NetApp Verified Architecture

Quantum StorNext with NetApp E-Series Systems Deployment Guide

NVA Deployment

Ryan Rodine, NetApp
January 2021 | NVA-1150-DEPLOY | Version 1.2

Abstract

This document provides details on how to deploy a StorNext parallel file system solution with NetApp® E-Series storage systems. This solution covers the NetApp EF280 all-flash array, the NetApp EF300 all-flash NVMe array, the NetApp EF600 all-flash NVMe array, and the NetApp E5760 hybrid system. It offers performance characterization based on Frametest benchmarking, a tool that is widely used for testing in the media and entertainment industry.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Solution overview

NetApp® E-Series plus a file system such as StorNext dramatically streamlines workflow and improves productivity. The combination creates a shared repository that supports flexible, high-performance streaming, even with high-bit-rate media content. This repository also includes:

- **A single namespace,** for virtually limitless bandwidth or capacity
- **Near-linear bandwidth scalability,** for both scale-up and scale-out configurations
- **Support for Linux, Microsoft Windows, and macOS clients**

Growing media and entertainment companies are challenged to find storage tier solutions that both satisfy their high-density, high-bandwidth requirements and optimize rack space power and cooling. NetApp EF280, EF300, EF600, and E5760 systems fulfill these roles.

The architecture that is demonstrated in this deployment guide shows the capabilities of the four options. First, an entry level all-flash option, using a single NetApp EF280 all-flash array. Second, a medium-tier option, using a single EF300 array with high performing NVMe drives. Third, a high-performance building block using a single NetApp EF600 all-flash NVMe array also with high performing NVMe drives. Fourth, a NetApp E5760 system using cost-effective SAS HDD drives. You can add additional similarly configured arrays to the StorNext namespace to allow virtually unlimited scale-out.

Each option also demonstrates value. The EF280 array is a low-cost entry into an all-flash storage array. The EF300 and EF600 have the ability to support a high number of high-resolution video streams with no dropped frames. The E5760 provides a lower-cost, large-capacity target in 8U of rack space. All options save the customer in footprint and power/cooling costs.

In addition, the EF300 and EF600 arrays support NVMe protocols, enabling the potential to upgrade to NVMe over Fabrics in the future.

Solution technology

The StorNext solution in this design guide consists of at least four StorNext clients. Each configuration runs the same tests but with a different target storage array in each case. All configurations use a StorNext metadata controller (MDC) with separate storage for the metadata and the journal volumes. In this example, an EF570 all-flash array is used for the MDC, but a StorNext MDC appliance can be used for the metadata and journal, as well.

The SAN consists of a 32Gb FC fabric, facilitated with ATTO CTFC-324E adapters running ATTO MultiPath Director on the clients.

The EF280 array has 24 flash drives and is provisioned into two 10+2 RAID 6 volume groups.

The EF300 and EF600 options each have 24 NVMe drives and are provisioned into two 10+2 RAID 6 volume groups.

The E5760 array has 120 SAS HDDs and is provisioned into twelve 8+2 RAID 6 volume groups.

Figure 1 through Figure 4 show the technical components of the EF280, EF300, EF600, and E5760 solutions.

Figure 1) Technical components of the solution with an EF280 array.

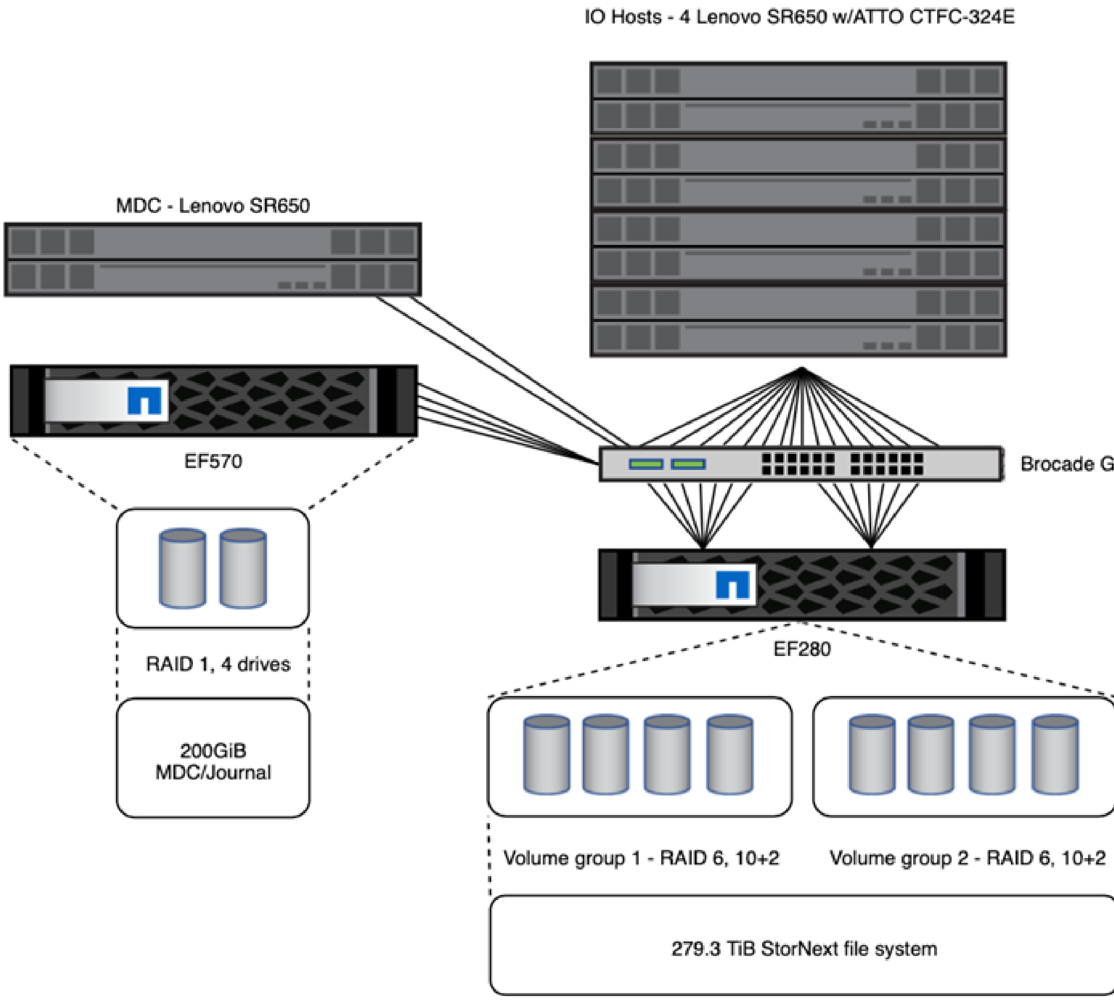


Figure 2) Technical components of the solution with an EF300 array.

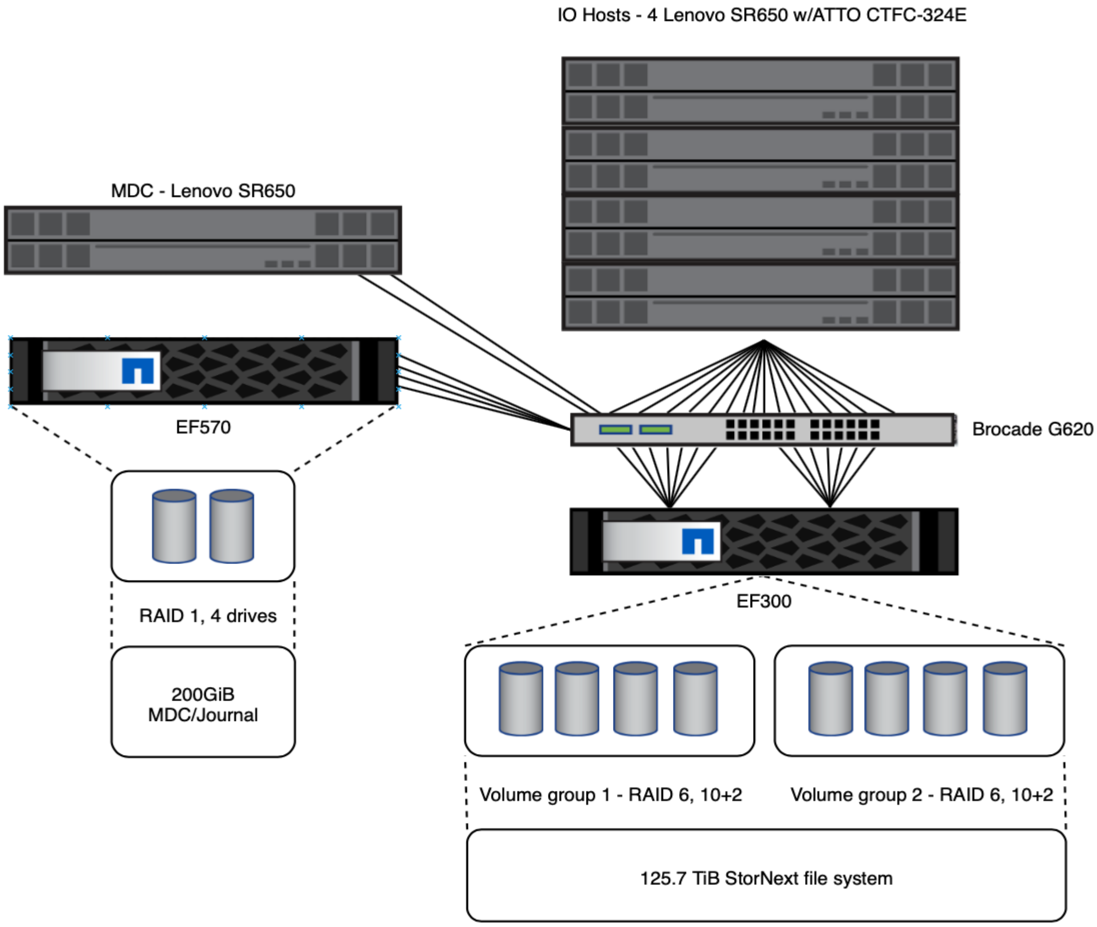


Figure 3) Technical components of the solution with an EF600 array.

