The Spider Center Wide File System; From Concept to Reality

Galen M. Shipman, David A. Dillow, Sarp Oral, Feiyi Wang

National Center for Computational Sciences, Oak Ridge National Laboratory
Oak Ridge, TN 37831, USA,
{gshipman,dillowda,oralhs,fwang2}@ornl.gov

Abstract

The Leadership Computing Facility (LCF) at Oak Ridge National Laboratory (ORNL) has a diverse portfolio of computational resources ranging from a petascale XT4/XT5 simulation system (Jaguar) to numerous other systems supporting development, visualization, and data analytics. In order to support vastly different I/O needs of these systems Spider, a Lustre-based center wide file system was designed and deployed to provide over 240 GB/s of aggregate throughput with over 10 Petabytes of formatted capacity. A multi-stage InfiniBand network, dubbed as Scalable I/O Network (SION), with over 889 GB/s of bisectional bandwidth was deployed as part of Spider to provide connectivity to our simulation, development, visualization, and other platforms. To our knowledge, while writing this paper, Spider is the largest and fastest POSIX-compliant parallel file system in production. This paper will detail the overall architecture of the Spider system, challenges in deploying and initial testings of a file system of this scale, and novel solutions to these challenges which offer key insights into file system design in the future.

1 Introduction

In 2008 the Leadership Computing Facility (LCF) at Oak Ridge National Laboratory (ORNL) deployed a 1.38 Petaflop Cray XT5 supercomputer [2], dubbed Jaguar, providing the world's most powerful HPC platform for open-science applications. In addition to Jaguar XT5 LCF also hosts an array of other computational resources such as Jaguar XT4, visualization systems and application development systems. Each of these systems require a high performance scalable file system for stable storage.

Meeting the aggregate file system performance requirements of these systems is a daunting challenge. Using system memory as the primary driver of file system bandwidth resulted in a requirement of 240 GB/sec throughput for Jaguar XT5. Achieving this level of performance requires a parallel file system that can utilize thousands of magnetic disks concurrently; the Lustre [4] parallel file system provides this capability on the Jaguar XT5 platform. Aggregate bandwidth of 200 GB/sec has been demonstrated using the Lustre file system on Jaguar XT5 and work is ongoing to reach our target (baseline) bandwidth of 240 GB/sec.

Parallel file systems on leadership class machines have traditionally been tightly coupled to a single simulation platform. This approach has resulted in the deployment of a dedicated file system for each computational platform at LCF. These dedicated file systems have created islands of data within LCF. Users working on a visualization system such as Lens cannot directly access data generated from a large scale simulation on Jaguar and must instead resort to transferring data over the LAN. Maintaining even two separate namespaces is a distraction for users. In addition to poor usability, dedicated file systems can make up a substantial percentage of total system deployment cost often exceeding 10%.

The LCF recognized that deploying dedicated file systems for each computational platform was neither cost effective nor manageable from an operational perspective. With this in mind the LCF initiated the Spider project to deploy a center wide parallel file system capable of serving the needs of the largest scale computational resources at the LCF. The Spider project had a number of ambitious goals:

1. To provide a single storage pool for all LCF computational resources.

2. To meet the and performance scalability requirements of the largest and all LCF platforms.

3. To provide resilience in the face of system failures both internal to the storage system as well as failures of external systems such as Jaguar XT5.

4. Allow upgrades with minimum reconfiguration effort to allow sustained growth of the storage pool independent of the computational platforms.

The LCF began evaluating the feasibility of meeting these goals in 2006. Initial work focused on developing prototype systems and integrating these systems within the LCF. While the LCF was the primary driver of this initiative, in order to achieve the technically ambitious goals dictated by the Spider project partnerships with Cray, Sun microsystems, and Data Direct Networks (DDN) were developed. The Lustre Center of Excellence (LCE) at ORNL was established as a result of our partnership with Sun. A primary activity of the LCE is to improve Lustre scalability, performance, and reliability for our leadership class computational platforms. To this end the LCE has 3 on-site Lustre file system engineers and 1 on-site Lustre application I/O specialist. Through the LCE and our partnership with Cray, Sun, and DDN, the LCF has worked through a number of complex issues in the development of the Spider file system which will be elaborated in Section 3.

This paper describes the Spider parallel file system and our efforts in taking this system from concept to reality. The remainder of this paper is organized as follows: Section 2 provides an overview of the Spider file system architecture. A number of key challenges and their resolutions are described in Section 3. Finally, conclusions are discussed in Section 4 followed by future work in Section 5.

