Dell HPC NFS Storage Solution – High Availability (NSS-HA) Configuration with Dell PowerVault MD3260/MD3060e Storage Arrays

A Dell Technical White Paper

Xin Chen, Garima Kochhar and Mario Gallegos

Dell HPC Engineering

May 2013| Version 2.0

May 2013 | Rev 2.0

Contents

Executive summary (updated May 2013) 5
1. Introduction 6
2. Overview of NSS-HA solutions 6
2.1. A brief introduction to NSS-HA solutions 6
2.2. NSS-HA offerings from Dell 7
3. Dell PowerVault MD3260 and MD3060e storage arrays 9
4. Evaluation 11
4.1. Method 11
4.2. Test bed 11
4.3. HA functionality 14
4.3.1. Potential failures and fault tolerant mechanisms in NSS-HA 15
4.3.2. HA tests for NSS-HA 15
5. NSS4.5-HA I/O Performance (updated May 2013) 17
5.1. IPoIB sequential writes and reads 18
5.2. IPoIB random writes and reads 18
5.3. IPoIB metadata operations 19
6. Conclusion 21
7. References 21
Appendix A: Benchmarks and test tools 22
A.1. IOzone 22
A.2. mdtest 24
A.3. Checkstream 25
A.4. The dd Linux utility 26

Tables

Table 1. NSS-HA Solutions[(1), (2), (3)] 8
Table 2. Storage components in NSS-HA 10
Table 3. NSS4.5-HA hardware configuration 12
Table 4. NSS4.5-HA software versions 13
Table 5. NSS4.5-HA firmware and driver versions 13
Table 6. NSS4.5-HA client configuration 14
Table 7. NSS-HA mechanisms to handle failures 15
Table 8. Appendix A — IOzone command line arguments 22

Figures

Figure 1. The infrastructure of the NSS-HA solution 7
Figure 2. PowerVault MD3260/MD3060e dense-enclosure storage array 9
Figure 3. NSS4.5-HA test bed 12
Figure 4. IPoIB large sequential write and read performance 18
Figure 5. IPoIB random write and read performance 19
Figure 6. IPoIB file create performance 20
Figure 7. IPoIB file stat performance 20
Figure 8. IPoIB file remove performance 21

Executive summary (updated May 2013)

This solution guide describes the Dell NFS Storage Solution - High Availability configurations (NSS-HA) with Dell PowerVault MD3260 and MD3060e storage arrays. The PowerVault MD3260 and MD3060e are high-density storage enclosures that are able to provide 60 3.5" drives in 4U of rack space. This solution guide presents a comparison between all available NSS-HA offerings so far, and provides performance-tuning best practices and performance results for a configuration with a storage system capacity of 360TB.

The NSS-HA solution described is designed to enhance the availability of storage service to the HPC cluster by using a pair of Dell PowerEdge servers and PowerVault storage arrays along with Red Hat HA software stack. As in previous versions of Dell NSS-HA solution guides, the goal of this solution guide is to improve storage service availability and maintain data integrity in the presence of possible failures or faults and to maximize performance in a failure-free scenario.

Version 2.0 of the white paper, dated May 2013, includes updated performance results.

1. Introduction

This solution guide provides information on the latest Dell NFS Storage Solution - High Availability configurations (NSS-HA) with Dell PowerVault MD3260 and MD3060e storage arrays. The solution uses Dell PowerEdge servers and PowerVault storage arrays along with Red Hat high Availability software stack to provide an easy to manage, reliable, and cost effective storage solution for HPC clusters. It leverages the latest Dell PowerVault Storage arrays (MD3260 and MD3060e) to offer denser storage solutions than previous NSS-HA solutions. This version of the solution is NSS4.5-HA.

The design principle for this release remains the same as previous Dell NSS-HA solutions. The major changes between the current and previous version of NSS-HA solution are the change from Dell PowerVault MD3200 and MD1200 storage arrays to the latest PowerVault MD3260 and MD3060e storage arrays, and the change from the RHEL 6.1 operating system to RHEL 6.3. For complete details, review this document along with the previous NSS-HA white papers [(1)] [(2)] [(3)].

The following sections describe the technical details, evaluation method, and the expected performance of the solution.

2. Overview of NSS-HA solutions

Along with the current version, four versions of NSS-HA solutions have been released since 2011. This section provides a brief description of the NSS-HA solution, and lists the available Dell NSS-HA offerings.

2.1. A brief introduction to NSS-HA solutions

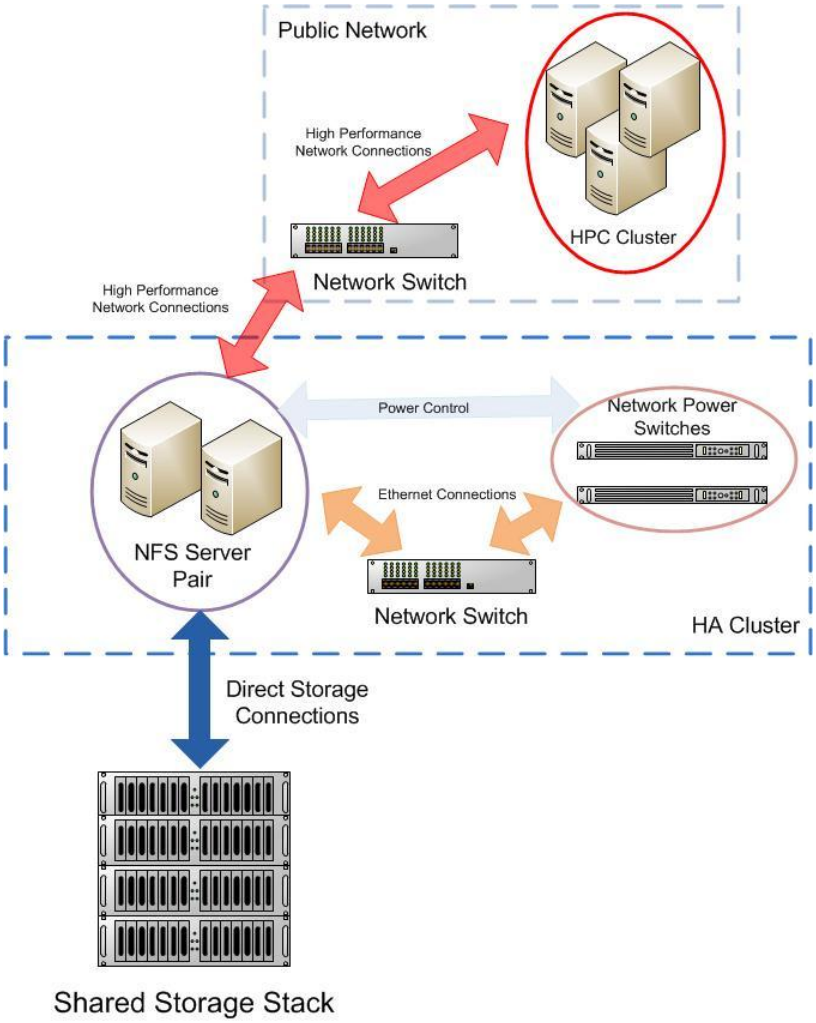
The design of the NSS-HA solution for each version is identical. In general, the core of the solution is a high availability (HA) cluster[(4)], which provides a highly reliable and available storage service to HPC compute clusters via a high performance network connection such as InfiniBand (IB) or 10 Gigabit Ethernet (10GbE).

