Dell HPC NFS Storage Solution – High Availability (NSS6.0-HA) Configuration with Dell PowerEdge 13th Generation Servers

A Dell Technical White Paper

Xin Chen

Dell HPC Engineering

November 2014| Version 1.0

November 2014 | Rev 1.0

Contents

Tables

Figures

Executive summary

This white paper describes the Dell NFS Storage Solution - High Availability configurations (NSS6.0-HA) with Dell PowerEdge 13th generation servers. It presents a comparison among all available NSS-HA offerings so far, and provides performance results for a configuration with a storage system providing 480TB of raw capacity.

The NSS-HA solution described here 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. The goal of the solution is to improve storage service availability and maintain data integrity in the presence of possible failures or faults and to optimize performance in a failure-free scenario.

1. Introduction

This white paper provides information on the latest Dell NFS Storage Solution - High Availability configurations with Dell PowerEdge 13th generation servers. 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 PowerEdge 13th generation servers and Red Hat Enterprise Linux 7.0 operating system (RHEL 7.0) to deliver more powerfull storage solutions than previous NSS-HA solutions. This version of the solution is NSS6.0-HA.

The design principle for this release remains the same as previous Dell NSS-HA solutions. The major changes between the current and the last version of NSS-HA solution(NSS5.5-HA) are

- the change from Dell PowerEdge 12th generation servers (R620) to the latest PowerEdge 13th generation servers (R630);
- the change from the RHEL 6.5 to RHEL 7.0;
- and the change from 3TB disks to 4TB disks.

With those changes, the NSS6.0-HA gains the following two improvements

- It supports larger storage capacity than previous NSS-HA releases. The maximum capacity is up to 500TB, while 300TB is the limit of previous NSS-HA releases.
- It delivers faster sequential read performance than NSS5.5-HA, 75% on average, and maintains similar write performance as NSS5.5-HA.

For complete details of the NSS-HA solution family, review this document along with the previous NSS-HA white papers and blogs[1] [2] [3] [4] [5] [6].

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 similar. In general, the core of the solution is a high availability (HA) cluster[7], 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).

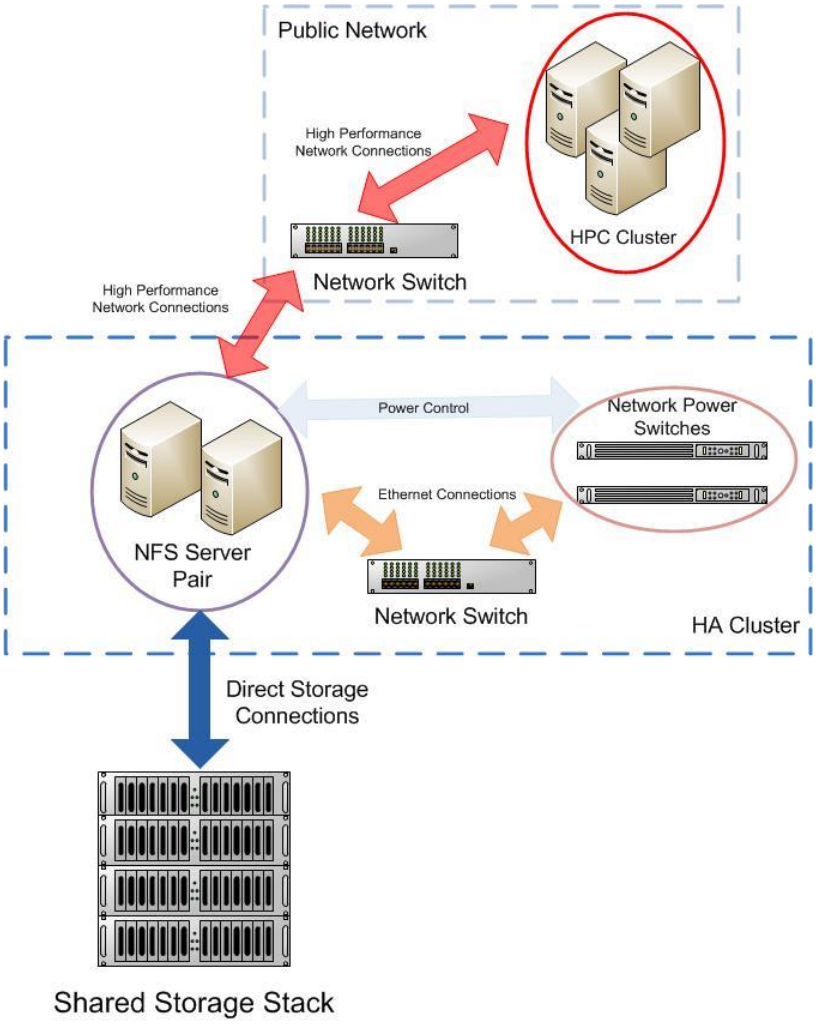
The HA cluster consists of a pair of Dell PowerEdge servers and a network switch. The two PowerEdge servers have shared access to disk-based Dell PowerVault storage in a variety of capacities, and both are directly connected to the HPC cluster via IB or 10GbE. The two servers are equipped with two fence devices: iDRAC7 Enterprise and an APC Power Distribution Unit (PDU). If failures such as storage disconnection, network disconnection, system crash, etc., occur on one server, the HA cluster will