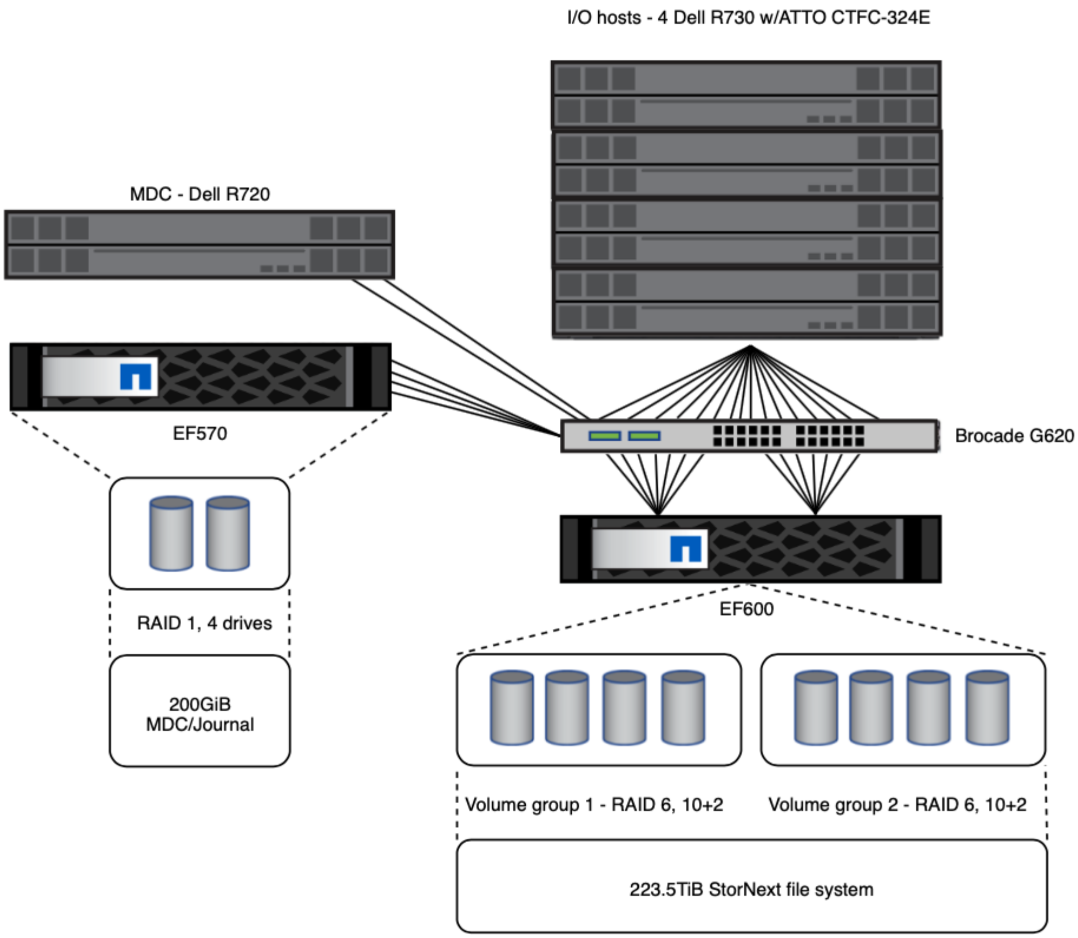
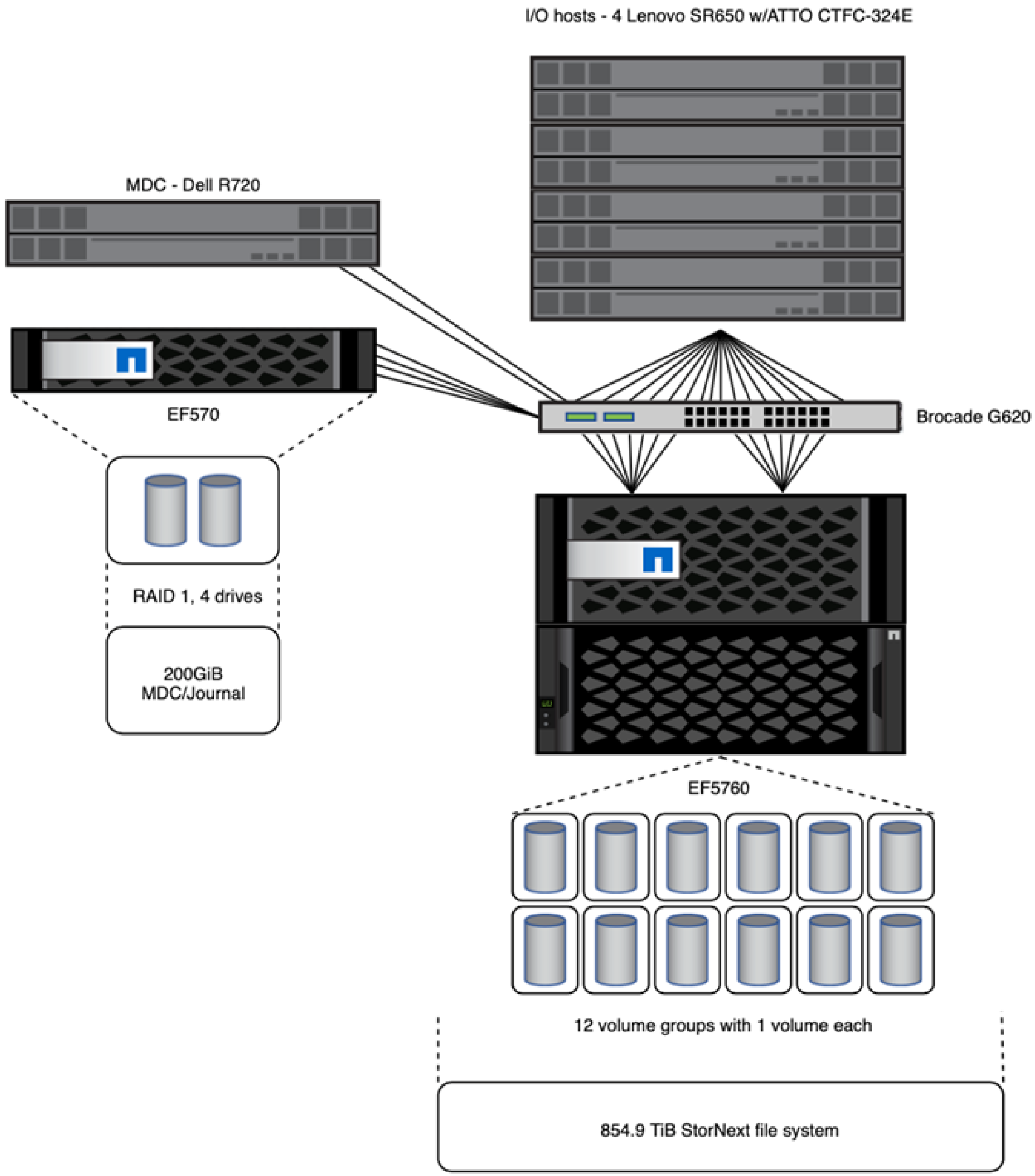


Figure 4) Technical components of the solution with an E5760 system.



Use case summary

This solution applies to the following use cases in a StorNext environment:

- Ingest and playout of high-bandwidth media streams
- The need for either a high-bandwidth streaming tier or a lower-cost, high-capacity streaming tier
- High-resolution media asset management
- Online media archiving
- Centralized storage to provision content creation workstations and to broadcast playout workstations

High-performance and cost-effective video streaming

Ingest and playout of media streams are the primary use cases for this solution. The idea is that the NetApp EF280 array can offer an entry into an all-flash array. The NetApp EF300 is a middle-tier entry. The EF600 array can fill the role of an extreme high-performance tier, and the E5760 system is a cost-effective, high-capacity option in a StorNext environment. The goal of this example solution is to prove that idea and to determine the maximum number of uncompressed 4K streams that the storage arrays can support without dropping frames.

