DATA STORAGE SOLUTIONS

Enterprise and Cloud IT
Intel Proof of Concept

ALL FLASH PARALLEL FILE SYSTEM SOLUTION:
UTILIZING LUSTRE* FILE SYSTEM FOR HIGH-PERFORMANCE ENTERPRISE

Gain Business Value Using Parallel File Systems on Dell EMC PowerFlex*-Ready Nodes with Solid-State Drives (SSDs)

Authors:
Bill Gallas
Sr. Solutions Architect,
Intel Corporation

Nash Kleppan
Performance Engineer,
Intel Corporation

Executive Summary

In the High Performance Computing (HPC) environment, many storage solutions implement Lustre* on an array of storage servers that are clustered together. A small Dell* and Intel team had identified a bottleneck in small object performance in Lustre* hampering greater market adoption. It was determined that it may be possible to decouple Lustre* from storage and improve small object (<1 MB) performance by integrating Dell's PowerFlex* technology as the backend of the storage array.

Dell* and Intel engineering teams worked collaboratively to develop this solution based on Dell R640* servers running Intel® Xeon® Scalable processors. Intel set up a cluster in the laboratory's data center and the team benchmarked the cluster extensively, noting excellent performance across a wide spectrum of object sizes. The system topology and Software (SW) solution can compete in both the HPC space as well as other more traditional market segments requiring target shared file system.

Using Dell's R640* servers, the testing has shown that high Bandwidth (BW), at very low latency, with durable storage can be attained at a fraction of the system cost and ongoing operations of lesser performing solutions. This proof of concept performs at a higher level than other known competing Lustre* solutions, at an attractive price point. The cluster shows best in class performance across key metrics when compared to other solutions presently in the market. By reducing or eliminating the common constraints of high cost, small object size performance and latency, while increasing BW and assuring

data availability on a system designed to serve in more than just the HPC segment, it is possible to see an incredible opportunity in many non-traditional segments, as providers can adopt this solution for their compute and fast, durable storage needs.

Table of Contents

Introduction

In high performance computing, the efficient delivery of data to and from the compute nodes is critical to system performance and is often complicated to evaluate. Multiple tasks from researchers can generate and consume data in HPC systems at such high speeds that the storage components become a major bottleneck. Recently, solid state devices such as Non-Volatile Memory Express* (NVMe*) have become affordable and will likely replace rotating hard drives as the block storage devices of choice for high-performance, Parallel File Systems (PFS). Getting maximum performance from a PFS requires a scalable storage solution and fast block storage like NVMe* devices.

Lustre* is an open-source PFS that is used in the largest super computers with extremely high throughput, and it is also capable of managing multiple petabytes of data. Although Lustre* is known for its applicability in HPC, the performance of the Lustre* solution discussed in this document highlights a solution for more general use in an enterprise.

PowerFlex* is a SW-only solution that uses existing servers' local disks and LAN to create a virtual SAN. PowerFlex* SW components are installed on the application servers and communicate via a standard LAN to handle the application Input/Output (I/O) requests sent to PowerFlex* block volumes. PowerFlex* is Hardware (HW) agnostic, the SW works efficiently with various types of disks, including: SAS*, SATA, or NVMe* Solid-State Drives (SSD).

PowerFlex* has three modules: Storage Data Server (SDS), Storage Data Client (SDC), and the Metadata Manager (MDM). PowerFlex* can be run on the Lustre* Object Storage Servers (OSS) and use local NVMe* SSDs attached to the OSS server to create elastic, scalable, and resilient virtual SANs. By using off-the-shelf servers, it is possible to lower costs and see reduced complexity over traditional SANs. Also, the R640*s are commercially available from Dell* EMC. With efficient CPU utilization and memory footprint, in combination with low latency and locally attached NVMe* SSDs), PowerFlex* can run on the OSS and greatly enhance the performance of Lustre* when compared to traditional Lustre* installations that typically use a RAID storage backend.

This whitepaper describes the architecture and configuration of an appliance based on Lustre 2.10.8* running on PowerFlex* storage devices, using local

NVMe* SSDs, and combining these two SW products to achieve very good to excellent performance for a PFS solution of this size. An optimized storage configuration using PowerFlex* as the backend for the Lustre 2.10.8* solution is also discussed and presented. Intel® Omni-Path Fabric (Intel® OP Fabric) was used to connect the Lustre* clients, the OSSs, and the Metadata Servers (MDSs), as well as 100 Gigabit Ethernet to connect the PowerFlex* clients and servers (OSSs and MDSs).

This paper describes the tested maximum performance of such a solution and demonstrates the functional operation of an accelerated Lustre* storage appliance.

To summarize, the objectives of the evaluation are as follows:

- Investigate a cost-effective, high-performance, SW-only Lustre* file system on an NVMe* solution.
- Take advantage of PowerFlex* as a durable and performant storage layer for Lustre*.
- Run all components on each of the nodes to ease deployment at scale.