2 Architecture

2.1 Spider

Spider, a Lustre-based center-wide file system, will replace multiple file systems on the LCF network with a single scalable system. Spider provides centralized access to petascale data sets from all LCF platforms, eliminating islands of data. File transfers among LCF resources will be unnecessary. Transferring petascale data sets between Jaguar and the visualization system, for example, could take hours between two different file systems, tying up bandwidth on Jaguar and slowing simulations in progress. Eliminating such unnecessary file transfers will improve performance, usability, and cost. Data analytics platforms will benefit from the high bandwidth of Spider without requiring a large investment in dedicated storage.

Unlike previous storage systems, which are simply high-performance RAID sets connected directly to the computation platform, Spider is a large-scale storage cluster. 48 DDN S2A9900 controllers provide the object storage which in aggregate provides over 240 GB/s of bandwidth, over 10 petabytes of RAID6 formatted capacity from 13,440 1-terabyte SATA drives.

The DDN S2A9900 is an update to the S2A9550 product. The couplet is composed of two singlets. Coherency is loosely maintained over a dedicated Serial Attached Storage (SAS) link between the controllers. This is sufficient for insuring consistency for a write-back cache disabled system. An XScale processor manages the system but is not in the direct data path. Host-side interfaces in each singlet can be populated with two dual-port 4x DDR IB HCAs or two 4Gb FC HBAs. The back-end disks are connected via ten SAS links on each singlet. For a SATA based system, these SAS links connect to expander modules within each disk shelf. The expanders then connect to SAS-to-SATA adapters on each drive. All components have redundant paths. Each singlet and disk tray has dual power-supplies where one power supply is powered by the house power and the other by the UPS. Figure 1 illustrates the internal architecture of a DDN S2A9900 couplet and Figure 2 shows the overall Spider architecure.

The DDN S2A9900 utilizes FPGAs to handle the pipeline of data from hosts to back-end disks. Different FPGAs handle segmenting the data, performing the Reed-Solomon (RAID6) encoding and de-

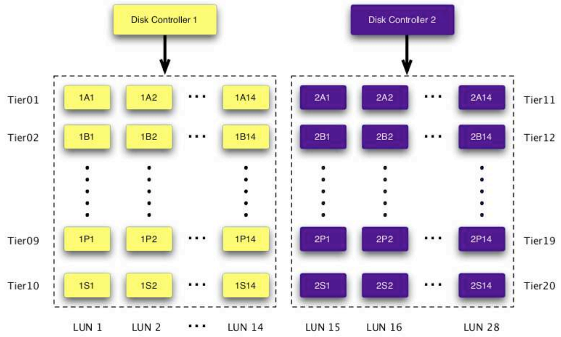


Figure 1: Internal architecture of a S2A9900 couplet

coding, and queuing the required disk commands. As a result of this architecture, the system is fairly rigid. The system requires all 10 back-end channels to be connected to disk trays for even the smallest configuration. The system only supports an 8+2 RAID configuration either in a RAID 5 or a RAID 6 configuration. Our system is configured with RAID 6. However, the system is highly efficient and since additional checks are performed in hardware on-the-fly, they introduce very little overhead. For example, enabling parity-check on read operations has shown to have negligible impact on read-performance. In fact, the check is enabled by default.

This object storage is accessed through 192 Dell dual-socket quad-core Lustre OSS (object storage servers) providing over 14 teraflops in performance and 3 terabytes of memory in aggregate. Each OSS can provide in excess of 1.25 GB/s of file system level performance. Metadata is stored on 2 LSI Engino 7900s (XBB2) and is served by 3 Dell quad-socket quad-core systems. These systems are interconnected via SION providing a high performance backplane for Spider.

Each DDN S2A9900 is configured with 28 RAID 6 8+2 tiers. Four OSSs provide access to these 28 tiers (7 each). OSSs are configured in failover pairs so that in the event of an OSS failure the designated partner can provide access to the failed OSSs OSTs. Each OSS is connected to both DDN controllers in a couplet so that it can access its OSTs in the event of a controller failure using DM multipath. Under such a controller failure case, while bandwidth is reduced, the storage system remains accessible to users. LCF platform is configured with Lustre routers to access Spider as if the storage was locally attached. All other Lustre components reside within the Spider infrastructure providing ease of maintenance as detailed above. Multiple routers are configured for each platform to provide performance and redundancy in the event of a failure.

