Dell™ | Terascala
HPC Storage Solution (DT-HSS4.5)

A Dell Technical White Paper

Mario Gallegos

Dell HPC Engineering

April 2013 | Version 1.0

Contents

Figures

Tables

1. Introduction

In High-Performance computing, the efficient delivery of data to and from the compute nodes is critical. The speed at which researchers can generate and consume data in HPC systems, is making storage a bottleneck; managing and monitoring complex storage systems is growing the burden on storage administrators and researchers. Data requirements around performance and capacity are increasing rapidly. Increasing the throughput and scalability of storage devices supporting compute nodes can require a great deal of planning and configuration.

The Dell | Terascala HPC Storage Solution, referred to as HSS for the rest of this document, is designed for researchers, universities, HPC users, and enterprises who need to deploy a fully-supported, easy-to-use, high-throughput, scale-out, and cost-effective parallel file system storage solution. HSS is a scale-out storage solution appliance capable of providing high throughput storage. Utilizing an intelligent and extensive, yet intuitive, management interface the solution greatly simplifies managing and monitoring all of the hardware and file system components. It is easy to scale in capacity or performance or both, thereby providing a convenient path to grow in the future. The storage appliance uses Lustre®, the leading HPC open source parallel file system.

The storage solution is delivered as a pre-configured, ready to go appliance and is available with full hardware and software support from Dell and Terascala. Utilizing the 12th generation of enterprise Dell PowerEdge™ servers and the latest high density PowerVault™ storage products, the latest Dell | Terascala HPC Storage Solution, referred to as HSS4.5 in the rest of the paper, delivers a superior combination of performance, reliability, density, ease of use and cost-effectiveness.

Due to the number and nature of the changes made in the fourth generation of the Dell | Terascala High-Performance Computing Storage Solution, it is important to evaluate the solutions' performance to determine its capabilities. This paper describes the latest solution, and outlines its performance characteristics.

The following sections of this paper describe the Lustre File System the HSS4.5 appliance, followed by performance analysis and conclusions. Readers may refer to "Appendix B: Lustre Client RPMs installation" for integration details of Lustre with compute nodes using a RHEL6.2 kernel.

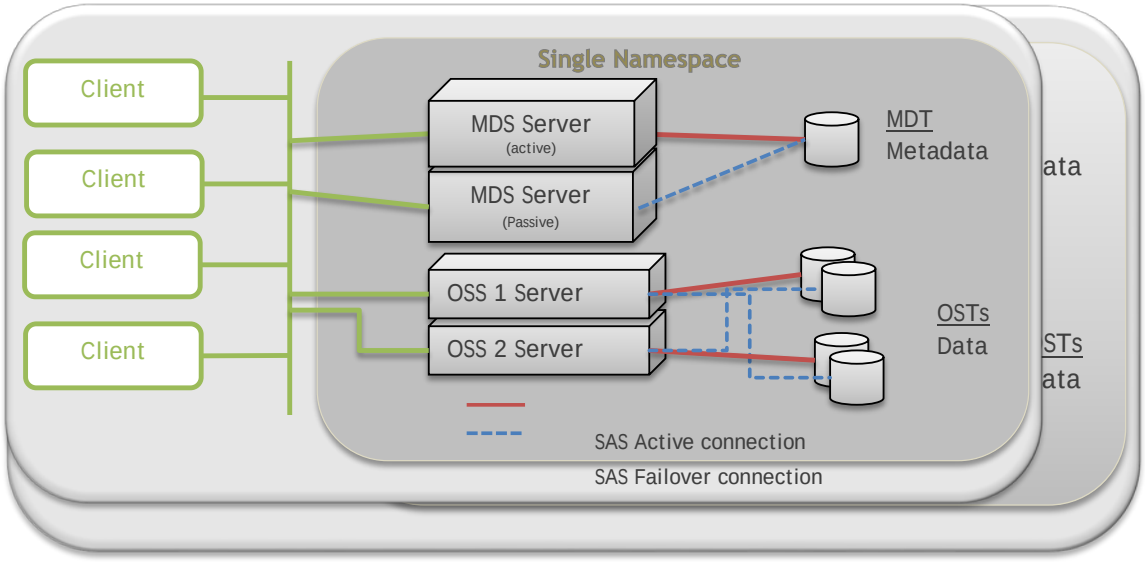
2. The Lustre File System

Lustre is a parallel file system, offering high performance through parallel access and distributed locking. In the HSS family, storage is provided using a single namespace that is easily accessed by the cluster's compute nodes, and managed through an extensive Web-based interface or a Java Management Console. A Lustre installation consists of three key elements: the metadata subsystem, the (data) object storage subsystem, and the compute clients that access and operate on the data.

The metadata subsystem is comprised of the Metadata Target (MDT) and a 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 MDS is a dedicated server that manages the MDT. In the HSS4.5 storage appliance, there is an active/passive pair of MDS servers providing a highly available metadata service to the cluster.

The object storage subsystem is comprised of one or more Object Storage Targets (OST) and one or more Object Storage Servers (OSS). The OSTs provides storage for file object data, while each OSS manages one or more OSTs. Typically, there are several active OSSs at any time. Lustre is able to deliver increased throughput by increasing the number of active OSSs (and associated OSTs). Each additional OSS increases the maximum networking throughput, while each additional OST increases the storage capacity. The HSS4.5 configuration evaluated in this study included one active/active pair of OSS servers that provide a highly available object data service to the cluster. Figure 1 shows the relationship of the MDS, MDT, OSS and OST components of the Lustre file system. Clients in the figure are the HPC cluster's compute nodes.

Figure 1: Lustre Overview



A parallel file system, such as Lustre, delivers performance and scalability by distributing data (using "striping" data) across multiple Object Storage Targets (OSTs), allowing multiple compute engines to access the data simultaneously.

The Lustre client software is installed on the compute nodes to allow access to data stored in the Lustre file system. To the clients, the file system appears as a single branch in the file system tree (single namespace). This single directory provides a simple starting point for application data access, and allows access via native client operating system tools for easier administration.

Lustre includes a sophisticated storage network protocol enhancement, referred to as lnet, which is capable of leveraging certain types of network features. For example, when the HSS4.5 utilizes InfiniBand as the network to connect the clients, MDS and OSSs, lnet's sophisticated features enable Lustre to take advantage of the RDMA capabilities of the InfiniBand fabric to provide faster I/O transport than experience with typical networking protocols.

In particular, this version of the HSS4.5 supports Mellanox ConnectX-3 InfiniBand FDR (56 Gb/s) adapters, which takes advantages of the PCIe 3.0 supported by Dell's 12th generation servers. Alternatively, HSS4.5 can support 10 Gb/s Ethernet as a media to connect to clients. Past studies (HSS3) have shown that the network very quickly becomes the bottleneck in 10 Gb/s Ethernet based

HSS solutions. Hence to characterize the maximum capabilities of the complete solution, this study focuses only on the performance of the InfiniBand based HSS4.5 configuration at FDR speeds.

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

- Metadata Target (MDT) – Stores the location of "stripes" of data, file names, time stamps, etc.
- Metadata Storage Server (MDS) – Manages the MDT providing Lustre clients access to files.
- Object Storage Target (OST) – Stores the data stripes or extents of the files on a file system.
- Object Storage Server (OSS) – Manages the OSTs providing Lustre clients access to the data.
- Lustre Clients – Access the MDS to determine where files are located, then access the OSSs to read and write data

Typically, Lustre deployments and configurations are considered very complex and time consuming tasks. Lustre installation and administration is normally done via a command line interface, requiring extensive knowledge of the file system operation, along with the auxiliary tools like lnet and the locking mechanisms. Such requirements, and the steep learning curve associated with them, may prevent Systems Administrators unfamiliar with Lustre from performing an installation, possibly preventing their organization from experiencing the benefits of a clustered file system. Even for experienced Lustre System Administrators, maintaining a Lustre file system can take a large portion of their time.

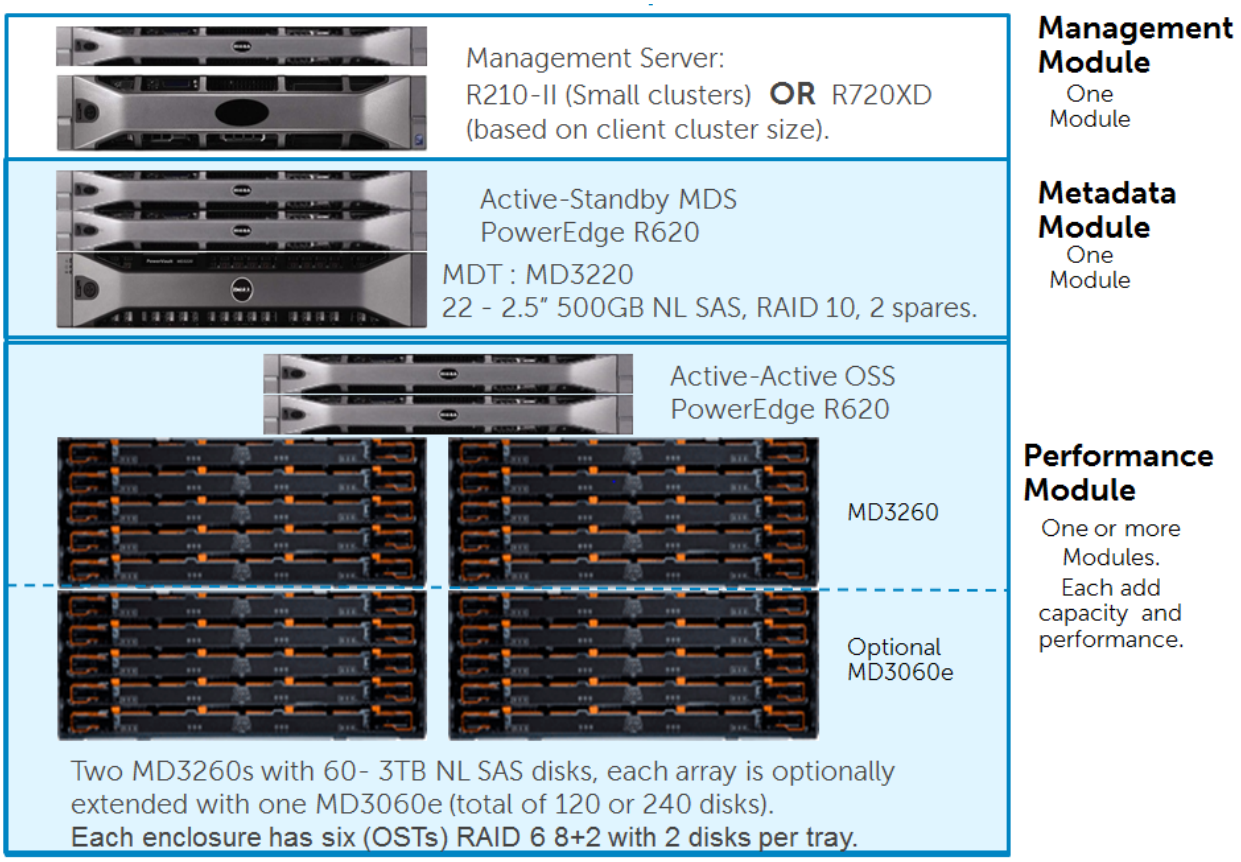
The Dell | Terascala HPC Storage Solution removes the complexities of installation, minimizing Lustre deployment and configuration time. It also automates the monitoring of the health and performance of the different appliance components. The decrease in time and effort speeds up the general preparations for production use. The automation of the appliance monitoring provides a better service to end users without increasing the burden for system administrators. In addition, the solution provides invaluable tools that help troubleshoot problems related to performance stemming from use or misuse of the file system by users and applications. Finally, the ability of keeping historical information for extended periods of time (years), allow for better planning for expansion/maintenance and upgrades of the storage appliance.

