OLCF's 1 TB/s, Next-Generation Lustre File System

Sarp Oral, David A. Dillow, Douglas Fuller, Jason Hill,
Dustin Leverman, Sudharshan S. Vazhkudai, Feiyi Wang,
Youngjae Kim, James Rogers, James Simmons, Ross Miller

Oak Ridge Leadership Computing Facility, Oak Ridge National Laboratory
Oak Ridge, TN 37831, USA
{oralhs,dillowda,fullerdj,hilljj,leverman,vazhkudaiss,fwang2,kimy1,
jrogers,simmonsja,rgmiller}@ornl.gov

Abstract

The Oak Ridge Leadership Computing Facility (OLCF) at Oak Ridge National Laboratory (ORNL) has a long history of deploying the world's fastest supercomputers to enable open science. At the time it was deployed in 2008, the Spider file system had a formatted capacity of 10 PB and sustained transfer speeds of 240 GB/s which made it the fastest Lustre file system in the world. However, the addition of Titan, a 27 PFLOPS Cray XK7 system, along with other OLCF computational resources, has radically increased the I/O demand beyond the capabilities of the existing Spider parallel file system. The next-generation Spider Lustre file system is designed to provide 32 PB of capacity to open science users at OLCF, at an aggregate transfer rate of 1 TB/s. This paper details the architecture, design choices, and configuration of the next-generation Spider file system at OLCF.

1 Introduction

The first-generation Spider file system (also known as Spider 1), deployed in 2008, was designed to be a center-wide, shared parallel file system, which represented a significant departure from the traditional approach of tightly coupling the parallel file system to a single simulation platform. This decoupled approach has allowed the Oak Ridge Leadership Computing Facility (OLCF) at Oak Ridge National Laboratory (ORNL) to utilize Spider 1 as the primary parallel file system for all major compute resources at the OLCF, providing users with a common scratch and project space across all platforms. This approach has also reduced operational costs and simplified the management of our storage environment.

The primary platform served by Spider 1 was Jaguar [1, 18, 8], a 3.3 Petaflop/s Cray XK6 [4] machine, and one of the world's most powerful supercomputers. Since then, Jaguar has been upgraded into Titan [2]. Titan is a Cray XK7 system [5] that couples the AMD 16-core Opteron 6274 processor, running at 2.4 GHz, with an NVIDIA "Kepler" K20 graphics processing unit (GPU) on each compute node. With 18,688 compute nodes, 710 TB of total system memory, and a peak performance of more than 20 Petaflops [8, 17], Titan is currently the No. 1 machine on the Top500 list of supercomputers. The combination of CPUs and GPUs is expected to allow Titan, and future systems, to overcome the power and space limitations inherent in previous generations of HPC machines. Figure 1 illustrates a Titan Cray XK7 compute node with Cray's Gemini interface.

In addition to Titan, the OLCF also hosts an array of other computational resources such as the visual-

This research used resources of the Oak Ridge Leadership Computing Facility, located in the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.

Notice: This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

ization, end-to-end, and application development platforms. Each of these systems requires a reliable, high-performance and scalable file system for data storage.

The next-generation Spider project (also known as Spider 2) was started in 2009, immediately after the commissioning of Spider 1. Spider 2 will continue to adopt the decoupled approach in order to provide a center-wide storage system.

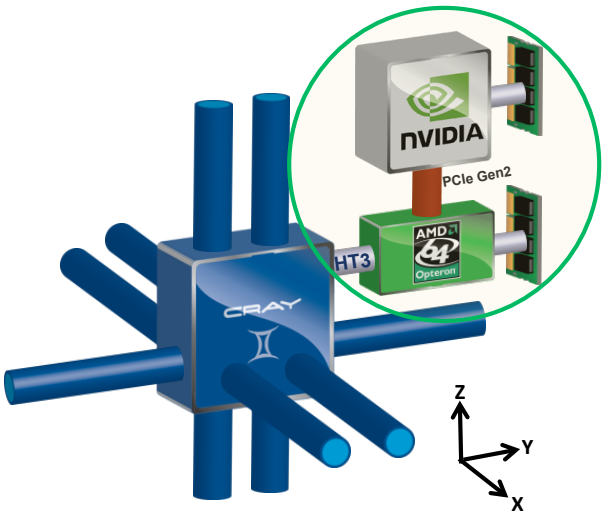


Figure 1: A Cray XK7 compute node block diagram connected to Cray Gemini interface. *Courtesy of Cray, Inc.*

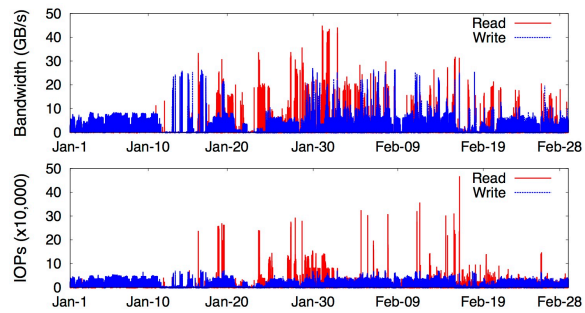
This paper presents our efforts towards deploying Spider 2. Much of our planning and execution has been based on lessons learned from the current system, operational experience, and projections based on the required capabilities for Titan. The remainder of this paper is organized as follows: Section 2 provides an overview of the lessons learned from Spider 1; Section 3 presents our preparations towards deploying and commissioning the new file system; Section 4 presents the hardware, network, and software architectures; Section 5 presents our conclusions and future work.

2 Lessons Learned from Spider 1

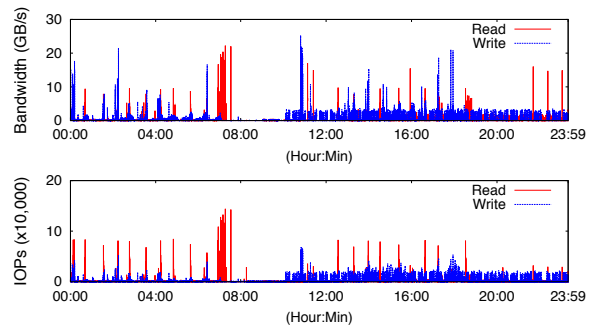
After commissioning Spider 1, we began analyzing the system from various aspects to better understand and optimize the I/O patterns and performance.