This paper presents the performance characteristics of two types of workloads using two popular HPC benchmarks: IOR 3.1.0 [1], which is used for evaluating sequential BW performance and MDtest 1.9.4-rc1 [2], which used to evaluate the file system metadata operation performance. Both benchmarks are using MPI (mvapich2-2.3b) to emulate storage-intensive HPC workloads. The test environment does not require any specialized HW. Therefore, any Intel Architecture-based servers configured as described could be expected to perform similarly.

IOR is a BW benchmark tool focused on using sequential or random large files with large network transfers. MDtest is a metadata-focused benchmark which drives smaller random IO. These two benchmarks complement each other to give an accurate overall system performance characterization.

The following text in this paper describes the Lustre* PowerFlex* storage solution with the HW and SW configurations implemented in this proof of concept. This section discusses the HW and the PowerFlex* backend configurations used. Further, the solution and the respective performance evaluation for sequential-BW HPC workloads are described. Also, the metadata server's configuration is discussed. The system performance is measured with IOR, and MDtest in detail. Finally, conclusions are drawn, and there are recommendations provided for building the Lustre* appliance using PowerFlex* SW Defined Storage to build block storage on NVMe* SSDs.

The Lustre* PowerFlex* Storage Solution

Lustre* is a parallel file system, offering high performance through parallel access to data and distributed locking. A Lustre* installation consists of three key elements: the metadata subsystem, the object storage subsystem (data), and the compute clients that access and operate on the data. The metadata subsystem is comprised of the Metadata Target (MDT), the Management Target (MGT), Management Server (MGS), and the MDSs. The MDT stores all metadata for the

file system including: file names, permissions, time stamps, and the location of data objects within the object storage system. The MGT stores management data such as configuration information and registry. The MDS and MGS each manage the MDT and MGT, respectively. In this cluster, the MDS and MGS are collocated.

In this configuration, a PowerFlex* pool on the eight NVMe* SSDs in each of the OSSs for metadata and data use was created. In the single MDS configuration, two devices were created, one each for the MDT and MGT using a single MDS/MGS server. In the dual MDS configuration, two MDTs of equal size and one MGT were created.

The object storage subsystem is comprised of multiple Object Storage Target (OST) devices and one or more OSS. The OSTs provide block storage for file object data, while each OSS manages four OSTs, using four PowerFlex* devices per OSS in our implementation. Each OST is built as one PowerFlex* mirrored volume using two 2.0 TB NVMe* SSDs in the OSS chassis. Typically, there are several active OSSs at any time; this test bed uses four.

Lustre* is able to deliver increased throughput by increasing the number of active OSSs (and associated OSTs). PowerFlex* allows similar scalability by adding more SDSs with their associated NVMe* SSDs. Each additional OSS increases the existing networking throughput, while each additional OST increases the storage capacity and BW (up to network limits). The compute clients are the HPC cluster's compute nodes; twelve were utilized in this test bed. The compute nodes were connected to the Intel® OP Fabric along with the MDSs and OSS components.

Test Environment

For this test cluster, PowerFlex* 3.0-100 was used to build a virtual SAN utilizing NVMe* SSDs local to the OSS nodes. The Lustre* file system was layered on top of the virtual SAN (see Figure 1). The primary test configurations used four physical servers running Lustre* OSS SW and four PowerFlex* server instances, one per OSS, and five or six PowerFlex* clients - one on each OSS server plus the MDSs (see Figure 1).

