiple double precision 2D arrays of size 3600x2400 are written and then read from disk. Each 2D



(a) IOR Read for a Single BG/L I/O Node



(b) IOR Write for a Single BG/L I/O Node

Fig. 5. IOR read and write throughput for GPFS, Lustre, and PVFS for all hardware configurations running on a single BG/L I/O node writing to independent files. This shows resource contention as additional tasks require the services of a I/O node.

array is decomposed across processors in block-cyclic form. For example, for a 500 processor run, each processor contains a tile of size 144x120. Each processor uses MPI-IO to write its tile of data to the correct location on disk.

The POPIO benchmark's I/O access pattern is similar to the worst case IOR benchmark where each task performs I/O and all tasks share a single file, but with much smaller and interleaved requests through the MPI-IO interface. We were surprised that the POPIO performance was substantially different than the IOR results (see Figures 6, 7, and 8). For example, while GPFS had the best overall performance using IOR, it had the worst write performance for POPIO.

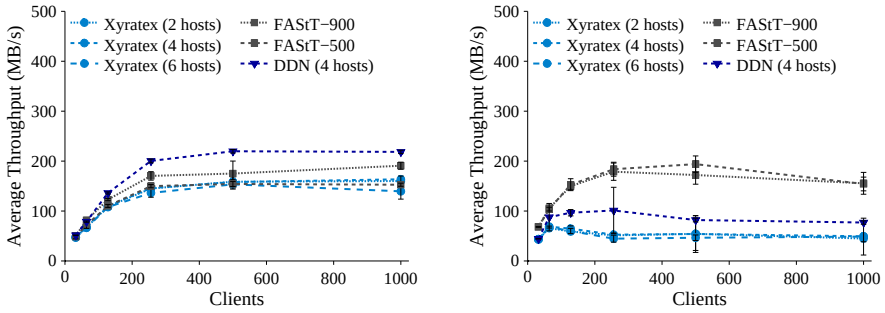
With Lustre, writes significantly outperformed reads, and using PVFS reads outperformed writes. This result is consistent with the IOR results as a general case, but while the consolidated results show clear trends for clusters of various sizes, comparing clusters of specific sizes produces divergent conclusions. For example, our two node Lustre storage cluster achieved the slowest IOR write performance of the tested Lustre configurations, but achieved the highest POPIO write throughput. The GPFS file systems running on the two FAStT storage clusters also exhibited this same divergent behavior, performing much better under POPIO than the other GPFS file systems, where under IOR the opposite was true.

4.4 File System Administrative Complexity

One area of interest is the complexity and time required to install and configure the parallel file systems. While Lustre has historically required a high installation overhead including substantial kernel-level maintenance, recent versions have substantially reduced the administrative burden. Once the operating system environment has been established, Lustre is painless to install and configure. The administrator creates the individual server backing store file systems, starts the servers, and starts the clients. GPFS is also a stable and robust software product, but its installation and configuration requires many more steps. The cluster, storage devices, file systems, and multi-cluster file system connections must be established independently. PVFS has a slightly more labor-intensive installation. While the installation scripts provide a basic starting point sufficient for small clusters, slightly more advanced configurations such as using multiple SCSI devices per storage host require manual configuration file manipulation and distribution.

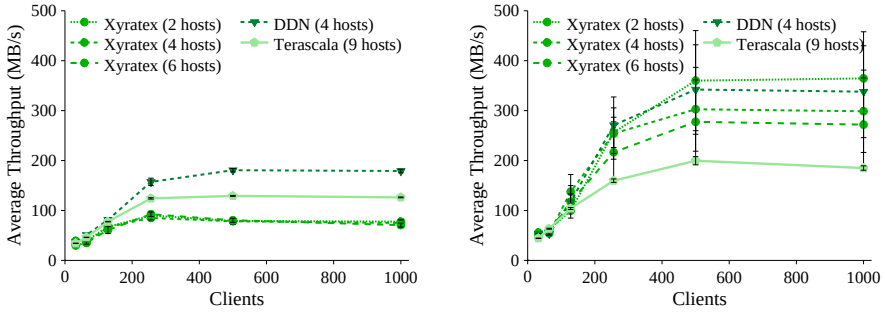
5 Discussion

Overall, our storage cluster construction experience was successful. Both the Lustre and GPFS file systems were able to achieve near 10Gbps line rate performance using 4 or 6 servers under maximum throughput testing. However, our results demonstrate that each of these three file systems have wildly divergent performance, even when running on identical hardware and similarly



