



# DELL EMC ISILON F800 AND H600 I/O PERFORMANCE

## ABSTRACT

This white paper provides F800 and H600 performance data. It is intended for performance-minded administrators of large compute clusters that access Isilon storage.

January 2018

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2018 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA. 1/18 White Paper

Dell EMC believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

TABLE OF CONTENTS

- ABSTRACT.....1
- EXECUTIVE SUMMARY .....4
- INTRODUCTION.....4
- DELL EMC ISILON.....4
- PERFORMANCE EVALUATION.....5
  - NETWORK CONNECTIVITY .....7
  - TEST TOOLS.....7
  - NETWORK BANDWIDTH TESTS .....7
  - I/O TESTS.....8
    - SEQUENTIAL I/O PERFORMANCE TESTS (N-TO-N) .....8
    - SEQUENTIAL I/O PERFORMANCE TESTS (N-TO-1).....10
    - RANDOM I/O PERFORMANCE TESTS .....12
    - I/O PERFORMANCE COMPARISONS: F800 VS. LUSTRE VS. NFS .....15
- REFERENCES.....19
- APPENDIX .....19

## EXECUTIVE SUMMARY

HPC applications vary widely in their I/O profiles. For instance, applications may read/write to and from files in an N-to-1 or N-to-N manner. A failure to match these I/O needs with the proper filesystem(s) can result in poor application performance, system underutilization and user frustration. This Dell EMC technical white paper describes sequential and random I/O performance results for Dell EMC Isilon F800 and H600 node types. The data is intended to inform administrators on the suitability of Isilon storage clusters for HPC various workflows.

## INTRODUCTION

The goal of this document is to present a performance comparison of F800 and H600 node types, and where possible, to compare them to other storage offerings like the Dell HPC Lustre Storage Solution and the Dell NFS Storage Solution (NSS). The same test methodologies and the same test hardware were used where possible to generate the results.

## DELL EMC ISILON

Dell EMC Isilon is a proven scale-out network attached storage (NAS) solution that can handle the unstructured data prevalent in many different workflows. The Isilon storage architecture automatically aligns application needs with performance, capacity, and economics. As performance and capacity demands increase, both can be scaled simply and non-disruptively, allowing applications and users to continue working.

Dell EMC Isilon OneFS operating system powers all Dell EMC Isilon scale-out NAS storage solutions and features:

- A high degree of scalability, with grow-as-you-go flexibility
- High efficiency to reduce costs
- Multi-protocol support such as SMB, NFS, HTTP and HDFS to maximize operational flexibility
- Enterprise data protection and resiliency
- Robust security options

A single Isilon storage cluster can host multiple node types to maximize deployment flexibility. Node types range from the Isilon F (All-Flash), to H (Hybrid) and A (Archive) nodes. Each provides a different optimization point for capacity, performance, and cost. Automated processes can be established that automatically migrate data from higher-performance, higher-cost nodes to more cost-effective storage. Nodes can be added “on the fly,” with no disruption of user services. Additional nodes result in increased performance (including network interconnect), capacity and resiliency.

Isilon OneFS also supports additional services for performance, security, and protection:

- **SmartConnect** is a software module that optimizes performance and availability by enabling intelligent client connection load balancing and failover support. Through a single host name, SmartConnect enables client connection load balancing and dynamic NFS failover and failback of client connections across storage nodes to provide optimal utilization of the cluster resources.
- **SmartPools** provides rule based movement of data through tiers within an Isilon cluster. Institutions can set up rules keeping the higher performing nodes like F and H available for immediate access to data for computational needs and A-series nodes used for all other data. It does all this while keeping data within the same namespace, which can be especially useful in a large shared research environment.
- **SmartFail** and **Auto Balance** ensure that data is protected across the entire cluster. There is no data loss in the event of any failure and no rebuild time necessary. This contrasts favorably with other file systems such as Lustre or GPFS as they have significant rebuild times and procedures in the event of failure with no guarantee of 100% data recovery.
- **SmartQuotas** help control and limit data growth. Evolving data acquisition and analysis modalities coupled with significant movement and turnover of users can lead to significant consumption of space. Institutions without a comprehensive data management plan or practice can rely on SmartQuotas to better manage growth.

Through utilization of common network protocols such as CIFS/SMB, NFS, HDFS, SWIFT and HTTP, Isilon can be accessed from any number of machines and by any number of users leveraging existing authentication services.

## PERFORMANCE EVALUATION

This Dell EMC technical white paper describes sequential and random I/O performance results for Dell EMC Isilon F800 and H600 node types and compares them to other storage offerings, like the Dell HPC Lustre Storage Solution and the Dell NFS Storage Solution (NSS), where possible. The data is intended to inform administrators on the suitability of Isilon storage clusters for various HPC workloads. The same test methodologies and the same test hardware were used where possible to generate the results.

### Storage configurations

Table 1 lists the configuration details of the three storage systems benchmarked. Details on the Dell NFS Storage Solution-High Availability (NSS-7.0-HA) can be found [here](#).

	F800	H600	NSS-7.0-HA
# of nodes	4	4	1
CPU per node	Intel(R) Xeon(R) CPU E5-2697A v4 @2.60GHz	Intel(R) Xeon(R) CPU E5-2680 v4 @2.40GHz	Dual Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz
Memory per node	256 GB	256 GB	128 GB
Storage Capacity	Total usable space: 166.8 TB 41.7 TB per node	Total usable space: 126.8 TB 31.7 TB per node	Total usable space: 306 TB
SSD L3 cache	NA	2.9 TB per node	NA
Network	Front end network: 40GbE Back end network: 40GbE	Front end network: 40GbE Back end network: IB QDR	Intel OmniPath
OS	Isilon OneFS v8.1.0.DEV.0	Isilon OneFS v8.1.0.0 B_8_1_0_011	Red Hat Enterprise Linux Server release 7.3 (Maipo) Kernel: 3.10.0- 514.el7.x86_64

**Table 1. Storage configurations. NFSv3 was used in all tests.**

### Compute nodes

64 nodes of the Zenith compute cluster<sup>1</sup> were used during the tests. Table 2 lists Zenith compute node configuration details.

DELL HPC INNOVATION LABS ZENITH COMPUTE CLUSTER

<i>Compute clients</i>	64 x PowerEdge C6320s
<i>Processor</i>	CPU: Intel(R) Xeon(R) CPU E5-2697 v4 @ 2.30GHz No. of cores = 18 per processor (36 per node) Processor Base Frequency: 2.3GHz AVX Base: 2.0GHz
<i>Memory</i>	128 GB @ 2400 MHz per node
<i>Operating System</i>	Red Hat Enterprise Linux Server release 7.2 (Maipo)
<i>Kernel</i>	3.10.0-327.13.1.el7.x86_64
<i>System BIOS Profile</i>	Max Performance <ul style="list-style-type: none"> <li>• Turbo mode: Enabled</li> <li>• C-states: disabled</li> <li>• Node interleave: disabled</li> <li>• Logical processor: disabled</li> <li>• Snoop mode: opportunistic snoop broadcast</li> <li>• I/O-Nonposted Prefetch: Disabled</li> </ul>
<i>Network</i>	1GbE, 10GbE, and Intel OmniPath

**Table 2. Zenith compute node details.**

## Network connectivity

The Zenith cluster and F800 storage system were connected via 8 x 40GbE links. Figure 2 shows the network topology used in the tests. The H600 was configured in the exact same way as the F800. Figure 3 shows the network configuration of the NSS-7.0-HA.

