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 with Dell* EMC* Software and Solid-State Drives

Authors:
Bill Gallas
Sr. Solutions Architect,
Intel Corporation

Nash Kleppan
Performance Engineer,
Intel Corporation

Executive Summary

Lustre* is an open source parallel file system (PFS) that is popular in high-performance computing (HPC). In traditional Lustre clusters, there are specialized hardware requirements and performance limitations that have prevented Lustre from being a viable platform for non-HPC applications. The cluster described in this document uses standard, 2U rack-mount servers and software to construct an extremely high performing Lustre cluster without the need for specialized RAID hardware or custom file systems such as ZFS. Instead of RAID or ZFS, this solution builds Lustre on top of Dell* EMC* VxFlex* operating system. The use of VxFlex OS allows for high performance mirrored storage without the need for expensive RAID support. To further enhance the performance of the software used in this solution, data is stored on Intel® Solid State Disks with NVM Express* (NVMe) drives and Intel Omni-Path* connectivity is used for Lustre networking. The cluster itself requires only 20U while delivering performance up to 20.1GBytes/sec IOR write, 52.4GBytes/sec IOR read, 121K MDtest create ops/sec, 510K MDtest stat ops/sec, 165K MDtest read ops/sec and 210K MDtest delete ops/sec. It is likely that the outstanding metadata performance of this cluster, as measured with MDtest, will enable the Lustre file system we have implemented to be used for more than just HPC. Moreover, the performance of this solution mitigates the impact of a user attempting to interact with Lustre as a simple, local filesystem. During scalability testing, benchmarking with IOR and MDtest showed almost ideal, linear scaling. If additional storage capacity or performance is required after initial deployment, additional nodes could be added to meet that need.

Table of Contents

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 research 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 NVMe* have become affordable and will likely replace rotating hard drives as the block 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 that is also capable of managing multiple petabytes of data. Although Lustre is known for its use in HPC, the performance of the Lustre solution discussed in this document may allow for more general use in an enterprise.

Dell* EMC* VxFlex* OS (formerly ScaleIO*) is a Dell EMC software product implemented as a Software Defined Storage solution, and it allows building SAN block storage devices using local SSD or NVMe flash storage attached to Linux* servers. VxFlex OS has three modules: Storage Data Server* (SDS), Storage Data Client* (SDC) and the Metadata Manager* (MDM). VxFlex OS can be run on the Lustre Object Storage Servers* (OSS) and use local NVMe attached to the OSS server to create elastic, scalable, and resilient virtual SANs with significantly lower cost and with reduced complexity over traditional SANs. Also, it is commercially available from Dell EMC. Due to its extremely low CPU and memory footprint -- in combination with low latency locally attached NVMe devices -- VxFlex OS can run on the OSS and greatly enhance the performance of Lustre SW when compared to traditional Lustre installations.

In this whitepaper we describe the architecture and configuration of an appliance based on Lustre 2.10.3 running on VxFlex OS storage devices, using local NVMe block devices, and combine these two software products to achieve best-in-class performance for a PFS solution of this size. We discuss and test an optimized storage configuration using VxFlex OS as the backend for the Lustre 2.10.3 solution. In this work we also use new and innovative network technologies to interconnect all the components of the solution and to connect the client nodes to the PFS. We use Intel® Omni-Path* fabric to connect the Lustre clients, the OSSs, and the MDSs, as well as 100 Gigabit Ethernet to connect the VxFlex OS clients and servers (OSSs and MDSs).

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

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

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

In this paper we present the performance characteristics of two types of workloads using two popular HPC benchmarks: IOR* 3.1.0 [1], which is used for evaluating sequential bandwidth (BW) performance and MDtest* 1.9.4-rc1 [2] used to evaluate 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 special features. Therefore, any Intel processor servers configured as described could be expected to perform similarly. IOR is a bandwidth benchmark tool focused on using sequential or random large files with large 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 sections of this paper describe the Lustre VxFlex OS storage solution with the hardware and software configurations implemented in the test cluster. Section 4 discusses the hardware and the VxFlex OS backend configurations used. Further, we describe the test bed and the respective performance evaluation for sequential-bandwidth HPC workloads. In Section 5 we discuss the metadata server's configuration. In section 6, we discuss the performance measured with IOR, and MDtest. Finally, we draw conclusions and provide recommendations for building the Lustre appliance using VxFlex OS SDS to build block storage on NVMe devices.

