Technical Report

Quantum StorNext with NetApp EF-Series

Solution with high-capacity flash drives

Mitch Blackburn, Philip Quang, NetApp

July 2025 | TR-5011

Abstract

This document provides details on how to design and deploy a StorNext parallel file system solution with NetApp® EF-Series storage systems. This solution covers the NetApp EF-Series EF300C and EF600C high-density all-flash arrays. 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

Executive summary

As commercial video industry technologies continue to grow and improve, editing and storage requirements also continue to grow, along with the need to produce video in higher resolutions. With these changes come greater production demands and the need for massive bandwidth from the underlying production storage systems. These changes can overwhelm the proprietary media storage systems that are traditionally used in these environments.

NetApp® EF-Series systems with StorNext provide broadcasters, media studios, and media content aggregators with a high-performance and high-capacity streaming tier that also optimizes rack space, power, and cooling usage.

The NetApp EF300C provides an all-flash NVMe array which can provide up to 18.7GBps of video throughput in a 2U building block and service up to 16 simultaneous uncompressed 4K read streams with zero dropped frames.

For the highest performance, the NetApp EF600C all-flash NVMe array, can provide up to 38.6GBps of video throughput in a 2U building block while serving up to 33 simultaneous uncompressed 4K read streams with zero dropped frames.

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

Solution overview

NetApp EF-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. EF300C and EF600C systems fulfill these roles.

The architecture that is demonstrated in this design guide shows the capabilities of these two arrays. The medium-tier option uses a single EF300C array with high performing NVMe drives and the high-performance building block using a single EF600C array also with high performing NVMe drives. Additional similarly configured arrays can be added to the StorNext namespace to allow virtually unlimited scale-out.

With high-capacity drives the EF300C and EF600C also demonstrate value. The EF300C and EF600C can support a high number of high-resolution video streams with no dropped frames. Both options save the customer in footprint and power/cooling costs.

In addition, the EF300C and EF600C arrays support both SCSI and NVMe host protocols, enabling the customer to start with a SCSI based protocol and then to upgrade to NVMe over Fabrics in the future.

Target audience

The target audience for the solution includes the following groups:

- Media content aggregators and distribution companies
- Broadcasters and networks
- Film studios
- Multimedia corporations
- Sports leagues

Solution technology

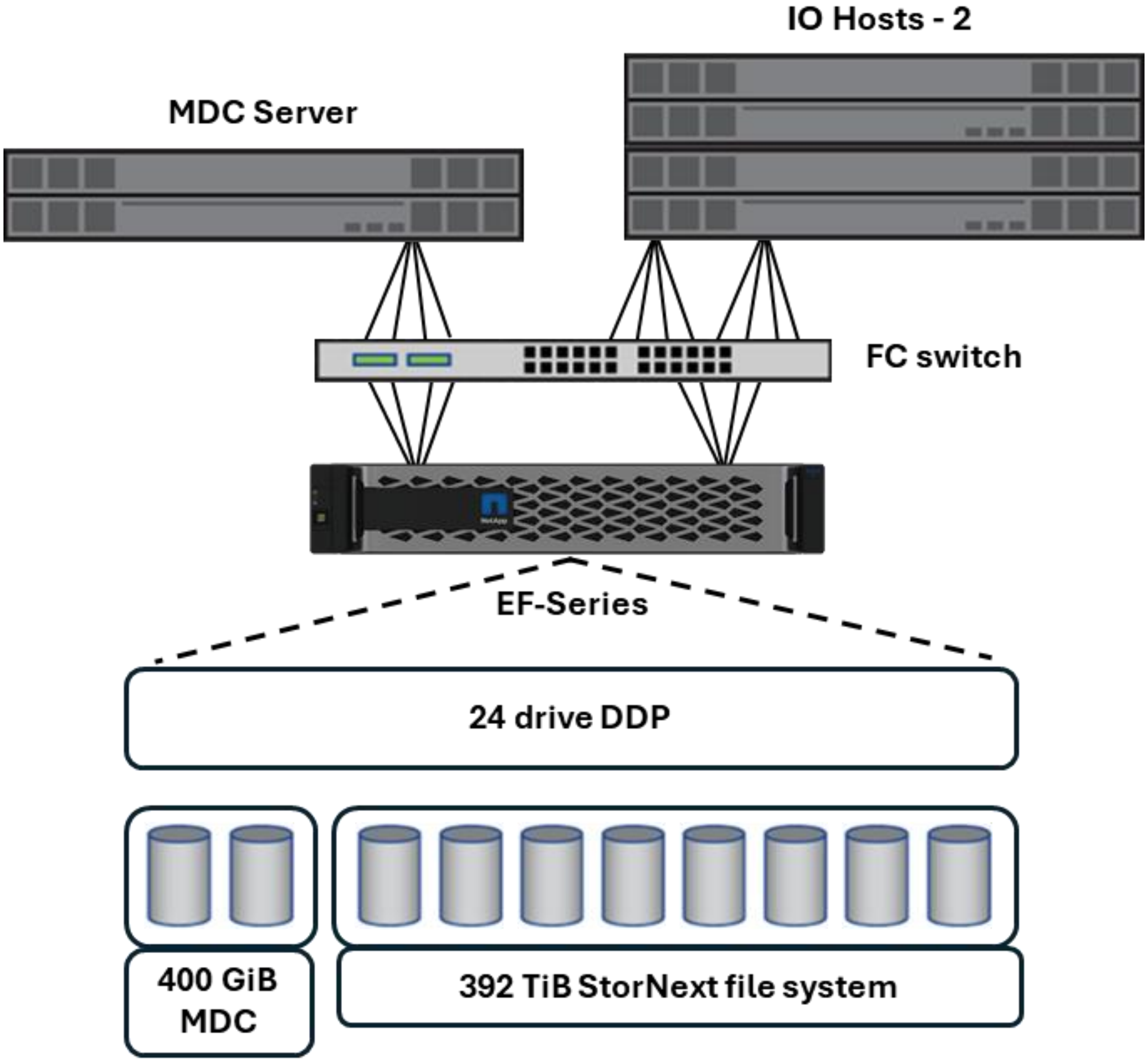
The StorNext solution in this design guide consists of at least two StorNext clients. Each configuration runs the same tests, but with a different target storage array in each case. The NetApp EF300C and EF600C all-flash arrays are used for both the MDC Metadata and the StorNext file system.