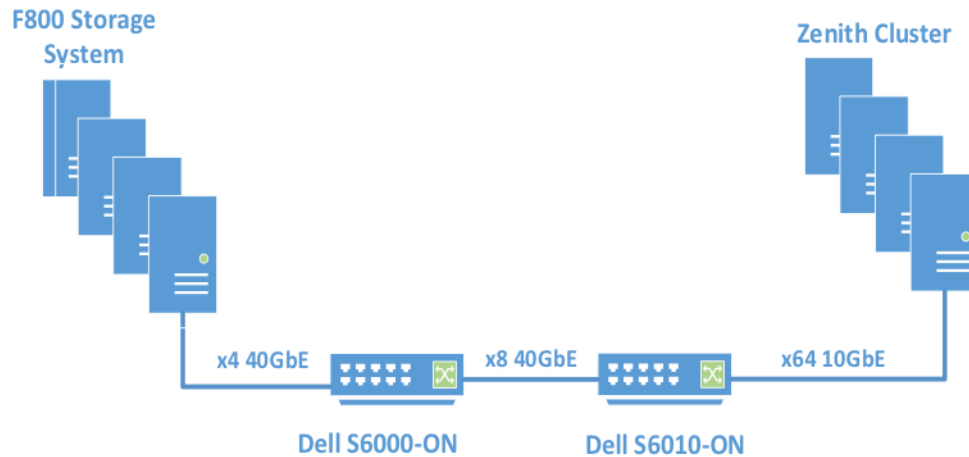


Figure 2. Network diagram of the F800 benchmark configuration

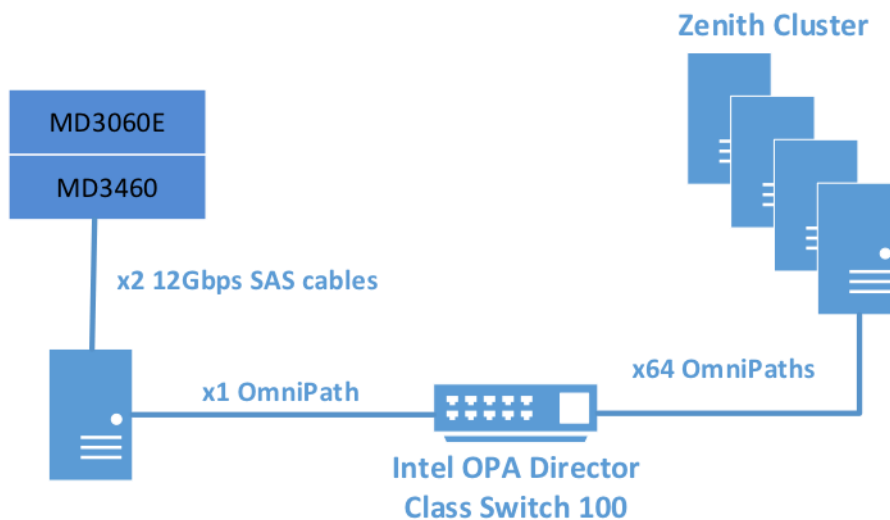


Figure 3. Network diagram of the NSS-7.0-HA benchmark configuration

## Test tools

- *iperf*<sup>2</sup> 2.0.5 was used for testing network bandwidth
- *IOR*<sup>3</sup> v.3.0.1 was used for sequential and random I/O tests

## Network Bandwidth Tests

Network bandwidth tests were run in order to understand network performance between the Zenith cluster and F800 nodes. The tool used to test network bandwidth was *iperf*. Bandwidth results:

- A single Zenith node to a single F800 node: ~9.9 Gbits/sec.
- A single F800 node to a Zenith node: ~9.8 Gbits/sec.

- Eight Zenith nodes to a single F800 node: ~40.06 Gbits/sec.
- Twenty Zenith nodes to four F800 nodes: ~154.29 Gbits/sec.

The *iperf* results demonstrate that the bandwidth between a single Zenith node and a single F800 node is approximately 9.9 Gbits/sec (or ~1.2 GB/sec), maximum bandwidth between Zenith and a single F800 node is approximately 40 Gbits/sec (or ~5GB/sec), and the aggregate bandwidth between the Zenith nodes and the F800 cluster is approximately 154.29 Gbits/sec (or ~19 GB/sec). Similar results were obtained for the H600:

- A single Zenith node to a single H600 node: ~9.9 Gbits/sec.
- A single H600 node to a Zenith node: ~9.8 Gbits/sec.
- Eight Zenith nodes to a single H600 node: ~39.81 Gbits/sec.
- Twenty Zenith nodes to four H600 nodes: ~120.83 Gbits/sec.

The *iperf* results demonstrate that the bandwidth between a single Zenith node and a single H600 node is approximately 9.9 Gbits/sec (or ~1.2 GB/sec), maximum bandwidth between Zenith and a single H600 node is approximately 40 Gbits/sec (or ~5GB/sec), and the aggregate bandwidth between the Zenith nodes and the H600 cluster is approximately 120.83 Gbits/sec (or ~15 GB/sec).

## I/O Tests

IOR v3.0.1 is the preferred I/O benchmark in the Dell HPC Innovation Labs for several reasons:

- Developed by Lawrence Livermore National Laboratory to evaluate I/O performance
- IOR scales much better than lozone<sup>4</sup> for multiple clients
- Can be used to measure I/O performance via
  - multiple access patterns
  - different storage configurations
  - several file sizes
- It can be used to evaluate the performance of different parallel I/O interfaces
  - POSIX, MPI-IO, HDF5, and NETCDF
- IOR makes "apples to apples" comparisons with other filesystems possible

Before executing any *IOR* commands the *iotest* directory was first deleted, then recreated. Before and between any IOR test run, the NFS share was first unmounted then mounted on each compute node used in the tests. NFSv3 was used for all tests.

- An example mount command:

```
mount -o tcp,vers=3 -t nfs <NFS_IP>:/ifs/iotest /mnt/F800
```

- An example unmount command:

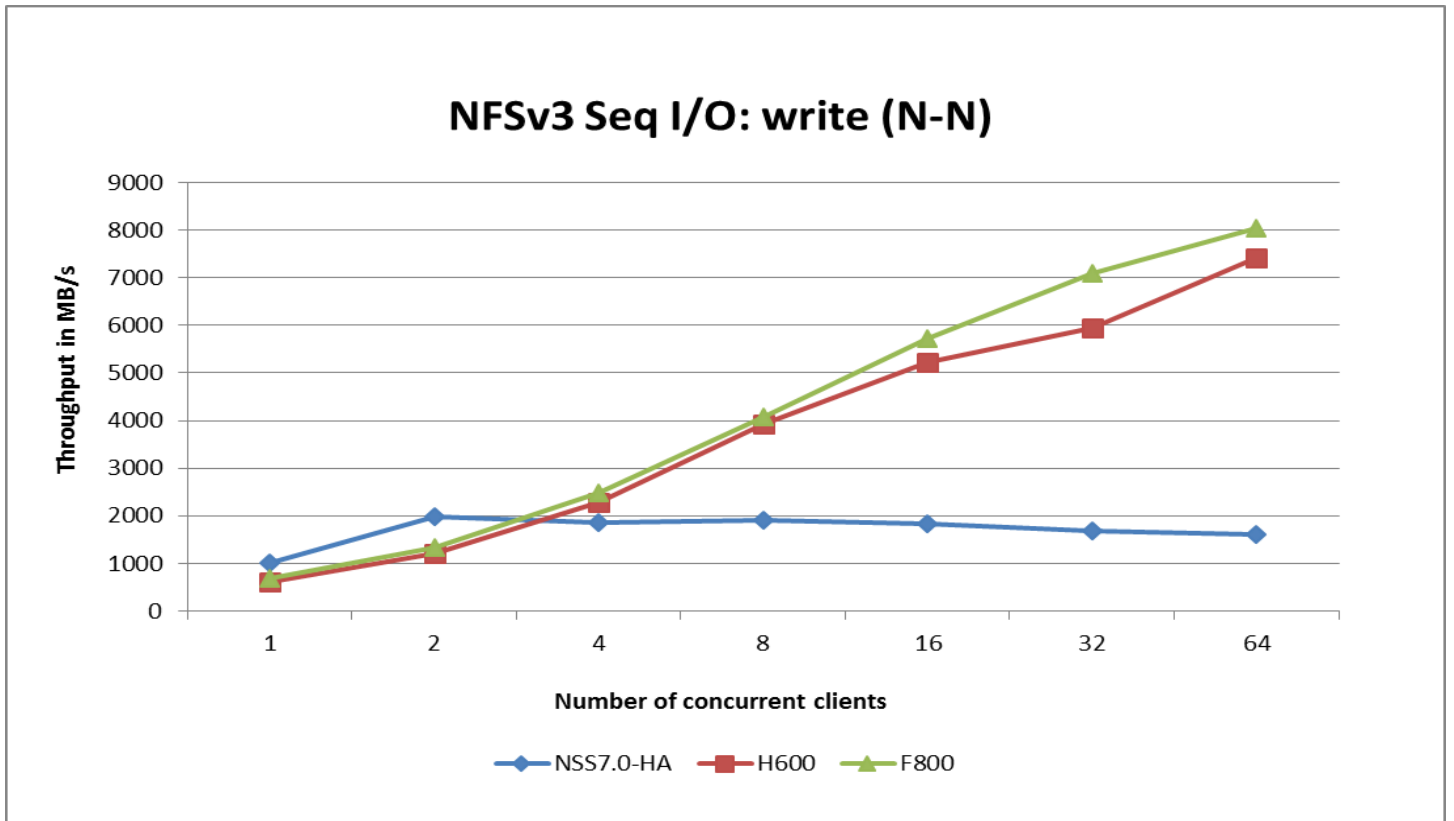
```
umount /mnt/F800
```

## Sequential I/O Performance Tests (N-to-N)

The F800 and H600 storage systems were benchmarked using the default OneFS configuration with enduring cache enabled, multi-writer enabled and coalescer disabled.