The Lustre* VxFlex* OS 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) and Management Server (MGS) and Metadata Server (MDS). 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 we created a VxFlex OS pool on one NVMe device in each of the OSSs for metadata use. In the single MDS configuration, we created two devices from the pool, one each for the MDT and MGT using a single MDS/MGS server. In the dual MDS configuration, we created two MDTs of equal size and one MGT.
The object storage subsystem is comprised of multiple Object Storage Target (OST) devices and one or more Object Storage Servers (OSS). The OSTs provide block storage for file object data, while each OSS manages four OSTs, using four VxFlex OS devices per OSS in our implementation. Each OST is built as one VxFlex OS mirrored volume using two 2.0TB NVMe drives in the OSS chassis. Typically, there are several active OSSs at any time, our test bed uses eight.

Lustre is able to deliver increased throughput by increasing the number of active OSSs (and associated OSTs). VxFlex OS allows similar scalability by adding more SDSs with their associated NVMe drives. Each additional OSS increases the existing networking throughput, while each additional OST increases the storage capacity and disk BW. The compute clients are the HPC cluster's compute nodes, twelve were utilized in our test bed. The compute nodes were connected to the Omni-Path* fabric along with the MDS(s) and OSS component pieces.

Test Environment

For our setup we used VxFlex* OS 2.5-1 to build a virtual SAN out of local NVMe drives on the OSS nodes. We layered the Lustre* file system on top of the virtual SAN (see Figure 1 below). The primary test configurations used eight physical servers running Lustre OSS SW and eight VxFlex OS server instances; one per OSS; and nine or ten VxFlex OS clients - one on each OSS server plus the MDS(s) (see Figure 1). To verify scaling, tests were also run using only four physical servers running Lustre OSS SW and four VxFlex OS server instances; one per OSS; and five or six VxFlex OS clients - one on each OSS server plus the MDS(s) (Figure 2 below).