3. Dell | Terascala HPC Storage Solution Description

The HSS4.5 solution provides a pre-configured storage solution consisting of a management server, Lustre Metadata Servers, Lustre Object Storage Servers, and the associated backend storage. Figure 2 shows an overview of a HSS4.5 system, including some basic information about the different components, which will be described later in this section. Note the three mayor components:

- A Management Module.
- A Metadata Module (or Metadata Server pairs).
- One or more Performance Modules (or Object Storage Server pairs).

Figure 2 HSS4.5 Components Overview



The appliance software images have been modified to support the Dell PowerEdge R620 as the Object Storage and Metadata Servers in the configuration. This PowerEdge R620, shown in Figure 3, allows for a significant improvement in the server density, performance and serviceability of these solution components with a decrease in the overall complexity of the solution itself.

Figure 3: Dell PowerEdge R620

3.1 Management Module

The Management Module is a single server connected to the rest of the HSS servers via an internal 1 GbE network. The server can be either a PowerEdge R210-II for small clusters or a PowerEdge R720-XD for large clusters, since client health information can also be monitored (via optional software) and historical data grows proportional to the client cluster size.

The management server is responsible for user interaction as well as systems health management and monitoring. All user-level access to the HSS4.5 appliance is via this device. While the management server is responsible for collecting data and management, it does not play an active role in the Lustre file system itself and can be serviced without requiring downtime on the file system. The management server presents the data collected and provides management via an interactive Java GUI called Terascala Management Console. Alternatively, a new and more powerful web GUI named TeraView

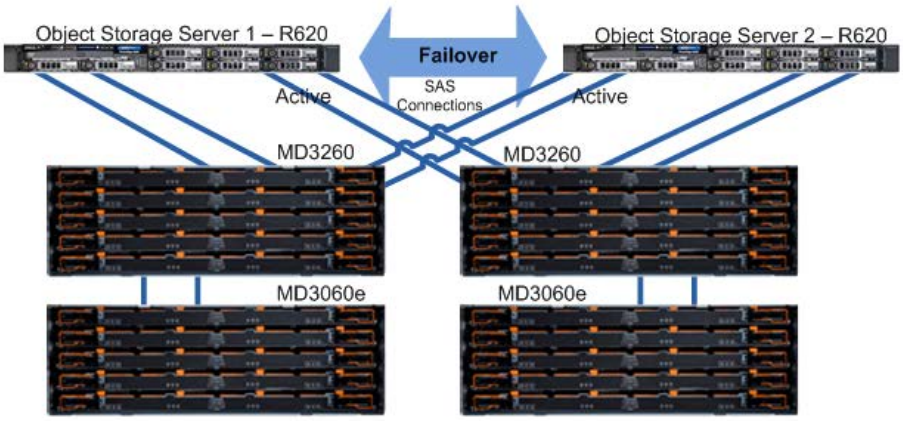
provides an increased level of monitoring including client cluster monitoring, but it is an optional product and out of the scope for this whitepaper. HSS4.5 users interact with the management server via Java Management Console or TeraView.

3.2 Metadata Module

In the latest HSS4.5 solution, the Metadata Server pair is comprised of two Dell PowerEdge R620 servers configured as an active/passive highly available cluster. Each server is directly attached to a Dell PowerVault MD3220 storage array housing the Lustre MDT. The Dell PowerVault MD3220 is fully populated with 24 - 500 GB, 7.2K, 2.5" near-line SAS drives configured in a 22 disks RAID10 with 2 hot spares, for a total available raw storage of 5.5 TB. In this single Metadata Target, the HSS4.5 provides about 4.8 TB of formatted space for file system metadata. The MDS is responsible for handling file and directory requests and routing tasks to the appropriate Object Storage Targets for fulfillment. With a single MDT of this size, the maximum number of files that can be served will be in excess of 1.45 billion. Storage requests are handled across Inet by either a single 56 Gb/s FDR InfiniBand or a single 10 Gb/s Ethernet connection.

Figure 4 Metadata Server Pair

Figure 5: Object Storage Server Pair



3.3 Performance Module

The hardware improvements also extend to the Object Storage Server. The PowerEdge R620 server is the new standard for this solution, providing 2X server density compared to the previous version, HSS3. In the HSS4.5, Object Storage Servers are arranged in two-node high availability (HA) clusters providing active/active access to two Dell PowerVault MD3260 high density storage arrays. Each PowerVault MD3260 array is fully populated with 60 - 3 TB 3.5" 7.2K near-line SAS drives (2 TB and 1 TB disks are also supported). Capacity of each PowerVault MD3260 array can be extended with one additional PowerVault MD3060e high density expansion array. Therefore, each OSS pair provides raw storage

capacity ranging from 360 TB up to 720 TB of raw storage. Since PowerVault MD3260 and MD3060e arrays hold 60 disks in 4U chassis, HSS4.5 provides 2.5X higher density compared to HSS3.

Object Storage Servers are the building blocks of the HSS solutions. With two dual port 6 Gb/s SAS controllers in each PowerEdge R620, the two servers are redundantly connected to each of two PowerVault MD3260 high density storage arrays.

The 60 - 3 TB drives in each PowerVault enclosure provide a total of 180 TB raw storage capacity per array. Each storage array is divided into six RAID 6 virtual disks, with eight data and two parity disks (using two disks per tray of the array) per virtual disk, to yield six Object Storage Targets per enclosure. By using RAID 6, HSS4.5 provides higher reliability compared to previous versions of the appliance, at a marginal cost on write performance (due to the extra set of parity data required by each RAID 6). Each OST provides almost 22 TiB of formatted object storage space. A single OSS pair has 12 OSTs at a minimum and can be expanded by an additional 12 OSTs by adding the PowerVault MD3060e expansion arrays, for a maximum of 24 OSTs per OSS module. The OSTs are exposed to clients with lnet via 56 Gb/s Infiniband FDR or 10 Gb/s Ethernet connections on the OSS.

When viewed from any compute node equipped with the Lustre client the entire namespace can be viewed and managed like any other file system, but with the enhancements of Lustre management. As an example, a typical 720 TB HSS4.5 configuration is shown in Figure 6.

Figure 6: 720 TB Dell | Terascala HPC Storage Solution typical configuration

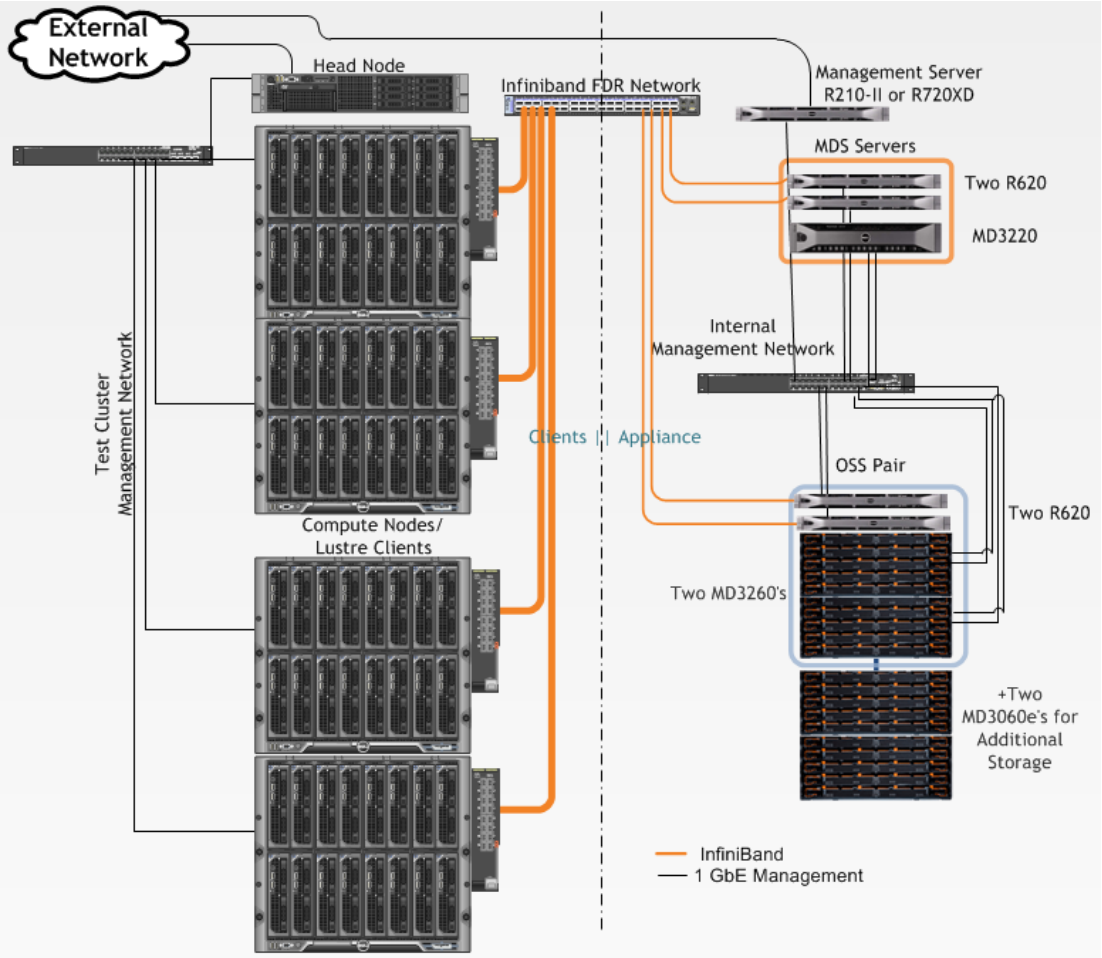
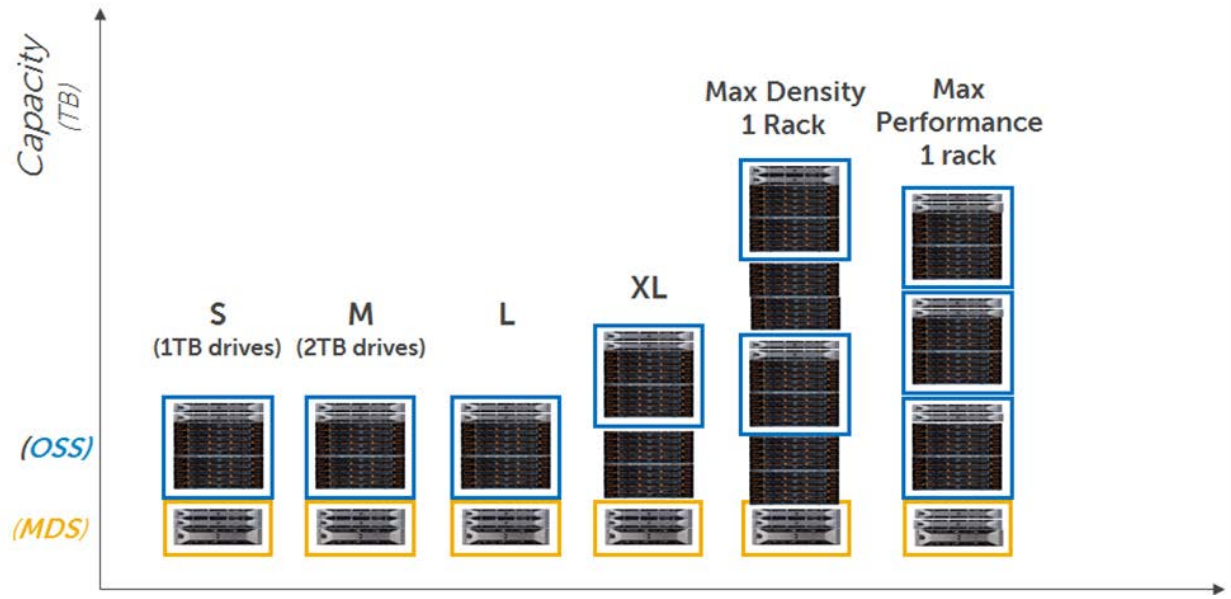