The SAN consists of a 32Gb Fibre Channel (FC) network running with generic Linux multipath on the clients.

The EF300C and EF600C options each have 24 high-capacity NVMe drives and are provisioned into one Dynamic Disk Pool (DDP).

Figure 1 shows the technical components of the solution with an EF-Series array.

Figure 1) Technical components of the solution with an EF-Series array.



Technology requirements

Hardware requirements

Servers for the NetApp EF300C and EF600C arrays:

One metadata controller (MDC) and two Fujitsu PRIMERGY RX2540 M5 with Fibre Channel (FC) adapters are used:

- Two Intel(R) Xeon(R) Gold 6244 CPU @ 3.60GHz
- 96GB RAM
- Client host bus adapters (HBAs): Emulex Corporation LPe35000/LPe36000
- MDC HBA: Emulex Corporation LPe35000/LPe36000

Storage configuration for the EF300C and EF600C:

Data volumes plus MDC/Metadata volumes on the array:

- 24 MZWMO30THCLF-0G5 - 30TB NVMe drives
- Single DDP with 8 data volumes and 2 metadata volumes (capacity is 20% overprovisioned per NVMe drive guidelines)

Switching

Brocade G620

Software requirements

This design includes StorNext file system version 6.4.0. 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 EF600C and EF300C storage targets running NetApp SANtricity® OS 11.90.

The clients in this design run Red Hat Enterprise Linux (RHEL) 7.9, with generic Linux Multipath enabled 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 implementation of the solution might vary based on customer requirements.

Table 1) Software requirements.

Software	Version or Other Information
StorNext	6.4.0
SANtricity OS	11.90
MDC OS	RHEL 7.9
Client OS	RHEL 7.9
Multipath Driver	OS default

Deployment procedures

Deploying the solution involves the following tasks:

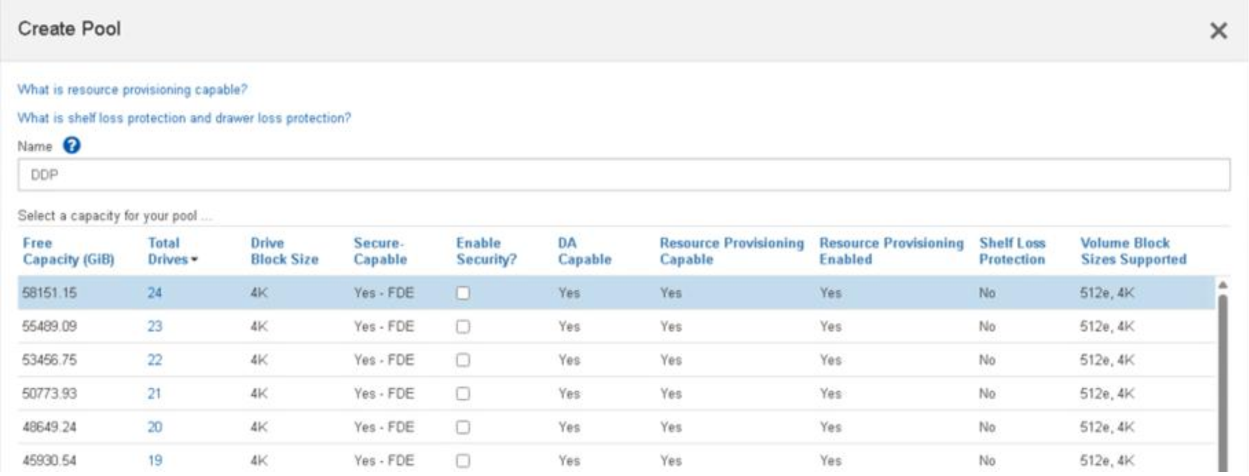
- Configure the NetApp EF-Series array
- Configure hosts and switch zoning
- Configure the StorNext file system
- Carry out performance tuning on the hosts
- Connect clients to the file system and run the Frametest benchmark

Volume configuration on the EF300C or EF600C array

The configuration for the EF300C and EF600C arrays is similar. For either one, complete the following steps

1. Create a single dynamic disk pool using all 24 drives and accept the default auto-provisioned quantity.

Note: This step may not be needed since the default configuration for EF300C and EF600C is a single DDP using all available drives.



Create Pool ✕

What is resource provisioning capable?

What is shelf loss protection and drawer loss protection?

Name

DDP

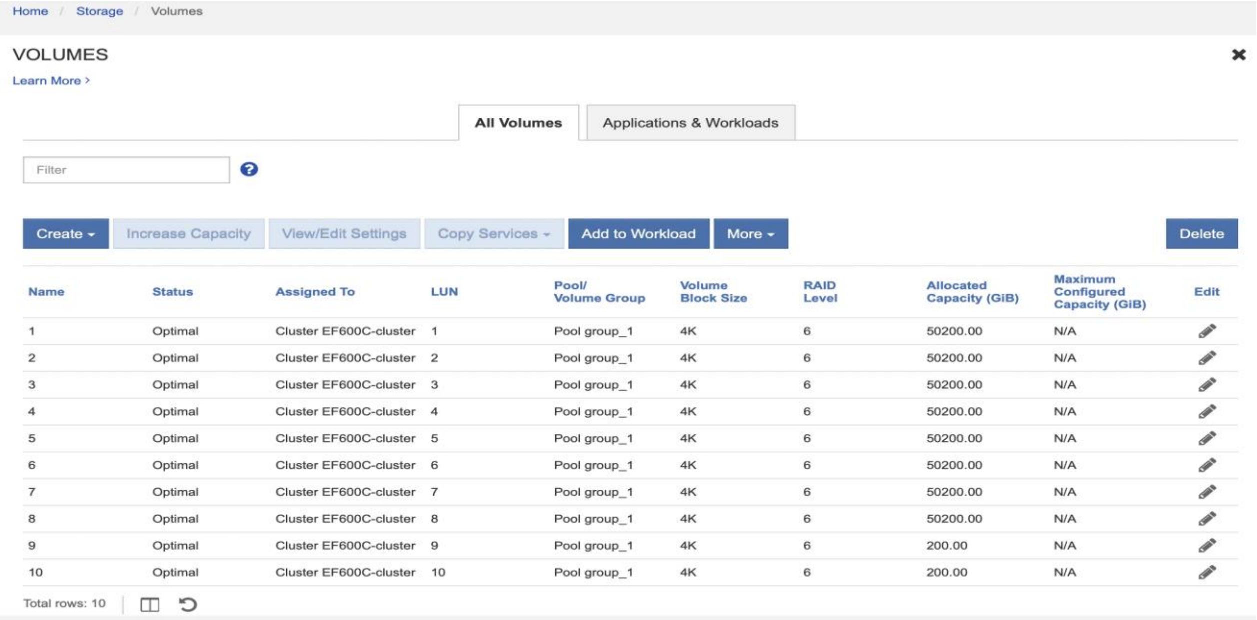
Select a capacity for your pool ...