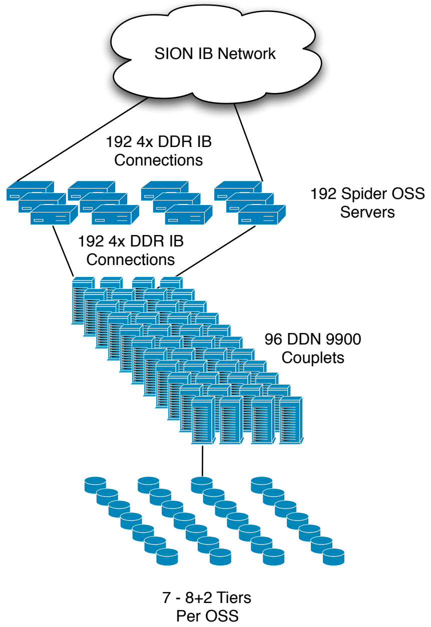


Figure 2: Overall Spider architecture

On the Jaguar XT5 partition 192 Cray service I/O (SIO) nodes, each with a dual socket AMD Opteron and 8 GBytes of RAM are connected to Crays SeaStar2+ network via HyperTranport. Each SIO is also connected to SION using Mellanox ConnectX HCAs and Zarlink CX4 optical cables. These SIO nodes are configured as Lustre routers to allow compute nodes within the SeaStar2+ torus to access the Spider filesystem at speeds in excess of 1.25 GB/s per XT5 compute node. The Jaguar XT4 partition is similarly configured with 48 Cray SIO nodes acting as Lustre routers. In aggregate the XT5 partition has over 240 GB/s of storage throughput while XT4 has over 60 GB/s. Other LCF platforms are similarly configured with Lustre routers in order to the requisite performance of a balanced platform.

Moving towards a centralized file system requires increased redundancy and fault tolerance. Spider is designed to eliminate single points of failure and thereby maximize availability. By using failover pairs, multiple networking paths, and the resiliency features of the Lustre file system, Spider provides a reliable high-performance centralized storage solution greatly enhancing our capability to deliver scientific insight.

2.2 SION

In order to provide a true integration between all systems hosted by the LCF, a high-performance large-scale IB network, dubbed as Scalable I/O Network (SION), has been deployed. SION is a multistage InfiniBand network and enhances the current capabilities of the LCF. Such capabilities include resource sharing and communication between the two segments of Jaguar and real time visualization as data from the simulation platform can stream to the visualization platform at extremely high data rates. Figure 3 illustrates the SION architecture.

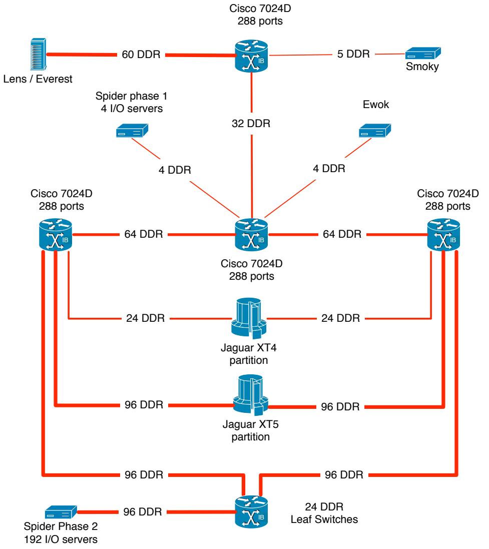


Figure 3: Scalable I/O Network (SION) architecture

SION currently connects both segments (XT4 and XT5) of the 1.645 PFLOPS supercomputer (Jaguar) and our centralized Lustre storage cluster (Spider), Lens (Visualization cluster), Ewok (end-to-end cluster), Smoky (application development and readiness cluster), HPSS [11] and GridFTP [1] servers.

As new platforms are deployed at the LCF, SION will continue to scale out providing an integrated backplane of services. Rather than replicating infrastructure services for each new deployment SION will allow access to existing services thereby reducing total costs, enhancing usability and decreasing the time from initial acquisition to production readiness.

SION is a high performance IB DDR network providing over 889 GB/s of bisectional bandwidth. The core network infrastructure is based on three 288-port Cisco 7024D IB switches and an additional fourth Cisco 7024D. One switch provides an aggregation link while the other two switches provide connectivity between the two Jaguar segments and the Spider file system. The forth 7024D switch provides connectivity to all other LCF platforms and is connected to the single aggregation switch. The Spider is connected to the core switches via 48 24-port Flextronics IB switches allowing storage to be accessed directly from SION. Additional switches provide connectivity for the remaining LCF platforms.

The LCF spans over 40,000 ft of raised floor space with platforms spread throughout the two-story center. In order to handle the distance requirements imposed by such a large-scale center, Zarlink IB optical cables in a number of lengths of up to 60 meters are used. These long length cables allowed connectivity between our two-story facility, an impossibility with copper cables. In total SION has over 3,000 InfiniBand (IB) [5] ports and over 3 miles of optical cables providing high performance connectivity. Future plans for SION include an upgrade for a 576 GB/s bandwidth increase.