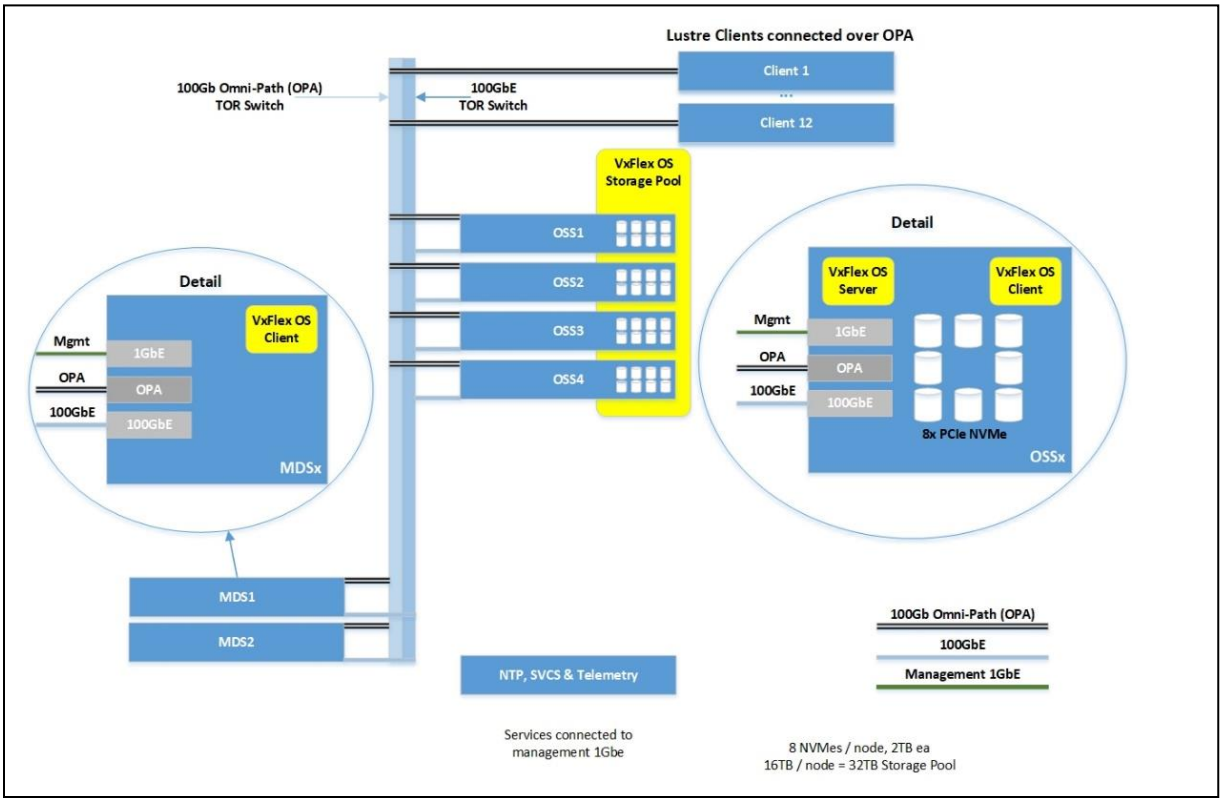

Figure 1. Diagram of Lustre*/PowerFlex* Cluster with Four OSSs and SDSs + Two MDSs

PowerFlex* is a SW-defined storage solution that uses existing servers' local block devices - in this case, NVMe* devices - and a 100-gigabit Ethernet network to interconnect the PowerFlex* data servers and clients, in order to create a virtual SAN that has all the benefits of external storage arrays. In running Lustre* as well as PowerFlex* on each node, it is possible to see a reduction in the cost and complexity over typical storage arrays. PowerFlex* utilizes the existing local block storage devices, creates shared storage pools from local NVMe* SSDs in multiple servers, and allocates block devices and Logical Unit Numbers (LUNs) from these pools.

The PowerFlex* virtual SAN consists of the following SW components:

- MDM: Configures and monitors the PowerFlex* system.
- SDS: Manages the capacity of a single server and acts as a backend for data access. The SDS is installed on all servers contributing storage devices to the PowerFlex* system. These devices are accessed through the SDS.
- SDC: A lightweight device driver that exposes PowerFlex* volumes as block devices to the application that resides on the same server on which the SDC is installed.

SDSs allocate and contribute storage to the overall storage pool to create a SW-defined, converged, shared SAN. This SW is media and server agnostic. It can be created on physical or virtual servers, and it utilizes SATA/SAS* SSDs, as well as NVMe* SSDs that are installed in the physical servers on which SDSs run. Different performance tiers may be configured, allowing the administrator to create a robust and manageable environment to fit the needs of the

enterprise. In this test cluster, Intel® SSD Data Center Family P4600 Series was used. These NVMe* SSDs were used for one pool that serves Lustre* OSTs, MDTs, and the MGT.

The Lustre* metadata subsystem is comprised of the MDT, the MGT, the MDS, and the MGS. The MDT stores all metadata for the file system including file names, permissions, time stamps, and the location of data objects within the object storage system. The MGT module stores management data such as configuration information and registry, and the MDS is a dedicated server that manages the namespace and the MDT. In this test bed, the Distributed Namespace Phase 2 features were advantageous; thus, allowing more than one MDS to be used with minimal additional configuration. Utilizing this feature greatly improves Lustre*'s metadata performance, as shown in the results.

The OSTs provide storage for file object data, while each OSS manages four OSTs. Each additional OSS increases the existing networking throughput, while each additional OST increases the storage capacity. In the evaluation, we tested four PowerFlex* servers with eight NVMe* SSDs each.
A parallel file system, such as Lustre*, delivers performance and scalability by distributing, or "striping," data across multiple OSTs. A key design consideration of Lustre* is the separation of metadata access from data access in order to improve the overall system performance. The Lustre* client SW is installed on the compute nodes and allows access to data stored on the Lustre* file system. To the clients, the file system appears as a single namespace that can be mounted for access. This single mount point provides a simple starting point for application data access, and allows access via native client OS tools for easier administration.

To summarize, the elements of the Lustre* file system are as follows:

- Metadata Storage Server (MDS): Manages the MDT, providing Lustre* clients access to files.
- Metadata Target (MDT): Stores the location of "stripes" of data, file names, time stamps, and so on.
- Management Server (MGS): Manages the MGT, providing Lustre* configuration data.
- Management Target (MGT): Stores management data such as configuration and registry.
- Object Storage Server (OSS): Manages the OSTs, providing Lustre* clients access to the data.
- Object Storage Target (OST): Stores the stripes of data or extents of the files on a file system.
- Lustre* Clients: Access the MDS to determine where files are located, and they access the OSSs to read and write data.