For sequential I/O tests with *IOR*, a block size of 1024 KB was used. There were 7 test cases (1-client, 2-client, 4-client, 8-client, 16-client, 32-client, and 64 client test cases). For each test *IOR* generated individual threads on all the compute nodes for reading/writing data and the total workload was 2 TB. For example, in the 1-client test case, one thread was generated on one compute node to read/write a 2 TB file. For the 2-client case, there are two compute nodes used in the test and each client node generated one thread to read/write a 1 TB file concurrently. The *IOR* commands used for these tests are listed in the Appendix. Figure 4 illustrates the sequential write performance.





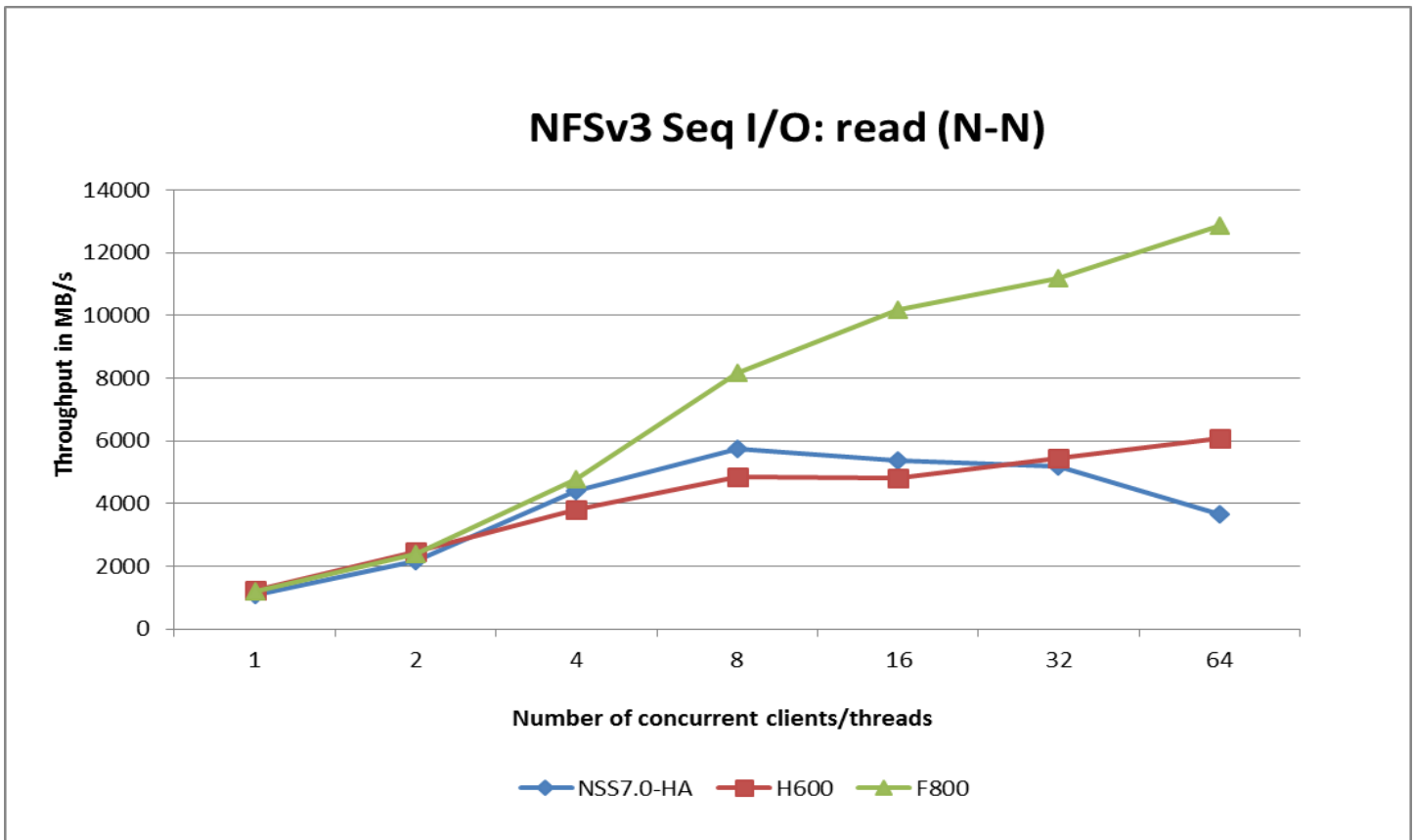
**Figure 4. Sequential write performance (N-N)**

Sequential write performance summary:

- Peak write performance of F800 was ~8GB/sec
- Peak write performance of H600 was ~7.4GB/sec
- Peak write performance of NSS-7.0-HA was ~2GB/sec
- Both F800 and H600 scale similarly

Future tests will utilize more than 64 client compute nodes in an attempt to maximize write throughput on the F800 and H600.

The IOR sequential read results for all three storage systems is shown in Figure 5.



**Figure 5. Sequential read performance (N-N)**

Sequential read performance summary:

- Peak read performance of F800 was ~12GB/sec
- Peak read performance of H600 was ~6GB/sec
- Peak read performance of NSS-7.0-HA was ~5.9GB/sec
- F800 scaled well

Future tests will utilize more than 64 client compute nodes in an attempt to maximize read throughput on the F800 and H600.

### Sequential I/O Performance Tests (N-to-1)

Similar to the sequential I/O N-to-N tests, there were also 7 test cases (1-client, 2-client, 4-client, 8-client, 16-client, 32-client, and 64 client test cases), the block size was 1024KB, and the total workload was 2TB. The difference between the two types of tests is that individual compute clients will read/write to separate files for the N-to-N tests, while individual clients will read/write to the same file for N-to-1 tests. For example, in the 2-client test case, the single file size is 2TB, one client node will read/write the first 1TB of the file and the other client will concurrently read/write the second 1TB of the file. All tests were run in the default configuration.

Figure 6 shows the *IOR* write performance for the N-to-1 tests.