The HA cluster consists of a pair of Dell PowerEdge servers and a network switch. The two PowerEdge servers have shared access to disk-based Dell PowerVault storage in a variety of capacities, and both are directly connected to an HPC cluster via IB or 10GbE. The two servers are equipped with two fence devices: iDRAC7 Enterprise and an APC Power Distribution Unit (PDU). When failures such as storage disconnection, network disconnection, system crash, etc., occurs on one server, the HA cluster will failover the storage service from the failed server to the healthy server with the assistance of the two fence devices, which also ensure that the failed server does not return to life without the administrator's knowledge or control.

The disk-based storage array is formatted as a Red Hat Scalable file system (XFS) and exported to the HPC cluster via NFS service of the HA cluster. Large capacity file systems (greater than 100TB) have been supported since the 2nd version of NSS-HA solution [(2)].

Figure 1 depicts the general infrastructure of the NSS-HA solution. For detailed information, refer to the previous NSS-HA white papers [(1)] [(2)] [(3)].

Figure 1. The infrastructure of the NSS-HA solution



Note: The iDRAC 7 enterprise is not shown in the figure, and it is installed on each NFS server for Dell NSS-HA solutions. The term of *Network Power Switches* refers to APC PDU (Power Distribution Unit) in Dell NSS-HA solutions.

2.2. NSS-HA offerings from Dell

Table 1 lists all of the Dell NSS-HA solutions with standard configurations. In addition to the standard configurations, a special NSS-HA configuration, *XL*, is available for all NSS-HA versions.

The NSS-HA-XL configuration provides two independent storage system services concurrently, along with the high availability functionality, but instead of one active-passive failover pair, the solution is designed as two active-passive failover pairs to host the two storage services. Each server is the active host for one of the storage arrays and passive for the other storage array. The NSS-HA-XL configuration uses the same hardware and software as required in the standard configurations, except it requires twice the number of PowerVault storage arrays than the NSS-HA standard configuration.

For more information about this special configuration, refer to the blog post titled *Dell NFS Storage Solution with High Availability - XL configuration* [5].

Table 1. NSS-HA Solutions[1], [2], [3]

	NSS2-HA Release (April 2011) [1]	NSS3-HA Release (February 2012) [2] "Large capacity configuration"	NSS4-HA Release (July 2012) [3] "PowerEdge 12G based solution"	NSS4.5-HA Release (October 2012) "PowerVault MD3260 based solution"
Storage Capacity	48TB and 96TB of raw storage space.	144TB and 288TB of raw storage space.		180TB to 360TB of raw storage space.
Network Connectivity	QDR InfiniBand or 10GbE connectivity.		FDR InfiniBand or 10GbE Connectivity.	
NFS servers	Dell PowerEdge R710 servers.		Dell PowerEdge R620 servers.	
Software	Red Hat Enterprise Linux 5.5 Red Hat Scalable File system (XFS) v2.10.2-7	Red Hat Enterprise Linux 6.1 Red Hat Scalable File system (XFS) v3.1.1-4		Red Hat Enterprise Linux 6.3 Red Hat Scalable File system (XFS) v3.1.1-7
Storage Devices	Dell PowerVault MD3200 and MD1200s. 2TB NL SAS drives.	Dell PowerVault MD3200 and MD1200s. 3TB NL SAS drives.		Dell PowerVault MD3260 and MD3060e. 3TB NL SAS drives.
Local Switch and Power Distribution Units (PDUs)	PowerConnect 5524. Two APC switched PDUs to manage high availability. Refer to Fence device and Agent Information for Red Hat Enterprise Linux for supported models of APC PDUs.			
Support and Services	3 years of Dell PRO Support for IT and Mission Critical 4HR 7x24 onsite pack. Dell deployment services are available to speed up installation, optimize performance and integrate NSS-HA solution with customer's HPC Cluster environment.			

Notes:

- There is no NSS1-HA release available on the market.
- Contact your Dell Sales Representative to discuss which solution would be suited for your environment. You can order any of the pre-configured solutions or a customized solution designed to address your needs. Based on the customization selected, some of the best practices discussed in this document may not apply.

3. Dell PowerVault MD3260 and MD3060e storage arrays

As compared to previous versions of the NSS-HA solution, a major change in the current version is the introduction of the Dell PowerVault MD3260 and MD3060e storage arrays. These arrays replace the PowerVault MD3200 and MD1200 storage arrays used in previous NSS-HA solutions.

With the introduction of the 4U, 60-drive PowerVault MD3260 and MD3060e dense-enclosure storage arrays, storage capacity can be increased significantly while reducing the NFS storage solution's footprint. The PowerVault MD3260 offers similar software, firmware, and management features as the PowerVault MD3200 series 2U arrays. However, for the same storage capacity, the PowerVault MD3260 and MD3060e storage arrays reduce the rack space footprint by 2.5 times (60 vs. 24).

The PowerVault MD3260 is an RBOD with two RAID controller modules in the storage array. The PowerVault MD3060e is also a 4U, 60 drive dense enclosure expansion box, but it is used as a JBOD (similar to the PowerVault MD1200) to extend the capacity of the PowerVault MD3260.

Figure 2 shows the front of a PowerVault MD3260/MD3060e dense-enclosure storage array. It has five drawers, and each drawer holds twelve 3.5 HDDs, for a total of 60 drives. With 3TB NL-SAS drives, each high-density storage array can provide up to 180 TB of raw capacity.

Figure 2. PowerVault MD3260/MD3060e dense-enclosure storage array

Due to the different physical characteristics of the PowerVault MD3260/MD3060e, some configuration parameters must be changed from previous NSS-HA versions to tailor the design to these new storage arrays. These parameters include the number of disks in a RAID virtual disk and the logical volume layout and stripe size. Details of these new parameter settings and the associated high availability functionality and I/O performance tests performed in the Dell HPC lab are provided below.