In all configurations, twelve servers as clients to the Lustre* PFS were used, on which the IOR and MDtest benchmarks were run. All systems are running open-source CentOS 7.6*.

Installation Details

PowerFlex* storage was configured using one shared pool containing eight NVMe* SSDs in each SDS, with a total capacity of 16 TB per OSS, as indicated in Figure 1, for an aggregate of 64 TB of raw storage.

It is important to note that PowerFlex* was configured to mirror the volumes, so the durable storage was one half of the raw storage: 32 TB total. Each OSS was also running both SDC and SDS. The MDS server was running only the SDC accessing the MDT pool.

Four 1.6-TB LUNs per OSS were configured as to have one OST per LUN. Two 552-GB LUNs for MDTs and one 8-GB LUN for the MGT from the MDT pool were carved. These LUNs were accessed by the PowerFlex* client running on the MDSs and served by the PowerFlex* SDSs on OSS1 to OSS4.

The following is a summary of the HW and SW components used to build the Lustre*/PowerFlex* Cluster.

MDS HW Configuration	
MDS Server	Dell PowerFlex Ready Node R640*
Processor	Two Intel® Xeon® Gold 6246 Processors
Memory	384 GB (24 x 16 GB DIMMs) 2667 MHz DDR4
Host Fabric Interface	Intel® OP Host Fabric Interface Adapter 100 Series, 100HFA016LS
Ethernet Network Interface Card (NIC)	Qlogic QL45611 100 GbE NIC
SATA SSD (for boot)	Intel® SSD DC S4500 Series, 480 GB
OSS HW Configuration	
OSS Server	Dell PowerFlex Ready Node R640
Processor	Two Intel® Xeon® Gold 6246 Processors
Memory	192 GB (12 x 16GB DIMMs) 2667 MHz DDR4
Host Fabric Interface	Intel® Omni-Path Host Fabric Interface Adapter 100 Series, 100HFA016LS
Ethernet NIC	Qlogic QL45611 100GbE NIC
SATA SSD (for boot)	Intel® SSD DC S4500 Series, 480 GB
NVMe* SSDs (for storage)	Eight Intel® SSD DC P4600 Series, 2.0 TB
Client HW Configuration	
Client Server (four blades in each chassis)	Chassis: Intel® Server Chassis H224XXKR2 Blades: Intel® Compute Module HNS2600TP
Processor	Two Intel® Xeon® E5-2695v4 Processors
Memory	128 GB (16 x 8 GB DIMMs) 2133MHz DDR4
Host Fabric Interface	Intel® OP Host Fabric Interface Adapter 100 Series, 100HFA016LS
SATA SSD (for boot)	Intel® SSD DC S3700 Series, 200 GB
Network Switches	
Ethernet Switch (PowerFlex*)	Arista 7060CX-32* 100 GbE Network Switch: MTU 9000
Fabric Switch (Lustre*)	48-port Intel® Omni-Path Edge Switch 100 Series, 100SWE48QF

Table 1. HW Components for the Lustre*/PowerFlex* Cluster

Storage Server Software	
OS	CentOS 7.6* x86_64
Kernel	3.10.0-957.1.3.el7_lustre.x86_64
Lustre*	2.10.8
PowerFlex*	3.0-100.134
Fabric Driver	IntelOPA-IFS.RHEL76-x86_64.10.9.3.1.1
Lustre* Client Software	
OS	CentOS 7.6* x86_64
Kernel	3.10.0-957.1.3.el7.x86_64
Lustre* Client	2.10.8
Fabric Driver	IntelOPA-IFS.RHEL76-x86_64.10.9.3.1.1

Table 2. Software Components for the Lustre*/PowerFlex* Cluster

Configuration Summary:

1) The Lustre* cluster:
    a. One system serving as the cluster NTP server and a Zabbix* server. This system is not in the data path.
    b. Two Lustre* MDSs, one of which is also serving as the MGS. The MDSs are connected to the PowerFlex* MDT pool using the PowerFlex* SDC. For some test cases, only one MDS was active.
    c. Four Lustre* OSSs (OSS1-4) connected to the PowerFlex* OSTs using the PowerFlex* volumes.
2) The PowerFlex* cluster:
    a. 32 x 2 TB NVMe* in the OSS/SDS servers – 64 TB total.
    b. Using PowerFlex*, one shared storage pool was configured using eight NVMe* drives per OSS for use as OSTs, MDTs, and the MGT.
    c. Three PowerFlex* volumes, one 8-GB size MGT and two 552-GB MDTs were created. One MDS node has 8-GB and one 552-GB volume while the other has only the second 552-GB volume mapped.
    d. 16 PowerFlex* volumes (each 1.6 TB in size) were created. Four volumes were mapped to each of the OSS servers.
3) All six servers (4x OSS + 2x MDS) and all 12 clients are connected to a 10-gigabit Ethernet Top of Rack (TOR) switch for management.
4) All six servers (4x OSS + 2x MDS) are connected to a 100-gigabit Ethernet TOR switch for PowerFlex* to use.
5) All six servers and 12 client nodes are connected to a 100-gigabit Intel® Omni-Path Architecture (Intel® OPA) switch for Lustre* to use.