failover the storage service to the healthy server with the assistance of the two fence devices; and 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 8 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 the available Dell NSS-HA solutions with standard configurations.

Table 1. NSS-HA Solutions[(1), (2), (3), (4), (5), (6)]

	NSS5.5-HA Release (April 2014) [(6)] **"PowerVault MD3460 based solution"**	**NSS6.0-HA Release (November 2014) "PowerEdge 13th generation server based solution"**
Storage Capacity	180TB to 360TB of raw storage capacity.	240TB and 480TB of raw storage capacity.
Network Connectivity	FDR InfiniBand or 10GbE Connectivity.	
Direct Storage Connection	12 Gbps SAS connections	
NFS servers	Dell PowerEdge R620 servers. CPU: Dual Intel Xeon E5-2695v2 @ 2.40 GHz, 12 cores per processor. Memory: 16 x 8GiB 1866 MHz RDIMMs.	Dell PowerEdge R630 servers. CPU: Dual Intel Xeon E5-2697v3 @ 2.60 GHz, 14 cores per processor. Memory: 16 x 8GiB 2133 MHz RDIMMs.
Software	Red Hat Enterprise Linux 6.5 Red Hat Scalable File system (XFS) v3.1.1-14	Red Hat Enterprise Linux 7.0 Red Hat Scalable File system (XFS) v3.2.0-0.10
Storage Devices	Dell PowerVault MD3460 and MD3060e. 3TB 7.2K NL SAS drives.	Dell PowerVault MD3460 and MD3060e. 4TB 7.2K NL SAS drives.
Local Switch and Power Distribution Units (PDUs)	PowerConnect 2848. 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:

- The table only lists NSS-HA versions currently available on the market. The NSS-HA versions before NSS5.5-HA are no longer 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 MD3460 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 4TB disks. In the previouse NSS-HA solutions[3], [4], [5], [6], 3TB disks were used. The PowerVault MD3460 and MD3060e storage arrays were used in NSS5.5-HA[6], and they are still being used in NSS6.0-HA.

Both PowerVault MD3460[8] and MD3060e[9] storage array are 4U, 60 drive dense enclosures. The major difference between them is that MD3460 has dual active/active RAID controllers and is used as an RBOD (Redundant Array of Inexpensive Disks Bunch of Disks), while, MD3060e is an EBOD (expansion bunch of disks), which is usually used to extend the capacity of the PowerVault MD3460.

NSS6.0-HA shares the same storage configurations as NSS5.5-HA except using 4TB disks instead of 3TB disks. Table 2 summarizes the supported storage configurations of the NSS6.0-HA.