- Virtual disk configuration:
    - A RAID-6 8+2 layout is adopted for the current version of the NSS-HA solution, while RAID-6 10+2 is used in the previous versions. The major reason for this change is to enhance data availability and reliability. In a PowerVault MD3260/MD3060e storage array, a virtual disk consists of ten disks across all five drawers, two disks per drawer. Thus, if a single drawer fails for any reason, the virtual disk can still work, because a RAID-6 virtual disk is able to tolerate two concurrent disk failures. In the previous versions, a single PowerVault MD3200/1200 storage array provided 12 disks. A RAID-6 10+2 design was the best way to utilize all of the disks in that configuration.
- Logical volume configuration:

- With the RAID-6 8+2 choice for the virtual disk configuration, six virtual disks can now be constructed for each PowerVault MD3260 or MD3060e storage array (60 / (8+2) = 6). An 180TB configuration consists of a single PowerVault MD3260 and the logical volume will contain six virtual disks. A 360TB configuration which consists of a PowerVault MD3260 and a PowerVault MD3060e will have a total of twelve virtual disks. The logical volume uses six virtual disks from the MD3260 and it is extended with the other six virtual disks from the MD3060e. As demonstrated by the results of the I/O performance tests, a stripe element size of 512 KiB can significantly enhance the sequential I/O performance in this configuration over the stripe element size of 1024 KiB used in the previous versions of NSS-HA solutions.

Section 4 and 5 provide detailed information about the high availability functionality and I/O performance tests.

Table 2 summarizes the differences in storage systems for the previous and current version of NSS-HA solutions.

Table 2. Storage components in NSS-HA

	NSS4-HA release (July 2012) "PowerEdge 12th Generation based solution"	NSS4.5-HA (September 2012) "PowerVault MD3260/MD3060e based solution"
Storage configurations	144TB: 1 PowerVault MD3200 + 3 PowerVault MD1200s. 288TB: 1 PowerVault MD3200 + 7 PowerVault MD1200s.	180TB: 1 PowerVault MD3260. 360TB: 1 PowerVault MD3260 + 1 PowerVault MD3060e.
Disks in storage array	3TB NL SAS.	
Virtual disk configuration	RAID-6 10+2, a virtual disk is created using all 12 disks in a storage array. Segment size 512KiB. Write cache mirroring enabled. Read cache enabled. Dynamic cache read prefetch enabled.	RAID-6 8+2, a virtual disk is created across all five drawers in a storage array, two disks per drawer. Segment size 512KiB. Write cache mirroring enabled. Read cache enabled. Dynamic cache read prefetch enabled.
Storage enclosure cabling	Asymmetric ring cabling scheme.	Chain cabling scheme.
Logical volume configuration	Stripe element size: 1024KiB. Number of stripes: 4. Number of virtual disks per logical volume: 4 for 144TB configuration 8 for 288TB configuration.	Stripe element size: 512KiB. Number of stripes: 6. Number of virtual disks per logical volume: 6 for 180TB configuration 12 for 360TB configuration.

4. Evaluation

The architecture proposed in this white paper was evaluated in the Dell HPC lab. This section describes the test methodology and the test bed used for verification. It also contains details on the functionality tests. Performance tests and results follow in Section 5.

4.1. Method

The NFS Storage Solution described in this solution guide was tested for HA functionality and performance. A 360TB NSS4.5-HA configuration was used to test the HA functionality of the solution. Different types of failures were introduced and the fault tolerance and robustness of the solution was verified. Section 4.3 describes these HA functionality tests and their results. HA functionality testing was similar to the work done in the previous versions of the solution [(3)].

4.2. Test bed

The test bed used to evaluate the NSS4.5-HA functionality and performance is shown in Figure 3.

- A 64 node HPC compute cluster was used to provide I/O traffic for the test bed.
- A pair of Dell PowerEdge R620 servers were configured as an active-passive HA pair and function as a NFS server for the HPC compute cluster (also called the clients).
- Both NFS servers were connected to a shared Dell PowerVault MD3260 storage enclosure extended with one Dell PowerVault MD3060e storage enclosure (Figure 3 shows a 360TB solution with two PowerVault MD storage arrays) at the backend. The user data resided on an XFS file system created on this storage. The XFS file system was exported using NFS to the clients.
- The NFS servers were connected to the clients using the public network. This network was either InfiniBand or 10GbE Ethernet.
- For the HA functionality of the NFS servers, a private Gigabit Ethernet network was configured to monitor server health and heartbeat, and to provide a route for the fencing operations using a PowerConnect 5524 Gigabit Ethernet switch.
- Power to the NFS servers was driven by two APC switched PDUs on two separate power buses.

Complete configuration details are provided in Table 3, Table 4, Table 5, and Table 6.