Free Capacity (GiB)	Total Drives ▾	Drive Block Size	Secure-Capable	Enable Security?	DA Capable	Resource Provisioning Capable	Resource Provisioning Enabled	Shelf Loss Protection	Volume Block Sizes Supported
58151.15	24	4K	Yes - FDE	☐	Yes	Yes	Yes	No	512e, 4K
55489.09	23	4K	Yes - FDE	☐	Yes	Yes	Yes	No	512e, 4K
53456.75	22	4K	Yes - FDE	☐	Yes	Yes	Yes	No	512e, 4K
50773.93	21	4K	Yes - FDE	☐	Yes	Yes	Yes	No	512e, 4K
48649.24	20	4K	Yes - FDE	☐	Yes	Yes	Yes	No	512e, 4K
45930.54	19	4K	Yes - FDE	☐	Yes	Yes	Yes	No	512e, 4K

2. Create eight volumes as data volumes within the disk pool. Select StorNext File System for the workload. Because the EF300C and EF600C arrays contain NVMe drives, we overprovisioned the volume groups in this solution by 20%. In this example with a maximum capacity of 401574GiB, subtract 20% and divide by the number of volumes (eight) to arrive at 50200GiB per volume.

   Note: With SANtricity OS 11.90, which the EF300C and EF600C were released with, a portion of the capacity will be reserved automatically, and this manual calculation is not required.

3. Create two volumes as metadata volumes within the disk pool. For this NVA, the metadata volumes were sized at 200GiB. With data volumes and metadata volumes, you should have total of 10 volumes. The graphic below shows the volumes in SANtricity.



Home / Storage / Volumes

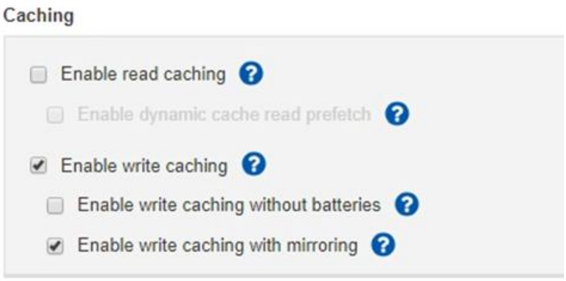
VOLUMES ✕

Learn More ›

Name	Status	Assigned To	LUN	Pool/ Volume Group	Volume Block Size	RAID Level	Allocated Capacity (GiB)	Maximum Configured Capacity (GiB)	Edit
1	Optimal	Cluster EF600C-cluster	1	Pool group_1	4K	6	50200.00	N/A	✎
2	Optimal	Cluster EF600C-cluster	2	Pool group_1	4K	6	50200.00	N/A	✎
3	Optimal	Cluster EF600C-cluster	3	Pool group_1	4K	6	50200.00	N/A	✎
4	Optimal	Cluster EF600C-cluster	4	Pool group_1	4K	6	50200.00	N/A	✎
5	Optimal	Cluster EF600C-cluster	5	Pool group_1	4K	6	50200.00	N/A	✎
6	Optimal	Cluster EF600C-cluster	6	Pool group_1	4K	6	50200.00	N/A	✎
7	Optimal	Cluster EF600C-cluster	7	Pool group_1	4K	6	50200.00	N/A	✎
8	Optimal	Cluster EF600C-cluster	8	Pool group_1	4K	6	50200.00	N/A	✎
9	Optimal	Cluster EF600C-cluster	9	Pool group_1	4K	6	200.00	N/A	✎
10	Optimal	Cluster EF600C-cluster	10	Pool group_1	4K	6	200.00	N/A	✎

Total rows: 10

4. 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.



Caching

- ☐ Enable read caching
  - ☐ Enable dynamic cache read prefetch
- ☑ Enable write caching
  - ☐ Enable write caching without batteries
  - ☑ Enable write caching with mirroring