The number and complexity of the LCF systems deployed required an in-house solution to the IB routing configuration. Out of the box, OpenSM was able to provide a better routing configuration for SION than the Cisco subnet manager, such as minimizing the number of hops per connection. However, OpenSM assigned two of the primary destinations on the Spider leaf switches to share a single uplink from the core Cisco 7024D switches and this resulted in a 33% overall performance reduction. To alleviate this shortcoming, OpenSM is provided with an ordered list of GUIDs that should have their forwarding entries placed before the general population. In this way, the standard algorithms that OpenSM uses to determine the minimum hop count and the sharing of links remains unchanged, yet the combination of techniques gives each primary destination a dedicated link with respect to other primary destinations. This solution allowed LCF to better utilize the deployed SION bandwidth.

3 Integration

To provide a scalable, high-performance, and reliable parallel file system at the LCF the Technology Integration Group began evaluation of a number of technologies in 2007. IB had gained considerable traction in the HPC space and had been demonstrated at scales exceeding our requirements [9]. The availability of double data rate (DDR) InfiniBand, high port count switches and long reach (up to 100 meter) optical cabling provided a plausible solution to our system area network requirements. Much of our early work on IB evaluation focused on optical cable testing [6] and porting the OpenFabrics OFED stack to the Cray service I/O (SIO) node. Cray later provided a productized version of this work and now fully supports IB on the XT series.

Working closely with DDN ORNL began evaluation of the DDN S2A9900 storage system in 2008. The LCF fielded one of the earliest examples of the S2A9900 platform for evaluation and worked with DDN to address a number of performance, stability and reliability issues. During this period a number of firmware and software level changes resulted in substantially improving the S2A9900 storage platform in order to meet the requirements of the Spider file system.

3.1 Reliability Analysis of the DDN S2A9900

Necessitated by both the scale and the uniqueness of the Spider system, our integration work also included analysis from a system reliability perspective. Our goal was to establish a failure model and a quantitative expectation of the system's reliability and availability. In addition to examining the AFR (Annual Failure Rate), impact of UBE (Uncorrectable Bit Errors), and RAID 8+2's impact on overall system reliability [3, 7], we also developed a detailed failure model for DDN S2A9900. Particular attention was given to the DDN S2A9900's peripheral components as they are often the deciding factors on the overall system reliability.

There are three other major components besides disk arrays which we considered: I/O module, DEM, and baseboard. The vendor supplied MTTF for these three components are shown in the following table. Our approach is to first define the possible *Spider failures*, then take into account the physical layout of our storage system and deduce the reliability graph for the couplet system: the composite system is composed of a mix of series and parallel component connections based on the failure model we defined, and we can reach the estimation for the reliability of the composite system (one couplet).

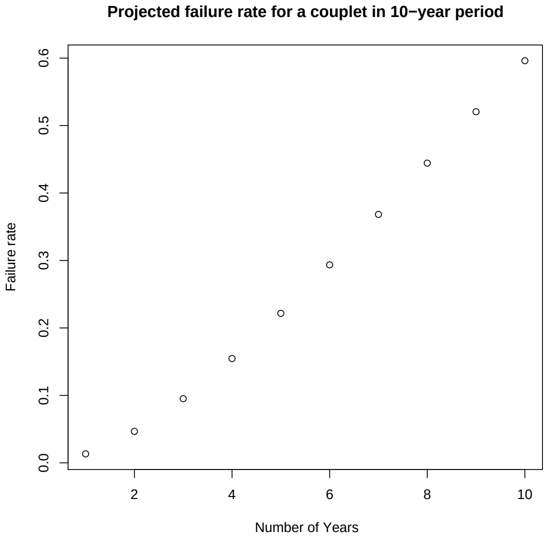


Figure 4: 10-year Projection of Failure Rate

Component	MTTF
I/O Module	1,263,856
DEM	1,552, 437
Baseboard	356,143

Based on this failure model (with enumeration of six defined failure cases) as well as known component MTTF listed in the table above, we can project the failure rate of one couplet over a 10-year period, as illustrated in Figure 4. We can also quantify the 10 year spread on each of the failure case in the form of the box-graph in Figure 5. It shows that on average (indicated by the middle bar in the box), case 1 and case 3 have the most significant impact on the overall failure rate.