Figure 3. NSS4.5-HA test bed

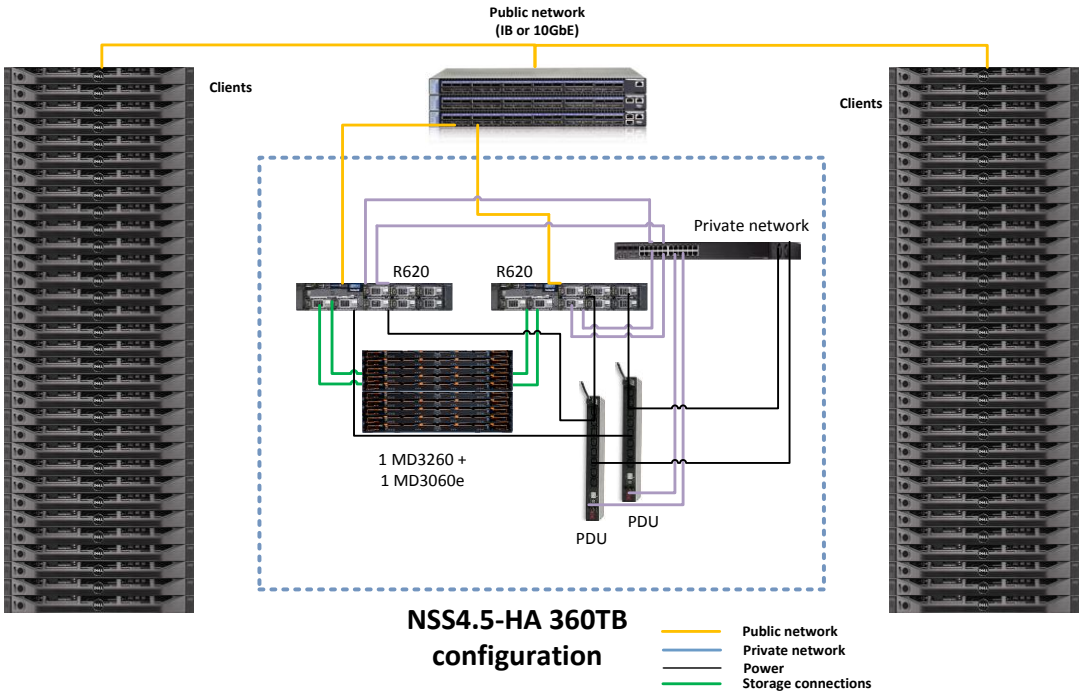


Table 3. NSS4.5-HA hardware configuration

Server configuration	
NFS server model	Two Dell PowerEdge R620s.
Processor	Dual Intel Xeon E5-2680 @ 2.70 GHz.
Memory	8 x 8GiB 1600 MHz RDIMMs. (The test bed used 64GiB; the recommendation for production clusters is to use 128GiB).
Local disks and RAID controller	PERC H710P with five 300GB 15K SAS hard drives. Two drives are configured in RAID-1 for the OS, two drives are configured in RAID-0 for swap space, and the fifth drive is a hot spare for RAID-1 disk group.
Optional InfiniBand HCA (slot 2)	Mellanox ConnectX-3 FDR PCI-E card.
1gbE Ethernet card (Daughter card slot)	Broadcom Corporation NetXtreme BCM5720 Gigabit Ethernet network daughter card.
External storage controller (slot 3 and slot 1)	Two 6Gbps SAS HBAs.
Systems Management	iDRAC7 Enterprise.
Power Supply	Dual Power Supply Units.

Storage configuration	
Storage Enclosure	One Dell PowerVault MD3260 enclosure and one MD3060e enclosure for the 360TB solution.
RAID controllers	Duplex RAID controllers in the Dell MD3260.
Hard Disk Drives	60 - 3TB 7200 rpm NL SAS drives per array.
Other components	
Private Gigabit Ethernet switch	Dell PowerConnect 5524.
Power Distribution Unit	Two APC switched Rack PDUs, model AP7921.

Table 4. NSS4.5-HA software versions

Software	
Operating system	Red Hat Enterprise Linux (RHEL) 6.3 x86_64
Kernel version	2.6.32-279.el6.x86_64
Cluster Suite	Red Hat Cluster Suite from RHEL 6.3
File system	Red Hat Scalable File System (XFS) 3.1.1-7
Systems management tool	Dell OpenManage Server Administrator 7.1.0
Storage management package	Consolidated Resource DVD (RDVD): 4.1.0.88

Table 5. NSS4.5-HA firmware and driver versions

Firmware and Drivers	
Dell PowerEdge R620 BIOS	1.1.2
Dell PowerEdge R620 iDRAC7 Enterprise	1.20.20
InfiniBand HCA firmware	2.10.700
InfiniBand driver	Mellanox OFED 1.5.3-3.1.0
10GbE Ethernet driver	ixgbe 3.6.7-k

Firmware and Drivers	
PERC H710P firmware	21.0.2-0001
PERC H710P driver	megaraid_sas 00.00.06.14-rh1
6Gbps SAS firmware	07.03.05.00
6Gbps SAS driver	mpt2sas 12.101.00.00

Table 6. NSS4.5-HA client configuration

Client / HPC Compute Cluster	
Clients	64 PowerEdge R410 compute nodes with Red Hat Enterprise Linux 6.2 x86-64.
InfiniBand	Mellanox ConnectX-2 QDR HCA. Mellanox OFED 1.5.3-3.0.0.
InfiniBand fabric	All clients connected to a single large port count InfiniBand QDR switch (Mellanox IS5100). Both PowerEdge R620 NFS servers also connected to an InfiniBand FDR switch (Mellanox SX6036), and the switch has three links each to the QDR switch (IS5100).
Ethernet	Onboard 1GbE Broadcom 5716 network adapter. bnx2 driver v2.1.6.
Ethernet fabric	Two sets of 32 compute nodes connected to two Dell PowerConnect 7048 Gigabit Ethernet switches. Both Dell PowerConnect 7048 switches have four links each to a 10 GbE Dell PowerConnect 8024F switch. Both PowerEdge R620 NSS-HA servers connected directly to the Dell PowerConnect 8024F switch. Flow control was disabled on the Dell PowerConnect 8024F switch and two Dell PowerConnect 7048 switches.

4.3. HA functionality

The HA functionality of the solution was tested by simulating several component failures. The design of the tests and the test results are similar to previous versions of the solution since the broad architecture of the solution has not changed in this release. This section reviews the failures and fault

tolerant mechanisms in NSS-HA solutions, then presents the HA functionality tests with regards to different potential failures and faults.

4.3.1. Potential failures and fault tolerant mechanisms in NSS-HA

There are many different types of failures and faults that can impact the functionality of NSS-HA. Table 7 lists the potential failures that are tolerated in NSS-HA solutions.

Note: The analysis below assumes that the HA cluster service is running on the *active* server; the *passive* server is the other component of the cluster.

Table 7. NSS-HA mechanisms to handle failures

Failure type	Mechanism to handle failure
Single local disk failure on a server	Operating system installed on a two-disk RAID 1 device with one hot spare. Single disk failure is unlikely to bring down server.
Single server failure	Monitored by the cluster service. Service fails over to passive server.
Power supply or power bus failure	Dual power supplies in each server. Each power supply connected to a separate power bus. Server continues functioning with a single power supply.
Fence device failure	iDRAC7 Enterprise used as primary fence device. Switched PDUs used as secondary fence devices.
SAS cable/port failure	Two SAS cards in each NFS server. Each card has a SAS cable to the shared storage. A single SAS card/cable failure will not impact data availability.
Dual SAS cable/card failure	Monitored by the cluster service. If all data paths to the shared storage are lost, service fails over to the passive server.
InfiniBand / 10GbE link failure	Monitored by the cluster service. Service fails over to passive server.
Private switch failure	Cluster service continues on the active server. If there is an additional component failure, service is stopped and system administrator intervention required.
Heartbeat network interface failure	Monitored by the cluster service. Service fails over to passive server.
RAID controller failure on Dell PowerVault MD3260 storage array	Dual controllers in the Dell PowerVault MD3260. The second controller handles all data requests. Performance may be degraded, but functionality is not impacted.

4.3.2. HA tests for NSS-HA

Functionality was verified for an NFSv3-based solution. The following failures were simulated on the cluster with the consideration of the failures and faults listed Table 7.

- Server failure.
- Heartbeat link failure.
- Public link failure.
- Private switch failure.
- Fence device failure.
- Single SAS link failure.
- Multiple SAS link failures.

The NSS-HA behaviors are outlined below in response to these failures.

- Server failure — simulated by introducing a kernel panic.
  When the active server fails, the heartbeat between the two servers is interrupted. The passive server waits for a defined period of time and then attempts to fence the active server. Once fencing is successful, the passive server takes ownership of the cluster service. Clients cannot access the data until the failover process is completed.