Technology requirements

Hardware requirements

Servers for the NetApp EF600 array: One MDC and four Dell R730 with ATTO FC adapters are used:

- Two Intel Xeon E5-2670 v3 processors @ 2.3GHz
- 128GB RAM
- Client host bus adapters (HBAs): ATTO CTFC-324E
- MDC HBA: Broadcom LPe32002

Servers for the NetApp EF280, EF300, and NetApp E5760 configurations: One MDC and four Lenovo SR650 with ATTO FC adapters are used:

- Two Intel Xeon Gold 6136 CPUs @ 3.00GHz
- 96GB RAM
- Client HBAs: ATTO CTFC-324E
- MDC HBA: Broadcom LPe32002

Storage: NetApp E-Series arrays:

- Data volumes on the EF280 array:
  - 24 MZILS15THMLS-0G4 - 15TB SSD drives
  - Two 10+2 RAID 6 volume groups
  - Four 34.9TiB volumes per volume group with 512KiB segment size
- Data volumes on the EF300 array:
  - 24 MZWLJ7T6HALA-0G5 - 7TB NVMe drives
  - Two 10+2 RAID 6 volume groups
  - Four 15.7TiB volumes per volume group with 512KiB segment size
- Data volumes on the EF600 array:
  - 24 MZWLL15THMLA-0G5 15.3TB NVMe drives
  - Two 10+2 RAID 6 volume groups
  - Four 27.9TiB volumes per volume group with 512KiB segment size (capacity is 20% overprovisioned per NVMe drive guidelines)
- Data volumes on the E5760 system:
  - 120 ST10000NM0096 10TB 7,200 RPM SAS drives
  - Twelve 8+2 RAID 6 volume groups
  - Twelve 71.2TiB volumes per volume group with 512KiB segment size
- MDC volumes on EF570:
  - Four PX02SMU080 800GB SSDs

- One 4-drive RAID 10 volume group
- Two 200GiB volumes with 128KiB segment size

Switching: Brocade G620

Software requirements

This design includes StorNext file system version 6.1.0. The Quantum StorNext file system is a versatile, high-performance shared file system that offers heterogeneous, block-based access to a single storage system or striped across multiple external storage systems.

This example consists of tests that were performed on separate EF280, EF600, and E5760 storage targets running NetApp SANtricity® OS 11.60. The EF300 array was running SANtricity OS version 11.70.

The clients in this deployment run Red Hat Enterprise Linux (RHEL) 7.7, with ATTO MultiPath Director for failover support; however, a wide variety of host operating systems are supported. For a full list of supported operating system combinations, consult the NetApp Interoperability Matrix Tool.

Table 1 lists the software components that are required to implement the solution. The software components that are used in any particular implementation of the solution might vary based on customer requirements.

Table 1) Software requirements.

Software	Version or other information
StorNext	6.1.0
SANtricity	11.60 and 11.70
MDC OS	RHEL 7.5
Client OS	RHEL 7.7
ATTO MultiPath Director	Driver 1.76 MP; firmware, March 26, 2019

Deployment procedures

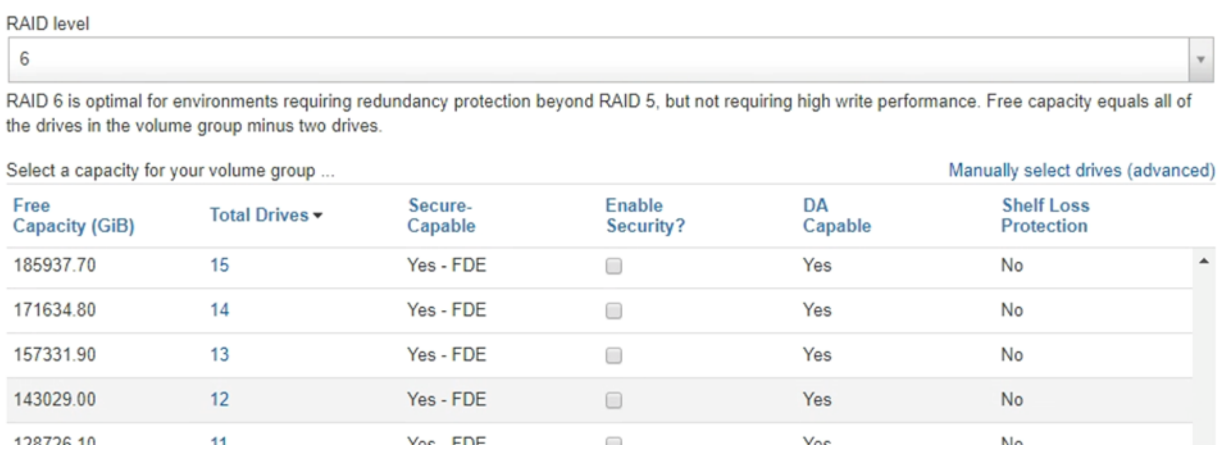
Deploying the solution involves the following tasks:

1. Configure the NetApp E-Series array.
2. Configure the hosts and switch zoning.
3. Configure the StorNext file system.
4. Carry out performance tuning on the hosts.
5. Connect clients to the file system and run the Frametest benchmark.

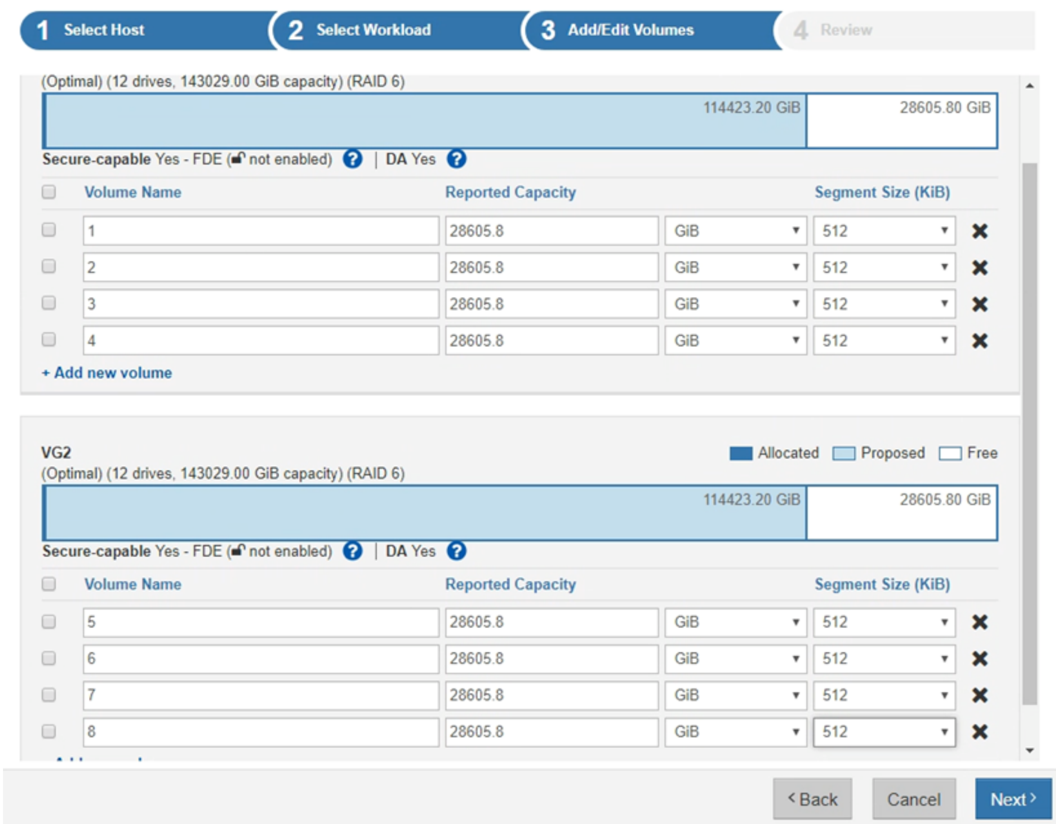
Volume configuration on the EF280, EF300, or EF600 array

The configuration for the EF280, EF300, and EF600 arrays is similar. For either one, complete the following steps:

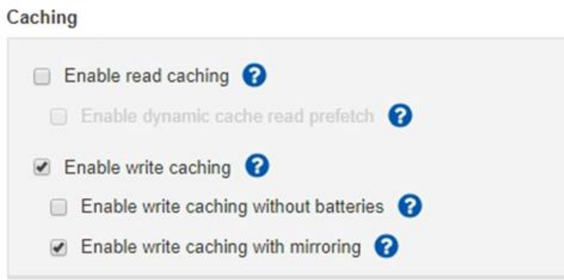
1. Create two RAID 6 volume groups with 12 drives each. You can select the auto provisioned quantity from the list that appears in the volume group creation window.



2. Create four volumes within each volume group. Select StorNext File System for the workload. The EF280 can use all of the available capacity within the volume group. Because the EF600 array contains NVMe drives, we overprovisioned the volume groups in this solution by 20%. In this example with a maximum capacity of 143029GiB, subtract 20% and divide by the number of volumes (four) to arrive at 28605.8GiB per volume. Note that as of SANtricity OS 11.70, which the EF300 was released with, a portion of the capacity will be reserved automatically, and this manual calculation is not required. Set the segment size to 512. The example below is from the EF600 configuration.