Table 2. Storage configurations in NSS6.0-HA

Storage configurations	240TB: One PowerVault MD3460. 480TB: One PowerVault MD3460 + One PowerVault MD3060e.
Disks in storage array	4TB NL SAS.
Virtual disk configuration	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	Chain cabling scheme.
Logical volume configuration	Stripe element size: 512KiB. Number of stripes: 6. Number of virtual disks per logical volume: 6 virual disks for 240TB configuration, 12 virtual disks for 480TB 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 480TB NSS6.0-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 [6].

4.2. Test bed

The test bed used to evaluate the NSS6.0-HA functionality and performance is shown in Figure 2.

- A 64 node HPC compute cluster was used to provide I/O traffic for the test bed.
- A pair of Dell PowerEdge R630 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 MD3460 storage enclosure extended with one Dell PowerVault MD3060e storage enclosure (Figure 2 shows a 480TB 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 to the clients via NFS.
- The NFS servers were connected to the clients using the public network. This network was either InfiniBand FDR 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 2848 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, and Table 5.

Figure 2. NSS6.0-HA test bed

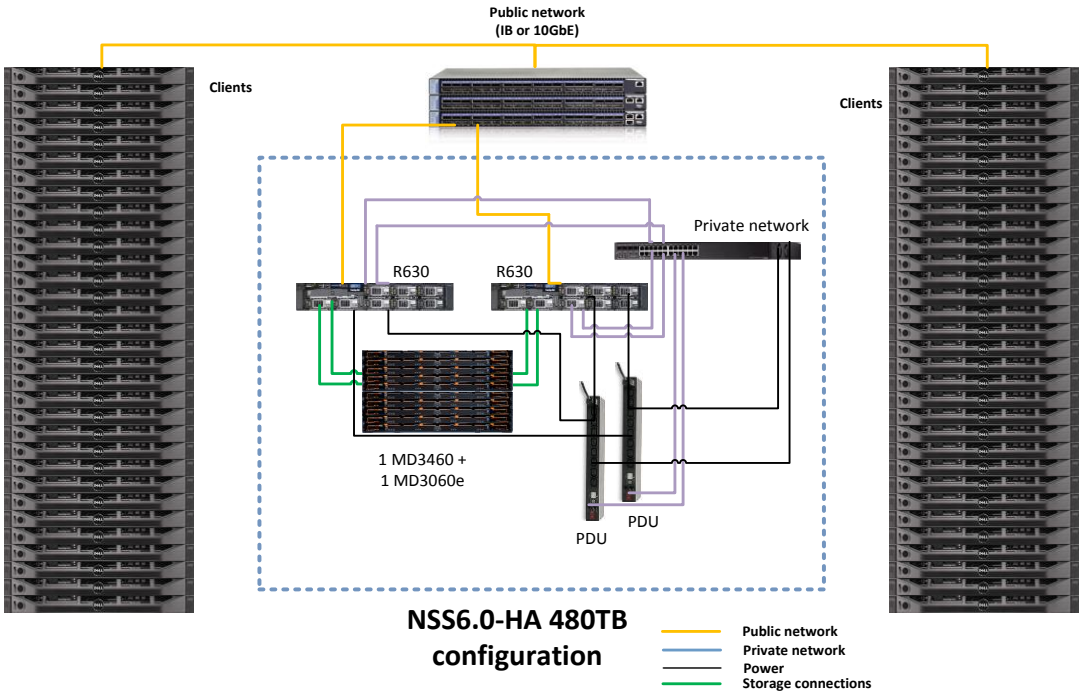


Table 3. NSS6.0-HA hardware configuration

Server configuration	
NFS server model	Two Dell PowerEdge R630s.
Processor	Dual Intel Xeon E5-2697 v3 @ 2.60 GHz, 14 cores per processor.
Memory	8 x 8GiB 2133 MHz RDIMMs. (The test bed used 64GB; the recommendation for production clusters is to use 128GB).
Local disks and RAID controller	PERC H730 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.
1/10 Gigabit Ethernet card (Network Daughter card slot)	Broadcom 57800 1/10 Gigabit Ethernet network daughter card.
External storage controller (slot 3 and slot 1)	Two 12Gbps SAS HBAs.