We have studied the workload of our storage cluster using the data collected from 48 DDN "Couplets" (96 DDN controllers) [21]. Currently, our storage cluster is composed of four file system partitions, called *widow1*, *widow2*, *widow3*, and *widow4*. Each widow partition forms $\frac{1}{4}$ of the 48 DDN "Couplets" and provide 60 GB/s and 2.6 PB of capacity. The maximum aggregate bandwidth over all partitions is approximately 240 GB/s.

Figure 2 shows the observed I/O bandwidth usage of *widow2* for January and February, 2013. Note that the observed utilized bandwidth is normally very low and only spikes of high bandwidth can be observed. For example, in Figure 2(a), we can observe high I/O demands, which can be over 40 GB/s at the end of January and early February, however other days show lower I/O bandwidth. For example, between January 1 and 10, the I/O demands never go above 10 GB/s. (The other three file system partitions show similar patterns.)



(a) I/O bandwidth usage for January to February in 2013



(b) I/O bandwidth usage for January 30, 2013

Figure 2: Observed I/O performance usage from *Widow2*

Figure 2(b) shows the I/O bandwidth usage of a single 24 hour period on January 30. It can be clearly observed that the normal bandwidth usage is quite low but has brief spikes throughout the time period. We can infer from the bandwidth data that the arrival patterns of I/O requests are bursty, the I/O demands can be tremendously high for short periods of time, and overall utilization can be dramatically lower than peak usage. This usage pattern can be consistently observed from applications such as checkpoint/restart workloads, which often require maximum bandwidth for a short time, but results in an average utilization much lower than the peak bandwidth. These observations motivate a tiering architecture for next generation systems where higher band-

width, low capacity media such as NVRAM is employed to build caching or buffering tiers backed by higher capacity but slower performance tiers of magnetic disk drives. However, we have not adopted this strategy for Spider 2 because of cost concerns. As SSD prices begin to drop as low price as HDD, it will become a viable alternative for next-generation storage systems.

We also have several interesting observations from our workload characterization studies [11, 21].

As of April 2013, we have collected monthly peak read and write bandwidth data for about three years. On all the widow file systems, we have observed that max read bandwidth is higher than max write bandwidth. This asymmetry in performance is common in storage media.

We have also studied the bursty properties of I/O requests and have plotted CDF (Cumulative Distribution Function) plots with the I/O bandwidth data. We also applied a curve-fitting technique to the CDF data and statistically demonstrated that the bandwidth distributions for reads and writes follow heavy long-tail distributions. These trends are observed across all widow file system partitions. Figure 3 shows the PDF and Figure 4 shows the CDF of the I/O bandwidth from Spider 1.

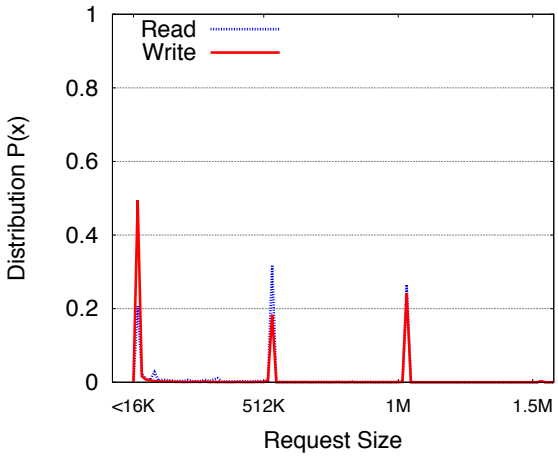


Figure 3: Spider 1 I/O request size PDF

Additionally, we have studied the percentage of read requests with respect to the total number of I/O requests in the system. Typically, HPC storage systems are write-dominant because HPC applications write a lot of checkpoint files for fault tolerance. However, we found that some widow partitions show higher read percentage than write, and even observed read percentages exceeding 80% of the total I/O in October 2010. Overall, average read percentage is around 30-40%, and we have seen that read percentages increases. For example, average read is 41% in 2011 whereas it was 24% in

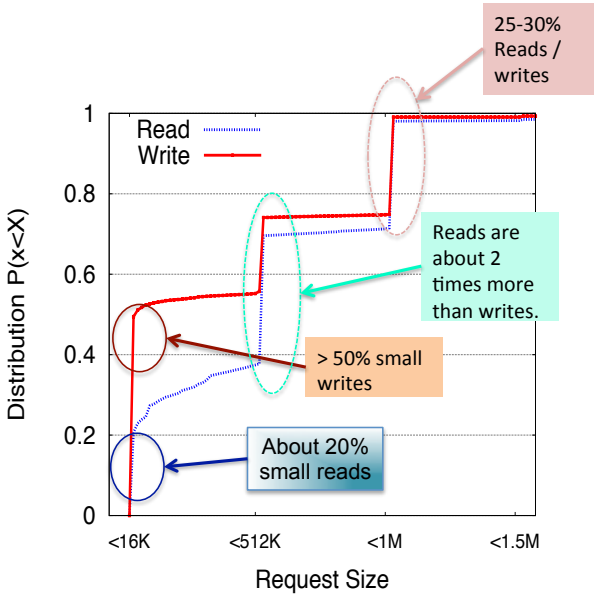


Figure 4: Spider 1 I/O request size CDF

2010. This phenomenon can be attributed to the center-wide, shared file system architecture of Spider 1, which supports both the simulation machine (Jaguar) and several data analysis, visualization, and application development clusters (Lens, Smoky, and Ewok). Jaguar typically runs HPC simulations, which are write-heavy (e.g., checkpoint I/O), while the data analysis clusters tend to support read-heavy workloads, processing the massive amounts of data produced by the simulation. Figure 5 shows the observed write ratio on OLCF's Spider 1.

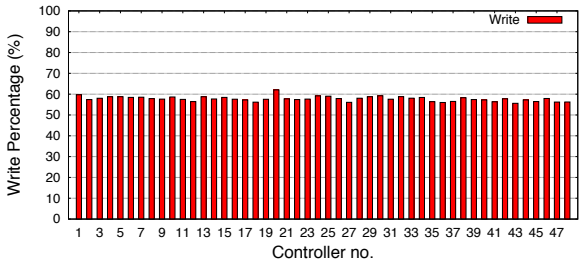


Figure 5: Spider 1 aggregate write I/O ratio

Request size distributions for reads and writes were also studied. More than 95% of total requests are 4-16 KB, 512 KB or 1 MB in length. This is because the request sizes cluster near 512 KB boundaries, imposed by the Linux block layer. Lustre also tries to send 1 MB requests to storage, and sometimes merges requests if opportunities are available. Also, storage server-side file systems do lots of meta data orations which are normally known to be small.