Figure 7 shows the different configurations we currently offer for HSS4.5, which are available with different hard disks sizes: 1, 2, or 3 TB. Also, Figure 7 shows the maximum footprint used on the rack, the raw capacity, disk drives used and estimated read and write sequential performance[2].

Figure 7: Current HSS4.5 Configurations



Total U[1]	16U	16U	16U	26U	42U	36U	*Custom*
Raw Capacity	120TB (1 OSS) 1 TB Drives	240TB (1 OSS) 2 TB Drives	360TB (1 OSS) 3 TB Drives	720TB (1 OSS) 3 TB Drives	1440TB (2 OSSs) 3 TB Drives	1080TB (3 OSSs) 3 TB Drives	*Custom (PB+)*
Peak Read[2] Performance	6.7 GB/s	6.7 GB/s	6.7 GB/s	6.7 GB/s	13.4 GB/s	20.1 GB/s	*Custom 10s GB/s*
Peak Write[2] Performance	3.5 GB/s	3.5 GB/s	3.5 GB/s	3.5 GB/s	7 GB/s	10.5 GB/s	*Custom 10s GB/s*

[1] Management Server not shown; it can be 1U (R210-II) for small clusters or 2U (R720XD) for large clusters. 2U size was assumed for all "Total U" calculations.
[2] The performance values listed are based on the studies performed ONLY on the XL configuration (3 TB disks); which are expected to yield the same throughput for the rest of the configurations.

3.4 Scalability

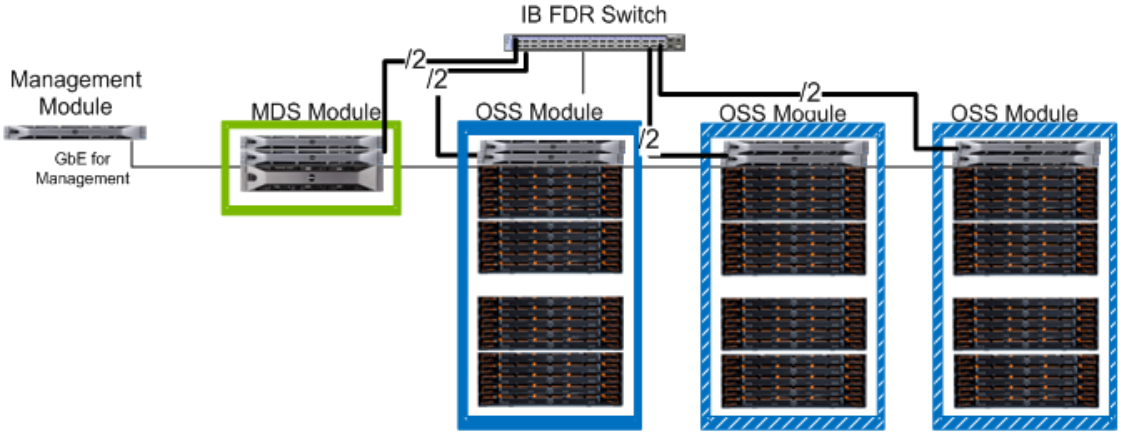
Providing the Object Storage Servers in active/active cluster configurations yields greater throughput and product reliability. This configuration provides high availability, decreasing maintenance requirements and consequently reducing potential downtime.

The PowerEdge servers provide greater performance and density. This 4th Generation solution provides 360 TB up to 720 TB of raw storage for each OSS pair. The HSS4.5 solution leverages the FDR InfiniBand interconnect for very high-speed, low-latency storage transactions or 10 Gb/s Ethernet can be used for high speed, lower cost and to allow the use of existing 10 GbE infrastructure. With this version of the solution, an upgrade of the OSS to the PowerEdge R620 takes advantage of the PCIe Gen3 interface for FDR InfiniBand helping achieve higher network throughput per OSS. Additionally the PowerEdge 620 supports faster 1600 MT/s memory and faster processors with more cores compared to previous servers.

An RPM based Lustre client for the RHEL6.2 kernel with Mellanox OFED v1.5.3.3 is available for access to HSS4.5 (for details see Appendix B: Lustre Client RPMs installation).

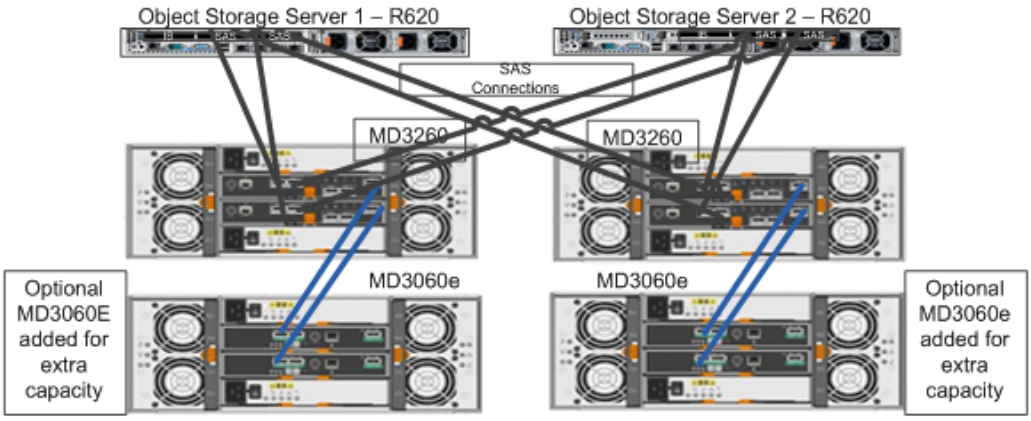
This combination of components from storage to client access is formally offered as the HSS4.5 appliance.

Figure 8: OSS Scalability



Scaling the HSS4.5 can be achieved in two ways. The first method, demonstrated in Figure 8, adds additional OSS pairs (inside the dotted blue boxes), thus increasing both the total network throughput and increasing the storage capacity at once. Also implied in Figure 8, the second method involves simply expanding the storage capacity of any single OSS pair by adding two PowerEdge MD3060e storage expansion arrays (bottom 2 arrays on each OSS module) to the 2 PowerVault MD3260s (top 2 arrays in each OSS module). This allows for an increase in the volume of storage available while maintaining a consistent maximum network throughput. This is better explained in Figure 9, where the SAS connections for a config that uses 120 disks are shown in black. Scaling the capacity of the OSS is achieved by using the MD3060e expansion arrays, with the expansion SAS connections showed in blue.

Figure 9: SAS Cabling Diagram for OSS



3.5 Networking

3.5.1 Management Network

The private management network (see Figure 6, on the right side) provides a communication infrastructure for Lustre and Lustre HA functionality as well as storage configuration, monitoring and maintenance. This network creates the segmentation required to facilitate day-to-day operations and to limit the scope of troubleshooting and maintenance. Access to this network is provided via the Management Server, which extends a single communications port for the external management network. The management server uses this network to interact with the different HSS4.5 components to query and collect systems health information as well as to perform any management changes initiated by administrators. Both OSS and MDS servers interact with the management server to provide health information, performance data, and to interact during management operations. Similarly, MD3220 and MD3260 controllers are accessed via the out of band (ethernet ports) to inform about systems health, and to perform any management actions initiated by administrators.

This level of integration allows even inexperienced operator to efficiently and effortlessly monitor and administer the appliance. Information provided is summarized for quick inspection, but users can zoom in to a level of details of HW error messages from server or storage components.