- Heartbeat link failure — simulated by disconnecting the private network link on the active server.
  When the heartbeat link is removed from the active server, both servers detect the missing heartbeat and attempt to fence each other. The active server is unable to fence the passive server since the missing link prevents it from communicating over the private network. The passive server successfully fences the active server and takes ownership of the HA service.

- Public link failure — simulated by disconnecting the InfiniBand or 10 Gigabit Ethernet link on the active server.
  The HA service is configured to monitor this link. When the public network link is disconnected on the active server, the cluster service stops on the active server and is relocated to the passive server.

- Private switch failure — simulated by powering off the private network switch.
  When the private switch fails, both servers detect the missing heartbeat from the other server and attempt to fence each other. Fencing is unsuccessful because the network is unavailable and the HA service continues to run on the active server.

- Fence device failure — simulated by disconnecting the iDRAC7 Enterprise cable from a server.
  If the iDRAC on a server fails, the server is fenced using the network PDUs, which are defined as secondary fence devices in the cluster configuration files.

For the above cases, it was observed that the HA service failover takes in the range of 30 to 60 seconds. In a healthy cluster, any failure event should be noted by the Red Hat cluster management daemon and acted upon within minutes. Note that this is the failover time on the NFS servers; the impact to the clients could be longer.

- Single SAS link failure — simulated by disconnecting one SAS link between the Dell PowerEdge R620 server and the Dell PowerVault MD3260 storage.
  In the case where only one SAS link fails, the cluster service is not interrupted. Because there are multiple paths from the server to the storage, a single SAS link failure does not break the data path from the clients to the storage and does not trigger a cluster service failover.

- Multiple SAS link failures — simulated by disconnecting all SAS links between one Dell PowerEdge R620 server and the Dell PowerVault MD3260 storage.
  When all SAS links on the active server fail, the multipath daemon on the active server retries the path to the storage based on the parameters configured in the `multipath.conf` file. This is set to timeout after 150 seconds by default. After this process times out, the HA service will attempt to failover to the passive server.

  If the cluster service is unable to cleanly stop the LVM and the file system because of the broken paths, a watchdog script reboots the active server after five minutes. At this point, the passive server fences the active server, restarts the HA service, and provides a data path again to the clients. This failover can therefore take anywhere in the range of three to eight minutes.

Impact to clients

Clients mount the NFS file system exported by the server using the HA service IP. This IP is associated with either an IPoIB or a 10 Gigabit Ethernet network interface on the NFS server. To measure any impact on the client, the `dd` utility and the `IOzone` benchmark were used to read and write large files between the clients and the file system. Component failures were introduced on the server while the clients were actively reading and writing data from/to the file system.

In all scenarios, the client processes completed the read and write operations successfully. As expected, the client processes take longer to complete if the process was actively accessing data during a failover event. During the failover period, when the data share is temporarily unavailable, the client processes were in an uninterruptible sleep state.

Depending on the characteristics of the client processes, they can be expected to either abort or sleep while the NFS share is temporarily unavailable during the failover process. Any data that has already been written to the file system will be available after the failover is completed.

For read and write operations during the failover case, data correctness was successfully verified using the `checkstream` utility.

5. NSS4.5-HA I/O Performance (updated May 2013)

This section presents the results of the I/O performance tests for the current NSS-HA solution. All performance tests were conducted in a failure-free scenario to measure the maximum capability of the solution. The tests focused on three types of I/O patterns: large sequential reads and writes, small random reads and writes, and three metadata operations (file create, stat, and remove).

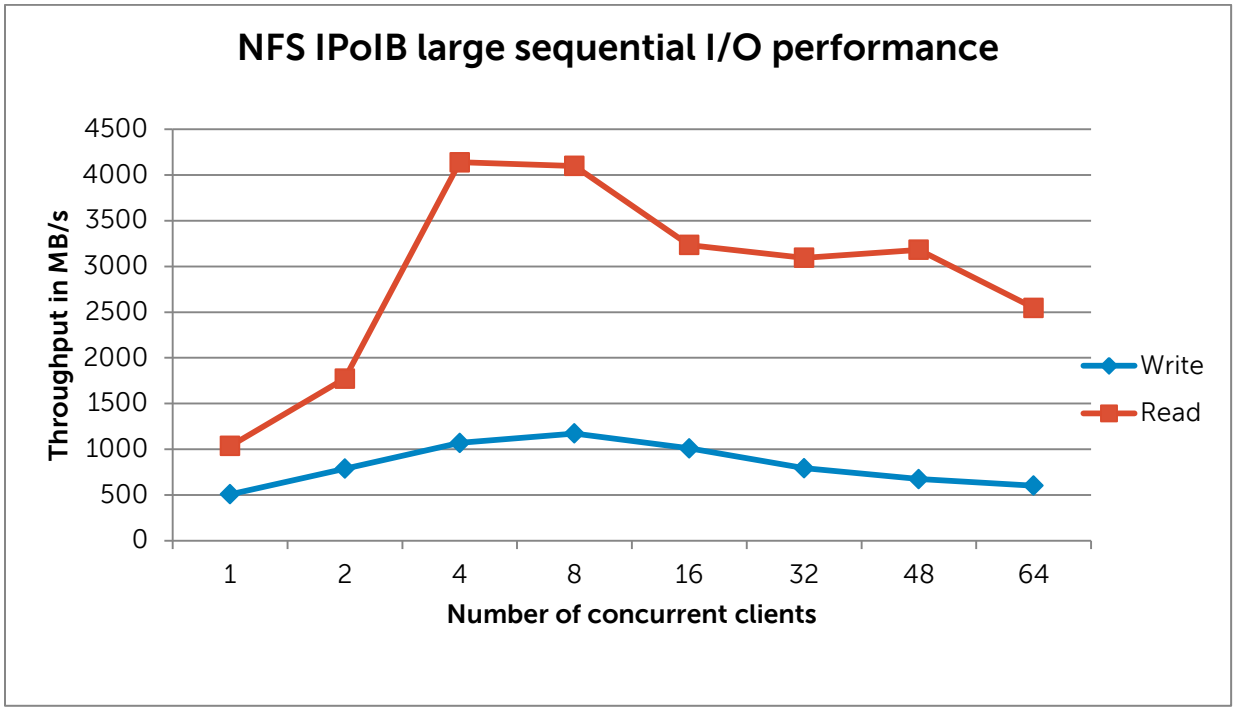
A 360TB configuration was benchmarked with IPoIB network connectivity. The 64-node compute cluster described in section 4.2 was used to generate workload for the benchmarking tests. Each test was run over a range of clients to test the scalability of the solution.

The `IOzone` and `mdtest` utilities were used in this study. `IOzone` was used for the sequential and random tests. For sequential tests, a request size of 1024KiB was used. The total amount of data transferred was 256GiB to ensure that the NFS server cache was saturated. Random tests used a 4KiB request size and each client read and wrote a 4GiB file. Metadata tests were performed using the `mdtest` benchmark and included file create, stat, and remove operations. Refer to Appendix A for the complete commands used in the tests.