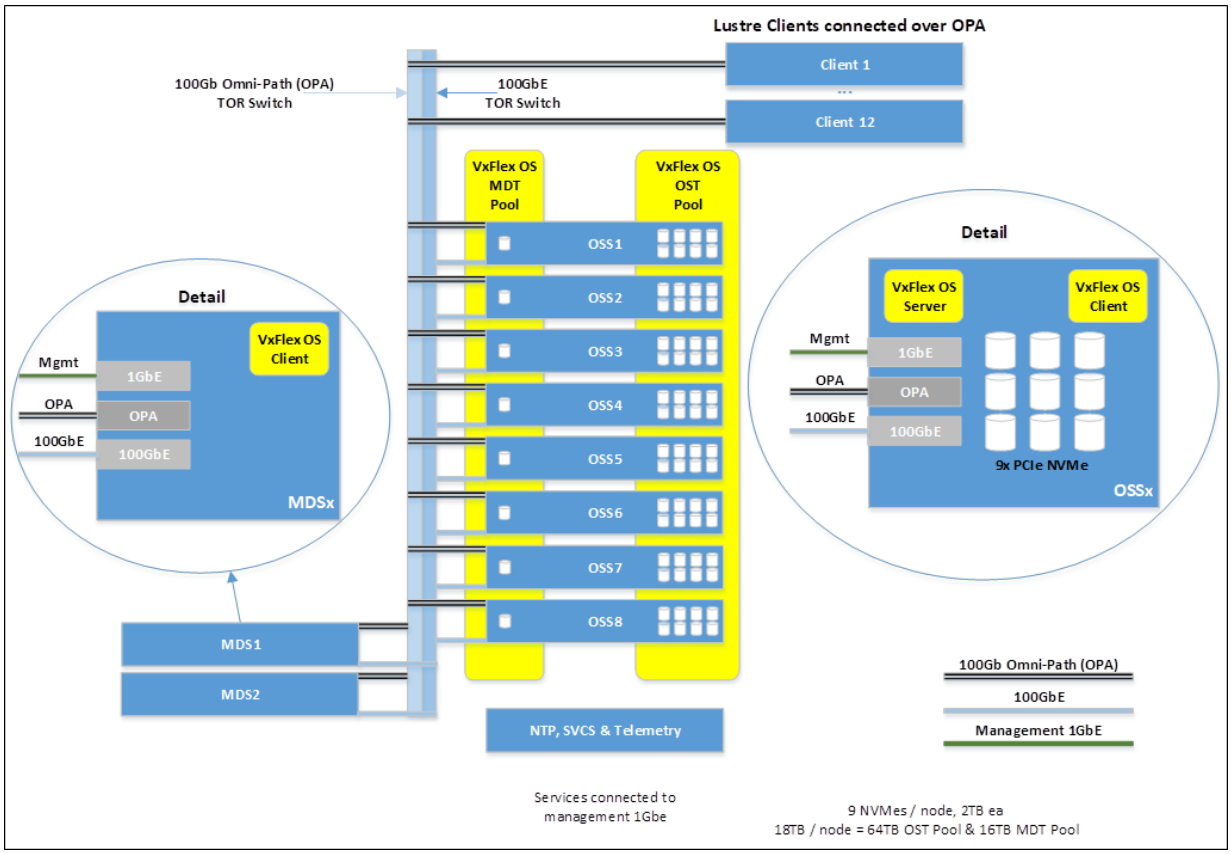

Figure 1: Diagram of Lustre/VxFlex OS cluster with 8 OSSs & SDSs plus 2 MDSs

VxFlex OS is a software-defined storage solution that uses existing servers' local block devices -- in our case, NVMe devices -- and a 100 gigabit Ethernet network to interconnect the VxFlex OS data servers and clients in order to create a virtual SAN that has all the benefits of external storage arrays at a fraction of the cost and complexity of typical storage arrays. VxFlex OS utilizes the existing local block storage devices and creates shared storage pools from local NVMe drives in multiple servers and allocates block devices, LUN's, from these pools.

Data Storage Proof of Concept

The VxFlex OS virtual SAN consists of the following software components:

- Metadata Manager—MDM: Configures and monitors the VxFlex OS system
- Storage Data Server—SDS: Manages the capacity of a single server and acts as a back-end for data access. The SDS is installed on all servers contributing storage devices to the VxFlex OS system. These devices are accessed through the SDS.
- Storage Data Client—SDC: A lightweight device driver that exposes VxFlex OS volumes as block devices to the application that resides on the same server on which the SDC is installed.

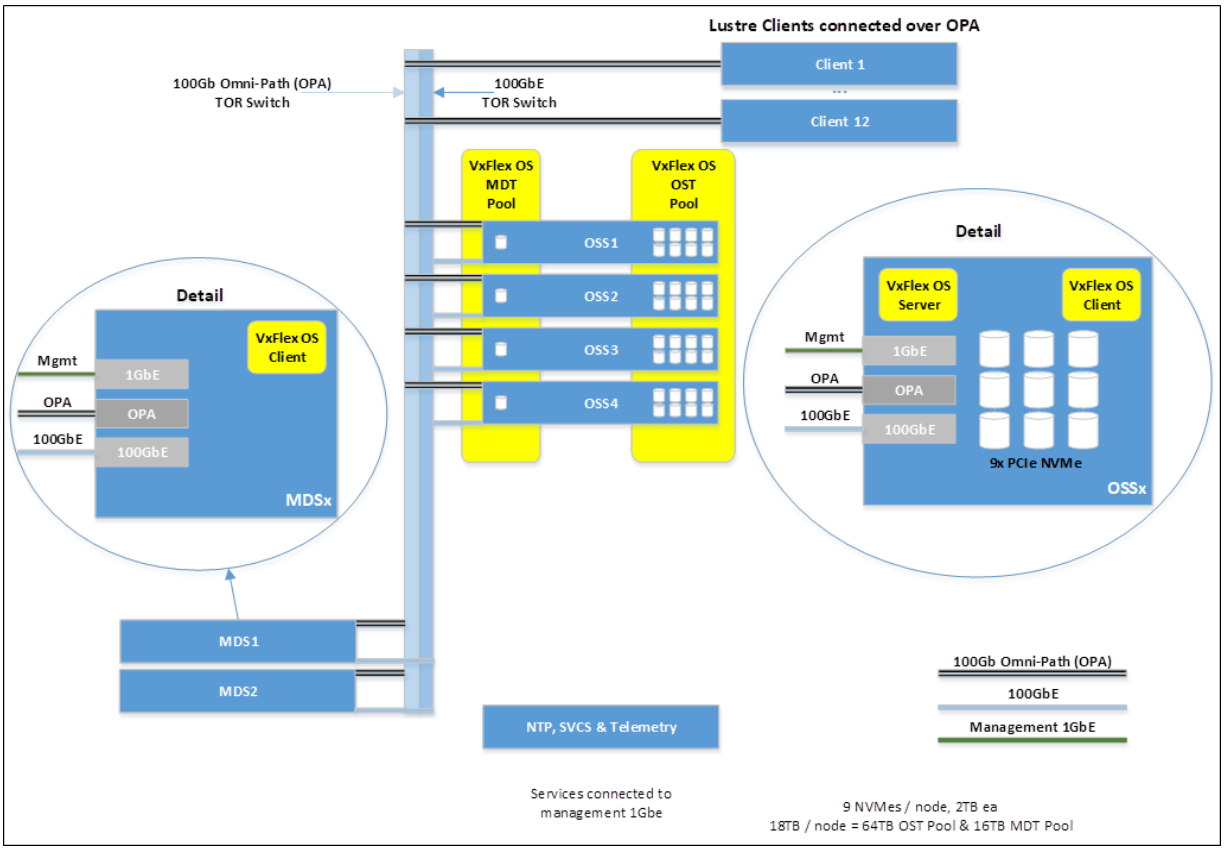

Figure 2: Diagram of Lustre/VxFlex OS cluster with 4 OSSs & SDSs plus 2 MDSs

Storage Data Servers (SDS) allocate and contribute storage to the overall storage SSD pool to create a software-defined, converged, shared SAN. This software is media and server agnostic; can be created on physical or virtual servers and it utilizes SATA/SAS SSD devices, as well as PCIe*-attached NVMe devices 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 our test bed we used Intel® SSD DC P4600 Series NVMe devices. These NVMe drives were used for both pools serving Lustre OSTs, MDTs, and the MGT.