Configure the hosts and switch zoning

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

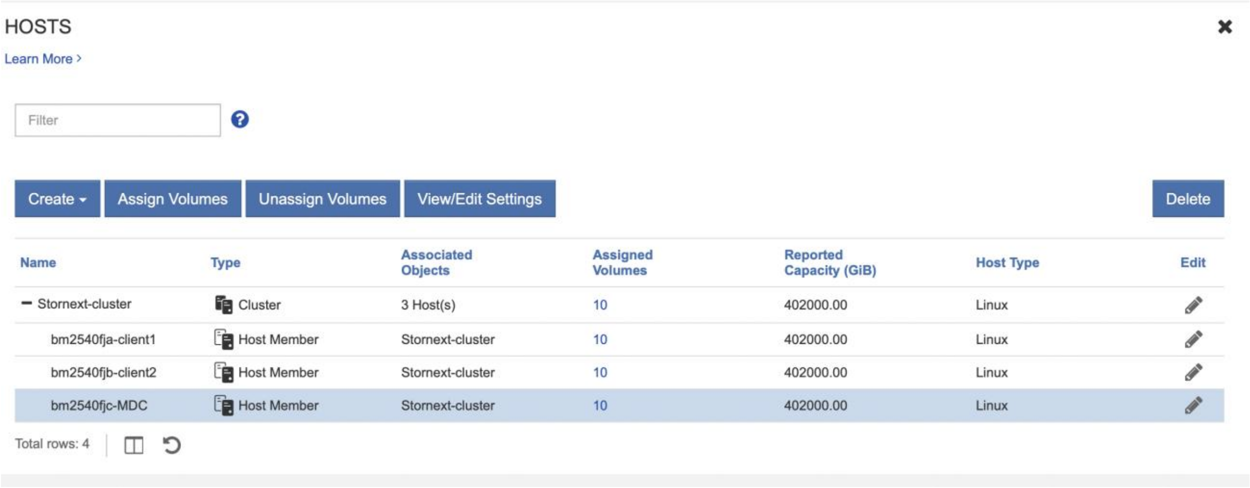
1. Ensure that the Multipath feature is enabled on the hosts. To ensure that your specific host combination is listed, check the NetApp Interoperability Matrix Tool, which also includes notes about any additional configuration that might be required.
2. Configure the zoning on the FC switch or switches so that the clients and StorNext MDC are zoned to the EF-Series array.
   a. 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.

   b. 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 EF300C array.

```
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
```

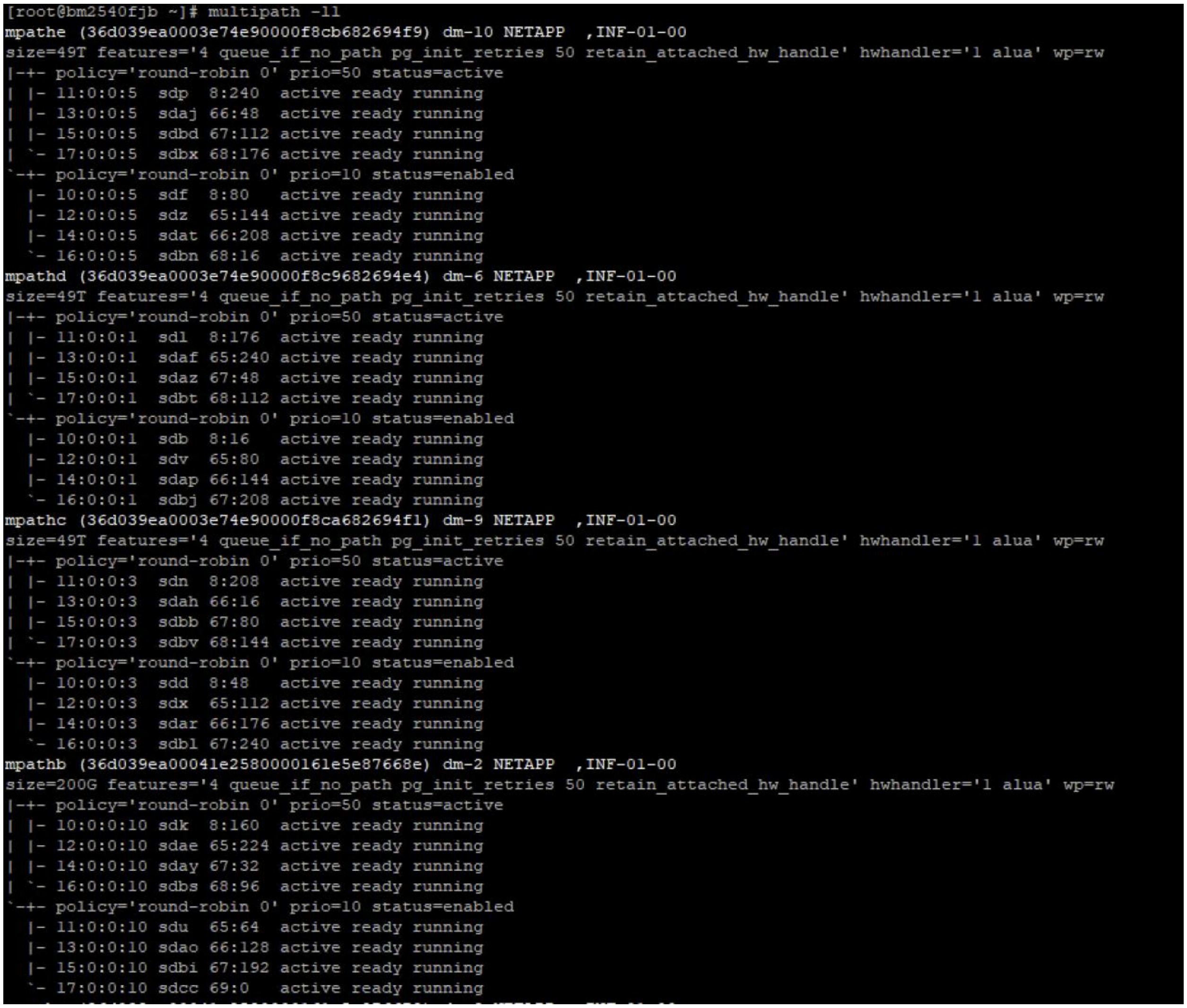
3. Create host mappings on the EF-Series array:
   a. Create a host mapping for each client, selecting the default host option.
   b. Create a host mapping for the MDC as well, specifying the OS version for that MDC. In this case, the MDC is running Linux with Device Mapper failover and Linux is selected as the host type.
   c. Create a cluster group and add each host to the group.