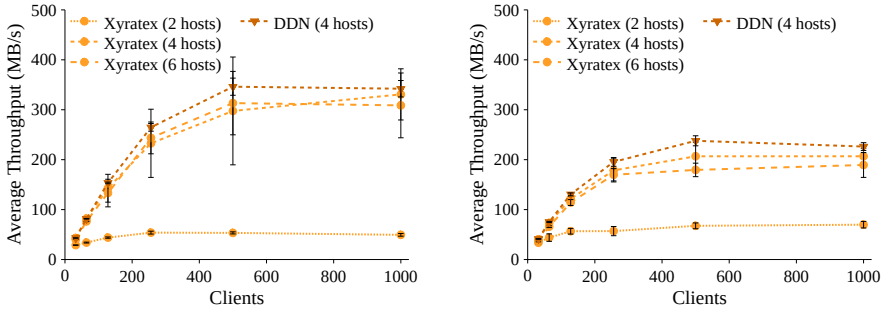
(a) POPIO GPFS Read (b) POPIO GPFS Write

Fig. 6. POPIO read performance and write performance on selected storage systems using the GPFS file system



(a) POPIO Lustre Read (b) POPIO Lustre Write

Fig. 7. POPIO read performance and write performance on selected storage systems using the Lustre file system



(a) POPIO PVFS Read (b) POPIO PVFS Write

Fig. 8. POPIO read performance and write performance on selected storage systems using the PVFS file system

tuned operating systems. Furthermore, each file system and hardware configuration can provide dramatically different performance for different benchmark and read/write situations. This variability suggests further avenues for exploration.

Of the file systems we tested, GPFS provides the highest aggregate throughput and most consistent performance between our BG/L test cases. However, the non-FAStT tests performed rather poorly at the POPIO tests. Unfortunately, this sole data point does not immediately suggest a course for improvement. The high performance seen on the IOR tests suggests that the servers are functioning properly, and the FAStT results indicate higher possible client performance. These cases run on substantially different server hardware (OpenPower 720 vs. Opteron), and we are attempting to identify a test case that may provide more data regarding fine-grained shared write performance.

Lustre provides highly scalable read and writes for test cases with no BG/L I/O node contention. For cases with contention, read performance drops substantially, and write performance less so. The favorable performance on the contention-free cases suggests that the hardware and file system are not at fault. Because the performance degradation is present only in reads with I/O node contention, we suspect that this issue arises because of interactions between the Lustre client and I/O node CIOD (e.g., non-optimal use of memory for readahead caching on the client). We intend to work with Sun to identify possible solutions.

PVFS provided rather poor performance when measured with IOR. From the GPFS and Lustre tests, we are confident that the underlying storage and server hardware can provide much better performance. However, PVFS excelled at fine-grained MPI-IO access with our POPIO benchmark.

6 Future Work

To more fully explore the performance of these parallel file systems in our environment, we intend to perform additional tuning and analysis. The primary performance constraint on the Blue Gene/L architecture is the I/O node architecture and its reliance on the CIOD daemon. As a complement to ZeptoOS, Argonne National Laboratory has developed the ZeptoOS I/O Daemon (ZOID) that attempts to increase I/O throughput with a multithreaded design. Having established performance baselines, we intend to deploy ZOID with all of the file systems, and anticipate higher performance from both PVFS and Lustre under the high contention scenarios.