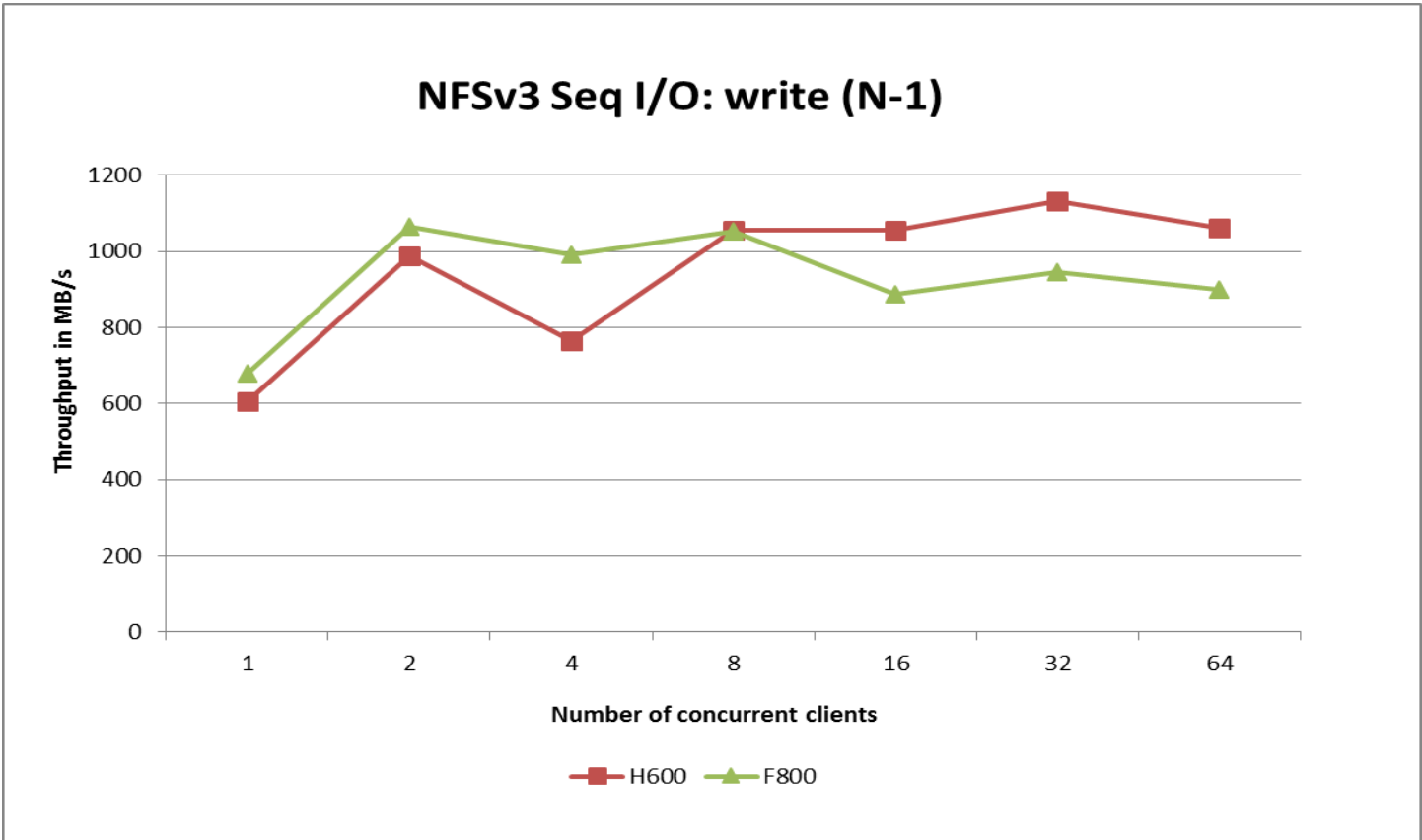


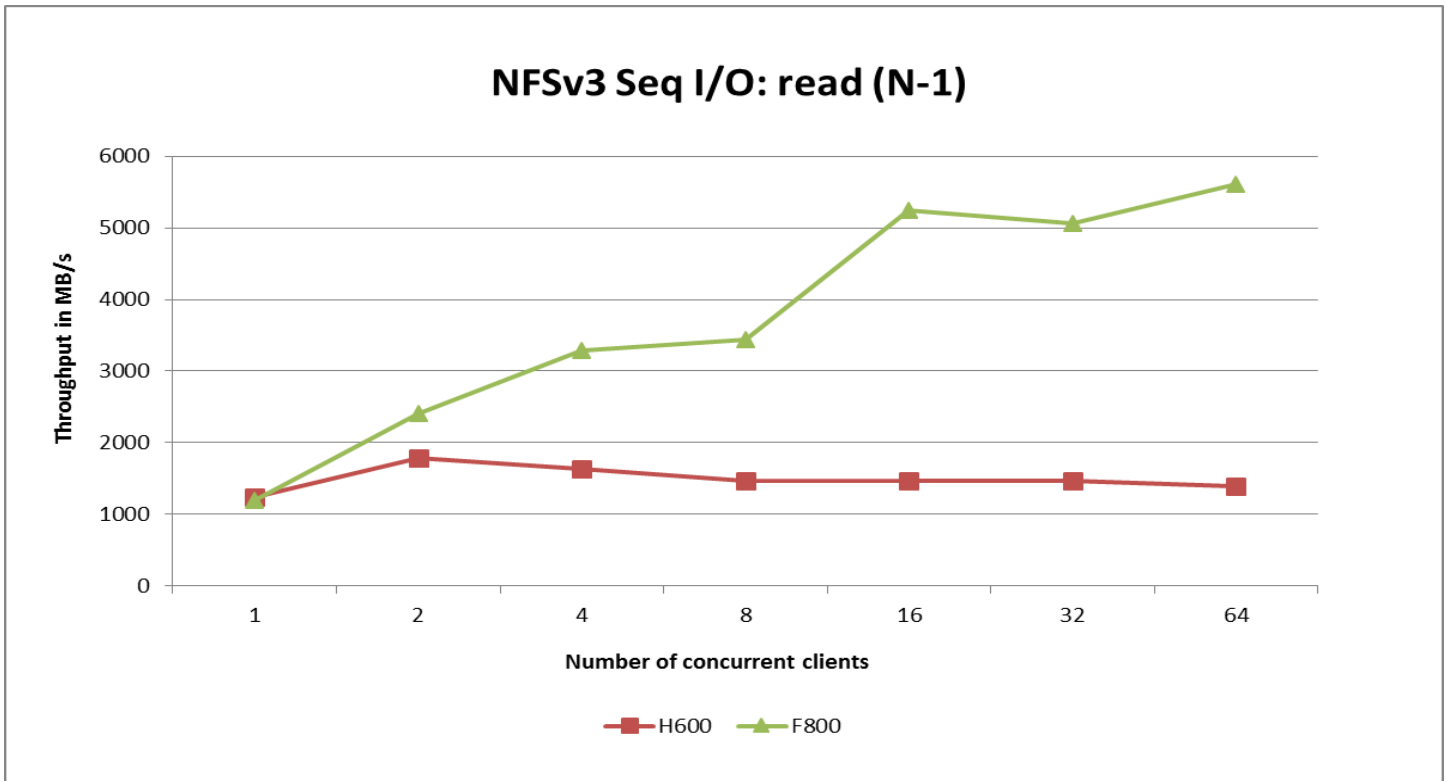
Figure 6. Sequential write performance (N-1)

Write performance for both the F800 and H600 peaked with the 2-client test case at approximately 1.1GB/sec and didn't change much as more compute nodes were added. It appears that file synchronization overhead limits write performance as an increasing number of client nodes attempt to concurrently write to the same file. In contrast, the N-to-1 read performance generally increases with an increase in client nodes and peak read performance was approximately 5.6 GB/sec (Figure 7). Future tests will utilize more than 64 client compute nodes in an attempt to maximize N-to-1 write throughput.

Sequential write performance summary:

- Peak write performance of F800 was ~1 GB/sec
- Peak write performance of H600 was ~1.1GB/sec
- Unlike sequential N-to-N write performance, sequential N-to-1 write performance of both H600 and F800 did not increase with more than 2 clients

Figure 7 shows the IOR read performance for the N-to-1 tests.



**Figure 7. Sequential read performance (N-1)**

Sequential read performance summary:

- Peak read performance of F800 was ~5.6 GB/sec
- Peak read performance of H600 was ~2 GB/sec
- Read performance of H600 did not increase with more than 2 clients

Future tests will utilize more than 64 client compute nodes in an attempt to maximize read throughput on the F800.

### Random I/O Performance Tests

Similar to the sequential *IOR* tests, there are 7 random *IOR* test cases (1-client, 2-client, 4-client, 8-client, 16-client, 32-client, and 64 client test cases), but the random F800 and H600 tests in Figure 8A used a 4 KB block size and each client writes a 32 GB file. The random H600 and NSS-7.0-HA tests in Figure 8B used a 4 KB block size and each client writes an 8 GB file. The F800 configuration was changed to enable better random I/O performance by executing the following OneFS commands affecting endurant cache, coalescer and multi-writer<sup>6</sup>:

- o `isi_sysctl_cluster efs.bam.ec.mode=0`
- o `isi nfs settings export modify --setattr-async=1 --commit-async=1`
- o `isi_sysctl_cluster efs.bam.coalescer.mutiwriter=0`
- o `isi set -c coal_only /ifs/iotest`



Figure 8A. Random write performance (N-N)