The Lustre metadata subsystem is comprised of the Metadata Target (MDT), the Management Target (MGT), the Metadata Server (MDS) and the Management Server (MGT). 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 namespace and the MDT. In this test bed, we take advantage of the recently developed Distributed Name Space phase two features; 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 eight VxFlex OS servers with nine NVMe devices each.

A parallel file system, such as Lustre, delivers performance and scalability by distributing, or "striping," data across multiple Object Storage Targets (OSTs). bA 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 software is installed on the compute nodes and allows access to data stored on the Lustre file system. To the clients, the filesystem 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 operating system tools for easier administration.

To summarize, the elements of the Lustre filesystem 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, etc.
- 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 data stripes or extents of the files on a filesystem.
- Lustre Clients – Access the MDS to determine where files are located, access the OSSs to read and write data.

In all configurations, we used twelve servers as clients to the Lustre PFS, on which we run the IOR and MDtest benchmarks. All systems are running open-source CentOS* 7.4.

Installation Details

VxFlex* OS storage was configured using 2 Pools: one pool using one NVMe* in each SDS, with a total capacity of 2TB per OSS, and another pool including the remaining eight NVMe devices in each node with a total capacity of 16TB per OSS as indicated in Figure 1 above, for an aggregate of 144TB raw storage -- 16TB in the MDT pool plus 128TB in the OST pool. It is important to note that VxFlex OS was configured to mirror the volumes, so the durable storage was one half of the raw storage; 72TB total, with 8TB in the MDT pool and 64TB in the OSS pool. Each OSS server was also running both SDC and SDS. The MDS server was running only the SDC accessing the MDT pool.

We configured four 1.7TB LUNs per OSS server carved from the OST pool for the OST devices, one OST per LUN. We also carved two 3.4TB LUNs for MDTs and one 8GB LUN for the MGT from the MDT pool. These LUNs were accessed by the VxFlex OS client running on the MDSs and served by the VxFlex OS SDSs on OSS1 to OSS8.

For the scalability test case, only four OSS/SDS nodes were in use, and all pool and LUN sizes, with the exception of the MGT LUN, are one half of the above. In this case, the aggregate raw storage was 72TB, with 8TB in the MDT pool and 64TB in the OST pool. VxFlex OS was again configured to mirror the volumes, so durable storage was 36TB total, with 4TB in the MDT pool and 32TB in the OST pool.

Data Storage Proof of Concept

The following is a summary of the hardware and software components used to build the Lustre/VxFlex OS Cluster:

MDS Hardware Configuration	
MDS Server	Intel® Server System R2224WFTZ
Processor	Two (2) Intel® Xeon® Gold 6142 Processors
Memory	768 GB (24 x 32GB DIMMs) 2667 MHz DDR4
Omni-Path HCI	Intel® Omni-Path* HCI PCIe* Gen3 x16 adapter. (100Gbps)
Ethernet NIC	Qlogic* QL45611 100GbE NIC
SATA SSD (for boot)	Intel® SSD DC S4500 series, 480GB
OSS Hardware Configuration	
OSS Server	Intel® Server System R2208WFTZ
Processor	Two (2) Intel® Xeon® Gold 6130 Processors
Memory	192 GB (12 x 16GB DIMMs) 2667 MHz DDR4
Omni-Path HCI	Intel® Omni-Path* HCI PCIe Gen3 x16 adapter. (100Gbps)
Ethernet NIC	Qlogic* QL45611 100GbE NIC
SATA SSD (for boot)	Intel® SSD DC S4500 series, 480GB
PCIe Switch (for NVMes)	Two (2) Intel® AXXP3SWX08040
NVMe	Nine (9) Intel® SSD DC P4600 series, 2.0TB
Client Hardware Configuration	
Client server (4 blades in each chassis)	Chassis: Intel® Server Chassis H224XXKR2 Blades: Intel® Compute Module HNS2600TP
Processor	Two (2) Intel® Xeon® E5-2695v4 Processors
Memory	128 GB (16 x 8GB DIMMs) 2133MHz DDR4
Omni-Path HCI	Intel® Omni-Path* HCI PCIe Gen3 x16 adapter. (100Gbps)
SATA SSD (for boot)	Intel® SSD DC S3700 series, 200GB
Network Switches	
100 Gigabit Ethernet switch (VxFlex OS)	Arista* 7060CX-32 100GbE Network Switch: MTU 9000
Omni-path Switch (Lustre)	48 port Intel® Omni-Path* Edge Switch 100 series, 100SWE48QF