We also intend to further explore using the Terascala RTS 1000 appliance while providing a native 10Gbps connection to each blade, using either 10Gbps Ethernet or Infiniband. The 1Gbps connection for each blade used in this evaluation was insufficient to achieve full to-disk performance, but was similar to the results of the other Lustre systems with comparable aggregate network throughput.

Finally, for this analysis, we sacrificed the initial design goal of server fault tolerance. In our original storage design, storage was deployed in fault tolerant

pairs of servers and RAID units. The original design used dual channel-bonded 1Gbps Ethernet, but as 10Gbps connections became affordable we decided to use fewer servers and discard the fault tolerance. We intend to revisit highly-available configurations in the future.

7 Conclusion

In this paper, we have identified several file system and storage hardware configurations that can provide a centralized storage resource for cluster and Grid computing. However, the out-of-box performance characteristcs are highly variable, and we believe that additional performance tuning and analysis is required for each file system in order to improve performance. After our tuning and deployment are complete, this storage system will provide capacity and performance required for NCAR's computational and Grid-based user community.

8 Acknowledgments

Computer time and support were provided by NSF MRI Grant #CNS-0421498, NSF MRI Grant #CNS-0420873, NSF MRI Grant #CNS-0420985, NSF sponsorship of the National Center for Atmospheric Research, the University of Colorado, and a grant from the IBM Shared University Research (SUR) program. We would like to thank the following people and corporations for their technical support and evaluation equipment: Tom Leinberger (Myricom), Lonnie Maynard (Aspen Systems), Tom Corpuz (Xyratex), Rick Scott and Keith Miller (Data Direct Networks), and Mike Barbour, Osvaldo Rentas, Rick Friedman, and Kevin Kelly (Terascala). Finally, thanks to John Dennis for critical feedback and the POPIO benchmark, Bobby House for his assistance with system administration and data management, and Adam Boggs for his work with Lustre on our IBM Blue Gene/L system.

References

1. IBM General Parallel File System (GPFS). http://www.ibm.com/systems/clusters/software/gpfs.html.
2. IOR HPC Benchmark. http://sourceforge.net/projects/ior-sio/.
3. Lustre. http://www.clusterfs.com.
4. Parallel Virtual File System 2 (PVFS2). http://www.pvfs.org/.
5. ZeptoOS: The small linux for big computers. http://www-unix.mcs.anl.gov/zeptoos/.
6. A. Boggs, J. Cope, S. McCreary, M. Oberg, H. M. Tufo, T. Voran, and M. Woitaszek. Improving cluster management with scalable filesystems. In *Proceedings of the 7th LCI International Conference on Linux Clusters: The HPC Revolution*, Norman, Oklahoma, May 2006.

7. J. Cope, M. Oberg, H. M. Tufo, and M. Woitaszek. Shared parallel file systems in heterogeneous Linux multi-cluster environments. In *Proceedings of the 6th LCI International Conference on Linux Clusters: The HPC Revolution*, Chapel Hill, North Carolina, Apr. 2005.
8. J. H. Laros, L. Ward, R. Klundt, S. Kelly, J. L. Tomkins, and B. R. Kellogg. Red Storm I/O performance analysis. In *Proceedings of the 2007 IEEE International Conference on Cluster Computing (Cluster 2007)*, Austin, Texas, July 2007.
9. R. Thakur. Parallel I/O benchmarks list. http://www-unix.mcs.anl.gov/~thakur/pio-benchmarks.html, June 2007.
10. H. Yu, R. K. Sahoo, C. Howson, G. Almasi, J. G. Castanos, M. Gupta, J. E. Moreira, J. J. Parker, T. E. Engelsiepen, R. Ross, R. Thakur, R. Latham, and W. D. Gropp. High performance file I/O for the Blue Gene/L supercomputer. In *Proceedings of the 12th International Symposium on High-Performance Computer Architecture (HPCA-12)*, Feb. 2006.