Along with our efforts to characterize the I/O workload on Spider 1, we have also investigated the congestion problem on the end-to-end I/O route between Jaguar and Spider 1. During the deployment of Jaguar at the OLCF, the interim direct-attached Lustre file system fell far short of the expected performance. A detailed analysis revealed that congestion on the SeaStar torus network had a significant effect on the realized performance. We characterized the bandwidth capacity of the SeaStar network links, and developed a placement strategy to pair clients to specific I/O server that are topologically close to each other, reducing the load on the common torus links and avoiding link saturation. With this approach, 92% of the raw back-end storage performance was achieved at the file system level. Furthermore, this level of performance (within 5%) was maintained when the choice of clients performing I/O was significantly limited. These results indicate that placement is a viable mechanism to increase aggregate I/O performance, not only for large-scale application invocations that span the entire Jaguar system, but also for applications that use a much smaller fraction of the available compute resources.

The performance benefits of placement did not automatically follow when the Lustre file system was transitioned from a direct-attached configuration to a routed configuration in support of center-wide access to Spider 1. The naive configuration in which all 192 routers were assigned the same weight, coupled with LNET's per-message round-robin selection policy, prohibited our placement optimization strategy. In addition to negating the benefits of reduced congestion on the torus, this configuration introduced substantial congestion within the Scalable I/O Network (SION) InfiniBand fabric. To regain opportunities for optimization via placement and to eliminate InfiniBand congestion, we have developed and evaluated three additional LNET routing configurations.

After weighing the benefits and drawbacks of alternate routing schemes, we selected a configuration for the production environment that "projected" the I/O servers into the torus. This configuration yielded over 90% of the raw back-end storage performance, and reduced the run-time of production scientific applications by up to 20.7%. When combined with specific placement strategies, this configuration demonstrated aggregate performance of 244 GB/s for both reads and writes when using the entire Spider 1 [7].

3 Preparing for Next-generation Spider

Based on the lessons learned from Spider 1 and our requirements, a long and multi-faceted preparation effort is needed for acquisition, deployment, and commissioning of Spider 2. This section describes our preparation efforts. The deployment and commissioning of the Spider 2 is not complete at the time of this writing. However, our past and current efforts can be classified into broad categories such as the evaluation efforts for the existing and emerging storage and file system technologies, development of a new benchmark suite based on the I/O workload characteristics of Spider 1 and our evaluation efforts, Lustre 2.4 testing, hardening, and scaling efforts, and finally, writing and releasing the RFP for Spider 2. As a generic guideline, we adopted an iterative design, development, and deployment cycle based on the lessons learned, evaluations, operations, and testing efforts. This is illustrated in Figure 6.

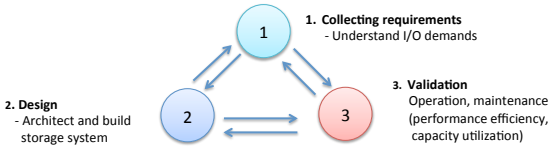


Figure 6: Spider file and storage system design, development, and deployment cycle

3.1 Pre-RFP Evaluations

In order to better prepare for writing the RFP, evaluating the responses, and to architect Spider 2, we conducted a series of technology evaluations. This effort started in late 2009 and lasted until the release of the RFP, in late 2012. We established partnerships with file and storage vendors and ran series of tests on their solutions.

Our evaluation efforts were mainly focused on assessing the block I/O performance, disk rebuild performance, meta data performance, reliability, management capability and interface, and user friendliness of the available technologies. We conducted extensive tests on a wide variety of technologies including both block-level and Lustre integrated solutions and provided results and feedback to the vendors through our established partnerships. Various magnetic disk and interconnect technologies were analyzed and evaluated in this period. In cases where technologies were unavailable to us to test in-house at OLCF, we made site visits to other HPC centers equipped with these resources to gain hands-on experience.

Our efforts, combined with our deeper understanding of Spider 1 I/O workloads, led to the development of OLCF's I/O benchmarking suite.

3.2 OLCF Benchmark Suite

The OLCF-3 Benchmark Suite was developed as part of our efforts toward procuring and deploying Spider 2 for Titan and other OLCF systems. The suite is publicly available and can be obtained at the OLCF website at [16].

The OLCF-3 Benchmark Suite is a comprehensive set of synthetic benchmarks and parameter sets designed to produce system-level and block-level performance metrics for a file or storage system. It also provides tools for quick visualization of the results, allowing head-to-head comparisons and detailed analysis of the target system's response to a variety of representative I/O scenarios.

Our experience with Spider 1 provided us with insight into the types of I/O scenarios to investigate using the benchmark suite. In particular, our I/O workload characterization effort (§2) determined that Spider 1 experiences a highly bursty, random workload with a heavy mix of small (under 512 KB) and large (512 KB and larger) read and write requests. The benchmark suite attempts to mimic this workload in order to yield a good approximation of real-world results.

Using `bash` scripts to administer a variety of I/O workloads, the suite uses the `obdfilter-survey` at the file system-level and the `fair-lio` [7], at the block-level as workload generators. Obdfilter-survey [20] is widely used and exercises the obdfilter layer in the Lustre I/O stack for reading, writing and rewriting Lustre objects. Fair-lio, developed in-house and based on libaio, generates parallel and concurrent block-level I/O with both sequential and random workloads to a set of specified local block-level targets.

The benchmark suite contains sections exercising block I/O and file system I/O. The block I/O section comprises of four benchmarks: single host scaling test (block-io-single-host-scale-up.sh), single host full-scale test (block-io-single-host-full-run.sh), scalable storage unit scaling test (block-io-ssu-scale-up.sh), and the scalable storage unit degraded mode full-scale test (block-io-ssu-degraded.sh). All four benchmarks use the `pdsh` and `dshbak` utilities to distribute and collect work. Each benchmark test includes random and sequential I/O requests, small and large request sizes, and read and write I/O operations. Each test may be parameterized to evaluate a variety of dimensions, including queue size, block size and I/O type (sequential write, sequential read, random write, or random read). Tests are administered in random order to eliminate caching effects on test nodes and the storage system, thus providing much more realistic results. Each individual iteration of the test is run for 30 seconds (with 15 seconds in between tests) to obtain statistically meaningful results.