Storage Server Software	
Operating system	CentOS* 7.4 x86_64
Kernel	3.10.0-693.11.6.el7_lustre.x86_64
Lustre	2.10.3
VxFlex OS	2.5-1
OPA driver	IntelOPA-IFS.RHEL74-x86_64.10.7.0.0.145
Lustre Client Software	
Operating system	CentOS* 7.4 x86_64
Kernel	3.10.0-693.11.6.el7.x86_64
Lustre Client	2.10.3
OPA driver	IntelOPA-IFS.RHEL74-x86_64.10.7.0.0.145

Storage 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 Metadata Servers, one of which is also serving as the MGS. The MDSs are connected to the VxFlex OS MDT pool using the VxFlex OS SDC. For some test cases, only one MDS was active.
    c. Eight Lustre Object Storage Servers (OSS1-8) connected to the VxFlex OS OST pool using local NVMe in the OSSs.
2) The VxFlex OS cluster
    a. 72 x 2TB NVMe in the OSS/SDS servers – 144TB total.
    b. Using VxFlex OS, two storage pools were configured. One pool was configured using eight NVMe drives per OSS for use as OSTs. A second pool was configured using one NVMe per OSS server for use as MDTs and MGT.
    c. Three VxFlex OS volumes, one 8GB size (MGT) and two 3.5TB (MDTs) from the MDT pool were created. One MDS node has the 8GB and one 3.5TB volume while the other has only the second 3.5TB volume mapped.
    d. 32 VxFlex OS volumes (each 1.7TB in size) were created in the OST pool. Four volumes were mapped to each of the OSS servers.
3) All ten servers (8xOSS + 2x MDS) and all 12 clients are connected to a 10 gigabit Ethernet top of rack (TOR) switch for management.
4) All ten servers (8xOSS + 2x MDS) are connected to a 100 gigabit Ethernet TOR switch for VxFlex OS to use.
5) All ten servers and 12 client nodes are connected to a 100 gigabit 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_aqhead_mb=1024
- osc.lustrefs-OST*.max_rpcs_in_flight=16

For VxFlex OS SDC, SDS, and MDM, only the parameter listed below was modified:

- profile high_performance

Performance Evaluation and Configuration Details

The performance study presented in this paper is using two popular benchmarks used to evaluate storage for HPC; IOR* [1] and MDtest* [2]. Both benchmarks are using MPI communication between the compute cluster nodes for synchronization of the benchmark. We used only POSIX* IO for this evaluation as MPIIO* is limited to specific collective IO's 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 markers:

- Throughput, 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 both for traditional HPC workloads and for IOPS intensive workloads seen in enterprise environments. For IOR, we used a file for each process of the benchmark. With MDtest, we used a subdirectory per process with a varying number of files in each subdirectory. Each set of tests were executed five times on all twelve clients of the solution, and the results reported are the averages of those five runs. Performance was recorded for IOR transfer sizes 128KiB, 512KiB, 2MiB, 8MiB, and 32MiB. Performance results were also collected for MDtest with 128 bytes, 4KiB, and 128KiB file sizes. With MDtest, we varied the number of files used by each process from 1024 to 262144 files per process.

In summary the objectives of the evaluation are as follows:

- Investigate a more cost-effective high performance software-only Lustre on NVMe solution.
- Take advantage of VxFlex OS 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 bandwidth testing was done with the IOR* benchmark tool version 3.1.0. Each node used 72 MPI processes, the same number of threads per node (2x36). We conducted our tests using eight OSSs and we ran IOR configured for peak performance. This included use of 2MIB transfer size, 4MiB stripe size, and the tunings listed in Section 4. A similar test was run with four OSSs to verify scaling.

Figure 3 below shows the maximum BW results for both write and read for the configuration described in Figure 1. The maximum BW performance was obtained by using 2MiB IO size. The peak Lustre* write performance was 20.1GB/sec (at transfer size 32MiB) and peak read performance was 52.4GB/sec (at transfer size 2MiB). The graph shows performance for both the read and write tests across a wide range of IO sizes, from 128KiB to 32MiB.