3.5.2 Data Network

The Lustre file system is served via a preconfigured LustreNet implementation on either InfiniBand FDR or 10 Gb/s Ethernet. This is the network used by the clients to access data. In the InfiniBand network, fast transfer speeds with low latency can be achieved. LustreNet leverages the use of RDMA for rapid file and metadata transfer from MDTs and OSTs to the clients. The OSS and MDS servers take advantage of the FDR InfiniBand fabric with single port Mellanox ConnectX-3 56 Gb adapters. The FDR InfiniBand HBAs can be integrated in to existing QDR or DDR networks if needed. With the 10 Gb/s Ethernet network, Lustre can still benefit from fast transfer speeds and take advantage of the lower cost and pervasiveness of Ethernet technology, allowing leverage of any existing 10 GbE infrastructure.

3.6 Managing the Dell | Terascala HPC Storage Solution

The Terascala Management Console (TMC) takes the complexity out of administering a Lustre file system by providing a centralized GUI for management purposes. The TMC can be used as a tool to standardize the following actions: mounting and unmounting the file system, initiating failover of the

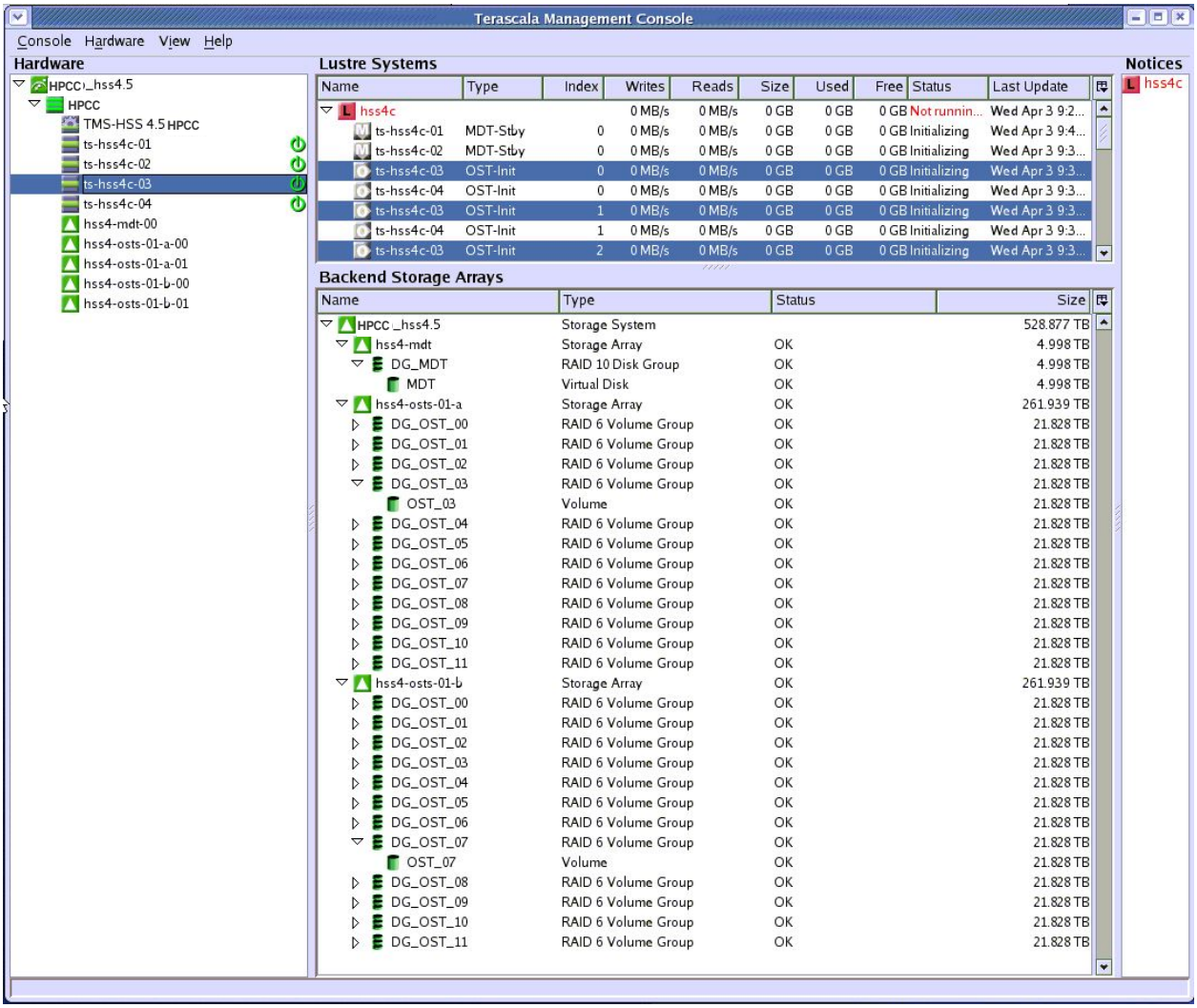
file system from one node to another (for either OSS or MDS), and monitoring performance of the Lustre file system and the status of its components. Figure 10 illustrates the TMC main interface.

The TMC is a Java-based application that can be run from any Java JRE equipped computer to remotely manage the solution (assuming all security requirements are met). It provides a comprehensive view of the hardware and file system, while allowing monitoring and management of the HSS solution.

Figure 10 shows the initial window view of a HSS4.5 system. In the left pane of the window are all the key hardware elements of the system. Each element can be selected to click down for additional information. In the center pane is a view of the system from the Lustre perspective, showing the status of the MDS and various OSS nodes. In the right pane is a message window that highlights any conditions or status changes. The bottom pane displays a view of the PowerVault storage arrays.

Using the TMC, many tasks that once required complex CLI instructions, can now be completed easily with a few mouse clicks. The TMC can be used to shut down a file system, initiate a failover from one MDS to another, monitor the storage arrays, etc.

Figure 10 Terascala Management Console



4. Performance Evaluation and Analysis

The performance studies presented in this paper profile the capabilities of a selected HSS4.5 configuration, the XL which has 240 – 3 TB disk drives (720 TB raw space). The goal is to quantify the capabilities of the solution, points of peak performance and the most appropriate methods for scaling for a variety of use cases. The client test bed used to provide I/O workload to test HSS4.5 solution is a Dell HPC compute cluster based on R410s, with configuration as described in Table 1.

A number of performance studies were executed, stressing a HSS4.5 XL configuration with different types of workloads to determine the limitations of performance and define the sustainability of that performance. InfiniBand was the network technology used for these studies since its high speed and low latency allows getting the maximum performance from HSS4.5, avoiding network bottlenecks.

Table 1: Test Client Cluster Details

Component	Description
Compute Nodes:	Dell PowerEdge R410, 64 nodes
Node BIOS:	1.2.4
Processors:	Two Intel Xeon™ E5540 @ 2.53 GHz quad core processors
Memory:	6 x 4 GiB 1333 MT/s RDIMM
Interconnect:	InfiniBand - Mellanox Technologies MT26428 (QDR)
Lustre:	Lustre 1.8.8 + Mellanox OFED Client
Cluster Suite:	ClusterCorp Rocks v6.0.1
OS:	Red Hat Enterprise Linux 6.2 (2.6.32-220.el6.x86_64)
IB SOFTWARE:	Mellanox OFED 1.5.3-3.0.0

Performance analysis was focused on three key performance markers:

- Throughput, data sequentially transferred in GB/s.
- I/O Operations per second (IOPS).
- Metadata Operations per second (OP/s).

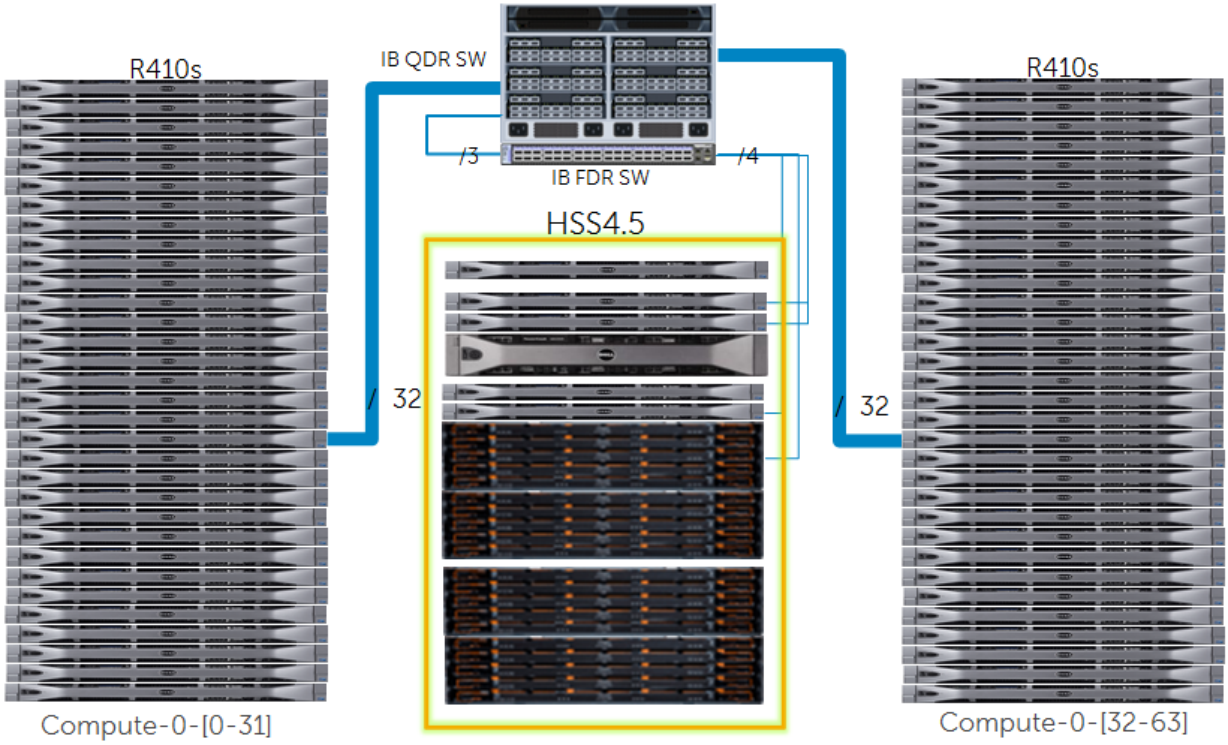
The goal is a broad but accurate review of the capabilities of the Lustre appliance. We selected three benchmarks to accomplish our goal: IOzone, IOR and MDtest.

There are two types of file access methods used with these benchmarks. The first file access method is N-to-N, where every thread of the benchmark (N clients) writes to a different file (N files) on the storage system. IOzone and IOR can both be configured to use the N-to-N file-access method. The second file access method is N-to-1, where every thread writes to the same file (N clients, 1 file). IOR can use MPI-IO, HDF5, or POSIX to run N-to-1 file-access tests. N-to-1 testing determines how the file system handles the overhead introduced with multiple concurrent requests when multiple clients (threads) write or read to the same file. The overhead encountered comes from threads dealing with Lustre's file locking and serialized writes. See Appendix A for examples of the commands used to run these benchmarks.