The block-io-single-host-full-run test is used to evaluate the scalability and performance of the storage system under evaluation when accessed from a single I/O server. A single Linux SCSI block device (sd device) from the target host is exercised in this test. The test matrix for this benchmark includes 720 individual tests, for a total run time of 9 hours. The output will contain the standard output of the individual tests, some additional information, and a set of comma-separated values (`.csv`) with the test results and computed statistics.

The script will also create a subdirectory containing individual raw test results for the range of request sizes (4 kB, 8 kB, 16 kB, 32 kB, 64 kB, 128 kB, 256 kB, 512 kB, 1 MB, 2 MB, 4 MB, and 8 MB), I/O patterns (sequential and random), I/O operations (write and read) and queue depths (4, 8, and 16). The script uses these parameters as command line arguments to drive the synthetic benchmark applications, launching them in random order.

The block-io-single-host-scale-up test uses the fair-lio benchmark to execute a randomized set of operations including sequential and random write and read tests. The reads and writes are tested for various block sizes and queue depths for all Linux SCSI block devices, available on a single I/O server attached to a given scalable storage unit for multiple iterations. Similar to the previous test, this benchmark also generates and randomizes test parameters, test modes and operations before execution. Our I/O workload study observed a consistent, high I/O load involving request sizes of 16 KB, 512 KB and 1 MB. These request sizes accounted for over 95% of all read and write requests received by Spider 1. Therefore, the parameter space for this test was reduced to save time. This test evaluates request sizes of 4 KB, 8 KB, 512 KB, and 1 MB. The total runtime depends on the number of target devices tested, but is approximately 1.8 hours times the binary logarithm ($\log_2$) of the number of devices tested. For example, if each host has 5 target devices, the total run time will be 7.2 hours.

The block-io-ssu-scale-up benchmark is designed to evaluate the maximum performance of a scalable storage unit. It exercises all configured Linux SCSI block devices on all target hosts using various I/O test modes and operations. Like its single-host counterpart, this test will run a randomized set of sequential and random write and read I/O tests using the fair-lio binary for various block and queue sizes, for all SCSI disk block devices devices for multiple iterations on all servers. The block and queue sizes selected for this benchmark is identical to those of the block-io-single-host-scale-up benchmark. The total run time of the benchmark is identical to that

of its single-host counterpart.

The degraded mode test, block-io-ssu-degraded, is similar to the ssu scale-up test. It uses identical sets of block sizes, queue depths, I/O modes, and operation parameters. This test uses all Linux SCSI block devices (i.e. RAID arrays or LUNs) on all test hosts and provides the performance profile of the SSU when 10% of the SCSI block devices are being rebuilt. This script will again run a set of sequential and random write and read I/O benchmarks using the fair-lio binary for various block and queue sizes for all Linux SCSI block devices. There are 144 tests in all, for a total runtime of 1.8 hours. For results to be valid, rebuild operations must be induced on 10% of the block devices to be tested and they must last for the duration of the test.

The benchmark suite also provides tools for parsing and plotting the results for the block-level benchmarks. The suite uses gnuplot and ps2pdf to produce graphical output. A sample block I/O plot is shown in Figure 7.

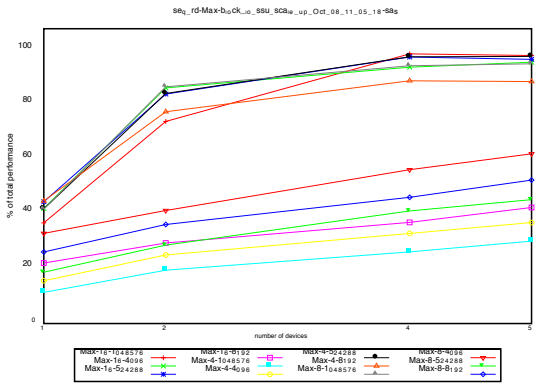


Figure 7: A sample block I/O benchmark plot

We use the Lustre obdfilter-survey benchmark to evaluate Lustre-level scalability and performance. Our tools produce a set of inputs and feed them to the obdfilter-survey. Similar to the block I/O portion of the benchmark suite, these parameters are based on our real-world experience with Spider 1 and our I/O workload characterization study. Our Lustre-level benchmark package includes a script titled obdfilter-survey-olcf. This script is modified from the one included with Lustre distributions and contains the set of parameters specified by OLCF.

To execute this benchmark, one only needs to modify the list of OSTs to be tested. There are no other modifiable variables or parameters. As it is a Lustre-level benchmark, a Lustre file system is required to be formatted on the test hardware in advance. Clients are not required. The benchmark uses Lustre version 1.8. To execute, the benchmark script requires passwordless ssh capability from the head node to the OSSes and between the OSSes, as well. A sample file-system-level benchmark plot is given in Figure 8.

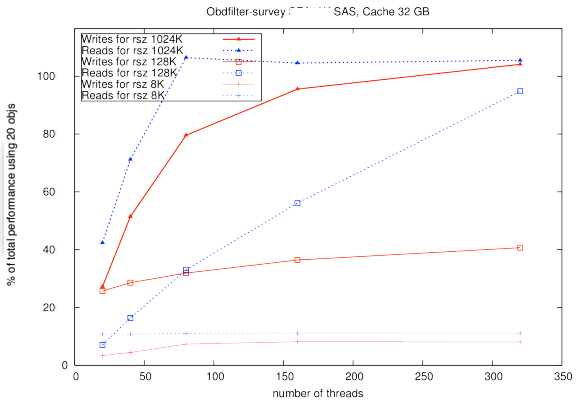


Figure 8: A sample file-system-level benchmark plot

OLCF's benchmark suite was shared with our partners and vendors in 2011 and is publicly available since early 2012. The feedback we have received has been very encouraging.

3.3 Lustre 2.4 Testing

Lustre was chosen as the file system for Spider 1 in 2009. Since then, OLCF has contributed to the Lustre community through testing and development efforts. Our development efforts were three fold: direct funding to Lustre developers for performance and scalability improvements and feature development efforts; support through OpenSFS [19] as a founding member; in-house Lustre development and bug fixing efforts. Most of the features, improvements, and fixes that OLCF funded and required from Lustre were landed in the Lustre version 2.4. Therefore, from early on OLCF's Lustre testing efforts were focused on Lustre 2.4.