Systems Management	iDRAC8 Enterprise.
Power Supply	Dual Power Supply Units.
Storage configuration	
Storage Enclosure	One Dell PowerVault MD3460 array and one MD3060e array for the 480TB solution.
RAID controllers	Duplex RAID controllers in the Dell MD3460.
Hard Disk Drives	60 - 4TB 7200 rpm NL SAS drives per array.
Other components	
Private Gigabit Ethernet switch	Dell PowerConnect 2848.
Power Distribution Unit	Two APC switched Rack PDUs, model AP7921.

Table 4. NSS6.0-HA software versions

Software	
Operating system	Red Hat Enterprise Linux (RHEL) 7.0 x86_64
Kernel version	3.10.0-123.8.1.el7.x86_64
Cluster Suite	Red Hat Cluster Suite from RHEL 7.0
File system	Red Hat Scalable File System (XFS) 3.2.0-0.10
Systems management tool	Dell OpenManage Server Administrator 8.0.1
Storage management package	Consolidated Resource DVD (RDVD): 5.1.0.9

Note:

The original kernel distributed with RHEL7.0 was 3.10.0-123.el7.x86_64. To fix the issues discovered with that kernel which impacts the functionality and performance of NSS6.0-HA, the kernel version was updated to 3.10.0-123.8.1.el7.x86_64.

Table 5. NSS6.0-HA client cluster configuration

Client / HPC Compute Cluster	
Clients	64 PowerEdge M420 blade servers 32 blades in each of two PowerEdge M1000e chassis Red Hat Enterprise Linux 6.4 x86-64.
Chassis configuration	Two PowerEdge M1000e chassis, each with 32 blades Two Mellanox M4001F FDR10 I/O modules per chassis Two PowerConnect M6220 I/O switch modules per chassis
InfiniBand	Each blade server has one Mellanox ConnectX-3 Dual-port FDR10 Mezzanine I/O Card Mellanox OFED 2.3-1.0.1
InfiniBand fabric for I/O traffic	Each PowerEdge M1000e chassis has two Mellanox M4001 FDR10 I/O module switches. Each FDR10 I/O module has four uplinks to a rack Mellanox SX6025 FDR switch for a total of 16 uplinks. The FDR rack switch has a single FDR link to the NFS server.
Ethernet	Each blade server has one onboard 10GbE Broadcom 57810 network adapter.
Ethernet fabric for cluster deployment and management	Each PowerEdge M1000e chassis has two PowerConnect M6220 Ethernet switch modules. Each M6220 switch module has one link to a rack PowerConnect 5224 switch. There is one link from the rack PowerConnect switch to an Ethernet interface on the cluster master node.

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 general 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 6 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 6. 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	iDRAC8 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 each controller in 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 MD3460 storage array	Dual controllers in the Dell PowerVault MD3460. 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 6.

- 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 in response to these failures are outlined below.

- 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 iDRAC8 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 during the configuration.

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 R630 server and the Dell PowerVault MD3460 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 R630 server and the Dell PowerVault MD3460 storage.
  When all SAS links on the active server fail, the HA service will attempt to failover to the passive server. 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 usually take less than two 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.

The information about `IOzone`, `checkstream`, and `dd` can be found in Appendix A.

5. NSS6.0-HA I/O Performance

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 two types of I/O patterns: large sequential reads and writes, and small random reads and writes.

A 480TB configuration was benchmarked with IPoIB cluster 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` tool was 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. Refer to Appendix A for the complete commands used in the tests.