Name	Type	Associated Objects	Assigned Volumes	Reported Capacity (GiB)	Host Type	Edit
− Stornext-cluster	Cluster	3 Host(s)	10	402000.00	Linux	✎
bm2540fja-client1	Host Member	Stornext-cluster	10	402000.00	Linux	✎
bm2540fjb-client2	Host Member	Stornext-cluster	10	402000.00	Linux	✎
bm2540fjc-MDC	Host Member	Stornext-cluster	10	402000.00	Linux	✎

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. You should see a total of 10 volumes – 8 data volumes and 2 mdc volumes. An abbreviated sample output of the multipath command, “multipath -ll”, is shown below.



```
[root@bm2540fjb ~]# multipath -ll
mpathe (36d039ea0003e74e90000f8cb682694f9) dm-10 NETAPP  ,INF-01-00
size=49T features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| |- 11:0:0:5  sdp  8:240  active ready running
| |- 13:0:0:5  sdaj 66:48  active ready running
| |- 15:0:0:5  sdbd 67:112 active ready running
| `- 17:0:0:5  sdbx 68:176 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 10:0:0:5  sdf  8:80   active ready running
  |- 12:0:0:5  sdz  65:144 active ready running
  |- 14:0:0:5  sdat 66:208 active ready running
  `- 16:0:0:5  sdbn 68:16  active ready running
mpathd (36d039ea0003e74e90000f8c9682694e4) dm-6 NETAPP  ,INF-01-00
size=49T features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| |- 11:0:0:1  sdl  8:176  active ready running
| |- 13:0:0:1  sdaf 65:240 active ready running
| |- 15:0:0:1  sdaz 67:48  active ready running
| `- 17:0:0:1  sdbt 68:112 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 10:0:0:1  sdb  8:16   active ready running
  |- 12:0:0:1  sdv  65:80  active ready running
  |- 14:0:0:1  sdap 66:144 active ready running
  `- 16:0:0:1  sdbj 67:208 active ready running
mpathc (36d039ea0003e74e90000f8ca682694f1) dm-9 NETAPP  ,INF-01-00
size=49T features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| |- 11:0:0:3  sdn  8:208  active ready running
| |- 13:0:0:3  sdah 66:16  active ready running
| |- 15:0:0:3  sdbb 67:80  active ready running
| `- 17:0:0:3  sdbv 68:144 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 10:0:0:3  sdd  8:48   active ready running
  |- 12:0:0:3  sdx  65:112 active ready running
  |- 14:0:0:3  sdar 66:176 active ready running
  `- 16:0:0:3  sdbl 67:240 active ready running
mpathb (36d039ea00041e2580000161e5e87668e) dm-2 NETAPP  ,INF-01-00
size=200G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| |- 10:0:0:10 sdk  8:160  active ready running
| |- 12:0:0:10 sdae 65:224 active ready running
| |- 14:0:0:10 sday 67:32  active ready running
| `- 16:0:0:10 sdbs 68:96  active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
  |- 11:0:0:10 sdu  65:64  active ready running
  |- 13:0:0:10 sdao 66:128 active ready running
  |- 15:0:0:10 sdbi 67:192 active ready running
  `- 17:0:0:10 sdcc 69:0   active ready running
```

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`.

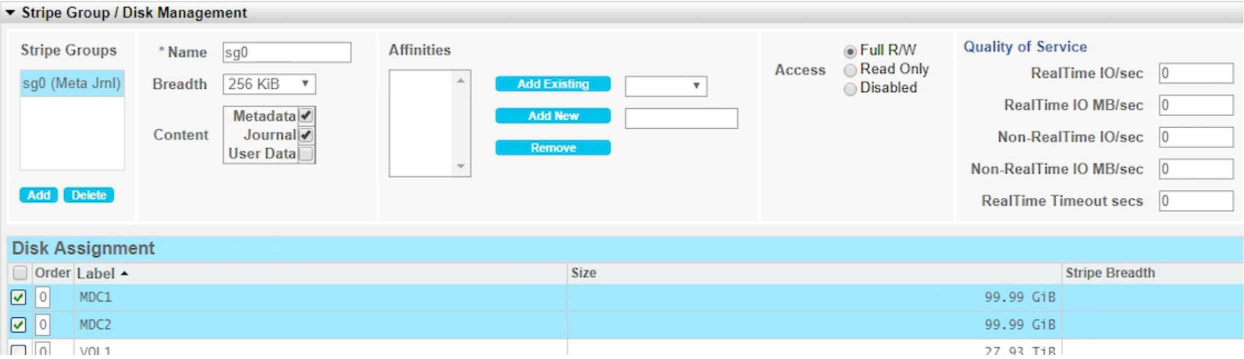
   When you label the disks, NetApp recommends that you label them according to the volume names on the storage array. Because volume ownership is staggered between controller A and controller B, it is important to keep the file system striping of the volumes in order. Because the example MDC is running 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.



Stripe Group / Disk Management

Stripe Groups: sg0 (Meta Jrnl)

* Name: sg0

Breadth: 256 KiB

Content: Metadata ☑ Journal ☑ User Data □

Access: ● Full R/W ○ Read Only ○ Disabled

Quality of Service

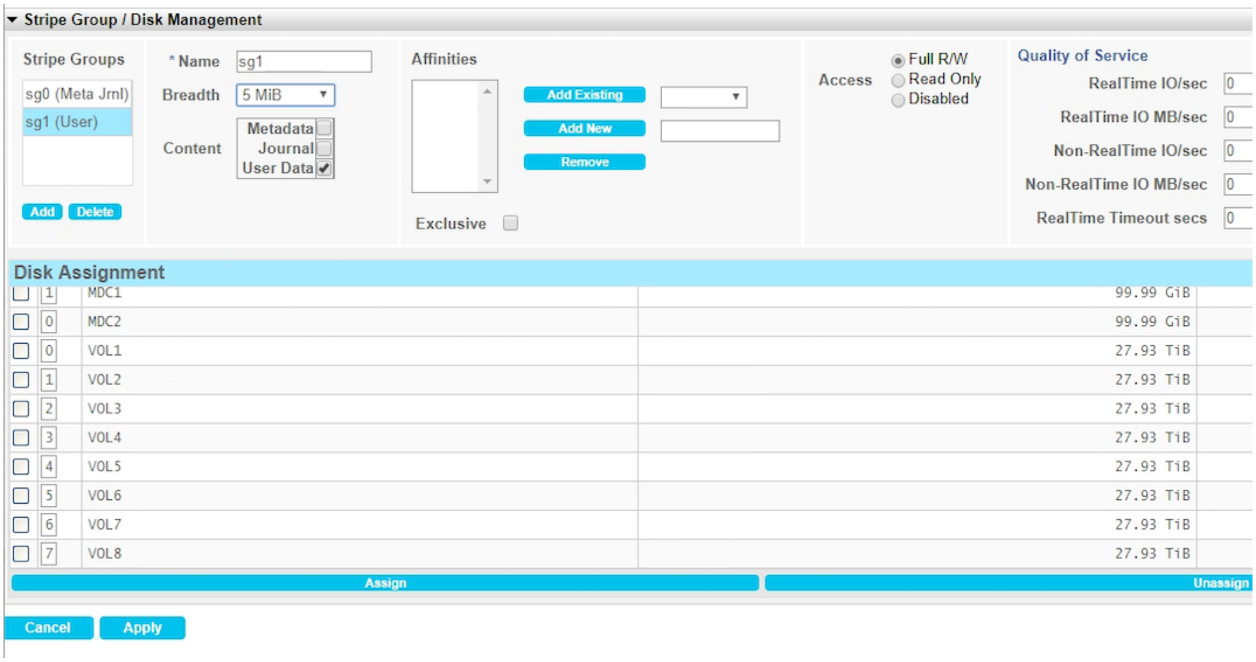
RealTime IO/sec	0
RealTime IO MB/sec	0
Non-RealTime IO/sec	0
Non-RealTime IO MB/sec	0
RealTime Timeout secs	0