Each set of tests was run on a range of clients to test the scalability of the solution. The number of simultaneous physical clients involved in each test was varied from one to 64 clients. The number of threads corresponds to the number of physical servers up to 64. That is to say that the number of threads is consistently 1 per client until 64. Total numbers of threads above 64 were achieved by

increasing the number of threads per client across all clients. For instance, for 128 threads, each of the 64 clients runs 2 threads. Figure 11 shows a diagram of the cluster configuration used for this study.

Figure 11: Test Cluster Configuration



The test environment for the HSS4.5 has a single *MDS* and a single *OSS Pair* (XL configuration) with a total of 720 TB of raw disk space. The OSS pair contains two PowerEdge R620s, each with 128 GiB of memory, two 6Gbps SAS controllers and a single Mellanox ConnectX-3 FDR HCA. The MDS has 128 GiB of memory, a single 6Gbps SAS controller and a single Mellanox ConnectX-3 FDR HCA.

The InfiniBand fabric comprised of a Mellanox IS5100 QDR InfiniBand 72 port switch for the client cluster and a 36 port Mellanox SX6036 FDR switch for the HSS4.5. 3 ports on the IS5100 switch were connected to the SX6036.

Table 2 shows the details about the characteristics for the different software and hardware components of HSS4.5.

Table 2: HSS4.5 Configuration

Configuration Size	720 TB XL
Lustre Server Version	1.8.8
Software Version	TeraOS 3.0.24 (Kernel 2.6.18-308.4.1.el5_lustre.1.8.8_1.3.3)
OSS Nodes	2 x PowerEdge R620 Servers
OSS Memory	128 GiB Memory 1600 MT/s
OSS Processors	2 x Intel Xeon™ E5-2660 @ 2.20 GHz eight cores
OSS Server BIOS	1.3.6
OSS Storage Array	2 x PowerVault MD3260, 2 x PowerVault MD3060e
Drives in OSS Storage Arrays	240 3.5" 3 TB 7200 RPM Nearline SAS
OSS SAS Controllers	2 x SAS 6 Gbps HBA
MDS Nodes	2 x PowerEdge R620 Servers
MDS Memory	128 GiB Memory 1600 MT/s
MDS Processors	2 x Intel Xeon™ E5-2660 @ 2.20 GHz eight cores
MDS Server BIOS	1.3.6
MDS Storage Array	1 x PowerVault MD3220
Drives in MDS Storage Array	24 - 2.5" 500 GB 7200 RPM Nearline SAS
MDS SAS Controller	1 x SAS 6 Gbps HBA
Data network - InfiniBand	
HSS4.5 Servers	Mellanox ConnectX-3 FDR HCA MT27500
Compute Nodes	Mellanox ConnectX-2 QDR HCA MT26428
Client QDR IB Switch	Mellanox 72 Port IS5100
HSS4.5 FDR IB Switch	Mellanox 36 Ports SX6036
IB Switch Connectivity	Clients: QDR Cables; Servers: FDR Cables 3 uplinks from the QDR switch to the FDR switch

To prevent inflated results due to caching effects, tests were performed with a *cold cache* established with the following technique. Before each test started, a *sync* was performed and the kernel is instructed to drop caches on all the clients with the following commands:

- `sync`
- `echo 3 > /proc/sys/vm/drop_caches`

In addition, to simulate a cold cache on the server, before each test started, on all the active servers (OSS and MDS) a "*sync*" was performed and the kernel is instructed to drop caches with the same commands used on the client. Also, after each test, a "sync" was performed.

In measuring the performance of the HSS4.5, all tests were performed within similar environments. The file system was configured to be fully functional and the targets tested were emptied of files and directories prior to each test.

4.1 N-to-N Sequential Reads / Writes

The sequential testing was done with the IOzone testing tools version 3.347 and throughput results are presented in GB/s (powers of 10). As explained next, the file size selected for this testing is 64 GiB per

file. All sequential reads and writes have an aggregate sample size of 64 GiB multiplied by the number of threads, up to a total 64 GiB * 128 threads or 8 TiB. In addition, to further thwart any client cache errors, the compute node responsible for writing the sequential file was not the same one used for reading the files. The block size for IOzone was set to 1 MiB to match the 1 MiB Lustre request size.

Each file written was large enough to minimize cache effects from OSS and clients, except for the 1 and 2 thread cases. However, the other techniques to prevent cache effects proved to avoid them. Files written were distributed evenly across the OSTs (Round Robin). This was to prevent uneven I/O load on any single SAS connection or OST in the same way that a user would expect to balance a workload.

Figure 12: Sequential Reads / Writes HSS4.5

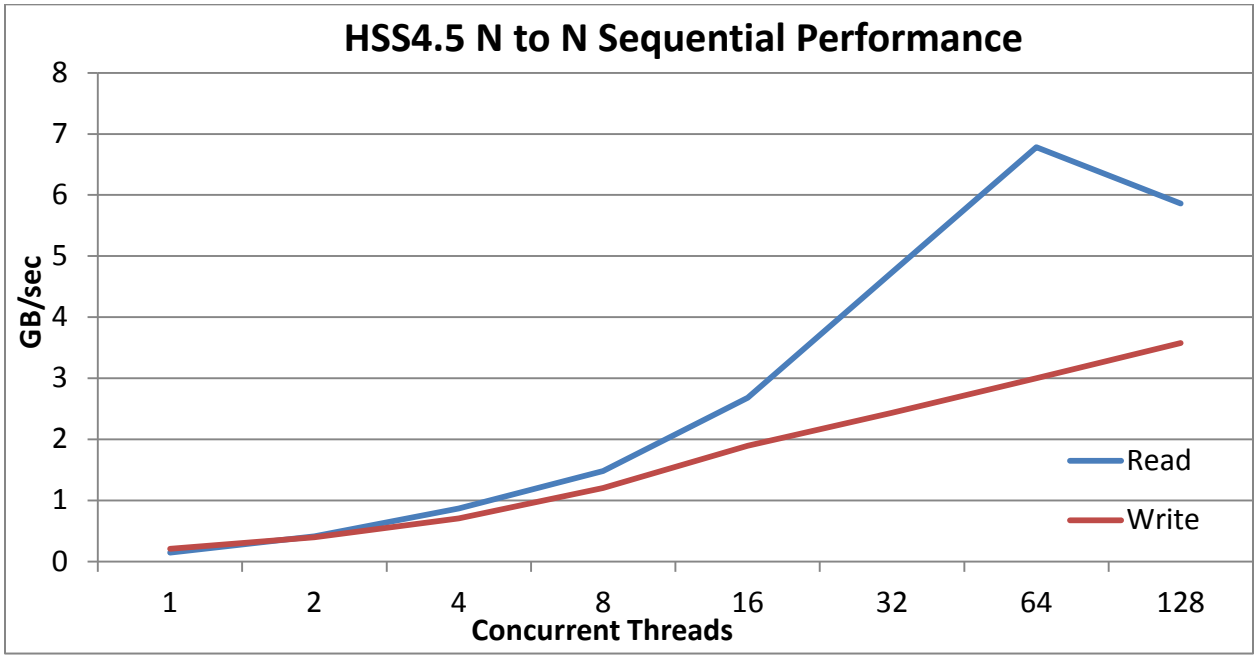


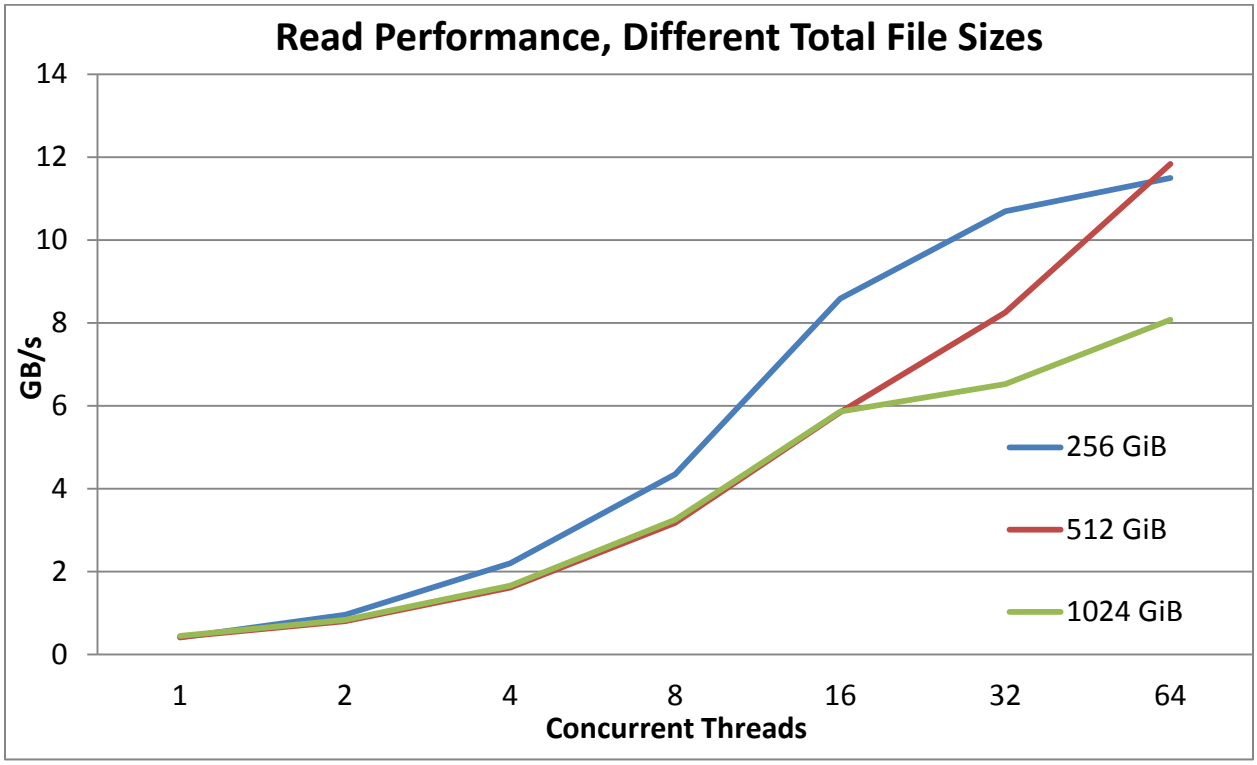
Figure 12 shows the sequential performance of the HSS4.5 XL (720 TB) test configuration. With the test bed used, write performance peaks at around 3.57 GB/sec at 128 concurrent threads but is likely to continue increasing with more threads since the plateau (saturation) point was not reached. Read performance peaks at 6.78 GB/sec for 64 simultaneous processes and continues to exceed 5GB/sec with 128 threads access individual OST's. The write and read performance rises steadily as we increase the number of process threads up to 64 for reads and 128 for writes. This is partially a result of increasing the number of OSTs that are utilized, as the number of threads is increased (up to the 24 OSTs in our system). By careful crafting of the IOzone hosts file, each added thread balances between compute nodes, SAS controllers, as well as keeping the number of files balanced on the OSTs, allowing a consistent increase in the number of files written while creating minimal locking contention between the files.