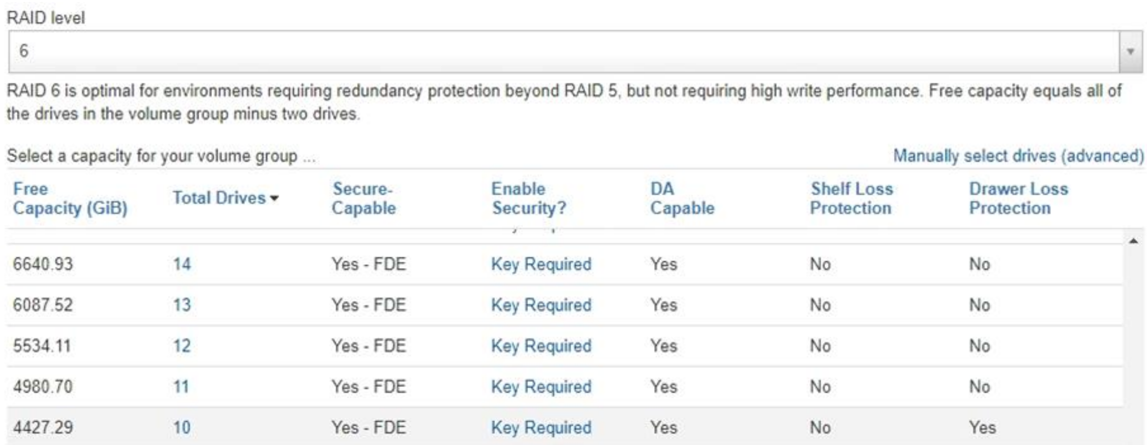
3. Disable read caching on each volume. Given the high performance of the drives, a read cache is not required, and disabling it allocates more cache to be available for writes. Access this setting in Volumes > [select volume] > View/Edit Settings > Advanced tab.



Volume configuration on the E5760 system

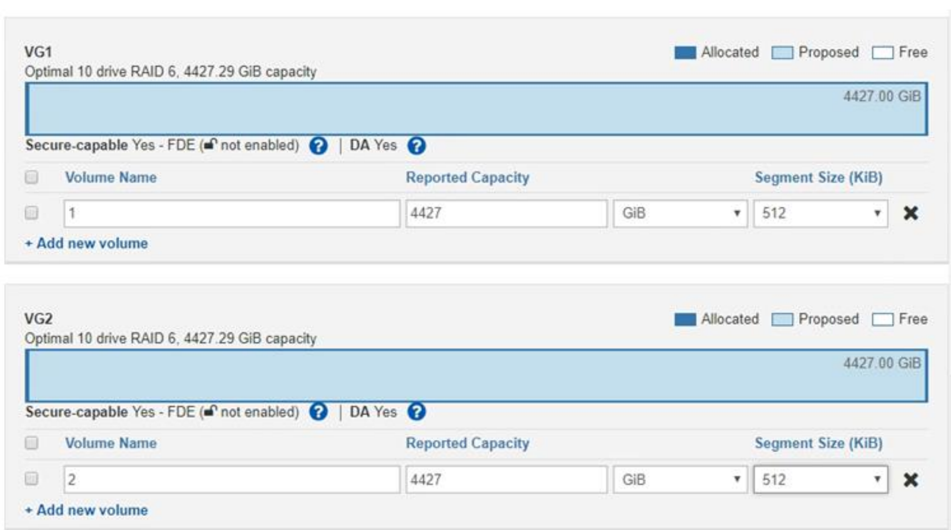
To configure the E5760 system, complete the following steps:

1. Create enough 8+2 RAID 6 volume groups to occupy all the available disks. This example has 120 drives, so you create 12 volume groups. You can select the auto provisioned quantity from the list that appears in the volume group creation window.



Free Capacity (GiB)	Total Drives	Secure-Capable	Enable Security?	DA Capable	Shelf Loss Protection	Drawer Loss Protection
6640.93	14	Yes - FDE	Key Required	Yes	No	No
6087.52	13	Yes - FDE	Key Required	Yes	No	No
5534.11	12	Yes - FDE	Key Required	Yes	No	No
4980.70	11	Yes - FDE	Key Required	Yes	No	No
4427.29	10	Yes - FDE	Key Required	Yes	No	Yes

2. Create one volume per volume group, using the full capacity of the volume group. Select StorNext File System for the workload. Set the segment size to 512.



3. Leave the cache settings at their default.

Configure the hosts and switch zoning

To configure the hosts and the switch zoning, complete the following steps:

1. Ensure that the ATTO MultiPath Director package is installed on each client. To ensure that your specific host combination is listed, check the [NetApp Interoperability Matrix Tool](), which also includes notes about any additional configuration that might be required.
2. Configure the zoning on the FC switch or switches so that the clients and the StorNext MDC are zoned to the E-Series array.

   Tip: When you configure the switch zoning, you can use the ATTO Configuration Tool to see which worldwide port name (WWPN) is associated to each host.

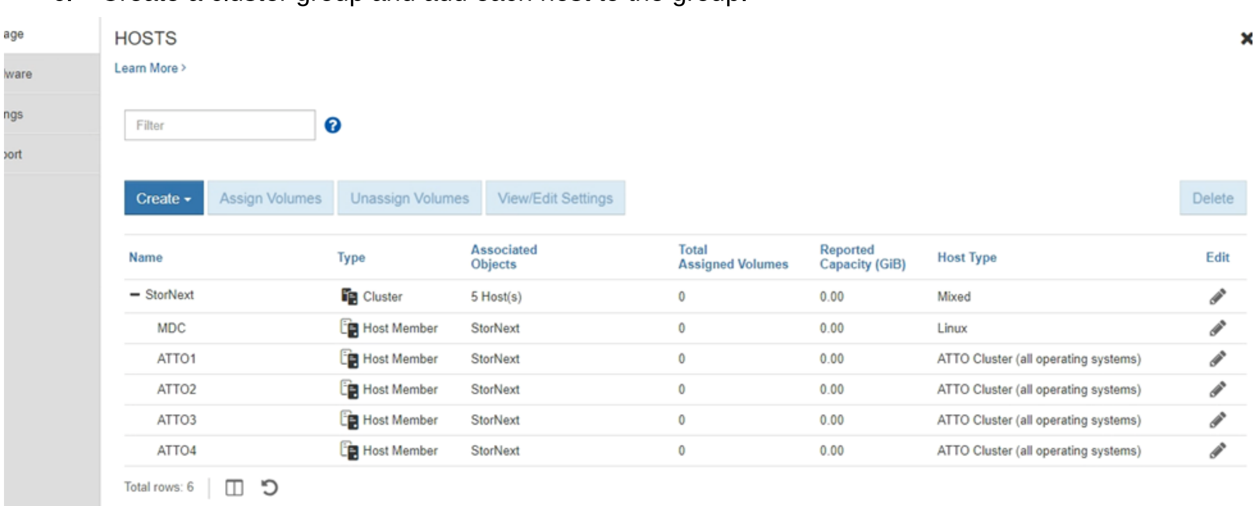
   – Each initiator port must be in a separate zone with all its corresponding target ports.

   Note: Because the storage controllers have a high number of FC ports per controller, NetApp does not recommend that you zone each host to every controller port. That approach results in a high number of paths that must be managed through host failover and does not provide any performance benefit.

   – In this solution, each host port is zoned with a corresponding port on the controllers (as if each host was direct-attached), alternating controllers between ports. Following is an example of the zoning for the first two hosts on the EF600 array. In this example, this pattern is repeated for each of the four hosts. The EF280, EF300, and E5760 zoning is similar, but with only four target ports on the arrays.

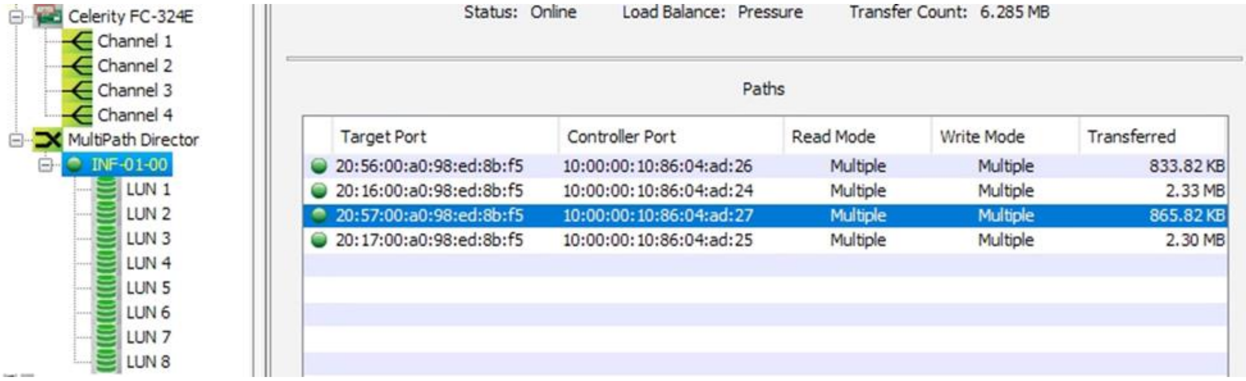
```
Zone 1 - Host 1 Port 1 -> Controller A, HIC 1, Port 1
Zone 2 - Host 1 Port 2 -> Controller B, HIC 1, Port 1
Zone 3 - Host 1 Port 3 -> Controller A, HIC 2, Port 1
Zone 4 - Host 1 Port 4 -> Controller B, HIC 2, Port 1
Zone 5 - Host 2 Port 1 -> Controller A, HIC 1, Port 2
Zone 6 - Host 2 Port 2 -> Controller B, HIC 1, Port 2
Zone 7 - Host 2 Port 3 -> Controller A, HIC 2, Port 2
Zone 8 - Host 2 Port 4 -> Controller B, HIC 2, Port 2
```

3. Create host mappings on the E-Series array:
   a. Create a host mapping for each client, selecting the ATTO cluster host option.
   b. Create a host mapping for the MDC as well, specifying the OS version for that MDC. In this case, the MDC is running Linux with Device Mapper failover, because it has an Emulex adapter instead of ATTO. Therefore, Linux is selected as the host type.
   c. Create a cluster group and add each host to the group.



   d. Next, assign the eight created volumes to the StorNext cluster group.