Disk Assignment

□	Order	Label	Size	Stripe Breadth
☑	0	MDC1	99.99 GiB	
☑	0	MDC2	99.99 GiB	
□	0	VOL1	27.93 TiB	

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

   In testing the EF300C and EF600C arrays, a setting of 5MiB provided the best performance. 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, DDP uses a 128k segment size in an 8 + 2 configuration. So the breadth size is 1MiB.



Stripe Group / Disk Management

Stripe Groups: sg0 (Meta Jrnl), sg1 (User)

* Name: sg1

Breadth: 5 MiB

Content: Metadata □ Journal □ User Data ☑

Exclusive □

Access: ● Full R/W ○ Read Only ○ Disabled

Quality of Service

RealTime IO/sec	0
RealTime IO MB/sec	0
Non-RealTime IO/sec	0
Non-RealTime IO MB/sec	0
RealTime Timeout secs	0

Disk Assignment

□	1	MDC1	99.99 GiB
□	0	MDC2	99.99 GiB
□	0	VOL1	27.93 TiB
□	1	VOL2	27.93 TiB
□	2	VOL3	27.93 TiB
□	3	VOL4	27.93 TiB
□	4	VOL5	27.93 TiB
□	5	VOL6	27.93 TiB
□	6	VOL7	27.93 TiB
□	7	VOL8	27.93 TiB

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 EF300C and EF600C settings:

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

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: 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	29-May-25
Time	11:54:45
Version	4.22
Operating system	Linux 3.10.0-1160.el7.x86_64 #1 SMP Tue Aug 18 14:50:17 EDT 2020
Host name	bm2540fja
Test path	`/stornext/snfs/host1_stream1`
Parameters	`-r -z49856 -n40000 -d90 -t1 -f24 -q24 -b5`
Frame rate	23.99fps
Bandwidth	1168.03MB/s
Dropped frames	0

Solution verification

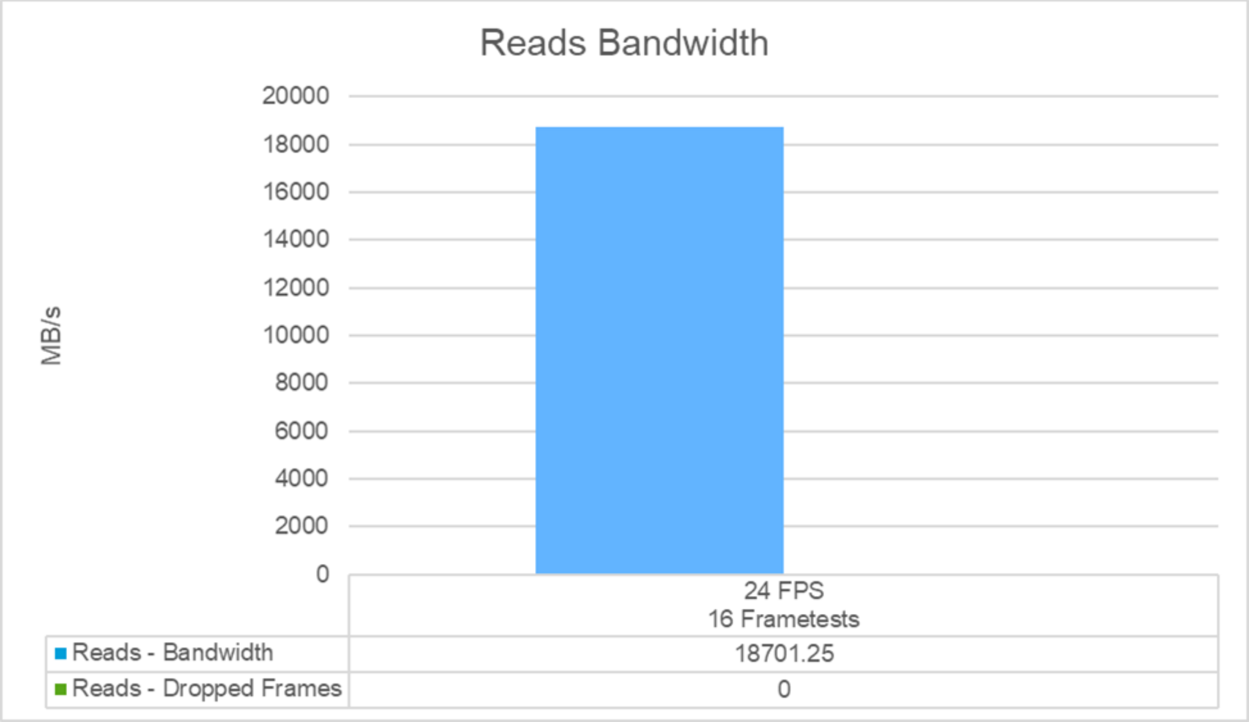
For this solution, the industry-standard Frametest benchmarking tool was used to generate uncompressed 4K streams at 24 frames per second (FPS). Multiple single-threaded Frametest streams were started on each host and were scaled up to see how many streams could run simultaneously without dropping frames.