The basic insights gained by this analysis are:

- The reliability of peripheral components present the most severe impact on the overall reliability, with the baseboard being the weak-

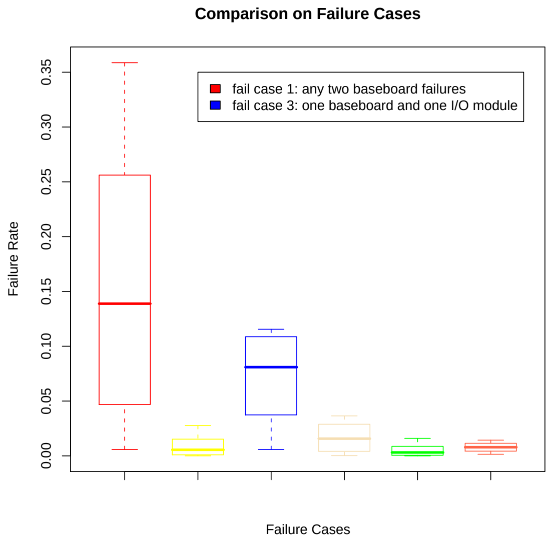


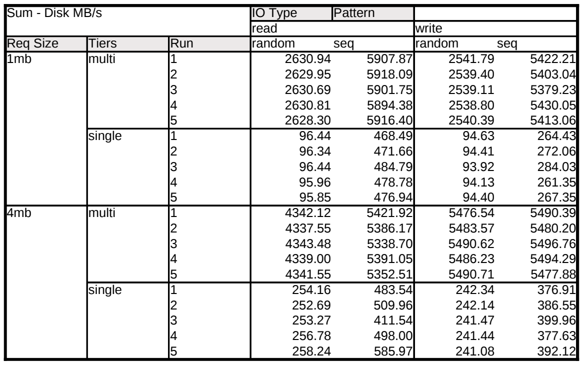
Figure 5: Failure Rate Distribution on Each Case

est link. It contributes to approximately 50% of the possible failure scenarios.

- For the first year, we can expect the number of failed couplets to be 0.64. However, by the end of year two, the number is approaching to 2.42, indicating a high probability that we will experience couplet failures.

3.2 Establishing a Baseline of Performance

In order to obtain a baseline of performance on the DDN S2A9900 the XDD benchmark [8] utility was used. XDD provides a mechanism to direct multiple processes to read and/or write blocks from/to a block device in a synchronized fashion. Readers and writers can exist on different machines with all processes synchronizing prior to initiating their block transfers. XDD can be run in sequential or random read or write mode. Our initial experiments focused on aggregate performance for sequential read or write workloads. Performance results using XDD from 4 hosts connected to the DDN via DDR IB are summarized in Table 6. The results presented are a summary of our testing and show performance of sequential read, sequential write, random read, and random write using 1MB and 4MB transfers. These tests were run using a single host with a single LUN and 4 hosts each with 7 LUNs which is labeled “multi” in the Tiers column. Performance results of 5 runs in each configuration are presented. Of particular interest is the dramatically improved performance of random read and random write operations when transfer sizes are increased from 1MB to 4MB as the cost of the head seek is amortized over a larger write. Efforts are ongoing to improve Lustre performance by utilizing 4MB transfers.



Sum - Disk MB/s			IO Type	Pattern		
			read		write	
Req Size	Tiers	Run	random	seq	random	seq
1mb	multi	1	2630.94	5907.87	2541.79	5422.21
		2	2629.95	5918.09	2539.40	5403.04
		3	2630.69	5901.75	2539.11	5379.23
		4	2630.81	5894.38	2538.80	5430.05
		5	2628.30	5916.40	2540.39	5413.06
	single	1	96.44	468.49	94.63	264.43
		2	96.34	471.66	94.41	272.06
		3	96.44	484.79	93.92	284.03
		4	95.96	478.78	94.13	261.35
		5	95.85	476.94	94.40	267.35
4mb	multi	1	4342.12	5421.92	5476.54	5490.39
		2	4337.55	5386.17	5483.57	5480.20
		3	4343.48	5338.70	5490.62	5496.76
		4	4339.00	5391.05	5486.23	5494.29
		5	4341.55	5352.51	5490.71	5477.88
	single	1	254.16	483.54	242.34	376.91
		2	252.69	509.96	242.14	386.55
		3	253.27	411.54	241.47	399.96
		4	256.78	498.00	241.44	377.63
		5	258.24	585.97	241.08	392.12

Figure 6: XDD Performance Results

3.3 Improve Filesystem Journaling

After establishing a baseline of performance using XDD we examined Lustre level performance using the IOR benchmark [10]. Testing was conducted using 4 OSSes each with 7 OSTs on the DDN S2A9900. Our initial results showed very poor write performance of only 1398.99MB/sec using 28 clients with each client writing to different OSTs. Lustre level write performance was a mere 25.8% of our baseline performance metric of XDD sequential writes with a 1MB transfer size. Profiling the I/O stream of the IOR benchmark using the DDN 9900 utilities revealed a large number of 4KB writes in addition to the expected 1MB writes. These small writes were eventually traced to ldiskfs journal updates. Journaling is widely used by modern file systems to increase file system robustness against meta data corruptions and to minimize file system recovery times after a file system crash. Lustre uses ldiskfs (a modified version of ext3) as the backend file system on the OST, MDT and MGT devices. Similar to ext3, ldiskfs journals only metadata *data journaling mode* by first writing the data blocks to disk followed by writing the metadata blocks to the journal. The journal is then written to disk and marked as committed. In the worst case this can result in 2 4KB writes (and 2 head syncs) with every 1MB write. Due to the poor IOP performance of SATA disks these additional head syncs and small writes substantially degraded performance.