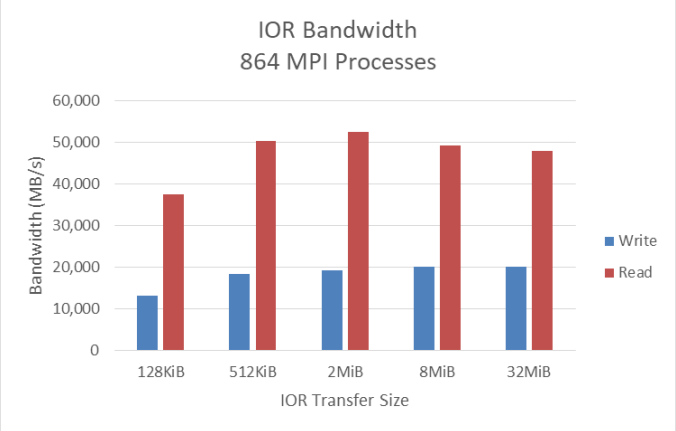

Figure 3: Lustre/ VxFlex OS IOR sequential BW across IOR transfer sizes

Figure 4 shows the benchmark results on a per-node basis. The peak write performance per OSS was 2.5GB/sec (at transfer size 32MiB), and the peak Lustre read performance per OSS was 6.5GB/sec (at transfer size 2MiB).

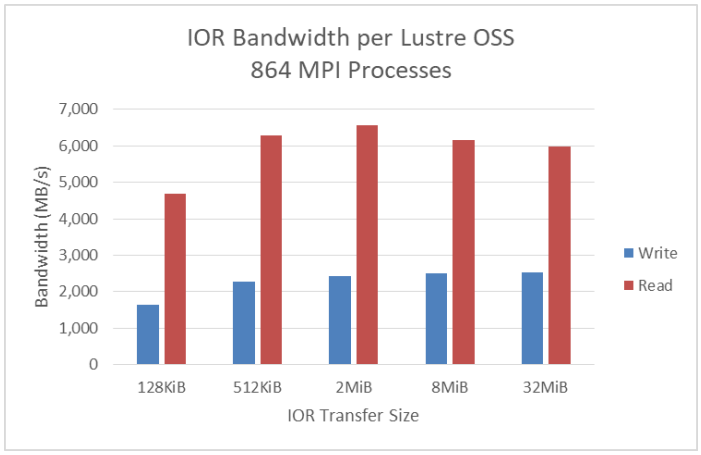

Figure 4: Lustre/ VxFlex OS IOR sequential BW per Lustre OSS across IOR transfer sizes

To verify scaling, the cluster was reduced to four Lustre OSS/ VxFlex* OS SDC nodes with otherwise identical configuration and the experiment was repeated for only the 2MiB transfer size. Figure 5 shows the four OSS IOR results graphed on a per-node basis. The Lustre write performance per OSS was 2.5GB/sec write and read performance per OSS was 5.5GB/s read, thus demonstrating that this solution scales from four to eight OSS/SDS nodes.

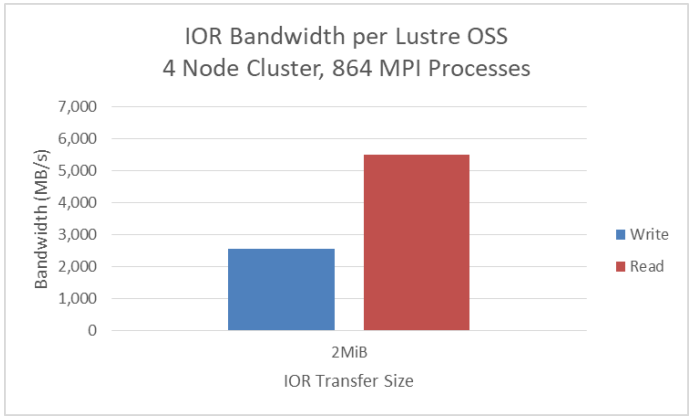

Figure 5: Four OSS/SDC Lustre/VxFlex OS IOR sequential BW per Lustre OSS across IOR transfer sizes

We were pleased to find minimal impact of IOR transfer size on the bandwidth of the cluster. Even at a transfer

size of 128KiB -- the lowest performing transfer size we tested -- the cluster was able to achieve 13.2GB/sec write and 37.4GB/sec read.

The performance results of IOR showed consistent results with good scale in a test range from 128KiB to 32MiB IOR transfer size and a 16GiB file size.

- Excellent I/O performance over wide IO sizes and ranges
- Low CPU utilization

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, we observed that the number of files per directory significantly affects the performance of the cluster. We measured the metadata performance while scaling up the number of files created in each directory. We measured this performance using 128 byte, 4KiB, and 128KiB files. We included the extremely small 128 byte and 4KiB files 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.

We varied the number of files per process from 1K to 256K files per process (442,368 files to 113,246,208 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. We used a maximum of 262,144 (256K) files per process as we were limited by the maximum number of inodes available with default ldiskfs parameters on the MDTs, and we felt that in most usage models, it will not be necessary to store over 100 million files.

