Technical Report

Implementing and configuring modern SANs with NVMe-oF

Michael Peppers, Martin George, NetApp
February 2023 | TR-4684

Abstract

This document describes how to implement and configure NVMe-oF transports (NVMe/FC and NVMe/TCP). It includes design, implementation, configuration, management guidelines and best practices to build highly available, high-performance modern SAN solutions using NVMe protocols and transports.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

What is NVMe?

NVMe—the NVM Express data storage standard—is emerging as a core technology for enterprises that are building new storage infrastructures or upgrading to modern ones.

NVMe is both a protocol optimized for solid-state storage devices, and a set of open-source architectural standards for NVMEM components and systems.

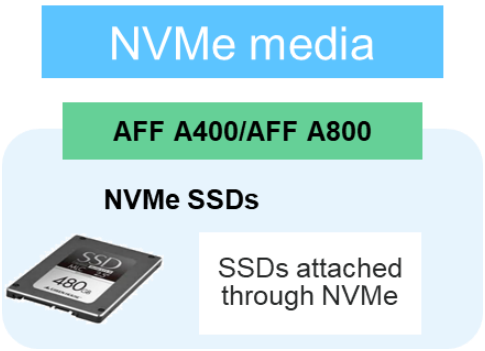
The NVMe standard is designed to deliver high bandwidth and low latency storage access for current and future memory technologies. NVMe replaces the SCSI command set with the NVMe command set and relies on PCIe, a high-speed and high-bandwidth hardware protocol that is substantially faster than older standards such as SCSI, serial-attached SCSI (SAS), and SATA.

SCSI was introduced almost 40 years ago and was designed for the storage technologies prevalent in that era: eight-inch floppy drives and file-cabinet-sized HDDs. It was also designed around the much slower single-core CPUs and smaller amounts of DRAM then available.

NVMe, on the other hand, was developed to work with nonvolatile flash drives, driven by multicore CPUs and gigabytes of memory. It also takes advantage of the significant advances in computer science since the 1970s, allowing for streamlined command sets that more efficiently parse and manipulate data.

NVMe as a term has been overloaded. With the introduction of NVMe over Fabrics (NVMe-oF) it is important to define whether we are talking about replacing media types. It is most often used to refer to the replacement of one media type with NVMe-attached disks (Figure 1). This use case could cover anything from a laptop, desktop, server or even storage array replacing a legacy media type with NVMe attached disk.

Figure 1) Defining NVMe.



Where does the NVMe standard come from?

The NVMe version 1.0 standard was ratified by NVM Express, Inc. in March 2011. There have been several updates to the standard since, with the latest being NVMe version 1.3a, ratified in November 2017. NVMe NVM Express published a complementary specification, the NVMe Management Interface (NVMe-MI), in November 2015. NVMe-MI is focused on both in-band and out-of-band management. The NVMe-oF specification was added in June 2016. NVMe-oF defined using the NVMe protocol over a network or fabric.

The NVMe 2.0 family of specifications is the current version of the NVMe specifications—it combines NVMe enhancements such as NVMe-oF, NVMe-MI, NVMe-KV, and several ratified technical proposals (TPs) to the new NVMe 2.0 family of specifications. The NVMe standard covers the software stack through to device access for modern storage devices. See NVM Express's Everything You Need to Know About the NVMe 2.0 Specifications and New Technical Proposals: https://nvmexpress.org/everything-you-need-to-know-about-the-nvme-2-0-specifications-and-new-technical-proposals/.

Note: A technical proposal is the NVMe NVM Express equivalent of the Internet Engineering Task Force (IETF) Request for Comment (RFC). A technical proposal is submitted for review by the NVM Express workgroup covering the area the that the technical proposal is focused on. Ultimately, technical proposals are voted on and potentially ratified as extensions to NVMe specifications and processes.

Who develops and controls the NVMe protocol?

NVMe is developed by a standards organization born from the Peripheral Component Interconnect PCI standards organization, the Peripheral Component Interconnect Special Interest Group (PCI-SIG). The NVMe standards organization is Non-Volatile Memory Express, Inc.

Where to find the NVMe standards documents

You can download the NVMe specifications, white papers, presentations, videos, and other collateral from the NVM Express, Inc. website.