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

EF300C results

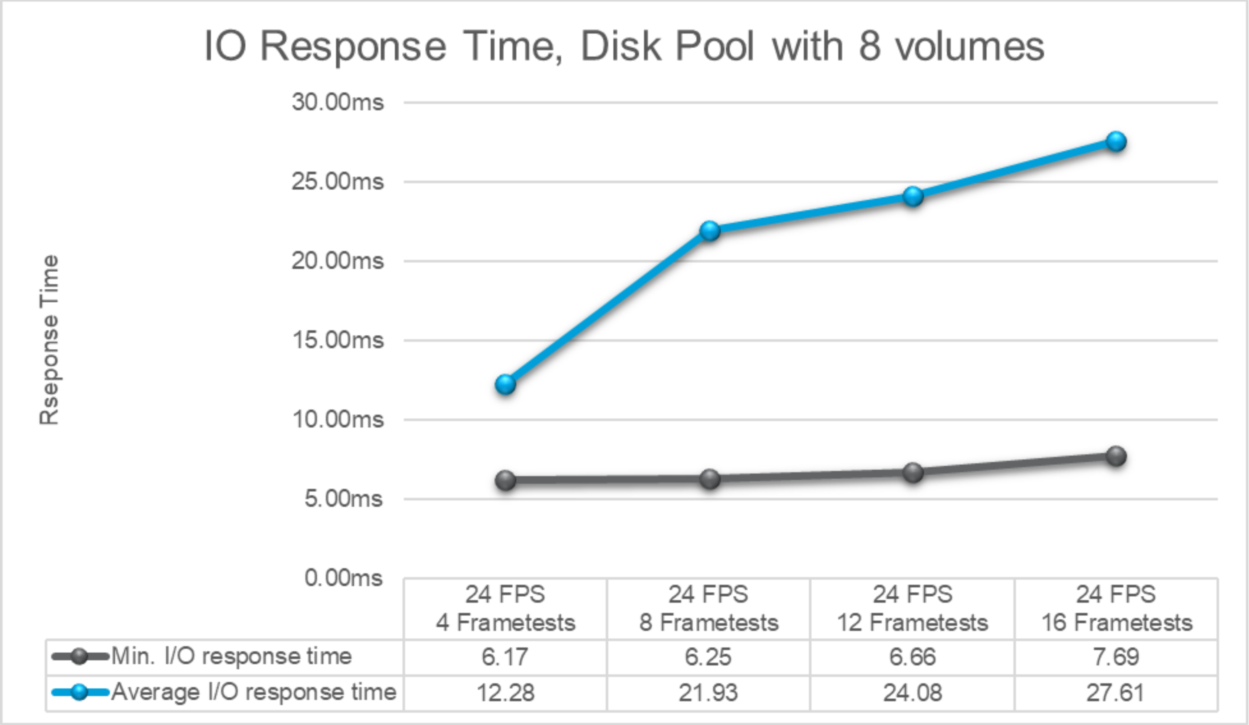
As Figure 2 shows, in testing the NetApp EF300C configuration achieved up to 16 simultaneous 4K read streams at a total of ~18.7GBps, with zero dropped frames on 100% reads.

Figure 2) Results from testing with an EF300C array.



As Figure 3 shows, I/O response times were low, with average response times of 27.61ms at 16 streams.

Figure 3) I/O Response times with an EF300C array.



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

At 100% writes, the EF300C array achieved up to five write streams with zero dropped frames, for a total of 5.8GBps writes. With a mixed workload, the EF300C array achieved up to eight read streams at 9.3GBps while simultaneously writing three streams at 3.5GBps with zero dropped frames. See Table 3.

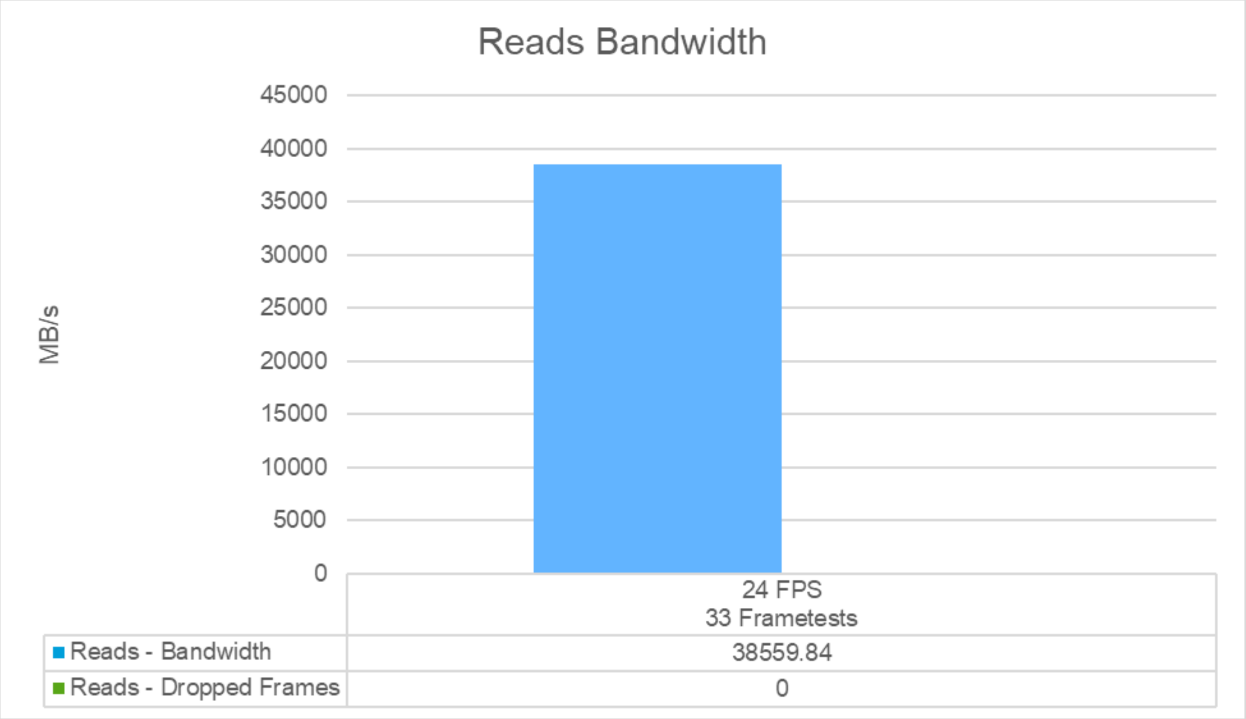
Table 3) Read, write, and mixed workload results for the EF300C array.