OLCF has a well-established Lustre testing methodology, capabilities and resources to perform small and large-scale Lustre tests. A dedicated testbed is available for small-scale testing. Around 64 white-box nodes are provided in the testbed as well as an array of back-end storage devices. Also, a single cabinet of a Cray XK7 system, named Arthur, is dedicated for testing purposes. Arthur has 20 compute clients and is connected to the testbed. These resources allow OLCF to conduct around the clock small-scale Lustre 2.4 tests. Our efforts concentrate on performance, reliability, scalability, recovery, regression, and feature testing.

We download the latest code base from the Lustre 2.4 repository and build its own client and server im-

ages. Identified bugs are reported to Lustre developers. When fixes or patches for these bugs are available (either through the Lustre community or through in-house OLCF Lustre developers), updated versions of Lustre images are re-tested.

After a series of small-scale tests on the white-boxes and Arthur, and when enough confidence is built on the tested Lustre version, a large-scale testing is scheduled. We perform large-scale tests roughly once a month. Titan and portions of Spider 1 are used for large-scale testing. (When formatting Spider 1 storage arrays, a small sliver was intentionally left unused on each array for future testing purposes. Our large-scale tests use these slivers to build the Lustre server test file systems.) An already tested Cray Lustre client image on Arthur is loaded on to the Titan compute nodes for large-scale tests as well. Since obtaining time on Titan is a challenging effort, our large-scale tests are usually condensed in time and happen after normal business hours and over weekends.

Since the beginning of 2013, we have conducted two large-scale tests, with five more planned prior to the commissioning of Spider 2.

The OLCF test set consists of S3D runs, meta data benchmarking, and several combinations of the Lustre file system level I/O test. For meta data testing, we use `mdtest` and `dir-bench`, a tool that was developed in-house. For Lustre file system tests, a combination of IOR runs, at various scales, are used. For all intents and purposes, ORNL tests were targeted for evaluating the pre-Lustre 2.4 functionality and reliability at a large scale.

3.4 Writing and Releasing the RFP

To achieve the best value for the acquisition budget, we used a competitive bidding process to acquire the file system hardware. A request for proposals (RFP) was drafted, containing the requirements and specifications needed for the new file system. In drafting detailed requirements and performance benchmarks, we released its RFP in draft form in August 2011 as a request for comments to acquire industry feedback. Feedback received throughout this process helped refine the specific technical and functional requirements as well as align them with the current technology climate.

As part of the comment period, we publicly released its file system benchmark suite. This enabled prospective bidders to evaluate their solution performance according to the metrics that would later be used for evaluation. In addition, it provided prospective bidders the opportunity to provide feedback on the tests and metrics before the benchmark suite became final. The public benchmark release also provided vendors who were considering entering the high-performance computing market an opportunity to gain insight into the benchmarking processes used by world-class computing facilities such as OLCF.

Shortly before we issued the RFC to the public, Tropical Storm Nock-ten made landfall in Thailand, causing one of the worst floods in the country's history. The flooding lasted six months in many parts of the country; estimates of economic losses attributable to the disaster were estimated at over 1.4 trillion baht ($ 45 billion). Due to Thailand's key role in the supply chain for disk drives, a global shortage of disk drives was anticipated as a result. Responding to the inevitable price shock triggered by the disruption, OLCF delayed the RFP release until prices re-stabilized.

Based on our experience deploying and operating Spider 1 and research into the state of the art of parallel file systems, we decided to deploy Lustre again for Spider 2. An analysis of the current market offerings in high-performance storage indicated that several vendors had available, fully integrated Lustre "appliances," storage systems that included servers with Lustre software installed and supported by the vendor. We structured the RFP to seek bids, offering either a large disk subsystem, with OLCF engineering the Lustre file system in-house, and/or a fully integrated Lustre deployment including vendor-supplied and supported Lustre hardware and software.

In addition to storing the simulation data, Spider 2 is required to function as a key asset in OLCF's data analysis pipeline, serving data to and from from Titan as well as OLCF's other computing and data analysis platforms. Additionally, the system will be required to function as a principal large-scale test platform for new Lustre features, many of which are directly funded, supported, and/or developed in-house. Therefore, significant flexibility is required of the system in order to accommodate these unique needs.

After a careful review of the proposals received, OLCF awarded the contract for the Spider 2 hardware to DataDirect Networks (DDN) for a system based on the SFA 12K40 platform (§4). Hardware for Lustre was separately procured and the engineering effort for Lustre will be conducted in-house. Due in part to the compressed timeline imposed by the delay in issuing the RFP, an aggressive delivery and deployment schedule was required.

OLCF possesses significant in-house Lustre expertise. In addition to engineering and deploying the Spider file system, OLCF contributes significant effort to Lustre in the form of software development, testing, and quality assurance. Regardless, Lustre is still a critical portion

of the OLCF infrastructure that remains under active development and Spider 2 will represent one of its largest deployments.

4 Deploying Next-generation Spider

4.1 Hardware Architecture

Based on the criteria set out in the RFP and our evaluation of the proposed solutions, the evaluation team selected the DDN SFA12K40 Infiniband connected Storage System as the winner. The evaluation team designed a complete filesystem capable of over 1.0 TB/s of aggregate performance with a capacity of 32.5 PB. The filesystems will be deployed using Scalable Clusters (SC) made up of Scalable Units (SU). A total of 4 Scalable Clusters, each containing 9 DDN SFA12K40 couplets and 72 Lustre OSS nodes, 9 FDR switches, and 5 Ethernet switches are to be deployed as individual filesystems. Figure 9 shows the high-level overview of the Spider 2 architecture and how it is connected to other OLCF resources. Below we talk about our initial deployment of this hardware and about future work that will change the deployment based on recent and upcoming additions to the Lustre server code base. We also describe the individual components, software architecture, and integration efforts that will be deployed at the OLCF.

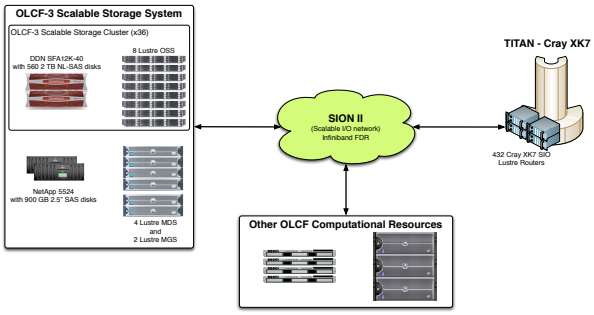


Figure 9: Overall architecture of Spider 2

4.1.1 Storage Backend

Each SFA12K40 storage subsystem is compromised of two SFA12K40 controllers, 10 SS7000 60-slot enclosures, an automatic transfer switch, two APC UPS units, two power strips, and a 50RU rack enclosure. The unit takes as power input 2 x 50A 3PH 208V inputs, with a maximum observed draw of 9kW of energy. There are 560 3.5 inch 2 TB Near-Line SAS drives in each couplet. Each couplet has 8 Lustre OSS nodes attached to each controller via FDR Infiniband. Each Disk enclosure is attached via 4 6 Gb/s SAS 4x ports to each SFA12K40 storage controller.