On the read operations, as the number of files per OST increases (in our case, 4 or more), the throughput starts to decline. This is most likely the result of issues related to the larger number of concurrent requests per OST. Positional delays and other overhead are expected as a result of the increased randomization of client requests on each OST. In order to maintain the higher throughput for

a greater number of files, increasing the number of OSTs is likely to help. A review of the storage array performance using the tools provided by the Dell PowerVault Modular Disk Storage Manager or using SMCli performance monitor was done to independently confirm the throughput values produced by the benchmarking tools.

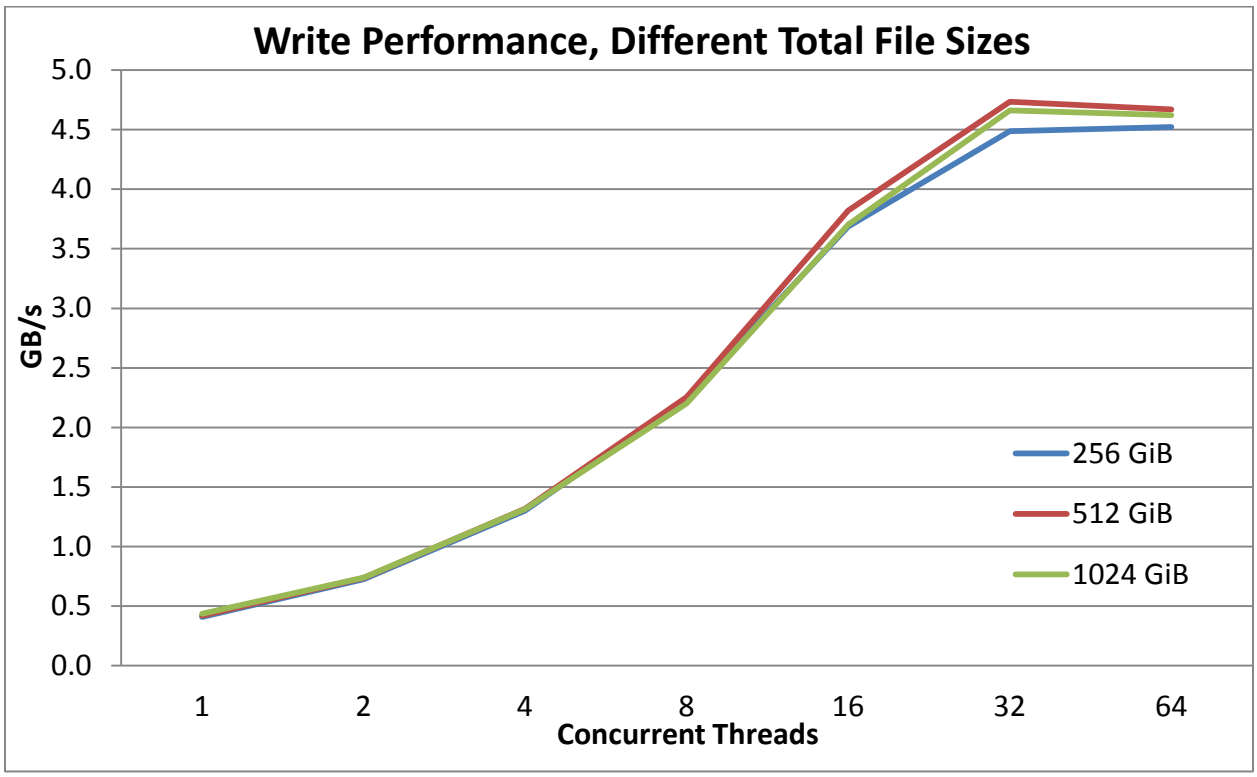
Figure 13 and Figure 14 illustrate the effects of caching on the HSS4.5 XL. The two graphs plot the results of N-to-N sequential write and read tests with no precautions taken to avoid server cache effects. Each line in the graph denotes the throughput measured with total file I/O of the size specified in the legend. For example, the curve titled 1024 GiB was the test case where total file I/O was 1024 GiB, i.e. Number-of-threads * File-size-per-thread was held constant at 1024 GiB. That is, for 4 threads, each thread accessed a 256 GiB file; for 64 threads, each accessed a 8 GiB file. In these two graphs, during testing each client was writing and reading the same file and the server cache was not dropped, but client cache was dropped between tests. Therefore the caching effect was mostly from the OSS servers.

Figure 13: HSS4.5 N to N Sequential Reads, Total File size distributed among all threads



As can be observed in Figure 13, HSS4.5 can very effectively use memory cache, providing a read throughput of almost 12 GB/s. This is almost double the value obtained with larger files for the test case presented in Figure 12 that were designed to avoid these caching effects.

Figure 14: HSS4.5 N to N Sequential Writes, Total File size distributed among threads used



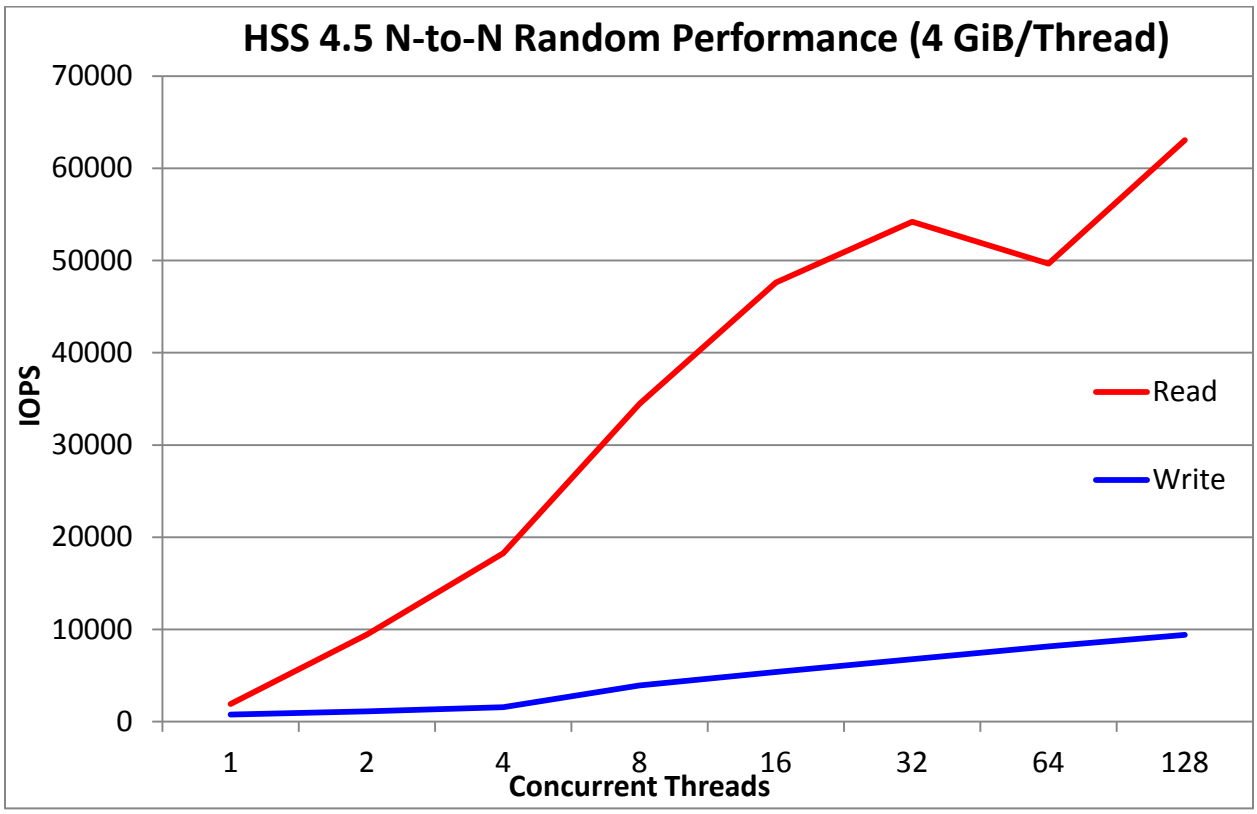
Similarly, Figure 14 shows that write throughput can benefit from cache effects and deliver over a throughput of over 4.6 GB/s.

4.2 Random Reads and Writes

The IOzone benchmark was used to gather random reads and writes metrics. In this case, a file size of 4 GiB was used per thread. The IOzone host file is arranged to distribute the work load evenly across the compute nodes. The storage is addressed as a single volume with an OST count of 24 and stripe size of 4 MiB. A 4 KiB request size is used because it aligns with this Lustre version's 4 KiB file system block size and is representative of small block accesses for a random workload. Performance is measured in I/O Operations per second (IOPS)

Figure 15 shows that the random writes peak at almost 10K IOPS with 128 threads. The IOPs of random reads increase rapidly from four to 16 threads and then continue to increase at a relatively steady rate. As the writes require a file lock per OST accessed, saturation is not unexpected. Reads take advantage of Lustre's ability to grant overlapping read extent locks for part or all of a file. Increasing the number of disks in the single OSS pair (for S, M or L configurations) or additional OSS pairs can increase the performance in terms of IOPS.

Figure 15: N-to-N Random reads and writes



4.3 IOR N-to-1 Reads and Writes

Performance review of the HSS4.5 with reads and writes to a single file was done with the IOR benchmarking tool. IOR accommodates MPI communications for parallel operations and has support for manipulating Lustre striping. IOR allows several different IO interfaces for working with the files; this testing used the POSIX interface to exclude more advanced features and associated overhead. This gives us an opportunity to review the file system and hardware performance independent of those additional enhancements.

IOR benchmark version 2.10.3 was used in this study. The MPI stack used for this study was Intel MPI version 4.1.0.030.