Minimal performance tuning was necessary for the configuration of the solution.

For all Lustre* OSSs and MDSs, the following Intel® Omni-Path Driver tunings were applied:

- `krcvqs` = 8
- `piothreshold` = 0

- `sge_copy_mode` = 2
- `wss_threshold` = 70

For the Lustre* clients, the following Intel® Omni-Path tunings were applied:

- `cap_mask` = 0x4c09a01cbba
- `krcvqs` = 4

For the Lustre* OSTs, only this Lustre* parameter was modified:

- `obdfilter.lustrefs-OST*.brw_size` = 16

For the Lustre* clients, the following Lustre* parameters were set:

- `osc.lustrefs-OST*.max_pages_per_rpc` = 4096
- `llite.lustre*.max_read_ahead_mb` = 1024
- `osc.lustrefs-OST*.max_rpcs_in_flight` = 16

For PowerFlex* SDC, SDS, and MDM, the following parameters were modified from default:

- SDS parameters:
  - `sds_number_os_threads` = 12
- SDC parameters:
  - `sdc_number_sockets_per_sds_ip` = 6
  - `sdc_number_network_os_threads` = 10
  - `sdc_max_inflight_requests` = 200
  - `sdc_max_infight_data` = 20

Performance Evaluation and Configuration Details

The performance study presented in this paper utilizes two popular benchmarks used to evaluate storage for HPC: IOR [1] and MDtest [2]. Both benchmarks use MPI communication between the compute cluster nodes for synchronization of the benchmark. Only POSIX IO was used for this evaluation, as MPI-IO is limited to specific collective IOs being enabled by the application, whereas the POSIX interface can be used without requiring any code changes.

A number of performance studies were executed, stressing the configuration with different types of workloads to determine the limitations of performance under different circumstances. The performance analysis was focused on two key performance indicators:

- BW, data transferred in MB/s to both Lustre* and backend storage.
- Metadata Operations per second (ops/sec).

The goal is a broad overview of the performance of this PFS to gauge how it can perform for both traditional HPC workloads and for IOPS intensive workloads seen in enterprise environments. For IOR, a file for each process of the benchmark was used. With MDtest, a subdirectory per process with a varying number of files in each subdirectory was implemented.

Each set of tests was conducted five times on all 12 clients of the solution, and the results reported are the averages of those five runs. Performance was recorded for IOR transfer sizes 128 KiB, 512 KiB, 2 MiB, 8 MiB, and 32 MiB. Performance results were also collected for MDtest with 128-byte, 4-KiB, and 128-KiB file sizes. With MDtest, the number of files used by each process from 1024 to 131072 files per process were varied.

In summary, the objectives of the evaluation are as follows:

- Investigate a more cost-effective high performance SW-only Lustre* on NVMe* solution.
- Take advantage of PowerFlex* as a durable and performant storage layer for Lustre*.
- Run all components on each of the nodes, for easier deployment at scale.

Benchmark Results

IOR Performance Evaluation:

The BW testing was done with the IOR benchmark tool version 3.1.0. The IOR MPI job using 12 client nodes as load generators was run. Each node used 72 MPI processes, the same as the number of threads per node (2x36). The tests were conducted using four OSSs and the IOR configured for peak performance was run. This included the use of 2-MiB transfer size and a 4-MiB stripe size.

Figure 2 shows the maximum BW results for both write and read for the configuration described in Figure 1. The maximum read BW was obtained by using 32-MiB transfer size. The peak Lustre* write performance was 11.5 GB/sec (at transfer size 8 MiB) and peak read performance was 30.8 GB/sec (at transfer size 32 MiB). The graph shows performance for both the read and write tests across a wide range of transfer sizes, from 128 KiB to 32 MiB.

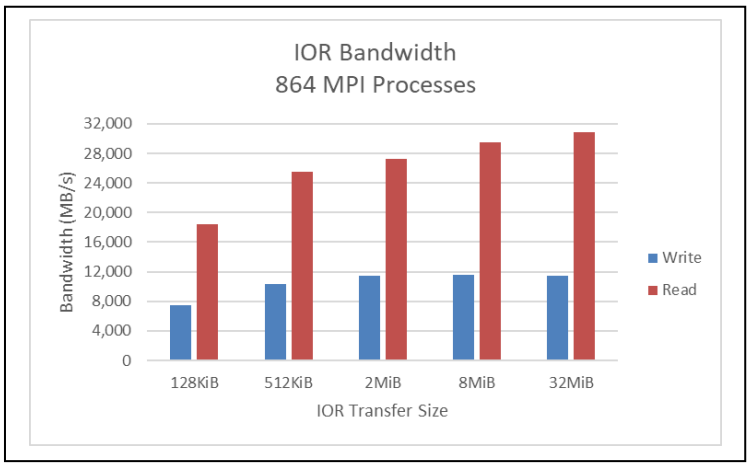

Figure 2. Lustre*/ PowerFlex* IOR Sequential BW Across IOR Transfer Sizes

Figure 3 shows the benchmark results on a per-node basis. The peak write performance per OSS was 2.8 GB/sec (at transfer sizes 2-32 MiB), and the peak Lustre* read performance per OSS was 7.7 GB/sec (at transfer size 32 MiB).