5.1. IPoIB sequential writes and reads

Figure 4 shows the sequential write and read performance. The peak read performance is 4138MB/sec, and the peak write performance is 1171MB/sec. The design choice of RAID-6 8+2 was to optimize the service availability and disk layout with the new storage enclosures.

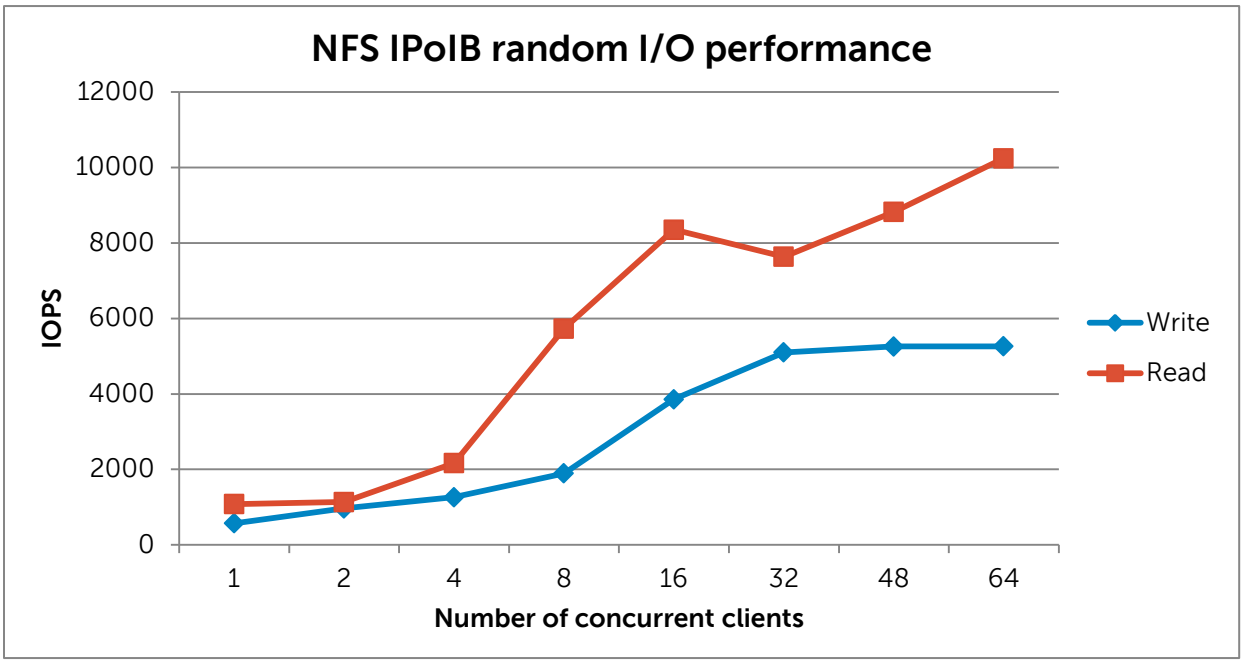
Figure 4. IPoIB large sequential write and read performance



5.2. IPoIB random writes and reads

Figure 5 shows the random write and read performance. From the figure, the random write performance peaks at the 32-client test case and then holds steady. In contrast, the random read performance increases steadily beyond going from 32, to 48 to 64 clients indicating that the peak random read performance is likely to be greater than 10244 IOPS (the performance for 64-client random read test case).

Figure 5. IPoIB random write and read performance



5.3. IPoIB metadata operations

Figure 6, Figure 7, and Figure 8 show the results of file create, stat, and remove operations, respectively. As the HPC compute cluster has 64 compute nodes, in the graphs below, each client executed a maximum of one thread for client counts up to 64. For client counts of 128, 256, and 512, each client executed 2, 3, or 4 simultaneous operations.

From the three figures, it is observed that file create and file remove operations have very similar performance trajectories, while the file stat operation has much higher performance numbers than the file create and remove operations. Such behaviors are expected, as file create and remove operations are write-intensive operations with a small request size, while a file stat operation is read-intensive, and RAID 6 has a larger overhead for small write operations than read operations.