The storage subsystem will be put through stringent acceptance tests that are based on the I/O benchmarks described in Section 3.2. Additional to peak performance tests, the system will be placed in various failure scenarios for performance and reliability testing. The system will be evaluated for data corruption, and finally evaluated for single points of failure in the data path.

4.1.2 File system I/O Servers

The Dell C6220 server platform was chosen to drive the DDN SFA12K40s. This platform offered 4 nodes in a 2 RU form factor. Each node contains an on-board FDR IB port as well as a single PCIe 3.0 x16 slot. This slot will contain a dual port Mellanox FDR Card that will connect to each of the 2 storage controllers. The Dell servers have 2 on-board Ethernet ports and an additional Ethernet port for dedicated IPMI/BMC. Each server chassis has 2 1200W power supplies. Two chassis units are dedicated to each DDN SFA12K40. These individual nodes are setup as fail-over pairs with the partner nodes residing in separate chassis.

4.2 New Scalable I/O Network

Using the information learned and briefly deployed on Jaguar and Spider 1, we will be implementing a Fine-Grained Routing [7] configuration for Spider 2. A total of 432 LNET Routers will connect to 36 1RU 36 port FDR switches that contain 8 Lustre OSS nodes. Each of those switches will contain 1 uplink to each of 2 108 port Aggregation switches. The aggregation switches contain direct connections for all the MDS, MGS, and file system management systems. In addition to these nodes, the Aggregation switches provide Inter-Switch Links (ISL) to another 108 port switch that will provide connectivity for the remaining OLCF resources such as Data Transfer Nodes, Visualization and Analysis platforms, and smaller compute clusters. This design shows a preference to Titan in terms of performance, but Titan is the only platform that has the ability to saturate more than the ISL capacity, so this trade-off is acceptable. The network has been designed such that additional compute clusters will only require the addition of more ISLs and LNET routers. Each new compute cluster will be purchased with LNET routers; existing systems will be accommodated and phased out as they are retired. Figure 10 shows a detailed diagram of the Spider 2 link and interconnect speeds.

The use of LNET routers will allow us to take a step forward in the SION; Spider 1 did not include

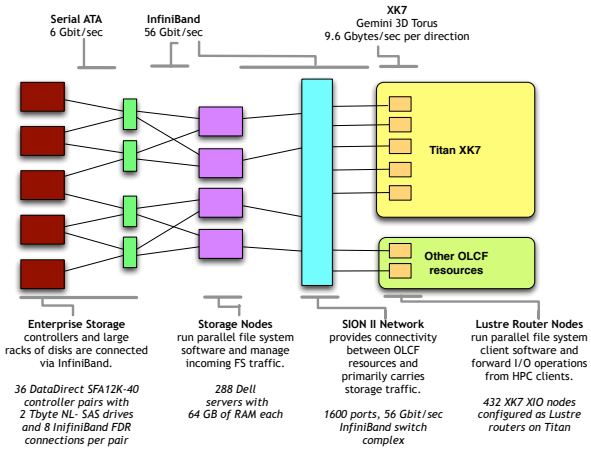


Figure 10: Spider 2 link speeds

LNET routers for the remaining OLCF resources, and that caused traffic such as MPI to traverse the core Infiniband fabric. With the addition of LNET routers, we effectively create several sub fabrics that have interconnection points and separate any job related traffic from the core Infiniband fabric, leaving it dedicated for Lustre traffic only. We feel that this is a large improvement.

Each of the 432 LNET routers for Titan will utilize a single port Mellanox ConnectIB FDR card in a Cray XIO node. In the base configuration, each of these LNET routers can pass 2880 MB/s of traffic. Testing has shown that if there were a mitigation for requiring the data to be checksummed (requirement imposed by Cray's Software stack), this number could be as high as 5400 MB/s. This work is left for the future as an enhancement to Lustre and the Cray Gemini interconnect.

4.3 Software Architecture

In this section we detail our plans for deploying Lustre at the OLCF on Spider 2. Additionally, we describe our tools for managing the storage, servers, and the filesystem itself.

4.3.1 Lustre

The OLCF, as a member of OpenSFS [19], is an active partner in the development of new features for Lustre. In addition to OpenSFS funded activities, the OLCF allocates non-recurring engineering (NRE) development funds to Lustre for priorities that are important to the users. The OLCF has seen an incredibly useful life out of the Lustre 1.8 code base, but as we begin the deployment of Spider 2, we look to the 2.x code branches to move forward. We have named the filesystems of the next generation of Spider "Atlas."

Initial Deployment Our initial deployment of the Atlas filesystems will be based on the Lustre 2.4 code base scheduled for release in early May 2013. As was discussed in Section 3.3, the OLCF is undertaking efforts to assist in the hardening of this release of the Lustre software. The 2.4 release has also been deemed a maintenance release – meaning that it will undergo a deeper inspection/testing process and will get regular bug fix-only updates over a longer duration. Currently maintenance releases receive quarterly bug fix updates for a duration of 18 months.

OLCF has 36 couplets of DDN SFA12K40 deployed in 4 SC's; each with 9 SU's. Our initial plan will to be deploy 3 production filesystems - two with 1 SC each, and one with 2 SCs. We are focusing on delivering production ready resources for Titan and then attacking potential scaling issues with future work noted below.

Future Deployment Options Released in the 2.4 release is a new feature called Distributed Namespace Phase I (DNE)[10]. DNE allows multiple Lustre Meta-Data Servers to exist in the same filesystem. This feature will not be deployed in the initial deployment as a management of risk. The undertaking of deploying Spider 2, while continuing to manage Spider 1, and then the eventual decommissioning of the Spider 1 hardware was a large enough risk without adding new features into the mix. The OLCF is working closely with the Lustre Developers and Community to understand the steps necessary to transition to DNE based filesystems in the near term.