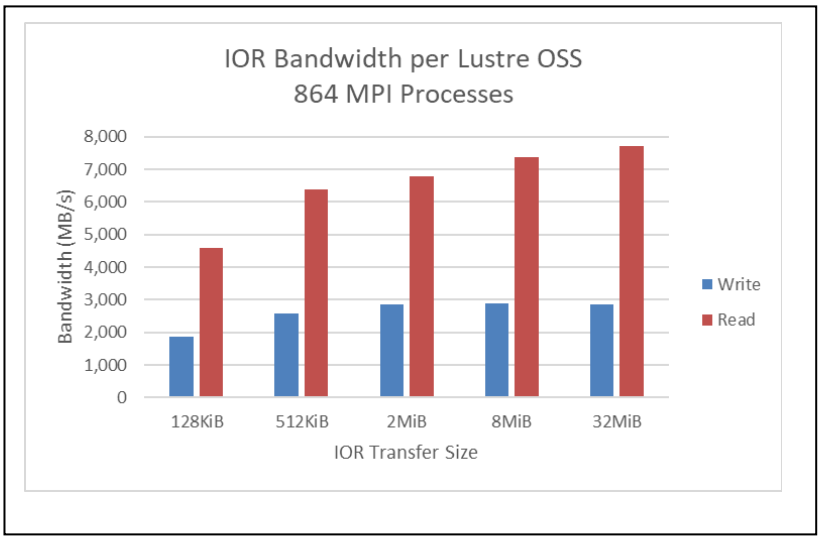

Figure 3. Lustre*/ PowerFlex* IOR Sequential BW per Lustre* OSS Across IOR Transfer Sizes

It was pleasing to find minimal impact of IOR transfer size on the BW of the cluster. Even at a transfer size of 128 KiB, the least performant transfer size that was tested, the cluster was able to achieve 7.4 GB/sec write and 18.3 GB/sec read.

The results of IOR showed consistent performance with good scale in a test range from 128 KiB to 32 MiB IOR transfer size and a 16 GiB file size. Additionally, it was possible to see:

- Excellent I/O performance over a wide range of IOR transfer sizes.
- Excellent application compatibility with the use of the standard POSIX interface.

MDtest Performance Evaluation

The experiments consisted of running MDtest against the file system, using all 12 clients and varying quantities of files per process. We use the DNE2* features recently added to Lustre* to stripe metadata operations across the two MDSs in a manner that is transparent to the clients once striping is set for the parent directory.

During the preliminary metadata testing, it was observed that the number of files per directory significantly affects the performance of the cluster. The metadata performance was measured while scaling up the number of files created in each directory. This performance using 128-byte, 4-KiB, and 128-KiB files was also measured. Additionally, the extremely small 128-byte and 4-KiB files were included to demonstrate the performance of the cluster when used in a manner not typical for Lustre*. Traditionally, Lustre* has been designed for large files with relatively few files per directory. The performance of this cluster suggests that it can be used for more than just traditional HPC.

The number of files per process was varied from 1K to 256K files per process (442,368 files to 56,623,104 files in total). For example, when testing 8192 files with 36 processes per client (432 total MPI processes), there are 3,538,944 files evenly spread across 432 subdirectories. A maximum of 131,072 (128K) files were used per process, as the test was

limited by the maximum number of inodes available with default `ldiskfs` parameters on the MDTs, and it was felt that in most usage models, it will not be necessary to store over 50 million files in a cluster of this size.

Figure 4 through Figure 7 show the MDtest results, in operations per second, for `create`, `stat`, `read`, and `remove` tasks on 128-byte, 4-KiB, and 128-KiB file sizes when two MDSs are in use for 1024 through 131072 files per process. At 65,536, 4-KiB files per process, MDtest is able to perform 98.3K file `create` ops/sec, 500K `stat` ops/sec, 197K `read` ops/sec, and 116K `remove` ops/sec. Peak performance for 4-KiB files occurs at 4,096 files per process, but there is a minimal reduction in performance for file counts as high as 65,536 files per process. In the most extreme case tested, 131,072 128-byte files per process (a total of 113,246,208 files), MDtest is able to perform 87.1K `create` ops/sec, 350K `stat` ops/sec, 181K `read` ops/sec, and 102K `remove` ops/sec.