4. Confirm that each host and the MDC can see the volumes that are provisioned on the storage array. For example:



Configure the StorNext file system

To configure the StorNext file system, complete the following steps:

Note: These steps assume that you have already installed the StorNext file system on the MDC and clients and that you have configured a management network to connect the clients to the MDC. Installation of the StorNext file system is outside the scope of this deployment guide. For installation instructions, consult the StorNext documentation.

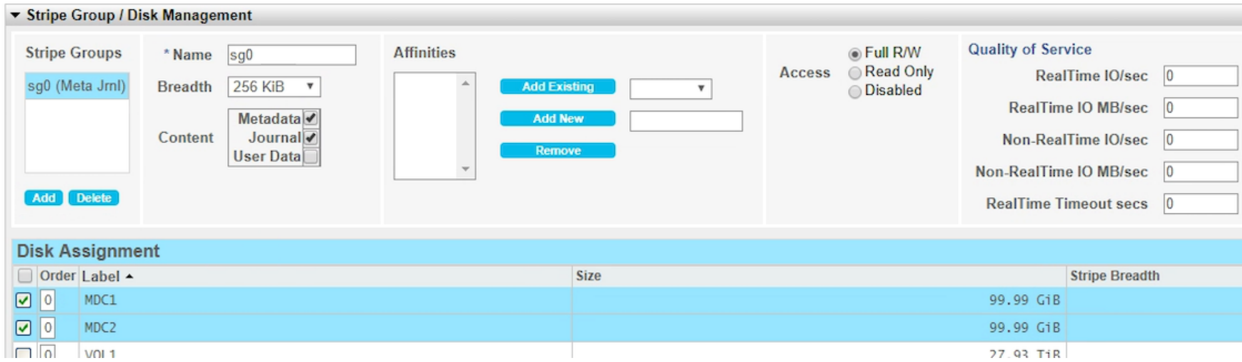
1. Log in to the StorNext MDC web interface to create a new file system. In this example, the file system is called `snfs2`.

   When you label the disks, NetApp recommends that you label them according to the volume names on the storage array. Because volume ownership is staggered between controller A and controller B, it is important to keep the file system striping of the volumes in order. Because the example MDC is running Device Mapper for failover, the volumes show up as `/dev/mapper/mpath[x]` devices.

   You can correlate which device matches up to each LUN on the storage array by executing the following command on the command line of the MDC. For example:

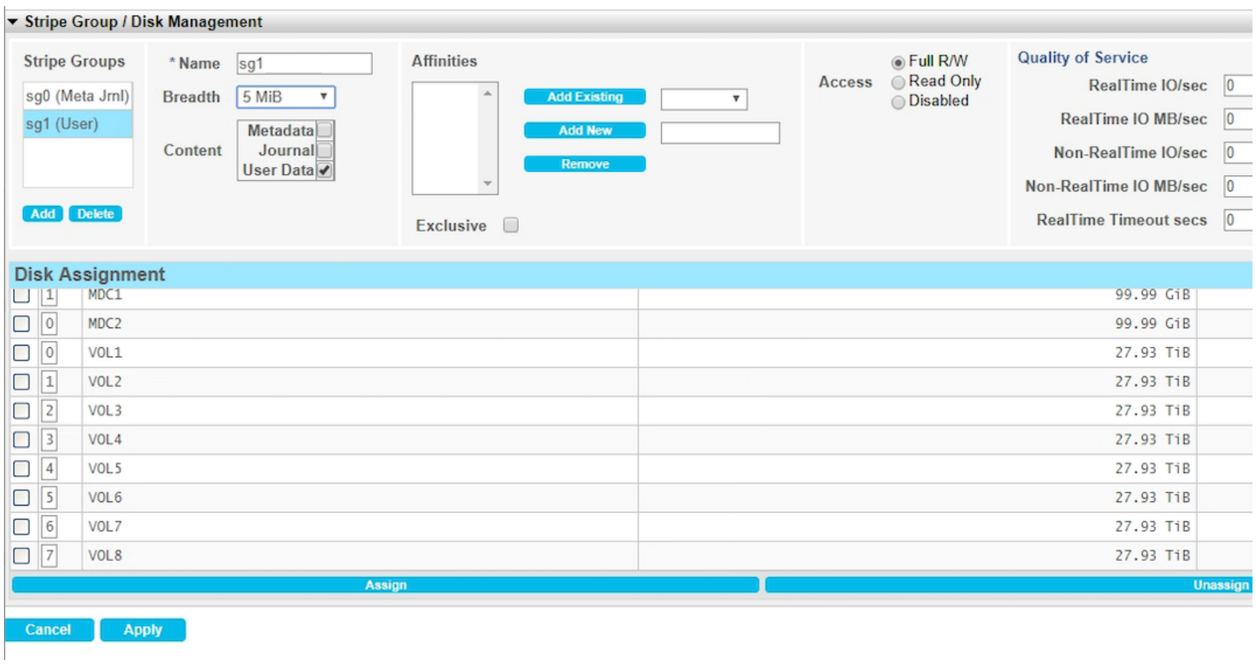
```
/usr/cvfs/bin/cvlabel -l -s
/dev/mapper/mpathd [NETAPP  INF-01-00       0861] SNFS-EFI "VOL1"Controller#: '200600A098ED8BF5'
Serial#: '600A098000EDC66000003D175E221A78' Sectors: 7498612475. Sector Size: 4096.  Maximum
sectors: 7498612475. Stripebreadth: 0.
/dev/mapper/mpathe [NETAPP  INF-01-00       0861] SNFS-EFI "VOL2"Controller#: '200600A098ED8BF5'
Serial#: '600A098000ED8BF50000159C5E221AAE' Sectors: 7498612475. Sector Size: 4096.  Maximum
sectors: 7498612475. Stripebreadth: 0.
```

2. Create a stripe group for the metadata and journal data. Assign the metadata and journal data to volumes that are on the separate redundant storage that is attached to the MDC. This example uses two 100GiB LUNs that are local to the MDC. And because each LUN has a 128KiB stripe size, this example also uses a stripe breadth of 256KiB. Your LUN size might differ, but the important thing is to make sure that the metadata and journal are assigned to the storage separate from the array on which your data will be stored.



3. Create another stripe group for the user data and assign it to the volumes from the E-Series array. Also specify the order of striping by numbering the boxes to the left of the volume labels (Zero-based numbering).

   In testing of the EF280, EF300, and EF600 arrays, a setting of 5MiB provided the best performance with their 10+2 RAID 6 volumes. For the E5760 system, the default setting of 4MiB provided the best performance with its 8+2 RAID 6 volumes. In general, NetApp recommends that you use breadth sizes of at least 1MiB or greater, while you also try to match the volume stripe size on the storage array. For example, an 8+2 RAID 6 volume with a 512K segment size has a stripe size of 4MiB.



Performance tuning on the hosts

In this example, several areas were tuned for performance on the clients, as explained in this section. Note that tuning on the MDC was left at the default settings. If you need information about MDC performance tuning, consult the StorNext documentation.

Disable CPU power-saving states

Quantum recommends this tuning for all StorNext clients. The following instructions are for RHEL. For other versions of Linux, consult the StorNext documentation.

1. Add the following text to the `GRUB_CMDLINE_LINUX` in `/etc/default/grub`:

```
intel_idle.max_cstate=0 processor.max_cstate=1
```

2. Run `grub2-mkconfig`, then reboot the host:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Enable blk-mq

In this example, multiqueue I/O scheduling was enabled specifically because the goal was to try to maximize the number of video streams per host, with only four hosts. In environments with more hosts, where the number of streams per host will be lower, this setting might not be required.

1. To enable multiqueue I/O scheduling, add the following text to the `GRUB_CMDLINE_LINUX` in `/etc/default/grub`:

```
scsi_mod.use_blk_mq=y
```

2. Run `grub2-mkconfig`, using the following command, and then reboot the host.

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

The deviceparams file

StorNext employs a `deviceparams` file, which can control the I/O scheduler and can tweak block-level device parameters. Create the file on each client under `/usr/cvfs/config/deviceparams`.