4.3.2 Cluster Management & Monitoring

GeDI, PowerMan, ConMan The OLCF will continue to use the tools that we used to deploy Spider 1 for Cluster Management. The Generic Diskless Installer [13] uses DHCP, PXE, TFTP, RSYNC, and then NFS read-only mounts the root filesystem. The PowerMan package provides a wrapper where we can interface with both the BMC on the motherboard of the Dell C6220 nodes, but also with the IPMI interface of the Geist power strips. ConMan allows us to access and store the output of the IPMI Console, saving costs on additional cables, adapters, and console servers.

High Availability The OLCF will be deploying Lustre fail-over when the Atlas filesystems transition into production. To manage fail-over the OLCF is using linuxHA with heartbeat version 3 [9]. The setup will contain 144 2-node fail-over clusters. The OLCF has developed scripts that will STONITH nodes as appropriate

to ensure data integrity during a fail-over. Several conditions must be met before a node will be STONITH'd, and the HA software is only configured to fail-over – fail-back will be a manual process.

Health Monitoring of Storage As part of SFAOS 1.5.0 and greater releases of the DDN firmware, a Python based API is available for configuring the storage as well as querying the status of the hardware. The OLCF is working to develop health monitoring for the storage controllers – power supply status, disk enclosure status, temperature, voltage, UPS status, Operating System drive mirror status, and more. This monitoring will be tied into our centralized monitoring service based on Nagios [14], that will send email and SMS alerts to the filesystem and hardware administrators based on the results of the check.

Infiniband Health Monitoring The OLCF HPC Operations staff are working to develop additional Infiniband Health Monitoring based on the tools that are part of OFED. The OLCF has already written HCA health monitors to ensure that they are at the correct speed and is being slightly modified to make it FDR aware. Additional efforts have been undertaken recently to provide a graphical representation of the IB fabric. Future work here will involve integrating the Port counters to display ports that are experiencing high traffic or errors. This could then be integrated into centralized monitoring and administrators could be notified automatically.

Server Monitoring Monitoring of both the C6220 chassis and the individual nodes in the chassis will be obtained using Dell's Open Manage [6] systems management. This work extends on the existing work to monitor the Spider 1 hardware. Additionally, monitoring the status of the multipath connections to the DDN SFA12K40s to ensure that the redundant paths are up and ready in case of a failure.

Lustre Monitoring The OLCF has already developed several pieces of monitoring infrastructure to determine if the Lustre filesystem is healthy. These include monitoring the flow of LNET messages through the LNET routers and Lustre Servers, determining if the proper devices are mounted, and the status of the /proc/fs/lustre/health_check file. Some of these checks will need to be modified to be fail-over aware – they currently will report "error" conditions if a Lustre server were to fail and its OSTs fail-over to a partner.

4.3.3 Tools

We use a variety of custom tools for monitoring and maintaining the current filesystem. Since we are moving to newer hardware and software, most of these tools will need at least some modification and a few will be replaced altogether.

DDNTool DDNTool [12] is a utility for monitoring the DDN controller hardware. It continuously polls the controllers and reports on things like failed drives and the progress of rebuilds. It stores its results in a database for easy access and analysis.

The original utility was a C++ program that communicated over DDN's S2A API. The DDN SFA hardware that Atlas will use has a completely different API and DDN provides a client library that simplifies access. The client library is written in Python, however, so the new version of the DDNTool will need to be rewritten in Python.

LustreDU LustreDU [3] is a tool designed to provide filesystem usage information without the performance issues associated with running the regular du utility from a Lustre client. It relies on the ne2scan [15] utility from Nick Cardo and consists of a single master process that runs anywhere and slave processes that run on each OSS. The master process reads the output from ne2scan and queries the appropriate slave for the size of each object in the filesystem. The results are posted to a database and a separate web application makes those results available to the users.

LustreDU was originally written for Lustre v1.8.x and expects a particular layout for the objects on the OST's. Since Atlas will run Lustre v2.x and the layout of the OST's has changed, the LustreDU slave processes will have to be modified to be able to locate objects in the new layout. We will also need to verify that ne2scan will work when run against a Lustre 2.x MDT. However, no changes should be needed for the LustreDU master process.

Robinhood On the current filesystem, stale files are subject to purging after 2 weeks of inactivity. This purging is handled by two utilities, called GenHit and Purge.[12] GenHit filters the output of ne2scan for a specific date range. Purge takes those results, double-checks that each file's mtime, ctime, and atime meet the requirements for purging and if so, unlinks the file.

The current utilities would continue to work on the new filesystem, however we are looking at moving to a new policy engine called Robinhood. Robinhood is

being developed by CEA for use on their Lustre filesystem and offers an advantage over ne2scan, GenHit and Purge by consuming the Lustre v2.x changelog, and thus keep itself updated in near real-time. That should eliminate the need to scan the filesystem periodically with ne2scan and LustreDU, although those utilities will be kept around in case they are needed.

4.4 Facilities

The Spider 2 file system resides in the National Center for Computational Sciences (NCCS) computing facility, adjacent to both the original Spider 1 and the Titan supercomputer. This space is a traditional 36" raised floor data center, with direct chilled water delivery to the large supercomputer systems, and forced air delivery from the below-floor plenum for air-cooled systems. The physical file system layout uses four rows, each with nine racks for DDN SFA12K-40 equipment, and one infrastructure rack. The total space required for the filesystem is 672 square feet. The equipment is installed in a hot-aisle/cold-aisle configuration, where adjacent inlet/supply-side systems face each other, and outlet/return-side heat is discharged into a common aisle. In addition, the cold aisles are fully contained, with both overhead panels and sliding doors at each end of the rows. This configuration eliminates hot air/cold air air mixing that can result in inconsistent supply temperatures both within a rack, and among adjacent racks. In addition, it reduces air flow requirements in to the cold aisles, since the cubic feet per minute (CFM) requirement is reduced to the total CFM requirement of the equipment in each set of 18 racks, with no superfluous air supply needed. 25% perforated tiles are used to allow cold air supply to the cold aisles. The cold aisle solution is fully compliant with the requisite National Fire Protection Association (NFPA) codes. Hot aisle waste heat is ejected to the computer room, and is recycled by a typical configuration of wall-positioned CRAC units.