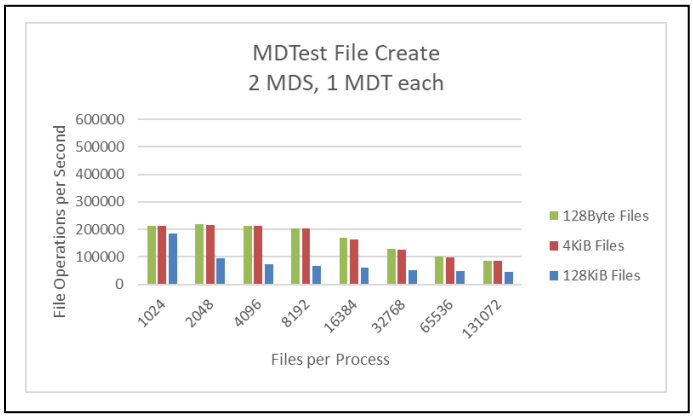

Figure 4. MDTest File Create

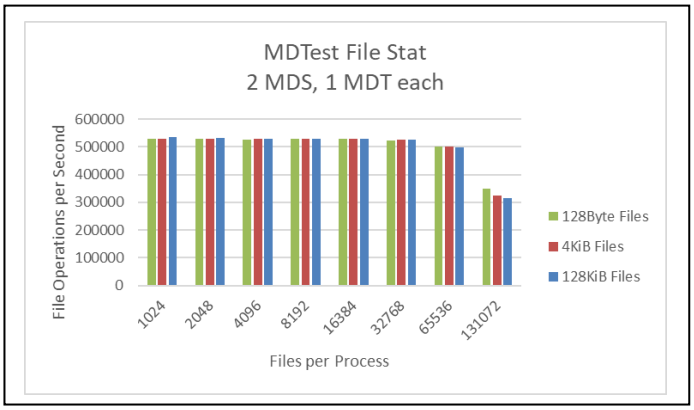

Figure 5. MDTest File Stat

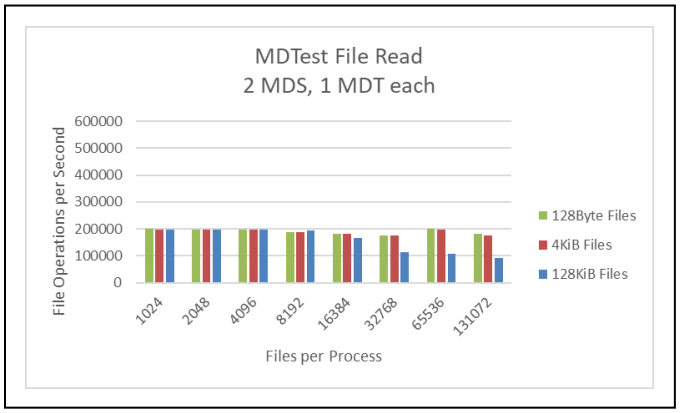

Figure 6. MDTest File Read

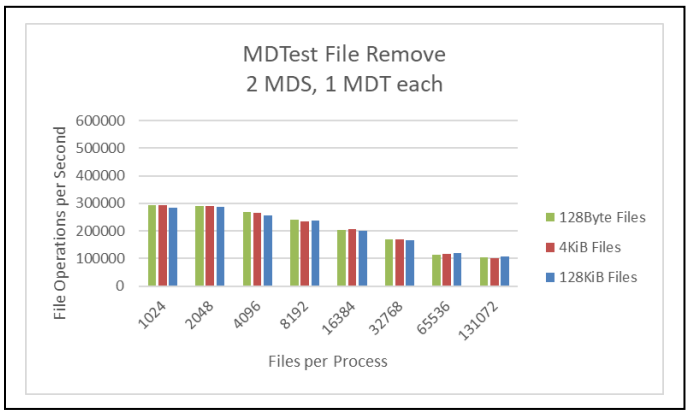

Figure 7. MDTest File Remove

To provide context to the performance figures previously shown, it is helpful to refer to the results that Oak Ridge National Laboratories* (ORNL*) obtained when they performed an evaluation of Lustre* with DNE2* enabled across eight MDSs. They ran an MDtest against a cluster they deemed representative of their production environment and published a graph of file ops/sec performance with 10,000 files per process. The graph published by ORNL* indicated approximately 100K `create` ops/sec, 40K `stat` ops/sec and 160K `delete` ops/sec [3]. The cluster described in this document, with just two MDSs, can deliver as much as 162K `create` ops/sec, 528K `stat` ops/sec, 181K `read` ops/sec, and 206K `delete` ops/sec when operating on a slightly larger dataset (16,384 4 KiB files per process instead of 10,000 files per process).

Additionally, the cluster ran with just one MDS to verify that MDS performance can be scaled. Due to the locking behavior of Lustre* and the high load applied with these tests, scaling from one MDS to two MDSs was as expected for all operations except file `create`. File creation is a much more demanding task for the file system than other file operations, and this was improved by 37% in the case of 65,536 4-KiB files per process.