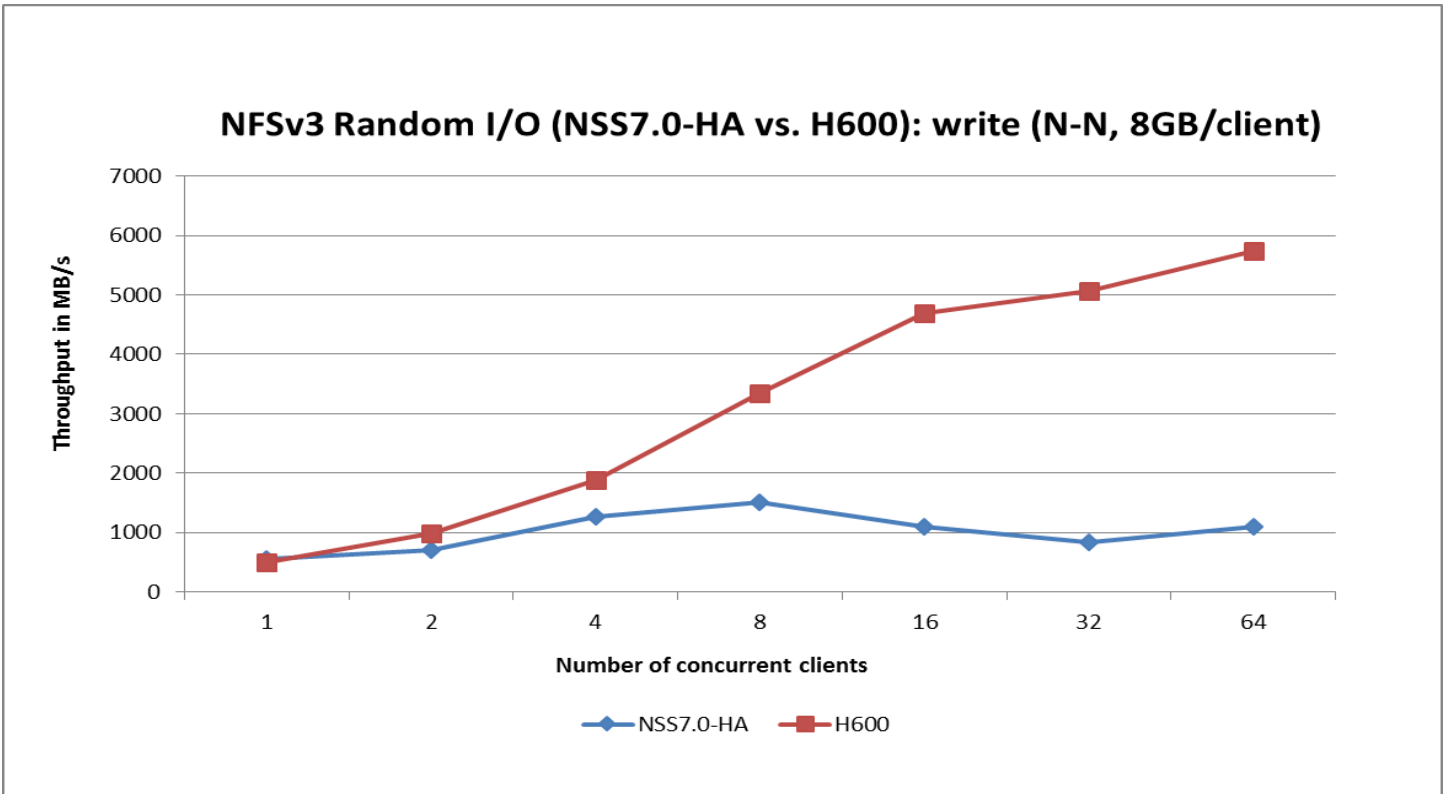


Figure 8B. Random write performance (N-N)

Random write performance summary:

- Peak write performance of F800 was ~1.6 GB/sec

- Peak write performance of H600 was ~800 MB/sec (32 GB file)
- Peak write performance of H600 was ~5.8 GB/sec (8 GB file)
- Peak write performance of NSS-7.0-HA was ~1.5 GB/sec
- H600 performed better with an 8 GB file vs. a 32 GB file.

Future tests will utilize more than 64 client compute nodes in an attempt to maximize write throughput on the F800 and H600.

The random F800 and H600 tests in Figure 9A used a 4 KB block size and each client reads a 32 GB file. The random H600 and NSS-7.0-HA tests in Figure 9B used a 4 KB block size and each client writes an 8 GB file.

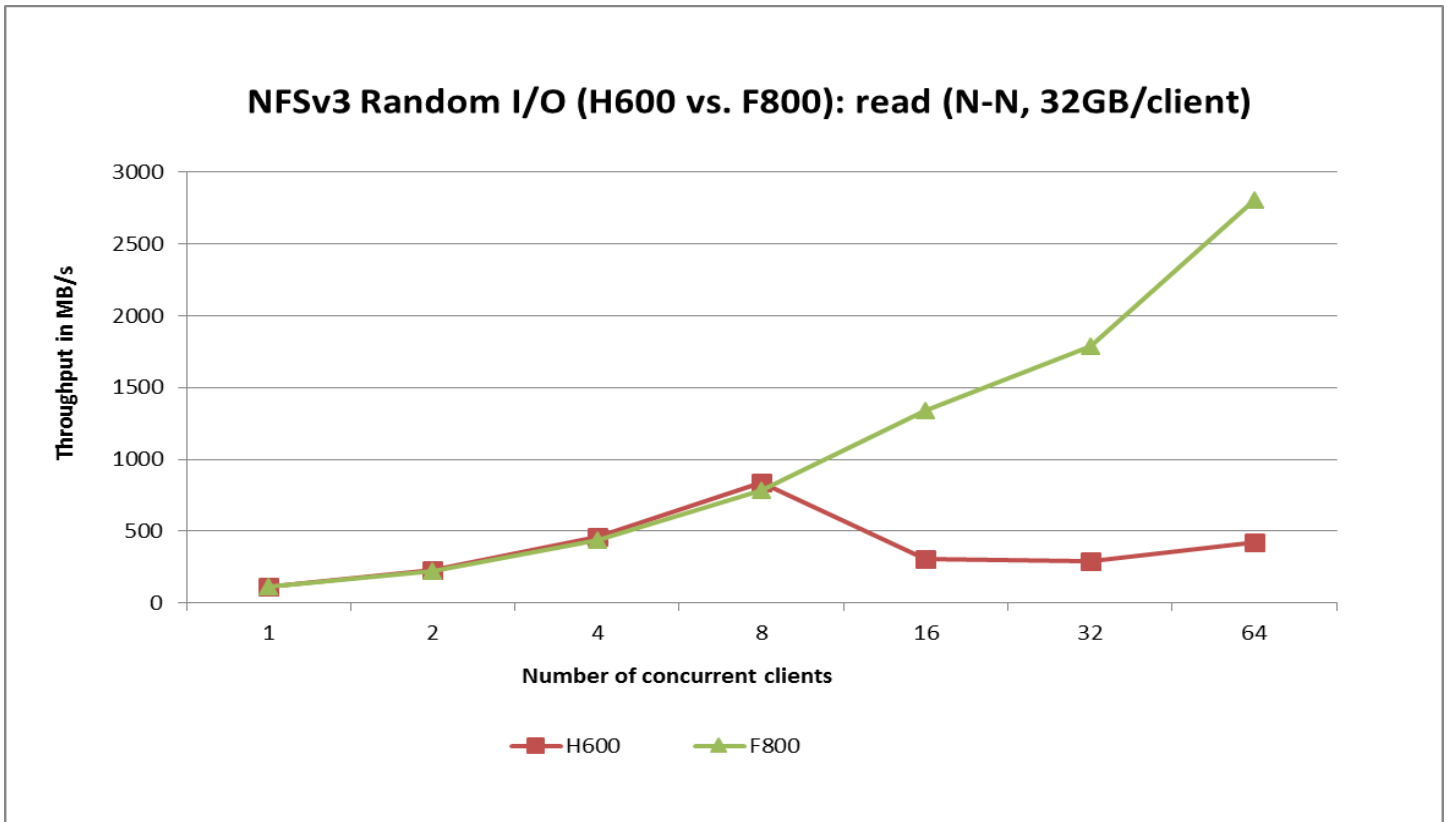
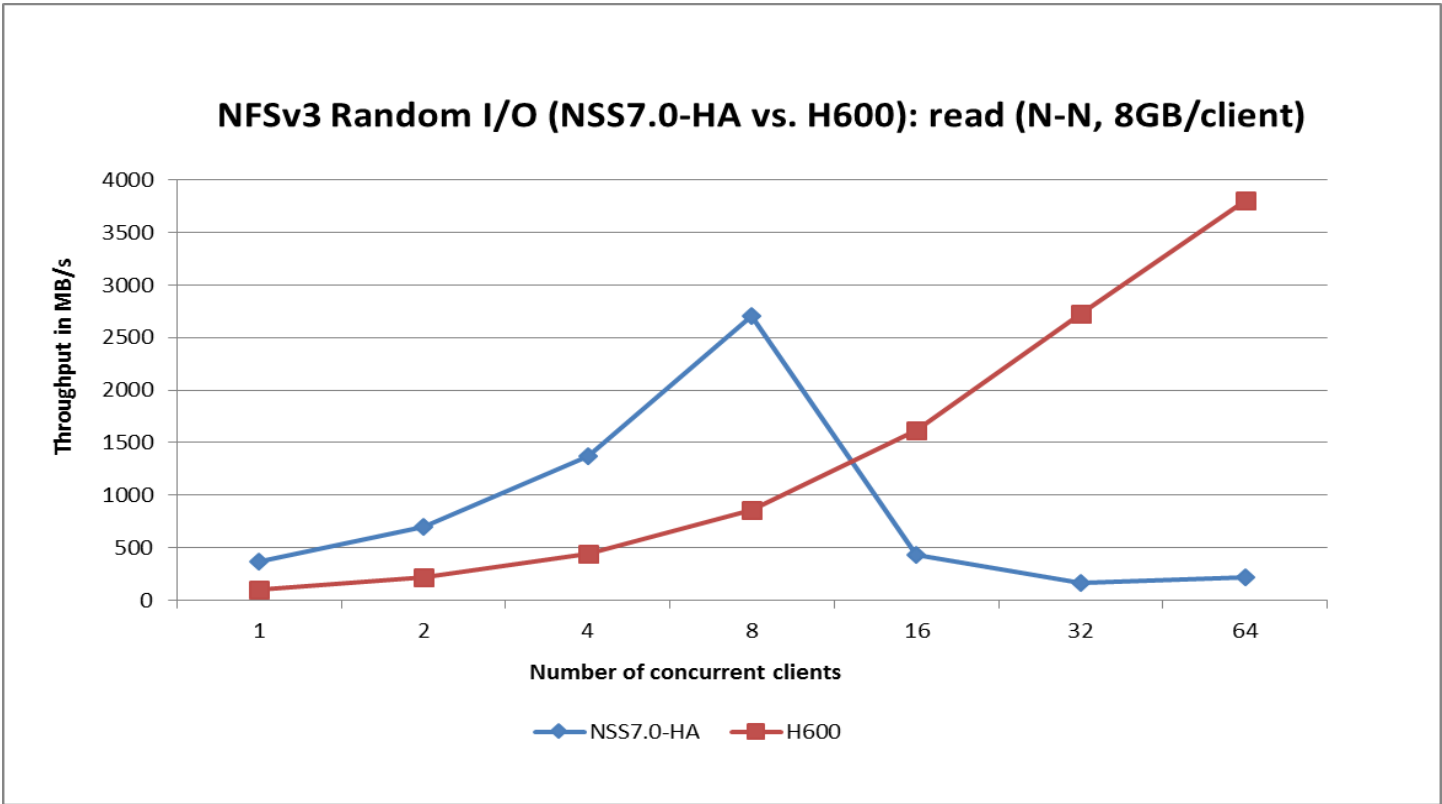