OLCF tested a single rack configuration under multiple scenarios and determined that the consumption on that single rack under multiple scenarios, including high volume sequential reads, sequential writes, random reads, random writes, and mixed workloads demonstrated a nominal load of 9kW per rack, with very little fluctuation. This allowed OLCF to very precisely engineer the electrical distribution system, and calculate the anticipated operational mechanical load. Total filesystem load, including the infrastructure racks is 400kW. Total cooling load is 114 tons. Each filesystem rack is fed with a pair of 208VAC 3-phase electrical feeds, protected by a 50A 100%-rated breaker. In addition, each pair of electrical connections is fed from two different transformer sources, i.e. the A-side is fed from one transformer, and the B-side is fed from a second transformer. Because the load on a single rack is less than (approximately 50%) of the capacity of a single electrical feed, and because the DDN SFA12K power distribution system is both load balanced and supports failover, OLCF can conduct both scheduled and unscheduled maintenance on one transformer without disrupting the filesystem operation. Neither electrical connection is protected by UPS.

4.5 Putting it all together

The OLCF has spent the last 2.5 years working through the process of developing requirements, developing benchmarks, waiting for hard disk prices to come back into line, issuing, and executing the RFP. We are currently underway with landing hardware on the computer room floor and beginning initial checkout. As we enter and complete acceptance we will be delivering a Lustre 2.4 filesystem that will be capable of over 1 TB/s to the scientific user community of the OLCF.

Through our partnerships with vendors we have were able to architect a solution that is a 400% improvement over Spider 1, fits very closely within the same power envelope, and has a smaller footprint in the data center. Lessons learned in the Spider 1 deployment shaped the facility requirements and how we will deploy the system. We are delivering not only a bandwidth increase for the scientific users of the OLCF, but a 3x capacity increase that is much needed. This increase in capacity should allow us to serve the needs of Titan for the life of the machine.

We are also deploying a system that has the capacity to expand its performance. With the potential of DNE deployments we can more finely target users and projects that have unique I/O patterns and attempt to minimize the impacts of those unique patterns on the remaining users/projects. As we designed in Spider 1, the deployment for Spider 2 is in Scalable Units and Scalable Clusters, allowing the OLCF to build upon if more performance or capacity is desired.

Each of the sections above describes an effort that is integral to providing a leadership class resource for the users of the OLCF. Without any one of these pieces the next generation Spider could fail to meet the demanding requirements of the users of the OLCF.

5 Conclusions and Future Work

Since the deployment of Spider 1, the OLCF has been actively engaged in planning and preparing for delivering Spider 2. Fitting within the same power envelope

and foot print as its predecessor, Spider 2 will deliver more than 1 TB/s aggregate I/O performance, while providing 32.5 TB of capacity to the scientific user community of the OLCF.

OLCF has reflected upon lessons learned from the deployment, commissioning, and operating Spider 1 for delivering Spider 2. While the process is not complete, we are making progress and are on track to start the acceptance phase of Spider 2. Acceptance, integration, commissioning, and user transition from Spider 1 to the Spider 2 are some of the tasks that need to be addressed soon. The scientific user community at OLCF will be able to begin using the advanced capabilities of Spider 2 in 2013.

References

[1] 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.

[2] A. S. Bland, J. C. Wells, O. E. Messer, O. R. Hernandez, and J. H. Rogers. Titan: Early experience with the Cray XK6 at Oak Ridge National Laboratory. In *Proceedings of Cray User Group Conference (CUG 2012)*, May 2012.

[3] A. G. Carlyle, R. G. Miller, D. B. Leverman, W. A. Renaud, and D. E. Maxwell. Practical support solutions for a workflow-oriented Cray environment. In *Proceedings of Cray User Group Conference (CUG 2012)*, 2012.

[4] Cray Inc. Cray XK6. `http://www.cray.com/Products/XK6/XK6.aspx`.

[5] Cray Inc. Cray XK7. `http://www.cray.com/Products/Computing/XK7.aspx`, 2012.

[6] Dell. Dell OpenManage systems management. `http://www.dell.com/content/topics/global.aspx/sitelets/solutions/management/en/openmanage?c=us&l=en&cs=04`, 2013.

[7] D. A. Dillow, G. M. Shipman, S. Oral, and Z. Zhang. I/O congestion avoidance via routing and object placement. In *Proceedings of Cray User Group Conference (CUG 2011)*, 2011.

[8] J. Dongarra, H. Meuer, and E. Strohmaier. Top500 supercomputing sites. `http://www.top500.org`, 2009.

[9] F. Haas. Linux high availability. `http://www.linux-ha.org`.

[10] Intel. Lustre distributed namespace. `https://wiki.hpdd.intel.com/display/PUB/DNE+1+Remote+Directories+High+Level+Design`, 2012.

[11] Y. Kim, R. Gunasekaran, G. M. Shipman, D. Dillow, Z. Zhang, and B. W. Settlemyer. Workload characterization of a leadership class storage. In *Proceedings of the 5th Petascale Data Storage Workshop Supercomputing '10 (PDSW'10) held in conjunction with SC'10*, November 2010.

[12] R. Miller, J. Hill, D. A. Dillow, R. Gunasekaran, G. M. Shipman, and D. Maxwell. Monitoring tools for large scale systems. In *Proceedings of Cray User Group Conference (CUG 2010)*, May 2010.

[13] M. Minich. GEneric Diskless Installer. `http://sourceforge.net/projects/gedi-tools/`, 2009.

[14] Nagios Enterprises. Nagios. `http://www.nagios.org`.

[15] Nicholas P. Cardo . Reaping the Benefits of Meta-Data. `http://wiki.lustre.org/images/d/df/Cardo_LUG2010.pdf`, 2010.

[16] Oak Ridge Leadership Computing Facility. OLCF I/O evaluation benchmark suite. `http://www.olcf.ornl.gov/wp-content/uploads/2010/03/olcf3-benchmark-suite.tar.gz`.

[17] Oak Ridge Leadership Computing Facility. Titan Cray XK7. `https://www.olcf.ornl.gov/computing-resources/titan-cray-xk7/`, 2012.

[18] Oak Ridge National Laboratory, National Center for Computational Sciences. Jaguar. `http://www.nccs.gov/jaguar/`.

[19] OpenSFS. OpenSFS. `http://www.opensfs.org/`, 2013.

[20] Oracle Inc. Benchmarking lustre performance (lustre I/O kit). `http://wiki.lustre.org/manual/LustreManual20_HTML/BenchmarkingTests.html`.

[21] G. M. Shipman, D. A. Dillow, D. Fuller, R. Gunasekaran, J. Hill, Y. Kim, S. Oral, D. Reitz, J. Simmons, and F. Wang. A Next-Generation Parallel File System Environment for the OLCF. In *Proceedings of Cray User Group Conference (CUG 2012)*, May 2012.