Figure 6. IPoIB file create performance

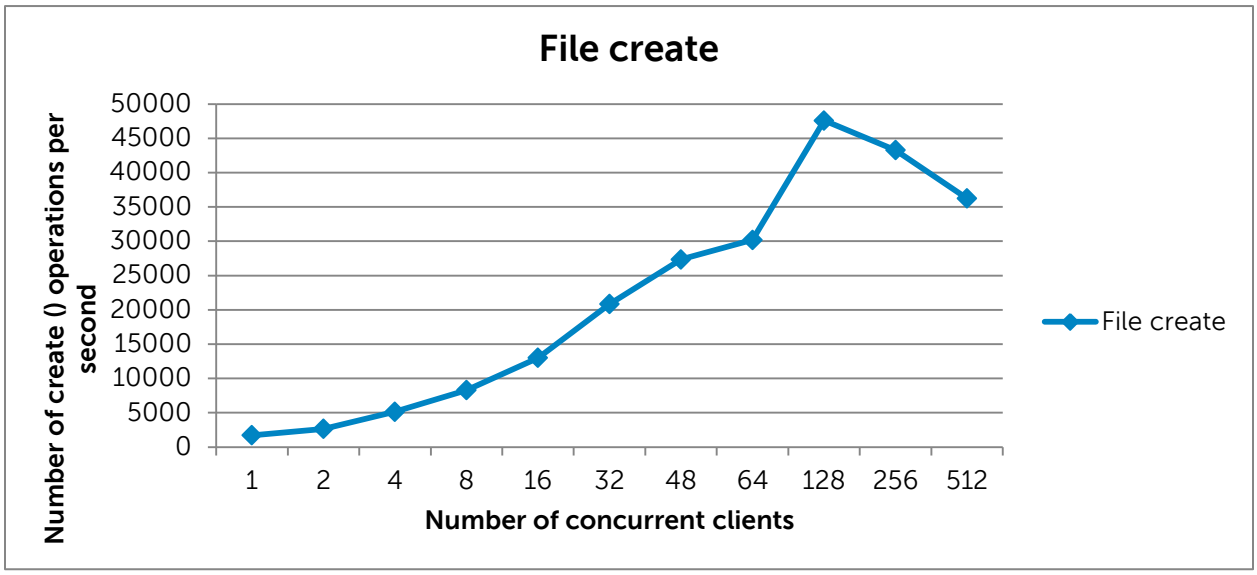


Figure 7. IPoIB file stat performance

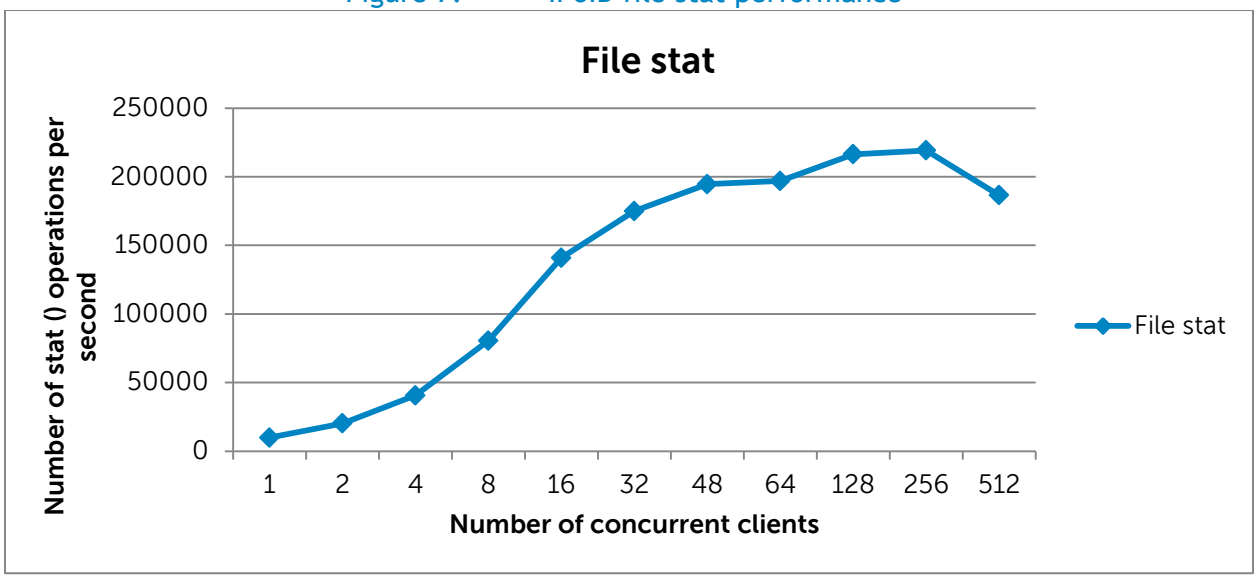
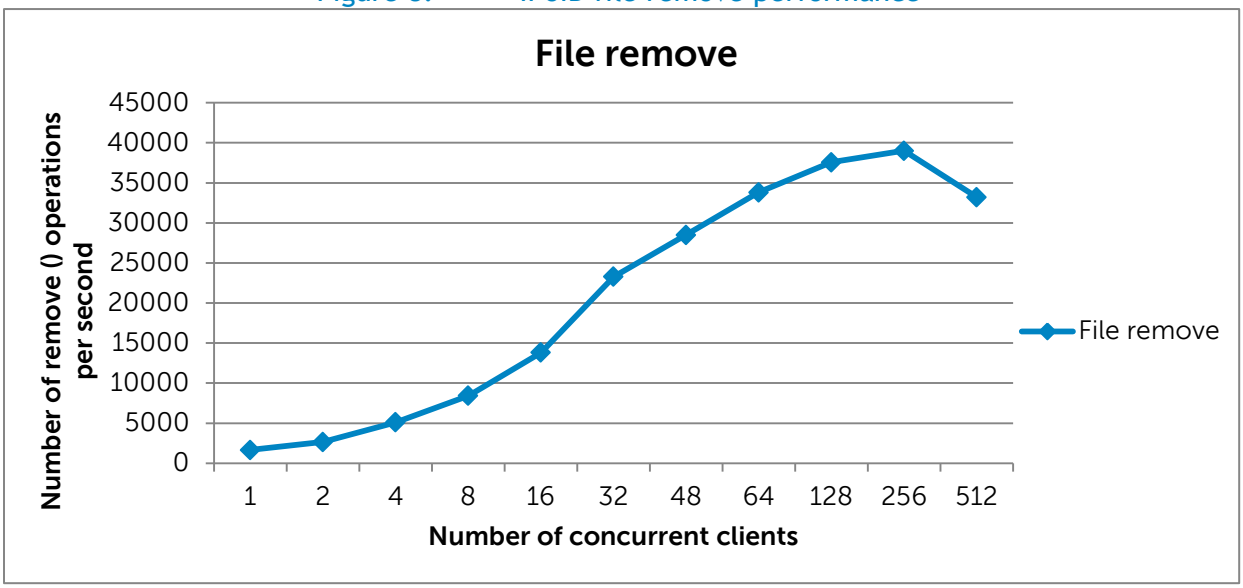


Figure 8. IPoIB file remove performance



6. Conclusion

This solution guide provides details of the latest Dell HPC NSS-HA Solution. With this release, the Dell NSS-HA solution reduces the solution footprint by 2.5 times as compared to the previous releases and improves its availability and reliability. The Dell NSS-HA solution is available with deployment services and full hardware and software support from Dell. This document provides complete technical details on the configuration and performance analysis of the solution.

7. References

1. Dell HPC NFS Storage Solution High Availability Configurations, Version 1.1
http://i.dell.com/sites/content/business/solutions/whitepapers/en/Documents/dell-hpc-nssha-sg.pdf

2. Dell HPC NFS Storage Solution – High availability with large capacities, Version 2.1
http://i.dell.com/sites/content/business/solutions/engineering-docs/en/Documents/hpc-nfs-storage-solution.pdf

3. Dell HPC NFS Storage Solution High Availability (NSS-HA) Configurations with Dell PowerEdge 12th Generation Servers, Version 1.0
http://www.dellhpcsolutions.com/assets/pdfs/NSS_HA_12G_final_July16.pdf

4. Red Hat Enterprise Linux 6 Cluster Administration – Configuring and Managing the High Availability Add-On.

   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Cluster_Administration/Red_Hat_Enterprise_Linux-6-Cluster_Administration-en-US.pdf

5. Dell NFS Storage Solution with High Availability - XL configuration.

   http://en.community.dell.com/techcenter/high-performance-computing/w/wiki/2299.aspx

Appendix A: Benchmarks and test tools

The `IOzone` benchmark was used to measure sequential read and write throughput (MB/sec) as well as random read and write I/O operations per second (IOPS).

The `mdtest` benchmark was used to test metadata operation performance.

The `checkstream` utility was used to test for data correctness under failure and failover cases.

The Linux `dd` utility was used for initial failover testing and to measure data throughput as well as the time to complete file copy operations.

A.1. IOzone

You can download `IOzone` from http://www.iozone.org/. Version 3.408 was used for these tests and installed on both the NFS servers and all the compute nodes.