Following are examples, along with an explanation of each parameter:

- Recommended EF280 and EF600 settings:

```
nr_requests=256
read_ahead_kb=10240
max_sectors_kb=5120
```

- Recommended E5760 settings:

```
nr_requests=256
read_ahead_kb=4096
max_sectors_kb=2048
```

nr_requests

For RHEL 7.7, with `blk-mq` enabled, 256 is the maximum setting. In cases in which `blk_mq` is not used, this parameter can be set to 4096.

read_ahead_kb

This parameter enables the host to “read ahead” as it is streaming video from the array.

max_sectors_kb

This parameter sets the maximum I/O size that the host sends to the storage array.

Further guidance

If you are not using `blk-mq`, then NetApp recommends that you specify `noop` as the I/O scheduler. Set this parameter only if `blq-mq` is not in use:

```
scheduler=noop
```

After you create the `deviceparams` file, trigger a rescan of the disks with the following command, which applies the settings:

```
/usr/cvfs/bin/cvadmin -e "disks refresh"
```

Connect clients to the file system and run the Frametest benchmark

To connect the clients to the file system and to run the Frametest benchmark, complete the following steps:

1. When you maximize the number of streams per host, as was done in this solution, NetApp recommends that you increase the buffer size on the host. This step enables the maximum number of streams to be processed. If you use more clients and fewer streams per client, then you do not need the `buffercachecap` and `cachebufsize` flags in the following mount command.

   In this example, the file system named `snfs2` is mounted to the directory `/stornext/snfs2`.

```
mount -t cvfs snfs2 /stornext/snfs2 -o
buffercachecap=2048,cachebufsize=2048k
```

2. Run the Frametest benchmark.

   Frametest is a free, commonly used benchmarking tool that simulates reading and writing of video streams. You can find more information about usage and available downloads here: https://support.dvsus.com/hc/en-us/articles/212925466-How-to-use-frametest

   The video files can be written with the following write command. You can queue up multiple processes by ending the command with `&`:

```
/root/frametest_linux -w 4k -f 24 -n40000 -t 1 -d 90 -x[filename].csv --name 1
/[stornext_directory_name]/host[x]_stream[x]
```

   The video files can then be read with the following read command. You can queue up multiple processes by ending the command with `&`:

```
/root/frametest_linux -r 4k -f 24 -n40000 -t 1 -d 90 -p15 -z49856 -x[filename].csv --name [x]
/[stornext_directory_name]/host[x]_stream[y]
```

   Replace the `[x]` variable in these examples with the host number and replace `[y]` with the stream number. Each flag is listed below with an explanation of what it does:

   – `-w` is the write command.
   – `-r` is the read command.
   – `4k` is the format that Frametest simulates.
   – `-f` is the frame rate (24 frames per second [FPS] in these examples).
   – `-n` is the number of frames; so, 40,000 is about 27 minutes.
   – `-t` is the number of threads; we kept this example at 1 and started multiple processes to increase threads (adding `&` at the end of the entire command, if you want to queue it),
   – `-d` is the number of files per directory; 90 was chosen for these examples.
   – `-p` is the number of seconds to pause before starting the test.
   – `-z` in the read command is used to match the size that Frametest sets on the write command (49856).
   – `-x` exports detailed timing to a CSV file.
   – `--name` is the name that is used in the file name pattern. These examples are numbered based on the host; here it is host 1, so file names will look like `1000000.tst`, `1000001.tst`, `1000002.tst`, and so on.
   – The final part is the directory where the frames are written. In these examples, a new directory was created for each host, and inside that directory are subfolders of 90 frames each, up to the maximum number of frames specified.

Frametest generates an `.out` text file with results for each stream. These example commands also specify the `-x` flag, which generates a CSV file that you can use for analysis. At the top of each CSV file is a summary of the run. Table 2 is an example of a single completed process:

Table 2) Summary of completed process.

Field.	Details
Date	1-Oct-20
Time	14:10:33
Version	4.22
Operating system	Linux 3.10.0-1062.el7.x86_64 #1 SMP Thu Jul 18 20:25:13 UTC 2019
Host name	bmcx400e
Test path	`/stornext/snfs2/host1_1`
Parameters	`-r -z49856 -n40000 -d90 -t1 -f24 -q24 -b5`
Frame rate	23.98fps
Bandwidth	1167.6MBps
Dropped frames	0

Solution verification

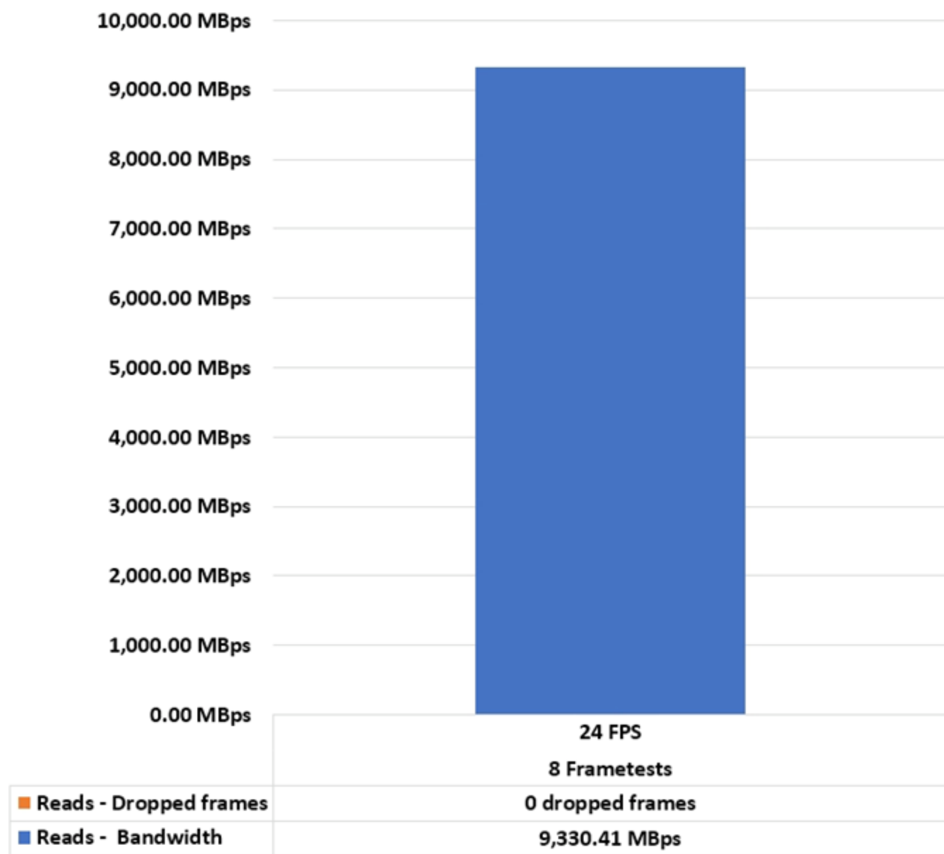
For this solution, the Frametest benchmarking tool was used to generate uncompressed 4K streams at 24 frames per second (FPS). Multiple single-threaded Frametest streams were started on each host and were scaled up to see how many streams could run simultaneously without dropping frames.

To confirm consistency, the streams were intentionally run for a long period, 27 minutes each.

EF280 results

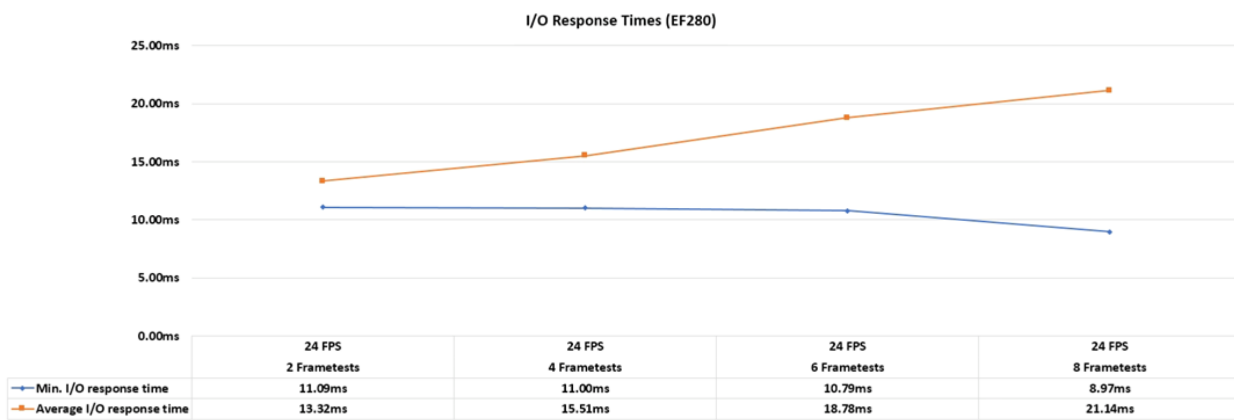
As Figure 5 shows, in testing, the NetApp EF280 configuration achieved up to eight simultaneous 4K read streams at a total of 9.3GBps, with zero dropped frames on 100% reads.