Figure 9A. Random read performance (N-N)



**Figure 9B. Random read performance (N-N)**

Random read performance summary:

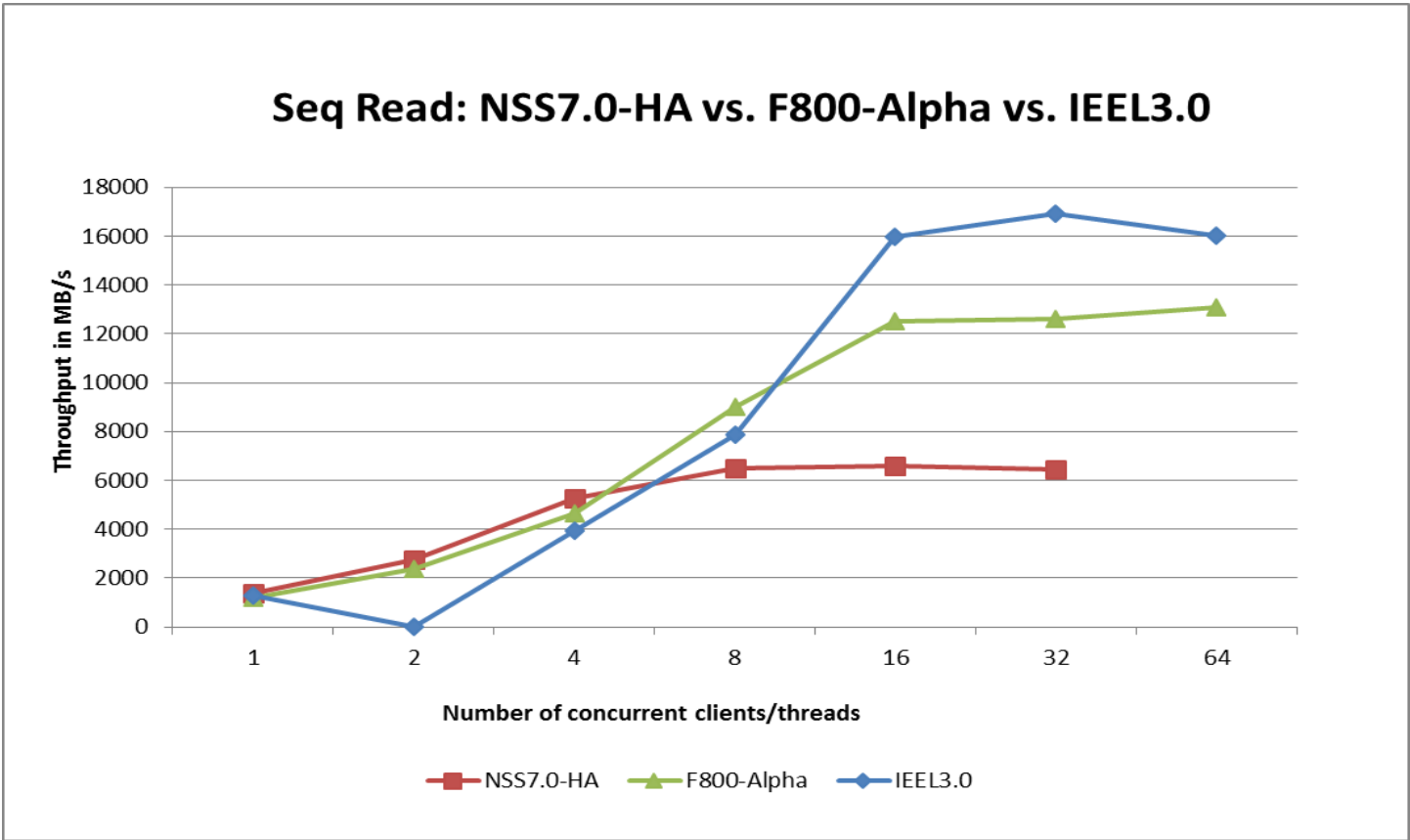
- Peak read performance of F800 was ~2.7 GB/sec
- Peak read performance of H600 was ~800 MB/sec (32 GB file)
- Peak read performance of H600 was ~3.7 GB/sec (8 GB file)
- Peak read performance of NSS-7.0-HA was ~2.6 GB/sec
- H600 performed better with an 8 GB file vs. a 32 GB file.

Future tests will utilize more than 64 client compute nodes in an attempt to maximize read throughput on the F800 and H600.

**I/O Performance Comparisons: F800 vs. Lustre vs. NFS**

*IOzone* has been used in the HPC Innovation Lab to benchmark the Dell HPC Lustre Storage with Intel Enterprise Edition Lustre filesystem<sup>5</sup> (IEEL3.0). IEEL3.0 is based on Lustre 2.7.15.3. While there are several important physical differences between the three filesystems, e.g. networking fabric, *IOzone* results run from similar test cases to the *IOR* runs can still illustrate approximate performance comparisons between the three storage systems.

