Deploying a Parallel File System for the World's First Exascale Supercomputer

Jesse Hanley
Oak Ridge National Laboratory
Oak Ridge, TN
hanleyja@ornl.gov

Dustin Leverman
Oak Ridge National Laboratory
Oak Ridge, TN
leverman@ornl.gov

Christopher Coffman
Oak Ridge National Laboratory
Oak Ridge, TN
coffmancd@ornl.gov

Bradley Gipson
Oak Ridge National Laboratory
Oak Ridge, TN
gipsonbm@ornl.gov

Christopher Brumgard
Oak Ridge National Laboratory
Oak Ridge, TN
brumgardcd@ornl.gov

Rick Mohr
Oak Ridge National Laboratory
Oak Ridge, TN
mohrrf@ornl.gov

***Abstract*—The world's first exascale supercomputer, OLCF's 'Frontier', debuted last year and is allocated for INCITE awards this year. OLCF partnered with HPE to design, procure and deploy a parallel file system to support the demands of this new machine. This file system is based on the ClusterStor E1000 storage platform and has been integrated into the OLCF site.**

With a useable namespace of 679PB, this cluster employs several newer features in Lustre to provide a solution that combines the performance of NVMe and the capacity of traditional hard disk drives. We present the architecture and configuration of this system and detail the steps taken to operationalize the file system cluster. The authors aim to provide the contents described as a community resource for others that are designing or deploying storage systems.

I. Introduction

The Oak Ridge Leadership Facility (OLCF) at the Oak Ridge National Laboratory (ORNL) is dedicated to providing researchers with leadership-class compute and data resources that enable groundbreaking scientific discoveries[1]. In this paper, we delve into the deployment strategies for the Orion storage cluster and discuss the acceptance process employed to qualify its components and ensure optimal performance. We discuss the design and layout of the storage cluster, detailing the various components that contribute to its functionality and efficiency. Moreover, we explore the tooling required for seamless site integration efforts, which play a crucial role in the successful implementation of Orion. Finally, we present highlights of the storage acceptance and testing process which provide insights into the thorough validation procedures for the reliability, functionality, and performance of the Orion cluster.

II. Design

The Orion cluster consists of a set of dedicated Lustre servers, a set of Lustre LNET routers, the supporting cluster infrastructure, and the associated network gear. There are five physically identical I/O Scalable Units (IOSUs) and a dedicated management cabinet. Each of the IOSUs consists of ten 50U rack cabinets that form a row in the data center.

A. Layout

For each IOSU, there is a central rack designated for infrastructure. In each of the infrastructure cabinets, there are 16 HPE Slingshot switches, 32 gateway router (RTR) nodes, and 4 Metadata Units (MDUs). Each RTR node is a HPE ProLiant XL225n Gen10 blade that fits in a 4 sled chassis, for a total of 8 chassis per infrastructure cabinet. Each MDU consists of an E1000 chassis that supports 2 Lustre Metadata Servers (MDSs) and 24 shared U.2 NVMe PCIe Gen4 slots for a total of 8 MDSs per infrastructure cabinet. The MDUs are populated with 30 TB KIOXIA NVMe solid state drives (SSDs), for a total raw capacity of 720 TB per MDU and an aggregate 14400 TB of raw capacity across the 40 MDSs.

The other 9 cabinets each contain 5 Scalable Storage Units (SSUs). Each SSU consists of the same E1000 chassis but the two nodes serve as 2 Lustre Object Storage Servers (OSSs). Additionally, each SSU is populated with 3.2 TB Samsung PM1733 NVMe SSDs for a total raw flash capacity of 76.8 TB per SSU and an aggregate 17280 TB of raw flash capacity across the 450 OSSs. Each SSU also has

two Serial Attached SCSI (SAS) hard drive disk enclosures. Each enclosure contains 106 18 TB Seagate Exos X18 hard disk drives (HDDs) for a total raw HDD capacity of 3816 TB per SSU and an aggregate 858600 TB of raw HDD capacity across the 450 OSS nodes.