The configuration for the write test included a directory set with striping characteristics designed to stripe across all 24 OSTs with a stripe size of 4 MiB. Therefore, all threads write to a file that is evenly striped across all OSTs. The request size for Lustre is 1 MiB, but in this test, a transfer size of 4 MiB was used to match the stripe size used on the target file.

In order to reduce the cache effects from server and client memory, it was decided to use a file size that was twice the memory size of the OSSs and the clients' memory, according to the following formula:

> **File Size** = 2 * (2 OSSs*128 GiB memory per OSS + Number of physical clients * 24 GiB memory per client).

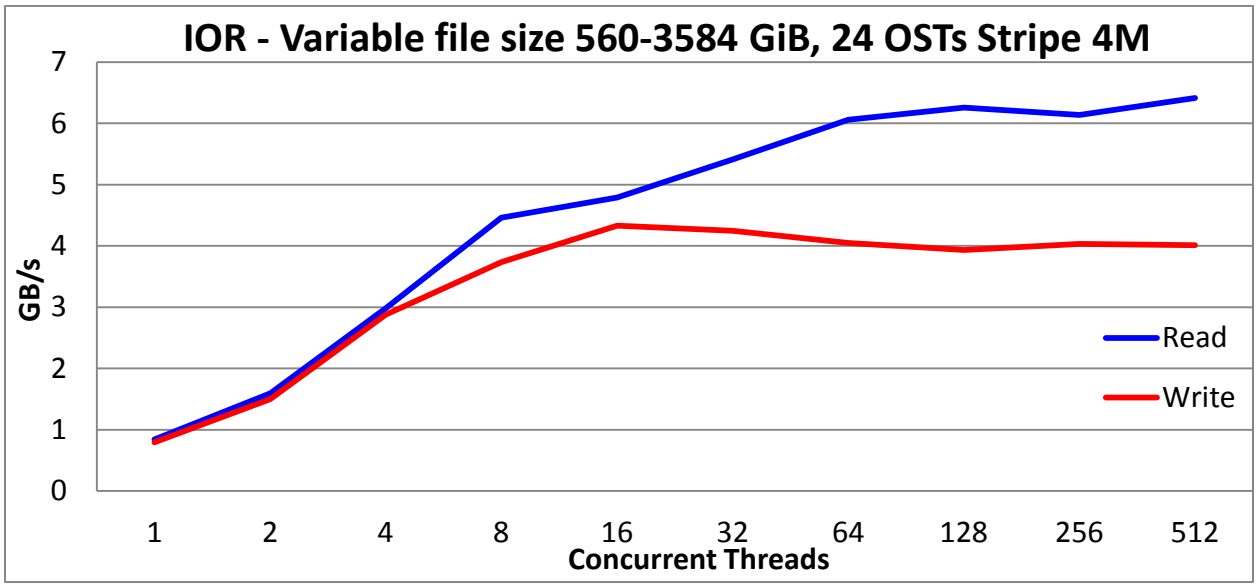
Table 3 shows the size of the data manipulated by each set of clients, the number of threads, and the size of the total shared file.

Table 3: IOR Shared File Size

Number of Threads	Number of Physical Clients	Data Written per Thread (GiB)	Shared File Size (GiB)
1	1	560	560
2	2	304	608
4	4	176	704
8	8	112	896
16	16	80	1280
32	32	64	2048
64	64	56	3584
128	64	28	3584
256	64	14	3584
512	64	7	3584

Figure 16 shows the IOR results, where reads have the advantage and peak at 6.4 GB/s at 512 threads, a value significantly near the sequential N-to-N performance. Write performance peaks within 15% of the sequential writes at 4.3 GB/s, but as the threads to OST ratio increases, the write performance settles around 4 GB/s, sustained. Reads are less affected and continues to slowly increase since the read locks for clients are allowed to overlap some or all of a given file.

Figure 16: N-to-1 IOR Read / Write



4.4 Metadata Testing

Metadata testing measures the time to complete certain file or directory operations that return attributes. *MDtest* is an MPI-coordinated benchmark that performs Create, Stat, and Remove operations on files or directories. This study used MDtest version 1.8.3. The MPI stack used for this study was Intel MPI version 4.1.0.030. The metric reported by *MDtest* is the time for completion in terms of operations per second (OP/sec). MDtest can be configured to compare metadata performance for directories and files. For this particular study, it was decide to analyze only metadata for file operations, leaving the analysis for directory metadata for a future study.

On a Lustre file system, OSTs are queried for object identifiers in order to allocate or locate extents associated with the metadata operations. This interaction requires the indirect involvement of OSTs in most metadata operations. In previous studies (refer to the HSS3 whitepaper in the References), it was found that using all OSTs stripped was more efficient for most metadata operations, especially for higher thread counts. For that reason, a preliminary study was conducted with 16, 32 and 128 clients for the following Lustre topologies (results not presented here), to select the most efficient configuration for the metadata performance tests for this version of the solution:

- 1 OST, 1 MiB Stripes (as a sanity check against previous studies).
- 24 OSTs, 1 MiB Stripes (to match the ideal Lustre transfer size).
- 24 OSTs, 4 MiB Stripes (the stripe size value used in previous tests in this and previous studies).

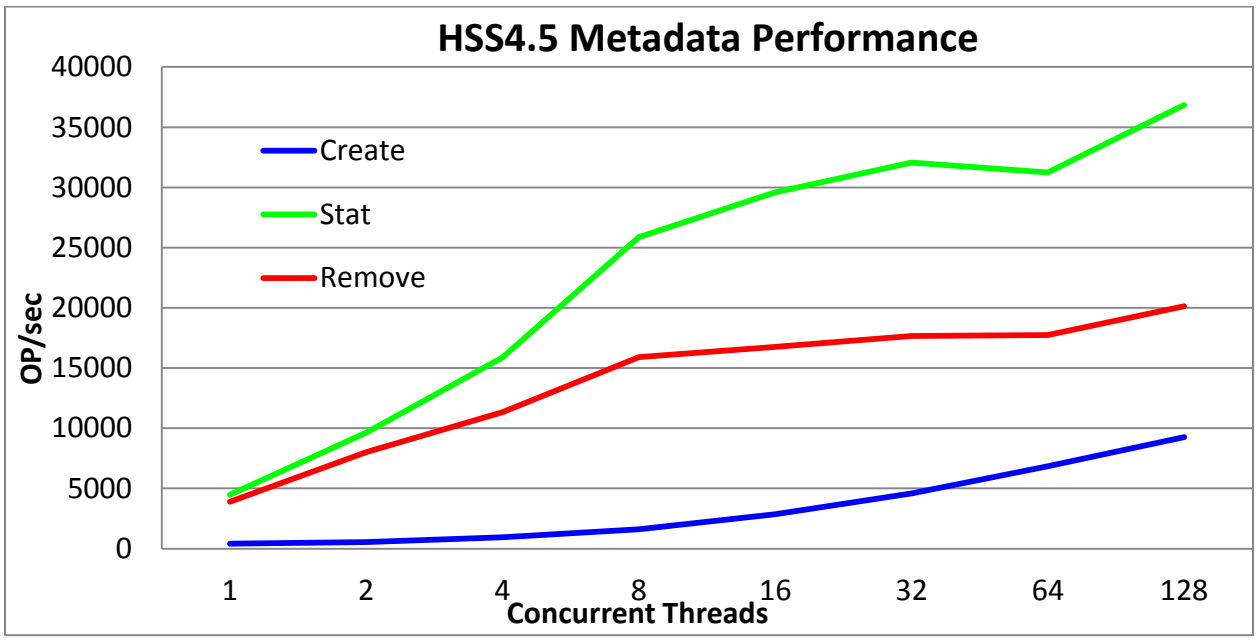
The most efficient configuration was found to be with 24 OSTs with 1 MiB stripes, therefore the results presented in this section are with such Lustre topology.

Also during the preliminary metadata testing, it was found that the number of files per directory significantly affects the results even while keeping constant the total number of files created. For example when testing with 64 threads creating 3125 files per directory in 5 directories per thread OR creating 625 files per directory in 25 directories per thread, both result in the creation of 1 million files, but the measured performance in IOPS is not the same. This is due to overhead of seeks performed on the OSTs when changing directories. In order to present coherent results, the number of files per directory was fixed at 3125 (5^6), which works nicely to divide the number of directories per thread from 1 to 64 (2^6) threads. Thus the total number of files was kept at 1 million whenever possible. The exceptions were the 1 and 2 thread cases where .5 million files were chosen to keep test time within reasonable limits and for the 128 thread case where 1.2M was the closest possible value. Table 4 provides the details of the values used.

Table 4: Parameters used on the MDtest

Number of threads (N)	Number of Files per directory	Number of Directories per thread	Total number of Files
1-2	3125	(160/N)-1	.5M
4-64	3125	(320/N)-1	1M
128	3125	2	1.2M

Figure 17: Metadata Operations



As explained before in Section 4, note that there are 64 physical compute nodes. Then, for thread counts from 1 to 64 each compute node runs a single thread, while 2 threads are run per compute node for the 128 threads test case.

Figure 17 shows the metadata results obtained from MDtest. Create is by far the most expensive metadata operation of the three, starting with less than .5K ops at 1 thread and scaling to over 9K ops with 128 concurrent threads. The performance of the MDT and OSTs are important to the Create operation; the fast multicore processors and larger amount of memory available enables the efficient creation of ~1 million files on HSS4.5. The solution was scaling without signs of a plateau up to 128 concurrent threads, indicating that either metadata operations were being cached, or the storage was not saturated yet. More analysis is needed to determine the limits of create metadata performance for the solution and to determine which component is the first bottleneck.

Stat is the lightest metadata operation of the three, providing almost 4K ops with 1 thread and scaling to almost 37K IOps with 128 concurrent threads without any signs of a plateau. The scaling of the stat operations is not as smooth as with create operations, perhaps due to the number of OSTs being 24 is not a power of 2 (as the number of threads used) creating some imbalances on the OSTs' load. Therefore, some sets of concurrent threads results in a better accumulation of metadata operations, filling the 1 MiB stripe more efficiently and providing the inflection points in the stat line in the graph.

Removal of the files is also limited by accesses to OSTs, like the create operation. However, the remove operation is less expensive than create, starting with a little less than 1K IOps at 1 thread and scaling to over 20K IOps with 128 concurrent threads.

Similarly to creates, more analysis is needed to determine the limits of stat and remove metadata performance for the solution and to determine which component is the first bottleneck for each case.