To alleviate the journaling overhead, we evaluated two potential solutions. The first was to move the file system journal to an external block device. The RamSan 400 was selected as a potential candidate for an external journal device. The RamSan 400 is an InfiniBand connected storage system which utilizes DRAM as the storage media. This provides an extremely high IOP solution and due to our relatively modest storage requirements of approximately $400MB$ per OST device the RamSan was a potential solution. By using the RamSan as an external journal device we were able to isolate all 4KB journal traffic to the high IOP storage device while allowing sequential data blocks to flow to the low IOP SATA disks. By utilizing external journals we were able to achieve $3292.6MB/sec$ or 60% of our baseline performance.

A second solution was later proposed to decrease the impact of journal updates. Rather than open the journal, write data blocks and then close the journal on every write, ldiskfs was modified to update the journal asynchronously after a potentially large number of writes. Utilizing this approach resulted in dramatically fewer 4KB updates (and fewer head seeks) which substantially improved performance to over $4625MB/s$ or 85% of our baseline performance.

3.4 Network Congestion Control

After improving Lustre level performance by over 330% on a single DDN S2A9900 we began to focus our efforts on examining performance at scale utilizing half of the available Spider storage and Jaguar XT5. Initially we configured the storage as a dedicated file system on Jaguar XT5. In this configuration each Cray SIO node acts as a Lustre OSS as opposed to a Lustre router. This allowed us to work through a number of stability and performance issues within Lustre without the additional complexity of a routed configuration. To baseline performance of the storage system from Jaguar XT5 we used XDD run from the Cray SIO nodes. In order to minimize network congestion on the IB fabric each Cray SIO node was paired with its associated DDN controller on an IB line card on the core switch. Each line card is a 24 port crossbar, by pairing the SIO node with its associated DDN controller on a crossbar congestion in the IB fattree is eliminated. This configuration allows each SIO node to achieve maximum bandwidth irrespective of the number of communicating peers which we verified by running XDD on 48 to 96 Cray SIO nodes in parallel. Performance scaled linearly. After baselining performance of the storage system from the Cray SIO nodes we then conducted a series of tests using IOR from the Cray compute nodes. Initial results showed a high variability in performance between successive runs even on a quiesced system. It was also noted that read performance was always substantially lower than write performance. Our suspicion was that placement of objects on OSTs relative to the corresponding clients location in the torus could result in pathological cases resulting in torus congestion. That is to say a Lustre client on a compute node may allocate objects on an OST that is topologically far away. In addition, it was noted that the dramatic performance difference between reads and writes may also be a result of torus congestion as the data paths over the network are not symmetric for read and write operations.

To test our theory that SeaStar+ network congestion was substantially impacting performance of the file system a mechanism was devised that allowed clients to allocate objects on OSTs that are topologically near the client. Performance was then measured using IOR with each client writing to a single file on an OST that was topologically near. Performance was improved substantially as illustrated in Figure 7. In Figure 7 “default read” and “default write” performance was obtained using the IOR benchmark using the default object allocation policy of Lustre. The performance results of both “placed read” and “placed write” were obtained using IOR and preallocating files on OSTs topologically near the client writing to this file.

Having demonstrated that network congestion can severely impact aggregate file system performance when the Cray SIO node was configured as a Lustre OSS we then began tackling this problem in the context of the routed Spider configuration. Lustre clients and servers spread load amongst routers based on queue depths on the router. Unfortunately there is no mechanism to detect congestion between a client and a router or a server and a router and prefer routers that minimize network congestion. Recognizing this as a severe architectural limitation we began working with Sun to pro-