The cluster system management units (SMU) leverage the same E1000 building block for shared SSD storage, but are populated with 6 of the Samsung PM1733 NVMe SSDs. Those drives are configured in a Linux software RAID6 using the MD (Multiple Devices) device drivers and then formatted as an EXT4 volume. The SMU nodes operate in a hot/warm capacity with one node mounting the volume at a time.

To minimize service interruptions, each subsystem was designed for redundancy while also keeping a careful balance of cost and performance efficiency. Each E1000 building block has two HPE Slingshot 200 Gbps network adapters while each of the RTR nodes has a single adapter. For the E1000 nodes, the two HPE Slingshot adapters are used with Lustre Mail-Rail. Additionally, the Lustre servers are configured with two Ethernet management ports that are configured as a bonded interface during system initialization. The Lustre servers run Pacemaker [2] and Corosync [3], which ensure the ZFS zpools are available across one of the nodes.

B. Management Software Stack

The SMU servers provide a hot/warm management stack. The active node handles primary responsibilities for cluster management with the secondary node in standby. These responsibilities include cluster provisioning, centralized log aggregation, remote power and console management, as well as some cluster health checks.

Orion uses a combination of open source tools for provisioning. The primary management server runs a `dnsmasq` [4] instance that provides DHCP, TFTP, and PXE services for the Orion cluster. A periodic `systemd.timer` [5] launches a service that scans the Ethernet management network. Given that each node plugs into a specific set of switch ports, this scanner builds out a mapping of switch port to node MAC addresses, which is then used to refresh the `dnsmasq` config. This periodic scanner allows for node replacement actions to be completed without manual configuration updates.

Storage images are built using container management tools. We use Buildah [6] to create a base operating system image, similar to a traditional chroot environment [7]. This image is then customized using our site local Puppet [8] configuration management tool before being compressed into an immutable `squashfs` image. As part of the image build process, we inject the Anchor [9] dracut module into the image's initrd. Additionally, the build process installs the HPE provided Linux kernel and relevant data path packages. Some of these include the Lustre, ZFS, High Availability (HA) and firmware packages.

The SMU servers also run common [10] cluster management tools such as `powerman`[11], `conman`[12], and `clustershell`. This stack allows for a clear audit trail of system configuration and a stable management environment while maintaining a total cluster boot time of around seven minutes.

C. Namespace Configuration

Analysis of the file distribution within the OLCF Alpine file system [13] led us to create a tiered layout for Orion that would give the best performance to the majority of users. In order to minimize initial operational risk, we chose to create a single namespace default layout that would make appropriate use of the metadata, performance, and capacity storage. Future work will consider data migration between the storage tiers. Orion uses three Lustre features to provide this namespace default configuration: Progressive File Layout (PFL), Data on MDT (DoM), and Self-Extending Layouts (SEL). Based on the analysis of Alpine and historic file system usage, we estimated that 70% of the files would live entirely in DoM if we sized that component to 256KB. Additionally, this distribution would mean that another 18% of files would not exceed an 8MB performance tier component, leaving 12% of the files extending into a capacity tier component. Though lower in total count, these files that extend into the capacity tier are expected to account for more than 98% of the total file system space usage. The size of the DoM and performance tier components were sized conservatively in order to minimize the future need for file restriping. This also reserves enough performance space capacity for projects that may need more fine-tuned striping. The SEL configuration is used as a safeguard against an OST that might run out of space, allowing user code to continue without failing with an `ENOSPC` error. The layout is set on the top level of the namespace with:

```
/usr/bin/lfs setstripe -E 256K -L mdt \
-E 8M -c 1 -S 1M -p performance -z 64M \
-E 128G -c 1 -S 1M -z 16G -p capacity \
-E -1 -z 256G -c 8 -S 1M -p capacity \
/lustre/orion/
```

III. System Observability

Orion is configured to monitor, self-repair, and alert appropriately in order to detect issues before science workloads are impacted. With such a large and complex system, there is a high likelihood of a failure occurring. Though there are not always strict distinctions between different components, monitoring is grouped into hardware, software, HPE provided tooling, and namespace health.