5.1. IPoIB sequential writes and reads

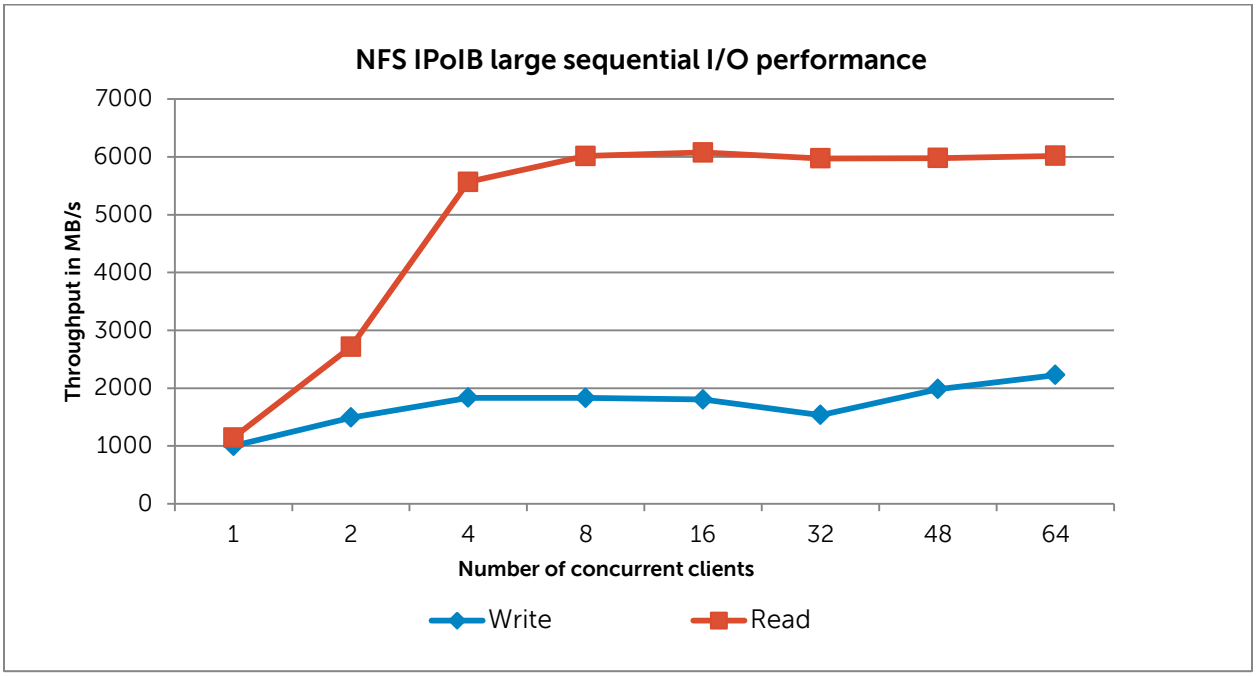
In the sequential write and read tests, the I/O access patterns are N-to-N, i.e., each client reads and writes to its own file. Iozone was executed in clustered mode and one thread was launched on each compute node. As the total transferred data size was kept constant at 256 GiB, the file size per client varied accordingly for each test case. For example, 256 GiB file was read or written in 1-client test case, 128 GiB file was read or written per client node in 2-client test case.

Figure 3 shows the sequential write and read performance. The figure shows the aggregate throughput that can be achieved when a number of clients are simultaneously writing or reading from the storage over the InfiniBand fabric.

Leveraging the latest PowerEdge R630 server and RHEL 7.0, significant sequential I/O performance improvements were observed during our tests:

- The peak read performance of NSS6.0-HA was up to 6.07 GB/sec; and there were on average around 75% improvement as compared to the read performance of NSS5.5-HA[6].
- The peak write performance of NSS6.-HA was up to 2.23 GB/sec, and there were on average around 14% improvement as compared to the write performance of NSS5.5-HA[6].