Figures 6a through 6d show the MDtest results, in operations per second, for create, stat, read, and remove tasks on 128 byte, 4KiB, and 128KiB file sizes when two MDSs are in use for 1024 through 262144 files per process.

At 65,536 4KiB files per process, MDtest is able to perform 68.8K file create ops/sec, 491K stat ops/sec, 138K read ops/sec and 119K remove ops/sec. Peak performance for 4KiB files occurs at 2,048 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, 262,144 4KiB files per process (a total of 113,246,208 files), MDtest is able to perform 61.4K create ops/sec, 266.K stat ops/sec, 151.4K read ops/sec, and 81.6K remove ops/sec.

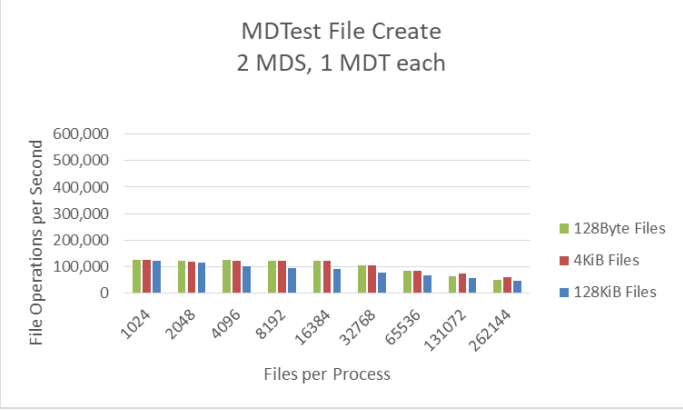

Figure 6(a)

Figure 6 (four graphs total): MDtest Performance for: (a) file create (b) file stat (c) file read, and (d) file remove

To provide context to the performance figures here, Oak Ridge National Laboratory* (ORNL*) performed an evaluation of Lustre with DNE2 enabled across eight

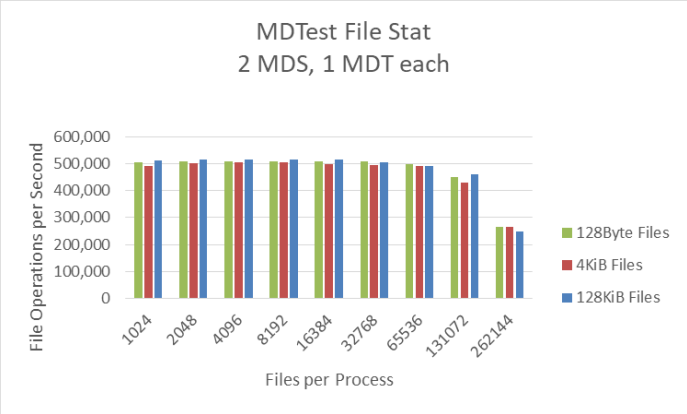

Figure 6(b)

MDSs. They ran 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 121K create ops/sec, 510K stat ops/sec, and 210K delete ops/sec when operating on a slightly larger dataset (16,384 files per process instead of 10,000 files per process).

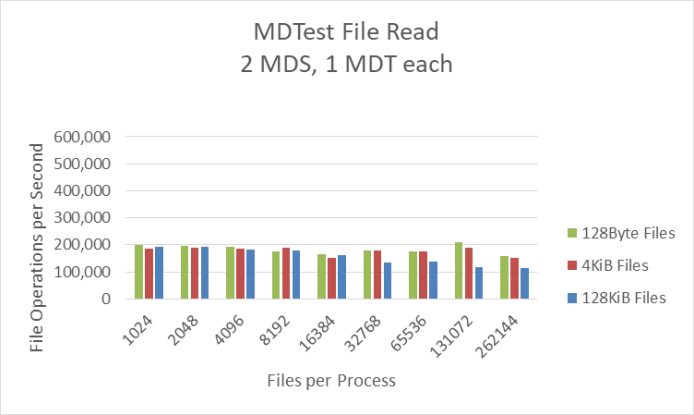

Figure 6(c)

We also ran MDtest against the four inode cluster to verify scalability in metadata performance. Figure 7 shows the results of this test when run with 65,536 4KiB files per process. With the only four OSS/SDC inodes, Lustre metadata performance was at least 85% the eight OSS/SDC cluster.