For the hardware components, a core resource is the Intelligent Platform Management Interface (IPMI) [14]. This, in conjunction with DMTF's Redfish [15] standard,

provides access to a variety of hardware level sensors and event logs. For the OSS nodes, we monitor various parts of the Serial Attached SCSI (SAS) subsystem, including cable health, error counter rates, and issues with negotiations between enclosures and the corresponding host. Equally important is the Hard Disk Drive (HDD) health, where we monitor for failing drives and slow drives. On all nodes that have NVMe storage, we verify that the drives have negotiated the correct PCIe communication speed and that they do not report errors. We build a catalog of firmware versions for each component in the system and compare that against a manifest of specified values.

Next, we deployed a series of health monitoring scripts that poll the current LNET stats of a node and compare it against a previous run. This allows us to detect network errors before they impact user jobs. Additionally, we verify that any routes and Lustre tunings are in place. We ensure that our configuration management tool runs according to our cyber security guidelines and alert if there are issues applying any of the configuration or if the agent has not run recently. Our configuration management tool ensures that standard services like `chronyd`, `crond` and `syslog` daemons are correctly started.

We leverage HPE's tooling for monitoring specifics to this hardware platform. Additionally, the HPE tooling provides a robust disk monitoring framework that proactively monitors, diagnoses and alerts on storage media issues. These tools integrate into the High Availability toolchain each Lustre server runs. In addition to the storage and host related tooling, there are diagnostic and monitoring tools for the HPE Slingshot fabric. We deployed health checks that convert the information available in the HPE tooling into a form our site-wide monitoring can ingest.

Finally, we have a series of namespace health checks. These measure space utilization, storage target availability, file system latency, and general health of the cluster. We leveraged our existing Nagios [16] infrastructure and configured each check with appropriate timeout conditions, dependency and retry logic, and alert notifications. This tooling allowed us to constantly monitor the system as a whole and escalate alerts as needed.

IV. Testing and Acceptance

Acceptance testing is a multifaceted process that ensures the delivered storage system achieves established performance goals and meets features needed for a transition to operations (T2O). Though some specifics are omitted from this paper, the authors provide a general overview of the process. Acceptance is comprised of a series of tests that were logically split into four different phases. Some tests had chronological overlap between phases which ensured maximum usage of staff and facility resources. Each phase has a series of tests and each test focuses on **hardware**, **functionality**, **performance**, or **stability**.

- **Hardware Test**: Physical installation
- **Functionality Test**: Demonstrate basic functionality meets resiliency, reliability, and operational needs
- **Performance Test**: Measure of hardware/software performance requirements
- **Stability Test**: Verification that the storage cluster can withstand a workload similar to operational conditions

A. Phase 1

The first phase of acceptance focused on physical delivery and installation. Before production systems arrived, OLCF staff deployed development systems to test cluster features and prepare configuration management data. Orion hardware was subject to the following tests as part of site integration efforts:

- **Cabinet Installation**: Members of the acceptance team compared each cabinet against the Bill of Materials (BOM). HPE staff loaded drives into the appropriate chassis, after which the cabinets were safely powered on. The cabinet containing the SMU was given priority during this process.
- **Cabinet Rear Door Heat Exchanger (RDHX)**: Site facility engineers integrated RDHX with appropriate water cooling systems and calibrated the cooling to meet anticipated demand.
- **Vendor Hardware Checkout**: Once the cabinets were powered on, HPE staff conducted a series of hardware verification tests to ensure the health of all components.
- **Cabinet Network**: As each cabinet finished Vendor Hardware Checkout, OLCF staff reprovisioned the delivered hardware to comply with site-specific cybersecurity policies.
- **Cabinet Boot**: After the network switches in the SMU cabinet were hardened, staff deployed the SMU node pair and verified that the nodes could pull an image over iPXE.
- **Firmware Upgrades**: The team ensured that firmware could be updated in a timely fashion, logged, and verified.
- **Cabinet Labeling and Location Verification**: A series of tests were performed to toggle location beacons and verify the physical location of hosts.
- **Drive Format**: The acceptance process included reformatting the drives and ensuring any drive features were consistently configured.
- **HDD Burn In**: Repeated runs of various I/O patterns enabled the acceptance team to identify slow, failing, or underperforming components.
- **Hardware Serviceability**: The team conducted pull tests to verify that disk enclosures and server components could be serviced.
- **Format Zpools**: The team verified that zpools could be built across disk enclosures.

- **LNET NIDs**: The team verified that kernel modules could be loaded successfully.

B. Phase 2 - Single Unit

Phase 2 of acceptance consisted of a mix of hardware, functionality, and performance tests that focused on a single storage building block. Some tests were executed on each building block, while others used a representative unit from each type of building block. This ensured that SMU, MDU, RTR, and SSU nodes were production ready. The tests were run with representative clients and the servers nodes were configured with their production-ready tunings.

1) Hardware: Following the initial acceptance phase, the team proceeded with more hardware-focused tests that ensured the performance and reliability of the Orion cluster. A subset of these tests included the following:

- **Power Distribution Units (PDUs)**: The team evaluated the PDUs to ensure they were functioning properly, could be reached over the network, and that the state of each port could be toggled.
- **Remote Power Control**: The team tested the ability to remotely control nodes through both PDU outlets and node BMCs, ensuring that any combination of nodes could be powered on, off, or reset.
- **Rear Door Heat Exchanger (RDHX) Health Monitoring**: Simulated failures allowed the team to verify appropriate monitoring was in place.
- **Monitoring**: Orion was integrated into the site monitoring solution so that the performance, utilization, and overall health of all components could be observed and appropriate actions could be automatically triggered.
- **Site Cybersecurity Integration**: The team integrated the cluster with site-specific configuration management and verified that the appropriate cybersecurity policies were enforced.
- **Switch/Server Power Feed Resiliency**: The team tested the resiliency of switches and servers power feeds for switches and servers, evaluating their ability to maintain uptime in the event of power fluctuations or failures.
- **SAS Network Resiliency**: The team assessed the resiliency of the Serial Attached SCSI (SAS) network and simulated a number of failures, including at the cable and IOM layer.
- **High-Speed Network (HSN) Resiliency**: The team evaluated the resiliency of the high-speed network infrastructure, ensuring its ability to handle traffic loads and maintain performance under various conditions.
- **Serviceability and Cable Management**: The team examined the overall serviceability of the cluster, focusing on cable management to ensure easy access, efficient maintenance, and reduced chances of cable-related issues.

These hardware-focused tests helped the team to further ensure that the Orion cluster was robust, reliable, and prepared to handle the demands of its users. This thorough testing allowed the team to identify and resolve potential issues before additional layers of complexity were introduced.

2) Functionality: After completing the hardware-focused tests, the team proceeded with functionality-focused tests that generally reflected tasks anticipated to occur during production. A subset of these tests included the following:

- **Power Cycle Resiliency**: The team tested the cluster's ability to survive repeated power cycles and still be serviceable.
- **Location Beacon Test**: The team tested the accuracy and reliability of the location beacons used for identifying the physical location of servers within the data center.
- **Command-line Based Firmware Upgrades**: The team verified the functionality and ease of use for command-line-based firmware upgrades, ensuring the process could be automated and logged.
- **Disk & NVMe Replacement and Fault Injection**: The team tested the process of replacing failed or malfunctioning disks and NVMe drives, as well as injecting faults to evaluate the system's ability to detect and handle drive failures.
- **Disk & NVMe Rebuild and Rebalance**: The team measured the system's ability to recover data in a zpool after a drive replacement was completed.
- **Disk Variability**: The team evaluated each building block's associated disk performance characteristics to identify misbehaving or problematic components.
- **ZFS Parity Check on Read**: The team tested the ZFS file system's ability to perform parity checks on data reads, ensuring data consistency and integrity.
- **High Availability**: The team assessed the overall high availability of the cluster, evaluating the SSU's ability to fail over resources and failback.

3) Performance: Following the functionality-focused tests, the team conducted single building block performance tests performance tests to evaluate the speed and efficiency of various components and layers within the Orion cluster. A subset of these tests included the following:

- **Individual NVMe/Drive Performance**: The team measured the performance of individual NVMe drives and hard disk drives, ensuring they met or exceeded expected benchmarks.
- **ZFS Dataset Performance**: Once individual disk performance concluded, the measured performance at the ZFS dataset layer.
- **Lustre Layer Performance**: The team assessed the performance of the Lustre file system using tools such as obdfilter-survey and mds-survey.
- **Metadata Performance**: The team evaluated the

performance of metadata operations, ensuring efficient handling of file and directory metadata.
- **Performance Tier**: The team assessed the performance of the high-speed storage tier, ensuring optimal speed and throughput for high-demand workloads.
- **Capacity Tier**: The team evaluated the performance of the capacity-focused storage tier, ensuring sufficient storage space and efficient data management.
- **Namespace Defaults**: The team tested the performance and functionality of the cluster's namespace defaults, ensuring smooth operations and optimal resource usage.

4) Phase 3 - Scale Up: As building blocks completed phase 2 of acceptance testing, they were added to a clustered namespace. This namespace was frequently reformatted. The team ran versions of Phase 2 tests with various numbers of participating SSUs to observe scaling behavior.

C. Phase 4 - Full-Scale tests

Once all building blocks completed Phase 3 acceptance, Phase 4 could begin. This phase was intended to test the system under simulated production workloads and day-to-day operations.

1) Functionality:

- **Online Lustre `lfsck`**: The team artificially aged the namespace and populated the file system with data and billions of files. They then verified that an online Lustre `lsfck` completed sucessfully within a reasonable time window.
- **Metric collection**: The team verified that key system data about the health and performance of the namespace were regularly collected. They also verified that the data was accurate, timely, and provided insight into the function of the file system.
- **System management**: Various cluster management functions were evaluated, such as creating immutable images, system upgrade and rollback, remote management and provisioning, and crashdump collection.
- **RTR failover**: Varous power events, such as power cycling and shutdown, were applied to a number of the Orion Lustre LNET routers while traffic was in flight. This test ensured that off-cluster clients did not receive I/O errors.
- **EPO**: The team verified the system could recover from a controlled Emergency Power Off (EPO), where nodes were not gracefully stopped. Recovery and normal system operations were evaluated to ensure an EPO would not lead to loss of system data.

2) Performance:

- **LNET selftests**: Various combinations and groups of Lustre servers, Frontier clients, and other miscellaneous endpoints were grouped together in batches of LNET Selftest runs. These groups were stressed using various communication patterns in order to verify the cluster could see sufficient network bandwidth in a variety of cases.
- **External cluster**: This test was designed to demonstrate the streaming performance, latency, and IOPS achievable from a cluster of Lustre clients external to the Frontier network fabric.
- **Metadata**: The various metadata performance tests completed in Phase 2 were scaled to the final namespace.
- **Performance and Capacity Tiers**: Similarly, the various tests that targetted the performance and capacity tiers in Phase 2 were scaled to the final system size. These include a range of I/O sizes, access patterns, and access methods against a freshly formatted and artifically aged namespace.
- **Namespace default**: The team ran the suite of I/O benchmarks against the default namespace configuration in order to verify the performance achievable and expected by science applications.