Figure 5) Results from testing with an EF280 array.



In addition, I/O response times were low, with average response times of 21.29ms at nine streams, as shown in Figure 6.

Figure 6) I/O response times with an EF280 array.



Some additional write and mixed workload testing was performed to see how many streams could be processed with no dropped frames.

At 100% writes, the EF280 array achieved up to two write streams with zero dropped frames, for a total of 2.3GBps writes. With a mixed workload, the EF280 array achieved up to six read streams at 6.9GBps while simultaneously writing one instance at 1.6GBps with zero dropped frames. See Table 3.

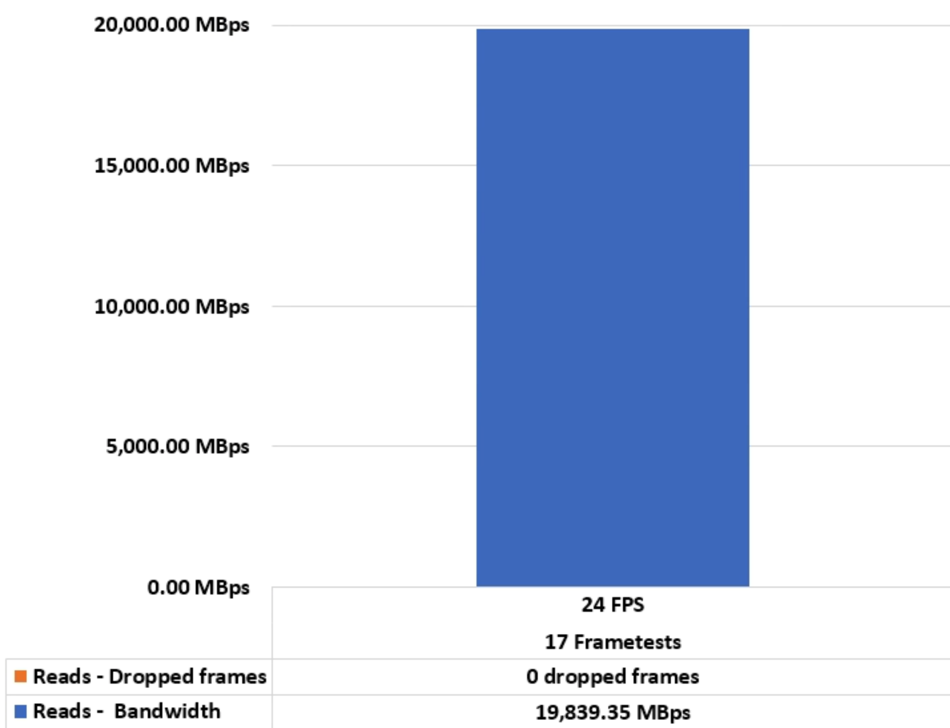
Table 3) Read, write, and mixed workload results for the EF280 array.

I/O Profile	Read Streams GBps	Write Streams GBps
100% reads	8 streams, 9.3GBps	–
100% writes	–	2 streams, 2.3GBps
Mixed reads/writes	6 streams, 6.9GBps	1 stream, 1.6GBps

EF300 results

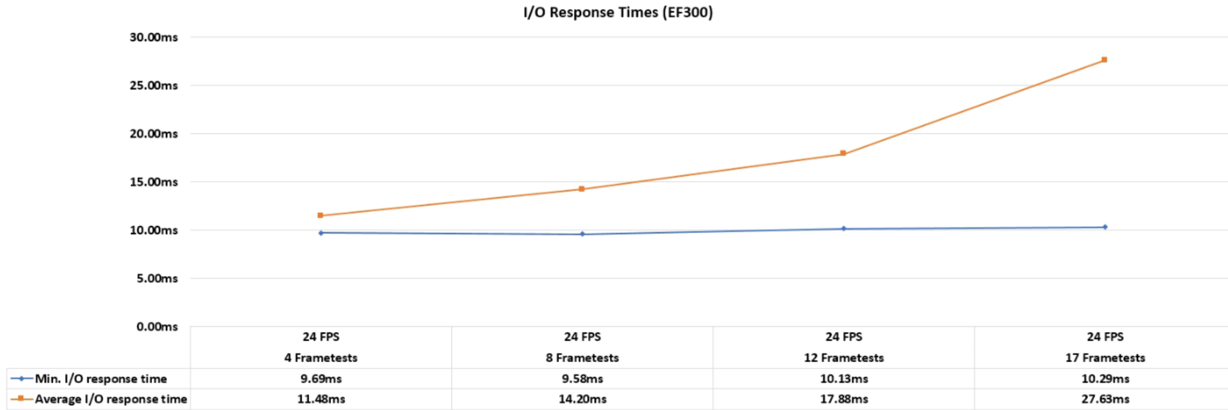
As Figure 7 shows, in testing, the NetApp EF300 configuration achieved up to 17 simultaneous 4K read streams at a total of 19.8GBps, with zero dropped frames on 100% reads.

Figure 7) Results from testing with an EF300 array.



As Figure 8 shows, I/O response times were low, with average response times of 27.63ms at 17 streams.

Figure 8) Response times with an EF300 array.



Some additional write and mixed workload testing was performed to see how many streams could be processed with no dropped frames.

At 100% writes, the EF300 array achieved up to five write streams with zero dropped frames, for a total of 5.8GBps writes. With a mixed workload, the EF300 array achieved up to eight read streams at 9.3GBps while simultaneously writing two streams at 2.3GBps with zero dropped frames. See Table 4

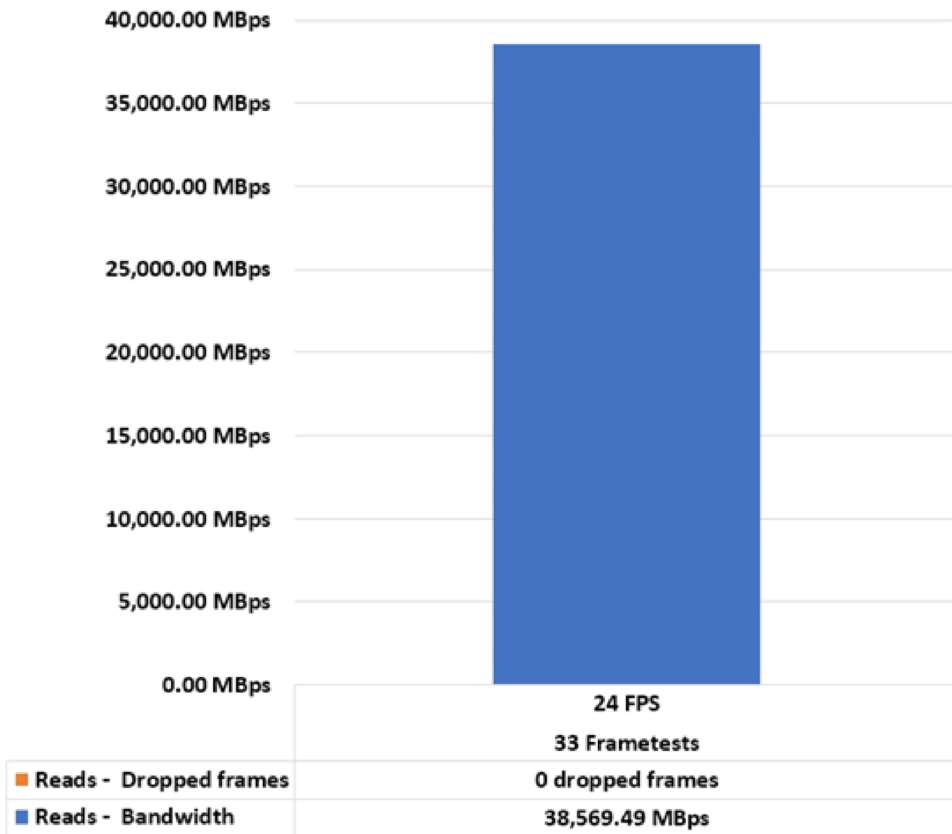
Table 4) Read, write, and mixed workload results for the EF280 array.