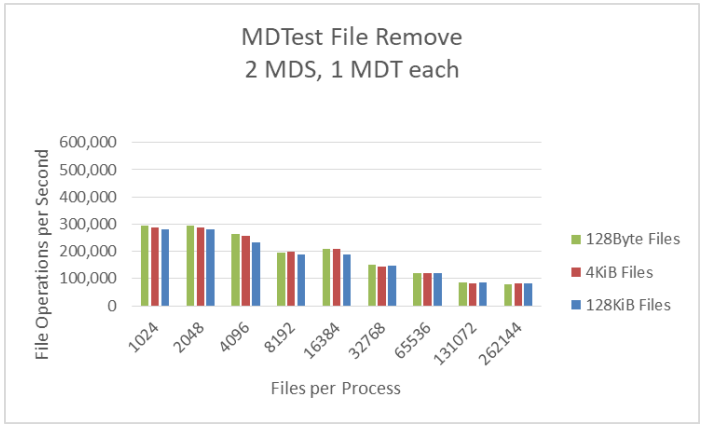

Figure 6(d)

Additionally, we ran the four inode cluster with just one MDS to verify that MDS performance can be scaled. Due to the locking behavior of Lustre and the high load we are applying with these tests, scaling from 1 MDS to 2 MDS was greater than 98% for all operations except file create. File creation is a much more demanding task for the file system than other file operations, and improved by 14%.

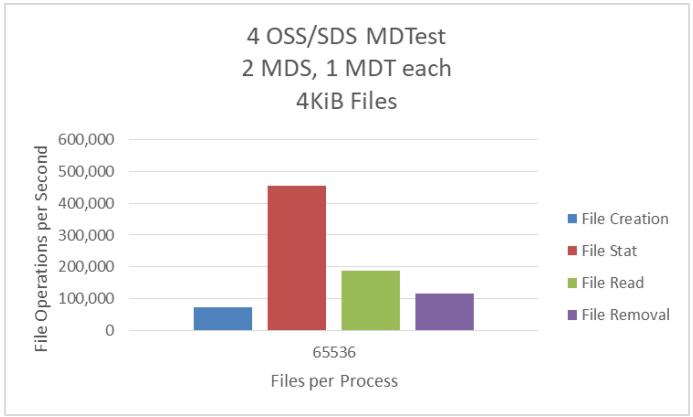

Figure 7: MDtest Performance with only four OSS/SDC nodes

These results, along with the four OSS/SDS IOR results demonstrate that the solution can have Lustre metadata and Lustre data performance scaled independently. Additionally, the eight OSS/SDS results demonstrate that this solution can provide great bandwidth (up to 20.1GB/sec IOR write and up to 52.4GB/sec IOR read) as well as outstanding Lustre metadata performance (up to 121K create ops/sec, 510K stat ops/sec, 165K read ops/sec and 210K delete ops/sec).

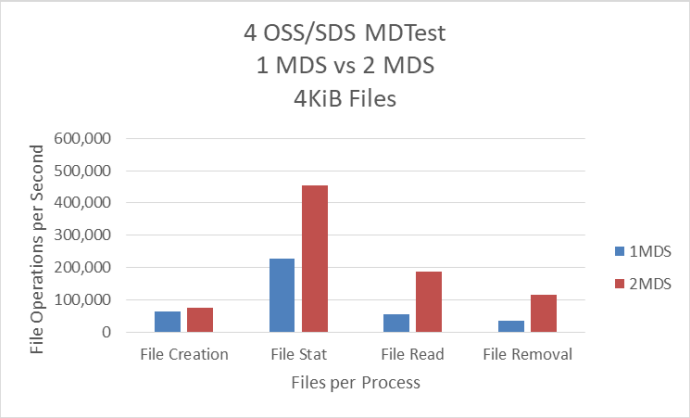

Figure 8: MDS scaling with only four OSS/SDS nodes

Conclusions

This Lustre*/ VxFlex* OS solution offers excellent performance in a compact form factor (20U using standard 2U 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. In a storage system where performance is paramount this solution provides very high read and write rates with very low latency and true data durability.

Intel engineers evaluated competing proprietary and open solutions on the market and from a limited study determined that this hybrid architecture enables a stable, cost effective high performance storage capability that will be hard to match. Moreover, the extremely high performance of this solution helps to mitigate the impact on a user attempting to interact with Lustre as if it is a simple local file system.

The Lustre/VxFlex OS solution discussed has also been shown to scale almost linearly in both Lustre Metadata Servers and in Lustre Object Storage Servers (also acting as VxFlex OS Storage Data Servers). If additional bandwidth or capacity is required, OSS/SDC nodes can be added. If additional metadata performance is required, additional MDS nodes can be added. This enables the cluster operator to more easily scale to meet growing performance and capacity demands without having to purchase an entirely new storage array.

References

[1] IOR Benchmark: http://www.csm.ornl.gov/essc/io/IOR-2.10.1.ornl.13/USER_GUIDE

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

[3] "Lustre Distributed Name Space (DNE) Evaluation at the Oak Ridge Leadership Computing Facility (OLCF)"; 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 VxFlex OS 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

We used 16GB files that will result in 6TB dataset. We also used the directIO option –B all tests.

The directIO command line parameter ("-B") allows us 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

We used same command 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, 4KiB, and 128