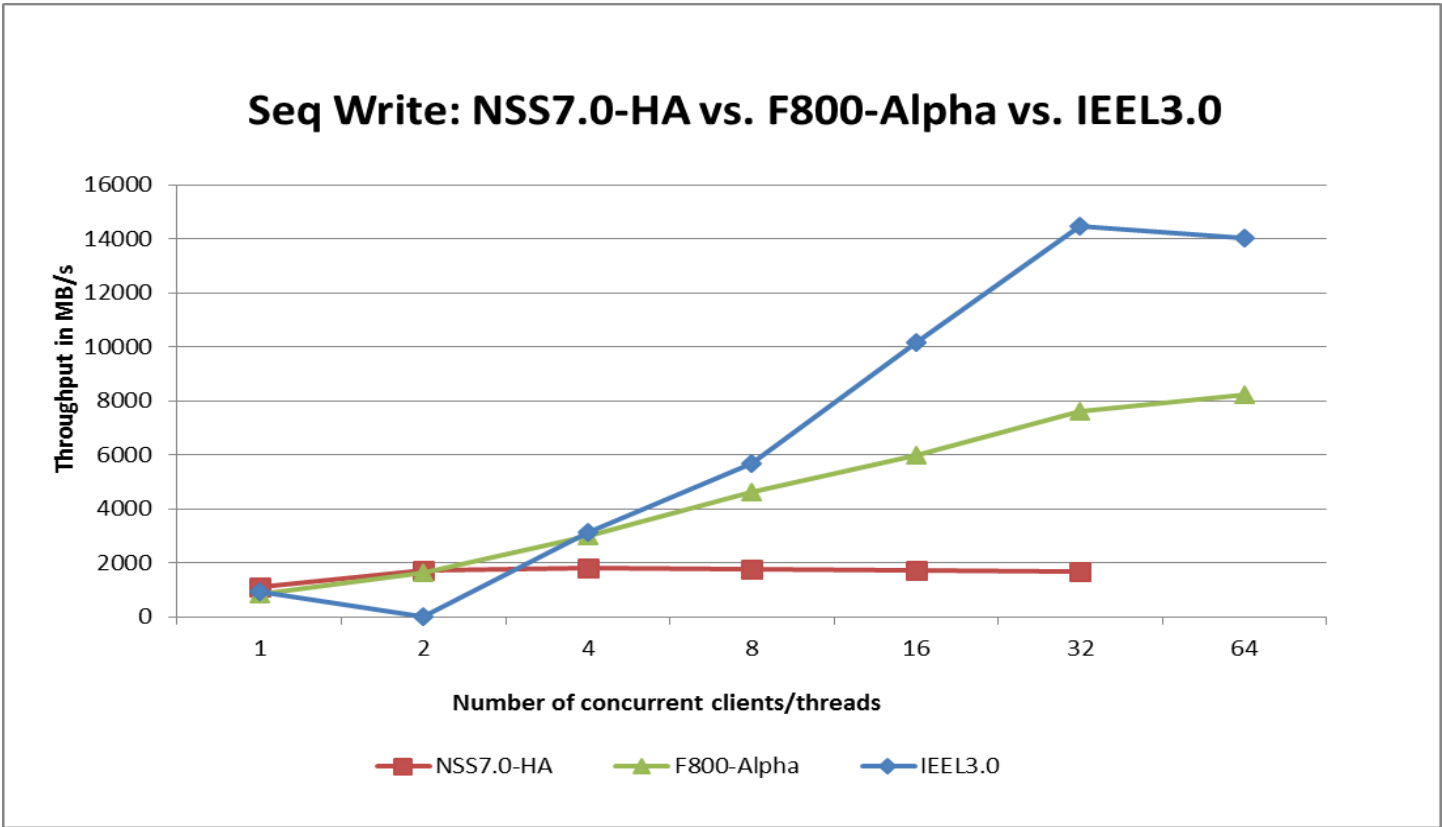
Figure 10 illustrates that sequential read performance on all three filesystems is similar up to the 8-client case. The F800 actually performs slightly better than the Lustre filesystem. As more clients are added, the Lustre filesystem improves to a peak of approximately 17GB/sec while the F800 peaks at 13 GB/sec and the NSS-7.0-HA peaks at approximately 6.2 GB/sec.



**Figure 10. Sequential Read Performance**

Figure 11 illustrates that sequential write performance on all three filesystems is similar up to the 2-client case, where the NSS peaks at approximately 1.9 GB/sec. As more clients are added, the Lustre filesystem improves to a peak of approximately 14.2 GB/sec while the F800 peaks at 8 GB/sec.





**Figure 11. Sequential Write Performance**

Figures 12 and 13 illustrate that F800 random read and write performance is better than Lustre and NSS for every test case. F800 random read performance is 7x greater than Lustre (140K vs 20K IOPS) at the 64-client case, while F800 random write performance is 3x greater than Lustre (45K vs 15K IOPS). Future tests will utilize more than 64 client compute nodes in an attempt to maximize F800 random read/write performance.