I/O Profile	Read Streams	Write Streams
100% reads	16 streams, 18.7GB/s	–
100% writes	–	5 streams, 5.8GB/s
Mixed reads/writes	8 streams, 9.3GB/s	3 streams, 3.5GB/s

EF600C results

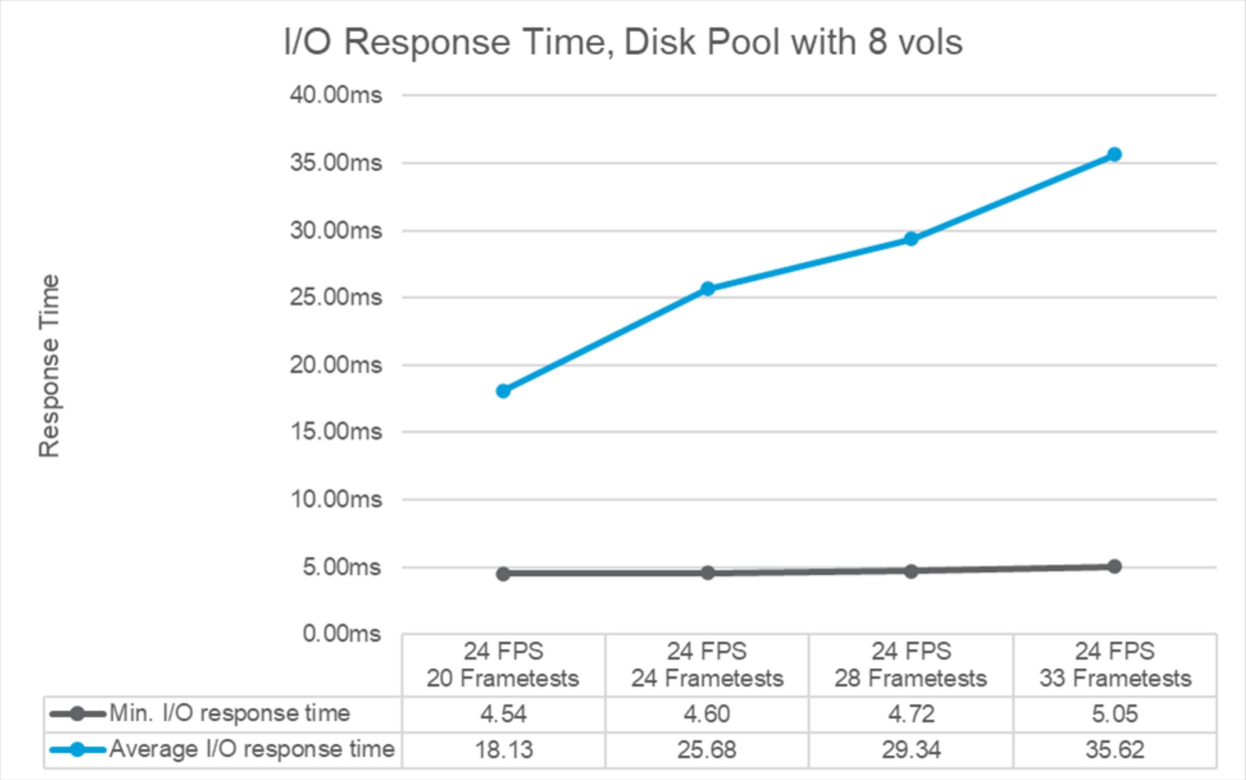
As Figure 4 shows, in testing, the NetApp EF600C configuration achieved up to 33 simultaneous 4K read streams at a total of 38.6GB/s, with zero dropped frames on 100% reads.

Figure 4) Results from testing with an EF600C array.



In addition, I/O response times were low, with average response times of 35.62ms at 33 streams. Figure 5 illustrates the minimum and average response times as the number of streams increased.

Figure 5) I/O response times with an EF600C array.



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

At 100% writes, the EF600C array achieved up to 9 write streams with zero dropped frames, for a total of 10.5GB/s writes. With a mixed workload, weighted toward reads, the EF600C array achieved up to 15 read streams at 17.5GB/s while simultaneously writing 5 streams at 5.8GB/s with zero dropped frames. See Table 4.

Table 4) Read, write, and mixed workload results for the EF600C array.

I/O Profile	Read Streams	Write Streams
100% reads	33 streams, 38.6GB/s	–
100% writes	–	9 streams, 10.5GB/s
Mixed reads/writes	15 streams, 17.5GB/s	5 streams, 5.8GB/s

Conclusion

Using NetApp high-capacity EF-Series arrays with up to 1.5PB of raw storage in your StorNext solution can provide a variety of options.

The NetApp EF300C array provides a midrange all-flash NVMe option, which can support up to 16 simultaneous 4k uncompressed streams without dropping frames.

Use of a NetApp EF600C array 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.

Where to find additional information

To learn more:

- NetApp Media and Entertainment Storage Solutions
  https://www.netapp.com/industries/media-entertainment/
- E-Series for Media Solution Brief
  https://www.netapp.com/us/media/ds-e-series-media.pdf
- NetApp E-Series Documentation
  NetApp E-Series documentation

Version history

Version	Date	Document Version History
Version 1.0	July 2025	Initial release.

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.

Copyright information

Trademark information

TR-5011-0725

NetApp