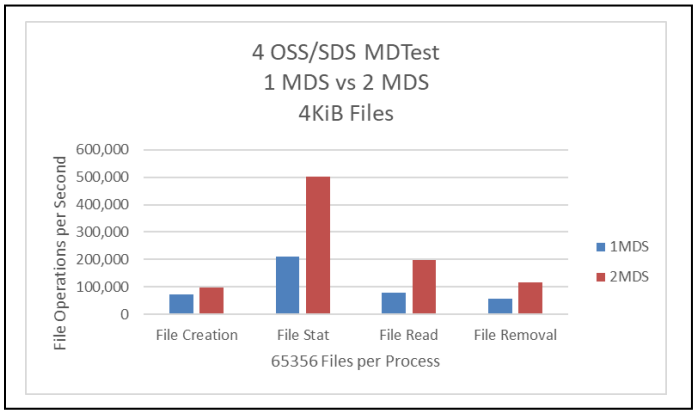

Figure 8. MDS Scaling

These results demonstrate that the solution can have Lustre* metadata and Lustre* data performance scaled independently. Additionally, these results demonstrate that this solution can provide great BW (up to 11.5 GB/sec IOR write and up to 30.8 GB/sec IOR read) as well as outstanding Lustre* metadata performance (up to 218K MDtest `create` ops/sec, 528K MDtest `stat` ops/sec, 197K MDtest `read` ops/sec and 291K MDtest `delete` ops/sec).

Conclusions

This Lustre*/PowerFlex* solution offers excellent performance in a compact form factor (6U using standard 1U servers) at a lower cost than with traditional storage appliances. The performance characteristics of this solution also suggest that it can be used in ways previously not possible with Lustre*, instead of just large file HPC workloads. Standard enterprise storage can benefit from the high-performance access to a large shared file system. In a storage system where performance is paramount, this solution provides very high read and write rates with excellent metadata performance and data durability through PowerFlex*.

Intel engineers evaluated competing proprietary and open solutions on the market, and from a limited study, they determined that this hybrid architecture enables a stable, cost-effective, high-performance storage capability that will be difficult to match. The team utilized COTs HW versus purpose built HW to realize HW cost reductions with this POC. Moreover, the extremely high performance of this solution helps to mitigate the impact of a user attempting to interact with Lustre* as if it were a simple local file system.

The Lustre*/PowerFlex* solution discussed has also been shown to scale in both Lustre* MDSs and in Lustre* OSSs (also acting as PowerFlex* SDSs). If additional BW or capacity is required, OSS or SDS nodes can be added. If additional metadata performance is required, additional MDS nodes can be added. PowerFlex* can be scaled in a similar manner. If additional BW or capacity is required, SDS nodes can be added. This enables the cluster operator to more easily scale to meet the growing performance and capacity demands without having to purchase an entirely new storage array.

References

[1] IOR benchmark, found at https://github.com/hpc/ior/blob/3.1.0/doc/USER_GUIDE

[2] MDtest benchmark, found at https://sourceforge.net/projects/mdtest/

[3] *Lustre Distributed Name Space (DNE) Evaluation at the Oak Ridge Leadership Computing Facility (OLCF)*, found at https://lustre.ornl.gov/ecosystem-2016/documents/papers/LustreEco2016-Simmons-DNE.pdf

Appendix A: Benchmark Command Reference

This section describes the commands used to benchmark the Lustre* with Dell EMC PowerFlex** storage solution.

A1. IOR benchmark

IOR write command

mpirun -np 864 -f hostfile IOR -a POSIX –b 16g -w -t 4m -o $Fname -F –k

IOR read command

mpirun -np 864 -f hostfile IOR -a POSIX –b 16g -r -t 4m -o $Fname -F –k

IOR Command Line Arguments	Description
-a POSIX	Type of IO access
-b 16g	Total file block size
-r	Read IO benchmark
-w	Write IO benchmark
-t 4m	Transfer Size
-o $Fname	File name used for each process
-F	Use N-to-N mode; one file per thread
-k	Preserve file after the test.
-B	Use DirectIO

16-GB files were used, which will result in a 6-TB dataset. Also, the directIO option –B all tests was used.

The directIO command line parameter ("-B") allows to bypass the cache *on the Lustre* clients* where the IOR threads are running. Note that the transfer size varied from test to test, 4m is used only as an example.

A2. MDtest Benchmark

MDtest – Metadata Files Operations

mpirun -np 432 –f hostfile mdtest -i 5 -F -w 4096 -L -n $Files -d $Dirname -v

MDtest Command Line Arguments	Description
-d $Dirname	the directory in which the tests will run
-v	verbosity (each instance of option increments by one)
-i	number of iterations the test will run
-F	perform test on files only (no directories)
-w 4096	file size in bytes
-L	files only at leaf level of tree
-n $Files	number of files per process/thread

The same command was used while varying the number of files per MPI process in the range: {1024, 4096, 8192 … 262144}. Note that the file size shown here is only an example, tests were run with file sizes 128 bytes, 4 KiB, and 128 KiB.