I/O profile	Read streams GBps	Write streams GBps
100% reads	17 streams, 19.8GBps	–
100% writes	–	5 streams, 5.8GBps
Mixed reads/writes	8 streams, 9.3GBps	2 stream, 2.3GBps

EF600 results

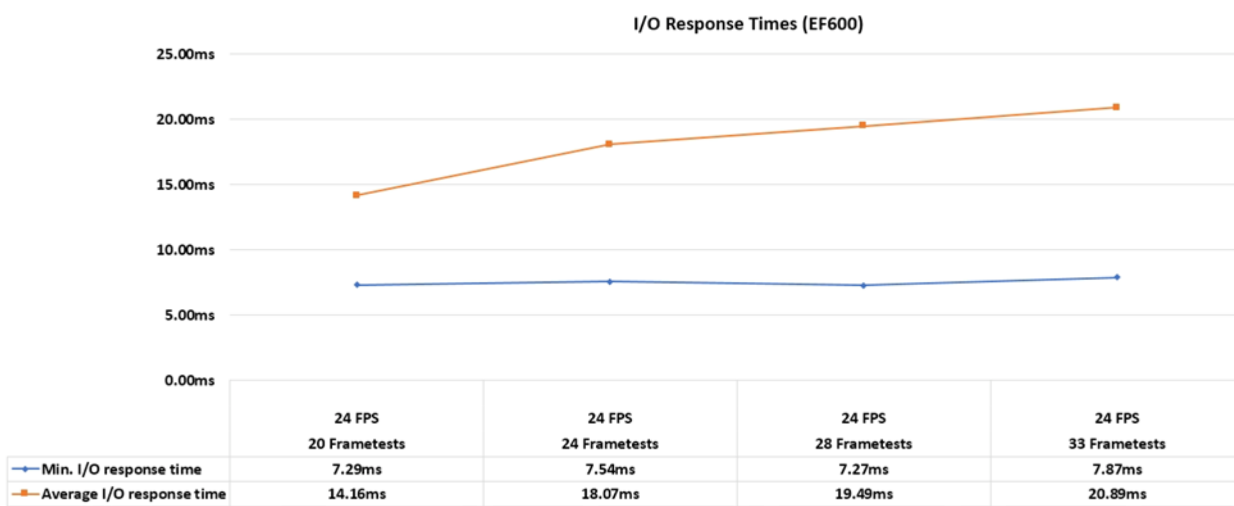
As Figure 9 shows, in testing, the NetApp EF600 configuration achieved up to 33 simultaneous 4K read streams at a total of 38.5GBps, with zero dropped frames on 100% reads.

Figure 9) Results from testing with an EF600 array.



In addition, I/O response times were low, with average response times of 20.89ms at 33 streams. Figure 10 illustrates the minimum and average response times as the number of streams increased.

Figure 10) I/O response times with an EF600 array.



Some additional write and mixed workload testing was performed to see how many streams could be processed with no dropped frames.

At 100% writes, the EF600 array achieved up to 9 write streams with zero dropped frames, for a total of 10.5GBps writes. With a mixed workload, weighted toward reads, the EF600 array achieved up to 16 read streams at 18.7GBps while simultaneously writing 5 streams at 5.8GBpsec with zero dropped frames. See Table 5.

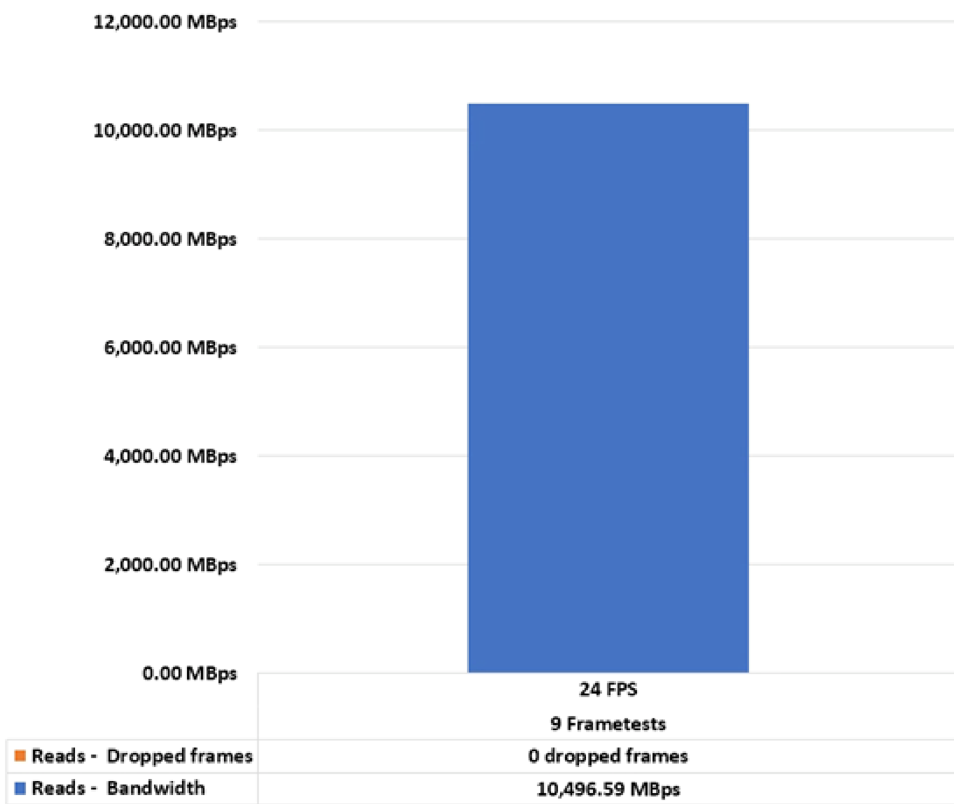
Table 5) Read, write, and mixed workload results for the EF600 array.

I/O profile	Read streams GBps	Write streams GBps
100% reads	33 streams, 38.5GBps	–
100% writes	–	9 streams, 10.5GBps
Mixed reads/writes	16 streams, 18.7GBps	5 streams, 5.8GBps

E5760 results

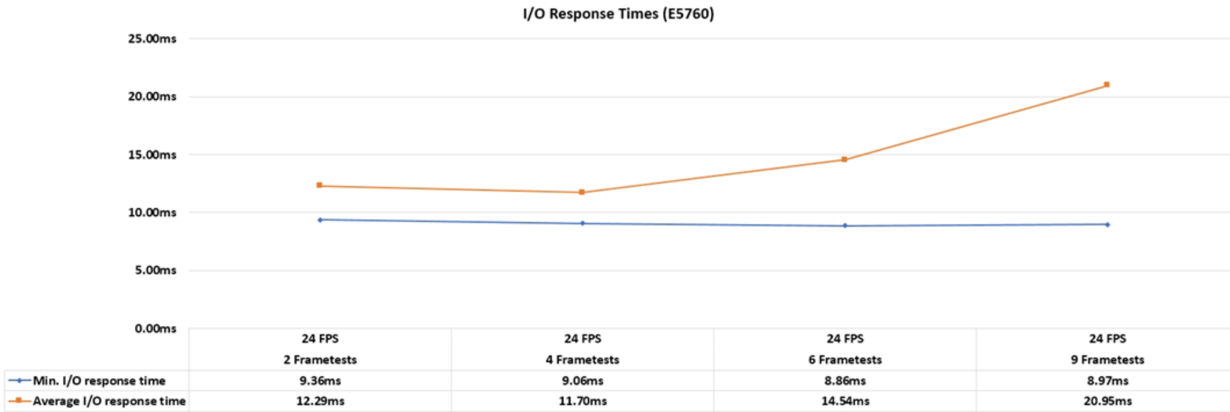
As Figure 11 shows, in testing, the NetApp E5760 configuration achieved up to nine 4K read streams at a total of 10.4GBps, with zero dropped frames on 100% reads.

Figure 11) Results from testing with an E5760 system.



I/O response times also remained low, up to nine streams. Figure 12 illustrates the minimum and average response times as the number of streams increased.

Figure 12) I/O response times with an E5760 system.



Some additional write and mixed workload testing was performed to see how many streams could be processed with no dropped frames.

At 100% writes, the E5760 system achieved up to seven write streams with zero dropped frames, for a total of 8.1GBps writes. With a mixed workload, weighted toward reads, the E5760 system achieved up to six read streams at 6.9GBps while simultaneously writing two streams at 2.3GBps with zero dropped frames. See Table 6.

Table 6) Read, write, and mixed workload results for the E5760 system.

I/O Profile	Read streams GBps	Write streams GBps
100% reads	9 streams, 10.4GBps	–
100% writes	–	7 streams, 8.1GBps
Mixed reads/writes	6 streams, 6.9GBps	2 streams, 2.3GBps

Conclusion

This document provides instructions on how to set up StorNext with the NetApp EF280, EF600, and E5760 systems to create a high-performance and cost-effective video streaming target.

The NetApp EF280 provides an entry-level all-flash option, which can support up to 8 simultaneous 4K uncompressed streams without dropping frames.

The NetApp EF300 array provides a midrange all-flash NVMe option, which can support up to 17 simultaneous 4k uncompressed streams without dropping frames.

The NetApp EF600 array with StorNext provides a high-performing shared storage system that is optimized for supporting many uncompressed 4K video streams. Testing shows that it can serve up to 33 simultaneous 4K uncompressed streams without dropping frames in only 2U of rack space.

The NetApp E5760 array provides a lower-cost, high-capacity tier that is efficient and cost-effective.

Where to find additional information

To learn more:

- NetApp Media and Entertainment Storage Solutions
  https://www.netapp.com/us/solutions/industry/media-entertainment-storage-solutions.aspx

- E-Series for Media Solution Brief
  https://www.netapp.com/us/media/ds-e-series-media.pdf
- NetApp Product Documentation
  https://docs.netapp.com

Version history

Version	Date	Document version history
Version 1.0	August 2020	Initial release.
Version 1.1	October 2020	Added reference to EF280.
Version 1.2	January 2020	Added reference to EF300.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NVA-1150-DEPLOY-0121