Figure 3. IPoIB large sequential write and read performance

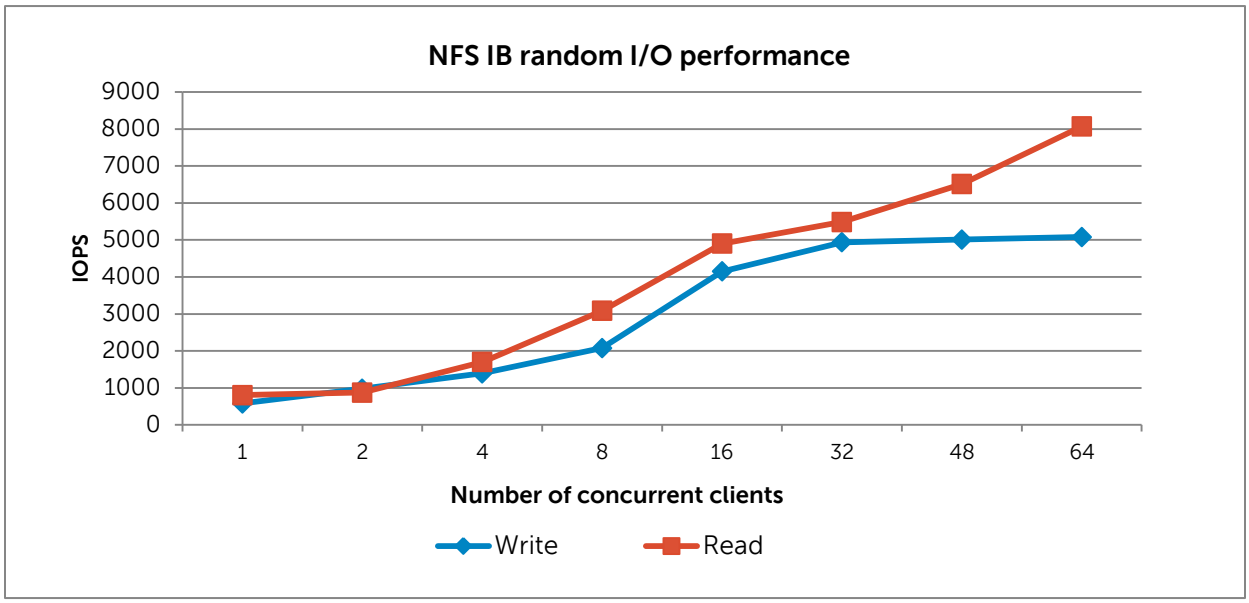


5.2. IPoIB random writes and reads

Figure 4 shows the random write and read performance. The figure shows the aggregate I/O operations per second when a number of clients are simultaneously writing or reading to/from the storage over the InfiniBand fabric.

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 from 1 to 64-client test cases, indicating that the peak random read performance is likely to be around 8069 IOPS or more.

Figure 4. IPoIB random write and read performance



6. Conclusion

This document provides details of the latest Dell HPC NSS-HA Solution, including the solution configuration, HA functionality evaluation and perfomrnace evaluation of the solution. With this version, the Dell NSS6.0-HA solution not only offers larger storage capacity but also delivers significant sequential I/O performance improvement when compared to the previous NSS-HA versions. The Dell NSS-HA solution is available with deployment services and full hardware and software support from Dell.

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. Dell NFS Storage Solution with High Availability: PowerVault MD3260/MD3060e, http://www.dellhpcsolutions.com/asset/147510210/96000/3710410/119266
5. Dell HPC NFS Storage Solution - High Availability Solution NSS5-HA configurations, http://en.community.dell.com/techcenter/high-performance-computing/b/hpc_storage_and_file_systems/archive/2013/10/28/dell-hpc-nfs-storage-solution-high-availability-solution-nss5-ha-configurations.aspx
6. Dell HPC NFS Storage Solution High Availability (NSS5.5-HA) Configuration with Dell PowerVault MD3460 and MD3060e Storage Arrays, Version 1.0 http://i.dell.com/sites/doccontent/shared-content/solutions/en/Documents/Nss5-5-ha-v1-0_final.pdf
7. Red Hat Enterprise Linux 7 High Availability Add-On Reference https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/High_Availability_Add-On_Reference/index.html
8. Manuals & documentation for your PowerVault MD3460 http://www.dell.com/support/home/us/en/19/product-support/product/powervault-md3460/manuals
9. Manuals & documentation for PowerVault MD3060e http://www.dell.com/support/home/us/en/19/product-support/product/powervault-md3060e/manuals

Appendix A: Benchmarks and test tools

The `IOzone` benchmark tool 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 `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 3.420 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 7. 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.
-s	File size.

IOzone Argument	Description
-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. 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.3. The dd Linux utility

`dd` is a Linux utility provided by the coreutils rpm distributed with RHEL 7.0. 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
```