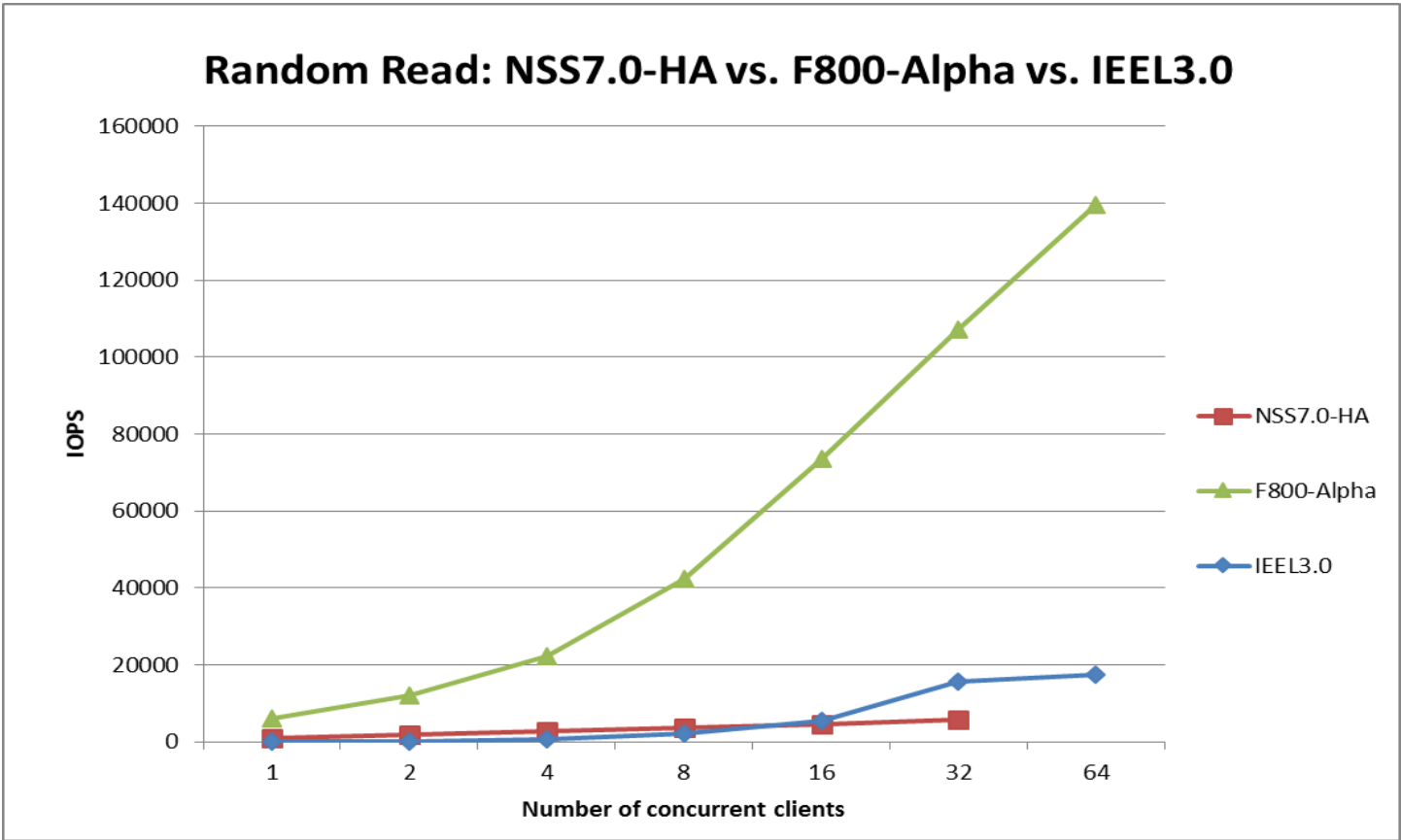


Figure 12. Random Read Performance

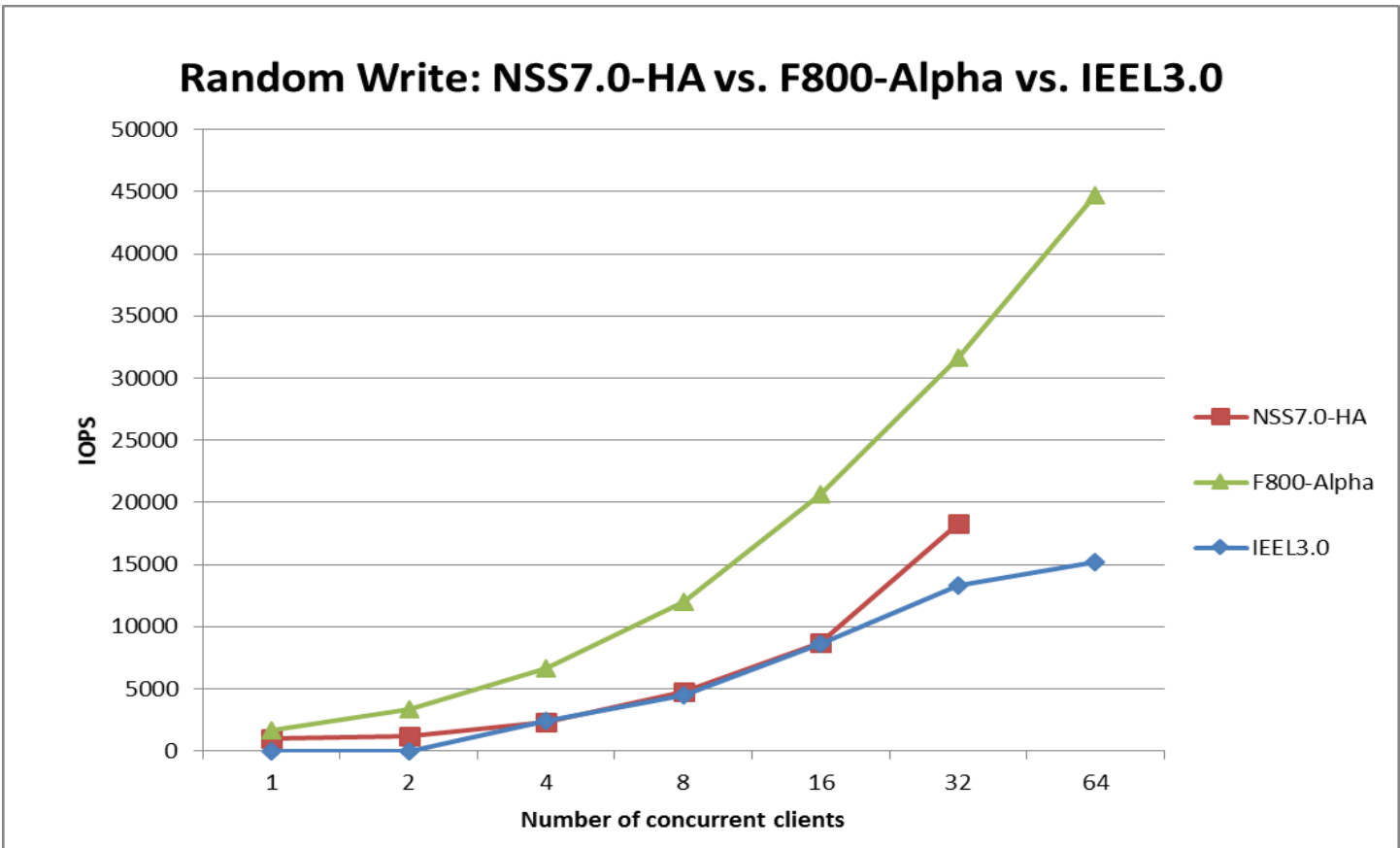


Figure 13. Random Write Performance

## Summary

- Sequential I/O Performance
  - N-to-N tests
    - F800 had better sequential read and write performance than NSS-7.0-HA and H600
    - F800 and H600 write performance up to 400% better than NSS-7.0-HA
    - F800 write performance up to 20% better than H600
    - F800 read performance up to 251% better than NSS-7.0-HA
    - F800 read performance up to 111% better than H600
    - H600 and NSS-7.0-HA had similar sequential read performance
  - N-to-1 tests
    - For write tests, both F800 and H600 had similar performance behavior
    - For read tests, F800 had ~300% better performance than H600
- Random I/O Performance (N-to-N)
  - F800 had up to 250% better random write performance than H600
  - H600 had up to 571% better random write performance than NSS-7.0-HA
  - F800 had up to 550% better random read performance than H600
  - H600 had up to 15x better random read performance than NSS-7.0-HA at 64-clients
  - F800 had 7x better random read performance than Lustre at 64-clients
  - F800 had 3x better random write performance than Lustre at 64-clients

The F800 and H600 consistently outperformed the NSS-7.0-HA and demonstrated good I/O performance scalability with both sequential and random workloads. Furthermore, the comparison between Lustre, NSS, and the F800 demonstrated that the F800 is the best choice for intensive random I/O workloads while Lustre was superior at sequential intensive workloads. While Lustre sequential I/O performance was better than the F800 by 25-75%, the F800 random I/O performance was 300-700% better than Lustre. So a case can be made that the F800 is the best overall choice for a high performance file system, particularly if the workload has a significant random I/O component. If features like backup, snapshots and multiple protocol (SMB/NFS/HDFS) support are required in addition to a mixed HPC workload, then Isilon is a better choice than Lustre or the NSS-7.0-HA. However, if the HPC workload is mixed and includes MPI-based or other applications that require low latency interconnects (Infiniband or OmniPath) in order to scale well, then Lustre is the better choice.

## REFERENCES

1. All benchmark tests were run on the Zenith cluster in the Dell HPC Innovation Lab in Round Rock, TX. Zenith ranked #292 in the Top 500 ranking as of November 2017: <https://top500.org/list/2017/11/?page=3>
2. Iperf: <https://sourceforge.net/projects/iperf/>
3. IOR: <https://github.com/LLNL/ior>
4. IOzone: <http://www.iozone.org/>
5. Dell HPC Lustre Storage Solution: [Dell HPC Lustre Storage with IEEL 3.0](#)
6. Stable Writes: <https://community.emc.com/community/products/isilon/blog/2016/10/18/stable-writes>; Accessed 1/24/18

## APPENDIX

*iperf* commands:

# On each F800 node the *iperf* command was:

```
iperf -s -w 2M -l 1M
```

# On each Zenith client node the *iperf* command was:

```
iperf -c <F800_IP> -w 2M -l 1M -N -t 30
```

*IOR* sequential write (N-to-N) commands:

#1-client write

```
mpirun --allow-run-as-root -np 1 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -w -s 1 -t 1m -b 2048g
```

#2-client write

```
mpirun --allow-run-as-root -np 2 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -w -s 1 -t 1m -b 1024g
```

#4-client write

```
mpirun --allow-run-as-root -np 4 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -w -s 1 -t 1m -b 512g
```

...

#64-client write

```
mpirun --allow-run-as-root -np 64 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -w -s 1 -t 1m -b 32g
```

*IOR* sequential read (N-to-N) commands:

#1-client read

```
mpirun --allow-run-as-root -np 1 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -r -s 1 -t 1m -b 2048g
```

#2-client read

```
mpirun --allow-run-as-root -np 2 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -r -s 1 -t 1m -b 1024g
```

#4-client read

```
mpirun --allow-run-as-root -np 4 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -r -s 1 -t 1m -b 512g
```

...

#64-client read

```
mpirun --allow-run-as-root -np 64 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -r -s 1 -t 1m -b 32g
```

*IOR* sequential writes (N-to-1):

#1-client write

```
mpirun --allow-run-as-root -np 1 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -w -s 1 -t 1m -b 2048g
```

#2-client write

```
mpirun --allow-run-as-root -np 2 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -w -s 1 -t 1m -b 1024g
```

#4-client write

```
mpirun --allow-run-as-root -np 4 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -w -s 1 -t 1m -b 512g
```

...

#64-client write

```
mpirun --allow-run-as-root -np 64 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -w -s 1 -t 1m -b 32g
```

*IOR* sequential reads (N-to-1):

#1-client read

```
mpirun --allow-run-as-root -np 1 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -r -s 1 -t 1m -b 2048g
```

#2-client read

```
mpirun --allow-run-as-root -np 2 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -r -s 1 -t 1m -b 1024g
```

#4-client read

```
mpirun --allow-run-as-root -np 4 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -r -s 1 -t 1m -b 512g
```

...

#64-client read

```
mpirun --allow-run-as-root -np 64 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -r -s 1 -t 1m -b 32g
```

*IOR* random reads and writes (N-to-N):

#1-client test case

```
mpirun --allow-run-as-root -np 1 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -w -r -s 1 -z -t 4k -b 32g
```

#2-client test case

```
mpirun --allow-run-as-root -np 2 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -w -r -s 1 -z -t 4k -b 32g
```

#4-client test case

```
mpirun --allow-run-as-root -np 4 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -w -r -s 1 -z -t 4k -b 32g
```

...

#64-client test case

```
mpirun --allow-run-as-root -np 64 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -F -k -o /mnt/nfs/test -w -r -s 1 -z -t 4k -b 32g
```

*IOR* random commands reads and writes (N-to-1):

#1-client test case

```
mpirun --allow-run-as-root -np 1 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -w -r -s 1 -z -t 4k -b 1g
```

#2-client test case

```
mpirun --allow-run-as-root -np 2 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -w -r -s 1 -z -t 4k -b 1g
```

#4-client test case

```
mpirun --allow-run-as-root -np 4 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -w -r -s 1 -z -t 4k -b 1g
```

...

#64-client test case

```
mpirun --allow-run-as-root -np 64 -npernode 1 -hostfile hosts -nolocal /home/xin/bin/ior -a POSIX -v -i 1 -d 3 -e -k -o /mnt/nfs/test -w -r -s 1 -z -t 4k -b 1g
```