3) Stability: For stability testing, the team launched a series of known I/O patterns against the namespace. All I/O used the namespace default configuration described above. The test simulated normal operational conditions and verified that scientific workloads could aggressively push the file system without I/O interruptions nor data integrity concerns.

V. Conclusion

By leveraging existing tooling, OLCF staff accelerated the integration and configuration of the Orion storage cluster. The file system is in production and is actively used. Several end-users of the system have reported significant I/O speed up. Orion leverages enhanced features of Lustre to provide a namespace that is performant without being overly prescriptive to a particular I/O workload. Using Lustre's DNE and PFL features, the namespace shards metadata across 40 metadata servers, stores the first part of a file alongside the metadata, and can support a variety of file sizes. Though Orion was deployed as a scratch namespace, the system incorporates various features, such as SEL within the Lustre namespace, and Corosync and Pacemaker at the server OS level to minimize the risk to user jobs. The detailed and strenuous acceptance process ensures that the namespace provides a consistent and performant environment for science workloads. As a large-scale storage resource, the transition to operations will undoubtedly present new challenges not exposed during the acceptance process. These challenges will inform future approaches, ensuring continued innovation and best practices.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research Program. This research used resources of the Oak Ridge Leadership Computing Facility,

which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

VI. References

[1] "Overview." https://www.olcf.ornl.gov/about-olcf/overview/

[2] "Pacemaker." https://github.com/ClusterLabs/pacemaker

[3] S. C. Dake, C. Caulfield, and A. Beekhof, "The corosync cluster engine," in *Linux symposium*, Citeseer, 2008, pp. 61–68.

[4] Kelley, Simon, "Dnsmasq." https://thekelleys.org.uk/dnsmasq/doc.html

[5] D. Both and D. Both, "Systemd," *Using and Administering Linux: Volume 2: Zero to SysAdmin: Advanced Topics*, 2020.

[6] "Buildah." https://buildah.io/

[7] "`chroot`: Run a command with a different root directory." https://www.gnu.org/software/coreutils/chroot

[8] "Puppet Infrastructure & IT Automation at Scale." https://www.puppet.com/

[9] "Anchor." https://github.com/olcf/anchor

[10] E. Leon *et al.*, "TOSS-2020: A commodity software stack for high-performance computing," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2020.

[11] "PowerMan." https://github.com/chaos/powerman

[12] "ConMan: The Console Manager." https://github.com/dun/conman

[13] V. G. Vergara Larrea *et al.*, "Scaling the summit: Deploying the world's fastest supercomputer," in *High performance computing: ISC high performance 2019 international workshops, frankfurt, germany, june 16-20, 2019, revised selected papers 34*, Springer, 2019, pp. 330–351.

[14] I. H.-P. N. Dell, "Intelligent platform management interface specification v2. 0 rev. 1.1." 2013. Available: https://www.intel.in/content/dam/www/public/us/en/documents/product-briefs/ipmi-second-gen-interface-spec-v2-rev1-1.pdf

[15] "REDFISH | DMTF." https://www.dmtf.org/standards/redfish

[16] W. Barth, *Nagios: System and network monitoring*. No Starch Press, 2008.

[17] "Cray ClusterStor E1000 Storage Systems Data Sheet." Available: https://www.hpe.com/psnow/doc/PSN1012842049USEN.pdf

[18] "A guide to mdadm." Available: https://raid.wiki.kernel.org/index.php/A_guide_to_mdadm

[19] Libby, Richard, "Effective HPC hardware management and Failure prediction strategy using IPMI,"

[20] "ClusterShell Python Library and Tools." https://github.com/cea-hpc/clustershell

[21] C. Wood, "Online monitoring for high-performance computing systems," 2021.