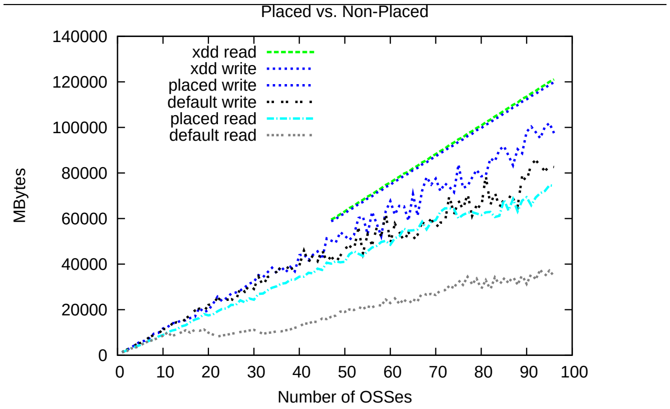


Figure 7: Performance on Jaguar XT5

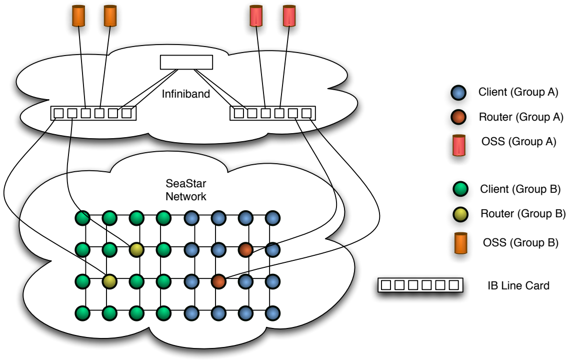


Figure 8: Lustre Fine Grain Routing

vide a mechanism for clients and servers to prefer specific routers. This mechanism would allow us to pair clients with routers within the SeaStar+ torus in order to minimize congestion within the torus. In essence we can view a set of 32 routers as a replicated resource providing access to every OST in the file system such that congestion in the IB network is minimized. With 192 total routers and 32 routers in each set we have 6 replicated resources within the SeaStar torus. By grouping these 32 routers appropriately we can then assign clients to these routers such that communication is localized to a sub 3-D mesh of the torus. This strategy should reduce contention in the SeaStar+ torus based on our previous results on OST placement. Work is ongoing to validate this theory. Figure 8 illustrates this concept using two routing groups on the SeaStar+ network each with two routers. Lustre clients in the first group indicated by the color green will prefer the routers indicated by the color yellow. The yellow routers can access all storage via an IB cross bar without resorting to traversing the IB fat-tree. In a similar fashion clients in blue can utilize the red routers to access any storage. Contention on both the SeaStar+ network and the IB network is therefor minimized.

3.5 Scalability

In order to verify the stability of the Spider file system at full scale testing was conducted using 4 major systems at the LCF which included the Jaguar XT5 and Jaguar XT4 partition, Lens and Smoky. All systems were configured to mount Spider concurrently equating to over 26,000 Lustre clients and over 180,000 processing cores. To our knowledge this is the largest number of clients to mount a single Lustre file system to date. In conducting this testing a number of issues were revealed. As the number of clients mounting the file system increased the memory footprint of Lustre grew at an unsustainable rate. As the memory footprint on the OSSes grew past 11GB we began to get out-of-memory errors (OOMs) on the OSS nodes. By analyzing the memory allocations from Lustre it was discovered that an extremely large number of 64KB buffers were being allocated. Reviewing the Lustre code base revealed that $40KB$ memory allocations were made, 1 for each client-OST connection which resulted in 64KB memory allocations within the Linux kernel. With 7 OSTs per OSS and over 26,000 clients this equated $26000 * 7 * 64KB = 11.1GB$ of memory per OSS for server side client statistics alone. As client statistics are also stored on the client, a much more scalable solution as each client would only store $7 * 64KB = 448KB$ in our configuration, we removed the server side statistics entirely. Figure 9 illustrates the server side memory footprint as a function of number of clients in our initial configuration.

3.6 Fault Tolerance

As Spider is a center wide resource much of our testing at full scale centered on surviving component failures. A major concern was the impact of an unscheduled outage of a major computational resource on the Spider file system. To test the impact of this we mounted the Spider file system on all the