The `IOzone` tests were run from 1-64 nodes in clustered mode. All tests were N-to-N, that is N clients would read or write N independent files.

Between tests, the following procedure was followed to minimize cache effects:

- Unmount NFS share on clients.
- Stop the cluster service on the server. This unmounts the XFS file system on the server.
- Start the cluster service on the server.
- Mount NFS Share on clients.

The following table describes the `IOzone` command line arguments.

Table 8. Appendix A — IOzone command line arguments

IOzone Argument	Description
-i 0	Write test.
-i 1	Read test.
-i 2	Random Access test.
-+n	No retest.
-c	Includes close in the timing calculations.
-t	Number of threads.
-e	Includes flush in the timing calculations.
-r	Records size.

IOzone Argument	Description
-s	File size.
-t	Number of threads.
+m	Location of clients to run IOzone when in clustered mode.
-w	Does not unlink (delete) temporary file.
-I	Use O_DIRECT, bypass client cache.
-O	Give results in ops/sec.

For the sequential tests, file size was varied along with the number of clients such that the total amount of data written was 256GiB (number of clients * file size per client = 256GiB).

IOzone Sequential Writes

```
# /usr/sbin/iozone -i 0 -c -e -w -r 1024k -s 4g -t 64 -+n -+m ./clientlist
```

IOzone Sequential Reads

```
# /usr/sbin/iozone -i 1 -c -e -w -r 1024k -s 4g -t 64 -+n -+m ./clientlist
```

For the random tests, each client read or wrote a 4GiB file. The record size used for the random tests was 4KiB to simulate small random data accesses.

IOzone IOPs Random Access (Reads and Writes)

```
# /usr/sbin/iozone -i 2 -w -r 4k -I -O -w -+n -s 4g -t 1 -+m ./clientlist
```

By using -c and -e in the test, IOzone provides a more realistic view of what a typical application is doing. The O_Direct command line parameter allows us to bypass the cache on the compute node on which we are running the IOzone thread.

A.2. mdtest

You can download `mdtest` from http://sourceforge.net/projects/mdtest/. Version 1.8.3 was used in these tests. It was compiled and installed on a NFS share that was accessible by compute nodes. `mdtest` is launched with `mpirun`. For these tests, OpenMPI version 1.4.3 was used. The following table describes the `mdtest` command-line arguments.

mpirun ARGUMENT	DESCRIPTION
-np	Number of Processes.
--nolocal	Instructs `mpirun` not to run locally.
--hostfile	Tells `mpirun` where the hostfile is located.
mdtest ARGUMENT	**DESCRIPTION**
-d	The directory `mdtest` should run in.
-i	The number of iterations the test will run.
-b	Branching factor of directory structure.
-z	Depth of the directory structure.
-L	Files only at leaf level of tree.
-I	Number of files per directory tree.
-y	Sync the file after writing.
-u	Unique working directory for each task.
-C	Create files and directories.
-R	Randomly stat files.
-T	Only stat files and directories.
-r	Remove files and directories left over from run.

As with the `IOzone` random access patterns, the following procedure was followed to minimize cache effects during the metadata testing:

- Unmount NFS share on clients.
- Stop the cluster service on the server. This unmounts the XFS file system on the server.

- Start the cluster service on the server.
- Mount NFS Share on clients.

Metadata file and directory creation test:

```
# mpirun -np 32 --nolocal --hostfile ./hosts /nfs/share/mdtest -d
/nfs/share/filedir -i 6 -b 320 -z 1 -L -I 3000 -y -u -t -C
```

Metadata file and directory stat test:

```
# mpirun -np 32 --nolocal --hostfile ./hosts /nfs/share/mdtest -d
/nfs/share/filedir -i 6 -b 320 -z 1 -L -I 3000 -y -u -t -R -T
```

Metadata file and directory removal test:

```
# mpirun -np 32 --nolocal --hostfile ./hosts /nfs/share/mdtest -d
/nfs/share/filedir -i 6 -b 320 -z 1 -L -I 3000 -y -u -t -r
```

A.3. Checkstream

The `checkstream` utility is available at http://sourceforge.net/projects/checkstream/. Version 1.0 was installed and compiled on the NFS servers and used for these tests.

First, a large file was created using the `genstream` utility. This file was copied to and from the NFS share by each client using `dd` to mimic write and read operations. Failures were simulated during the file copy process and the NFS service was failed over from one server to another. The resultant output files were checked using the `checkstream` utility to test for data correctness and ensure that there was no data corruption.

Below is a sample output of a successful test with no data corruption.

```
checkstream[genstream.file.100G]: ----------------------------------------------
checkstream[genstream.file.100G]: valid data for 107374182400 bytes at offset 0
checkstream[genstream.file.100G]: ----------------------------------------------
checkstream[genstream.file.100G]: end of file summary
checkstream[genstream.file.100G]: [valid data] 1 valid extents in 261.205032
seconds (0.00382841 err/sec)
checkstream[genstream.file.100G]: [valid data] 107374182400/107374182400 bytes (100
GiB/100 GiB)
checkstream[genstream.file.100G]: read 26214400 blocks 107374182400 bytes in
261.205032 seconds (401438 KiB/sec), no errors
```

For comparison, here is an example of a failing test with data corruption in the copied file. For example, if the file system is exported via the NFS async operation and there is an HA service failover during a write operation, data corruption is likely to occur.

```
checkstream[compute-00-10]: ----------------------------------------------
checkstream[compute-00-10]: valid data for 51087769600 bytes at offset 45548994560
checkstream[compute-00-10]:
checkstream[compute-00-10]: ----------------------------------------------
checkstream[compute-00-10]: end of file summary
```

```
checkstream[compute-00-10]: [valid data] 1488 valid extents in 273.860652 seconds
(5.43342 err/sec)
checkstream[compute-00-10]: [valid data] 93898678272/96636764160 bytes (87 GiB/90
GiB)
checkstream[compute-00-10]: [zero data] 1487 errors in 273.860652 seconds (5.42977
err/sec)
checkstream[compute-00-10]: [zero data] 2738085888/96636764160 bytes (2 GiB/90 GiB)
checkstream[compute-00-10]: read 23592960 blocks 96636764160 bytes in 273.860652
seconds (344598 KiB/sec)
checkstream[compute-00-10]: ------------------------------------------------
checkstream[compute-00-10]: encountered 1487 errors, failing
```

A.4. The dd Linux utility

`dd` is a Linux utility provided by the coreutils rpm distributed with RHEL 6.3. It was used to copy a file. The NFS file system was mounted at `/mnt/xfs` on the clients.

To write data to the storage, the following command line was used.

```
# dd if=/dev/zero of=/mnt/xfs/file bs=1M count=90000
```

To read data from the storage, the following command line was used.

```
# dd if=/mnt/xfs /file of=/dev/null bs=1M
```