5. Conclusions

There is a well-known requirement for scalable, high-performance clustered file system solutions. The Dell | Terascala HSS4.5 addresses this need with a well-designed appliance based solution that is pre-configured, easy to manage, associated with quantified performance metrics and fully–supported. The solution includes the added benefit of proven Dell PowerEdge™ 12th generation servers and PowerVault™ products and Lustre® technology, the leading open source solution for a parallel file system. The Terascala Management Console (TMC) unifies the management of the multiple solution components and associated utilities into a single control and monitoring panel for ease of use.

The latest generation Dell | Terascala HPC Storage Solution continues to offer the same advantages as the previous generation solutions, but with greater processing power, memory speed and capacity, as well as enhanced reliability and higher density. The scale of the raw storage, from 360 TB to 720 TB per Object Storage Server Pair and up to 6.7 GB/s of read throughput in a packaged component, is consistent with the needs of the high performance computing environments. The HSS4.5 is also capable of scaling in throughput as easily as it scales in capacity.

The performance studies demonstrate a high throughput for both reads and writes for N-to-N as well as N-to-1 file type access. Results from MDtest show an elevated capacity for the metadata file operations. With the PCI-e 3.0 interface, IB FDR HCAs contributes to excel in both high IOP and high bandwidth applications.

The continued use of generally available, industry-standard benchmark tools like IOzone, IOR and MDtest provide an easy way to match current and expected growth with the performance outlined for the HSS4.5. The profiles reported from each of these tools provide sufficient information to align the configuration of the HSS4.5 with the requirements of many applications or group of applications.

In short, the Dell | Terascala HPC Storage Solution delivers all the benefits of a scale-out parallel file system-based storage for your high performance computing needs.

Appendix A: Benchmark Command Reference

This section describes the commands used to benchmark the Dell | Terascala HPC Storage Solution 3.

IOzone

IOzone Sequential Writes –

```
/share/iozone -i 0 -c -e -w -r 1024K -I -s 64G -t $Thread -+n -+m /share/clntcnf/lustlist.$Thread
```

IOzone Sequential Reads -

```
/share/iozone -i 1 -c -e -w -r 1024K -I -s 64G -t $Thread -+n -+m /share/clntcnf/lustlist.$Thread
```

IOzone IOPS Random Reads / Writes –

```
/share/iozone -i 2 -w -c -O -I -r 4K -s 4G -t $Thread -+n -+m /share/clntcnf/lustlist.$Thread
```

IOzone Command Line	Description
-i 0	Write test
-i 1	Read test
-i 2	Random IOPS test
-+n	No retest
-c	Includes close in the timing calculations
-e	Includes flush in the timing calculations
-r	Records size
-s	File size
-+m	Location of clients to run IOzone on when in clustered mode
-I	Use O_Direct
-w	Does not unlink (delete) temporary file
-+n	No retests selected
-O	Return results in OPS

The O_Direct command line parameter ("-I") allows us to bypass the cache *on the compute nodes* where the IOzone threads are running.

IOR

IOR Writes –

```
mpirun -np $Threads -rr --hostfile /share/ior_clients/iorlist.$Threads /share/IOR -a POSIX -v -i
1 -d 3 -e -k -o /mnt/lustre/perf_tst24-4M/mytestfile -w -s 1 -t 4m -b $SizePerThread
```

IOR Reads –

```
mpirun -np $Threads -rr --hostfile /share/ior_clients/iorlist.$Threads /share/IOR -a POSIX -v -i
1 -d 3 -e -k -o /mnt/lustre/perf_tst24-4M/mytestfile -r -s 1 -t 4m -b $SizePerThread
```

IOR Command Line Arguments	Description
-a S	api -- API for I/O [POSIX|MPIIO|HDF5|NCMPI]
-v	verbose -- output information (repeating flag increases level)
-I N	repetitions -- number of repetitions of test
-d N	interTestDelay -- delay between reps in seconds
-e	fsync -- perform fsync upon POSIX write close
-k	keepFile -- don't remove the test file(s) on program exit
-o S	testFile -- full name for test
-w	writeFile -- write file
-r	readFile -- read existing file
-s N	segmentCount -- number of segments
-t N	transferSize -- size of transfer in bytes (e.g.: 8, 4k, 2m, 1g)
-b N	blockSize -- contiguous bytes to write per task (e.g.: 8, 4k, 2m, 1g)

MDtest - Metadata

Create Files -

```
mpirun -np $Threads -rr --hostfile /share/mdt_clients/mdtlist.$Threads /share/mdtest/mdtest.intel
-v -d /mnt/lustre/perf_test24-1M -i $Reps -b $Dirs -z 1 -L -I $Files -y -u -t -F -C
```

Stat files -

```
mpirun -np $Threads -rr --hostfile /share/mdt_clients/mdtlist.$Threads /share/mdtest/mdtest.intel
-v -d /mnt/lustre/perf_test24-1M -i $Reps -b $Dirs -z 1 -L -I $Files -y -u -t -F -R -T
```

Remove files -

```
mpirun -np $Threads -rr --hostfile /share/mdt_clients/mdtlist.$Threads /share/mdtest/mdtest.intel
-v -d /mnt/lustre/perf_test24-1M -i $Reps -b $Dirs -z 1 -L -I $Files -y -u -t -F –r
```

MDtest Command Line Arguments	Description
-d	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
-b	branching factor of hierarchical directory structure
-z	depth of hierarchical directory structure
-L	files only at leaf level of tree
-I	number of items per directory in tree
-y	sync file after writing
-u	unique working directory for each task
-t	time unique working directory overhead
-F	perform test on files only (no directories)
-C	only create files/directories
-R	randomly stat files (optional argument for random seed)
-T	only stat files/directories
-r	only remove files or directories left behind by previous runs

Appendix B: Lustre Client RPMs installation

Dell High Performance Computing Clusters support Mellanox adapters which require the use of the Mellanox OFED software to fully leverage its capabilities. Hence a Lustre client that supports a kernel modified by the Mellanox OFED is required.

Such software is provided in the form of two RPMs specifically compiled for the RHEL6.2 kernel with the Mellanox OFED 1.5.3-3.0.0 installed. During this study, StackIQ Rocks version 6.0.1 was running on the compute nodes that were installed with the Lustre Client RPMs to allow Lustre access to the cluster. Bright Cluster Manager with the same kernel and OFED should be able to use the same procedure described here, and it will be tested and verified in a future effort.
The following procedure was performed on one compute node as an example, but it needs to be repeated for each node that requires Lustre access. Users can utilize StackIQ cluster management software based on Rocks+ or Bright Computing Cluster Manager commands to propagate commands to all the compute nodes.
Both RPMs must be installed by the root user on each client with the command:

```
[root@compute-0-0 share]# rpm –ivh lustre-client-1.8.8-wc1_2.6.32_220.el6.x86_64_gbc88c4c.x86_64
\ lustre-client-modules-1.8.8-wc1_2.6.32_220.el6.x86_64_gbc88c4c.x86_64
```

Next, create a file /etc/modprobe.d/lnet.conf and make sure the content is the following:

```
[root@compute-0-0 share]# cat /etc/modprobe.d/lnet.conf
options lnet networks=o2ib0(ib0)
```

Then, make sure the InfiniBand adapter is properly configured and the state is up (use `ifup ib0` if it is not up):

```
[root@compute-0-0 share]# ip addr show ib0
4: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc pfifo_fast state UP qlen 1024
    link/infiniband 80:00:00:48:fe:80:00:00:00:00:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
    inet 10.10.10.1/16 brd 10.10.255.255 scope global ib0
    inet6 fe80::XXXX:XXXX:XXXX:XXXX:XXXX/64 scope link
       valid_lft forever preferred_lft forever
```

At this point, we can make certain that Lustre and LNET kernel modules are loaded, that Lustre LNET client can be enabled and that we can ping any other machine in the cluster. Notice the use of the LNET ping to validate that the Lustre client is working and the use of a known good IP address with LNET access (in this case the address used by HSS4.5 to export Lustre: 10.10.255.40):

```
[root@compute-0-0 ~]# modprobe lnet
[root@compute-0-0 ~]# modprobe lustre
[root@compute-0-0 ~]# lctl network up
LNET configured
[root@compute-0-0 share]# lctl ping 10.10.255.40@o2ib0
12345-0@lo
12345-10.10.255.40@o2ib
```

As a counter example of a failed ping operation:

```
[root@compute-0-0 share]# lctl ping 10.10.255.240@o2ib0
failed to ping 10.10.255.240@o2ib: Input/output error
```

The next step is to tune the client for better performance; the actual tuning steps can vary, so the following items are provided as an example only. Also, keep in mind that this tuning is dynamic and does not persist with client reboots, meaning it must be done again once the Lustre kernel module is loaded.

```
[root@compute-0-0 share]# cat set_params
for i in /proc/fs/lustre/osc/*/checksums; do echo 0 > $i; done
for i in /proc/fs/lustre/osc/*/max_dirty_mb; do echo 512 > $i; done
for i in /proc/fs/lustre/osc/*/max_rpc*; do echo 48 > $i; done
[root@compute-0-0 share]# bash set_params
```

At this point the client is ready to be used.

References

Dell | Terascala HPC Storage Solution Brief
http://i.dell.com/sites/content/business/solutions/hpcc/en/Documents/Dell-terascala-hpcstorage-solution-brief.pdf

Dell | Terascala HPC Storage Solution 2 Whitepaper
http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/Dell-terascala-dt-hss2.pdf

Dell | Terascala HPC Storage Solution 3 Whitepaper
http://i.dell.com/sites/doccontent/business/solutions/hpcc/en/Documents/dell-terascala-dt-hss3.pdf

Dell PowerVault MD3220
http://www.dell.com/Learn/us/en/04/content/powervault-md3200-whitepapers

Dell PowerVault MD3260 & MD3060e
http://www.dell.com/us/business/p/powervault-md32x0-series/pd?~ck=anav

Lustre Home Pages
http://www.whamcloud.com/lustre/
http://wiki.lustre.org/index.php/Main_Page

Transitioning to 6 Gb/s SAS
http://www.dell.com/downloads/global/products/pvaul/en/6gb-sas-transition.pdf

Dell HPC Solutions Home Page
http://www.dell.com/hpc

Dell HPC Wiki
http://www.HPCatDell.com

Terascala Home Page
http://www.terascala.com

Mellanox Technologies Home Page
http://www.mellanox.com

Bright Computing Cluster Manager
http://www.brightcomputing.com/Linux-Cluster-Management-GUI.php

StackIQ Enterprise HPC
http://www.stackiq.com/products/stackiq-enterprise-hpc/