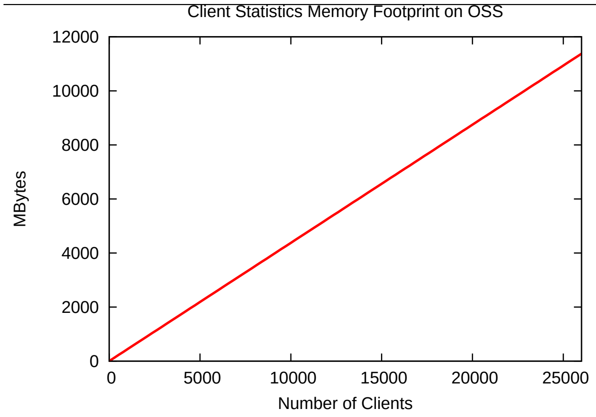


Figure 9: Memory footprint of client statistics

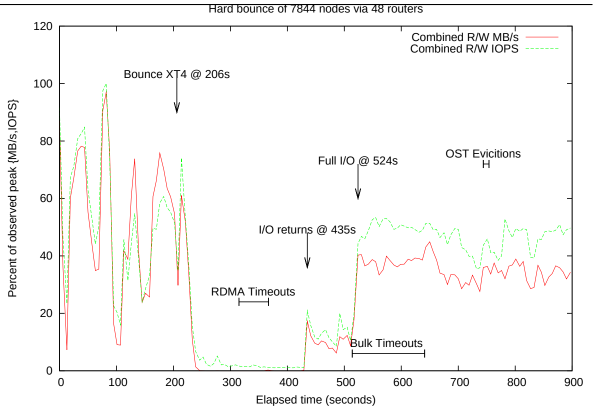


Figure 10: Impact of Jaguar XT4 Reboot

compute nodes spanning the Jaguar XT4 partition, Lens and Smoky. With an I/O workload active on Jaguar XT4, Smoky, and Lens the Jaguar XT4 system was rebooted. Shortly thereafter the file system became unresponsive. Postmortem analysis of this event showed that the OSSes spent a substantial amount of time processing client evictions. Using the DDN S2A9900 performance analysis tools we observed a large number of small writes to each OST during this eviction processing with very little other I/O progressing on the system. This was later tracked down to a synchronous write to each OST for each evicted client resulting in a backlog of client I/O from Lens and Smoky. Changing the client eviction code to use an asynchronous write resolved this problem and in later testing allowed us to demonstrate the file systems ability to withstand a reboot of either Jaguar XT4 or Jaguar XT5 with minimal impact to other systems with active I/O. Figure 10 illustrates the impact of a reboot of Jaguar XT4 with active I/O on Jaguar XT5, Lens and Smoky. The y-axis shows the percentage of peak aggregate performance throughout the experiment. At 206 seconds elapsed time Jaguar XT4 is rebooted. RPCs timeout and performance of the Spider file system degrades substantially for approximately 290 seconds. Aggregate bandwidth improves at 435 seconds and steadily increases until we hit the new steady state performance at 524 seconds. Aggregate performance does not return to 100% due to the mixed workload on the systems and the absence of Jaguar XT4 I/O load.

4 Conclusions

In collaboration with Cray, SUN, and DDN the Technology Integration group at ORNL has successfully architected, integrated and deployed a center wide file system capable of supporting over 26,000 clients and delivering in excess of 200 GB/sec of file system bandwidth. To achieve this goal a number of unique technical challenges were met. Designing a system of this magnitude required careful analysis of failure scenarios, fault tolerance mechanisms to deal with these failures, scalability of system software and hardware components, and overall system performance. Through a phased approach of deploying and evaluating prototype systems, deployment of a large scale dedicated file system followed by a transition to the Spider file system, we have delivered one of the world's highest performance file systems. The Spider file system is now in limited access production use at the LCF by "early science" projects on the Jaguar XT5 partition. Spider will transition to full production in the Summer of 2009.

5 Future Work

While a large number of technical challenges have been addressed during the Spider project a number still remain. Performance degradation during an unscheduled system outage may last for up to 5 minutes as detailed in Section 3. Technology Integration is working closely with file system engineers in the LCE in order to minimize or eliminate this degradation entirely. Fine grain routing in the LNET layer is currently accomplished by creating multiple distinct LNET networks. We have begun to test this approach and have had some success although the complexity of the configuration is daunting. Rather than using separate LNET networks to achieve fine grained routing a routing table approach may ease the complexity of configuration while giving tighter control over individual routes. This would require significant changes within LNET and as such each approach must be evaluated carefully.

References

[1] W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link. The Globus Striped GridFTP Framework and Server. In *SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 54, Washington, DC, USA, 2005. IEEE Computer Society.

[2] A. Bland, R. Kendall, D. Kothe, J. Rogers, and G. Shipman. Jaguar: The worlds most powerful computer. In *Proceedings of the Cray User Group Conference*, 2009.

[3] G. A. Gibson and D. A. Patterson. Designing disk arrays for high data reliability. *J. Parallel Distrib. Comput.*, 17(1-2):4–27, 1993.

[4] S. M. Inc. Luste wiki. `http://wiki.lustre.org`, 2009.

[5] Infiniband Trade Association. Infiniband Architecture Specification Vol 1. Release 1.2, 2004.

[6] M. Minich. Inniband Based Cable Comparison, June 2007.

[7] D. A. Patterson, G. A. Gibson, and R. H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). Technical report, Berkeley, CA, USA, 1987.

[8] T. Ruwart. XDD. `http://www.ioperformance.com/`, 2009.

[9] Sandia National Laboratories Technical Report. Thunderbird Linux Cluster ranks 6th in Top500 supercomputing Race. `http://www.sandia.gov/news/resources/releases/2006/thunderbird.html`.

[10] H. Shan and J. Shalf. Using IOR to analyze the I/O performance of XT3. In *Proceedings of the 49th Cray User Group (CUG) Conference 2007*, Seattle, WA, 2007.

[11] The HPSS Collaboration. HPSS. `http://www.hpss-collaboration.org/`.