The NVMe/FC standard is further defined by the Fibre Channel Industry Association (FCIA) in the International Committee for Information Technology Standards (INCITS) T11 Committee FC-NVMe standard, T11-2017-00145-v004 FC-NVMe.

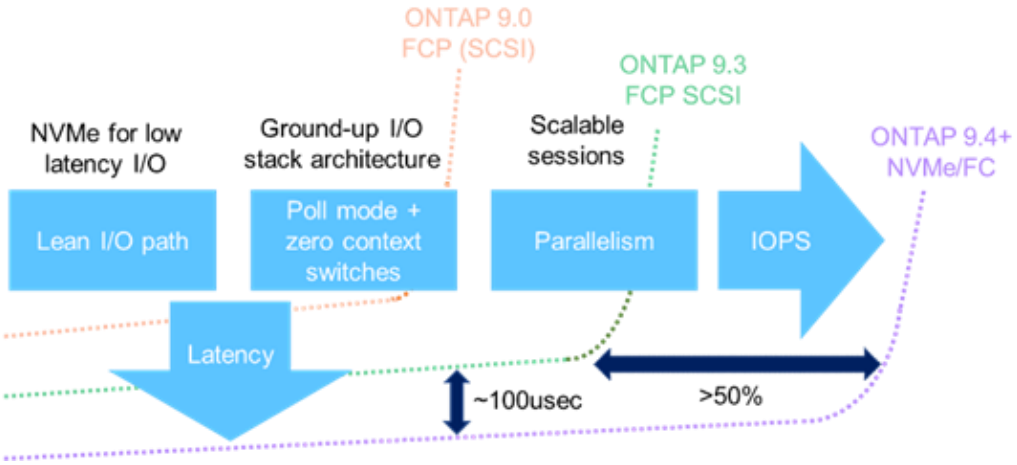
NVMe-oF is fast

NVMe will become an essential part of the modern data center, because it addresses four crucial attributes of data storage performance: IOPS, throughput, latency, and CPU utilization:

- **IOPS** is a measure of how many read or write operations a device can perform per second. It's different for reads than for writes, for sequential versus random operations, and depending on the size of the blocks being stored or retrieved. Most devices will report higher IOPS values when working with small block I/O sizes, such as 4KB or 8KB blocks. But real-world applications often require higher I/O sizes, such as 32KB or 64KB blocks, so it's important to base assessments on relevant I/O characteristics.
- **Throughput** is a measure of how quickly a storage device can read or write data, often specified in gigabits per second. It's typically higher for larger I/O sizes and varies with I/O direction and access type (random or sequential), so again it must be assessed with regard to your real-world operating environment.
- **Latency** is the time between the beginning and the completion of a read or write operation. Storage latency depends on the size of the data being transferred, whether it's sequential or random, whether it's being read or written, and the speed of the network. Low storage latency, especially low read latency, is essential for providing users with a responsive and engaging experience.
- **CPU utilization** is a measure of how many CPU cycles are used to generate the I/O within the latencies and with sufficient throughput to support a given workload or workloads. Unlike the other measures listed above, NVMe's efficiencies reducing CPU utilization is a less well-known benefit of migrating to NVMe from a SCSI-based protocol such as FCP or iSCSI. The benefits of reducing CPU utilization can include being able to consolidate more workloads on a given storage controller and being able to potentially reduce the number of servers that are required to host a specific workload. Both the storage and host-side reductions increase the IT investments and can create rapid returns on investments (ROI) for projects that migrate to NVMe.

Figure 2 and Figure 3 illustrate why NVMe is so efficient and, therefore, fast.

Figure 2) Why is NVMe/FC so fast?



The IOPS and bandwidth improvements are primarily the result of NVMe's flexibility and its ability to take advantage of fast transport technologies to move NVMe commands and data. These transports include:

- **FCP.** Currently available in speeds of 32Gbps and 64Gbps and soon 128Gbps.
- **Remote direct memory access (RDMA) protocol:**
  - Data center fast Ethernet: currently available in 25, 40, 50, 100, and 200Gbps.
  - InfiniBand (IB): currently available with speeds up to 100Gbps.
- **PCI Express 3.0.** Supports eight gigatransfers per second (GTps), which translates to approximately 6.4Gbps.
- **TCP which allows NVMe commands and payloads to use ubiquitous Ethernet networks.** Currently higher-speed Ethernet networks are available in 10, 25,40, 50, 100, and now 200Gbps

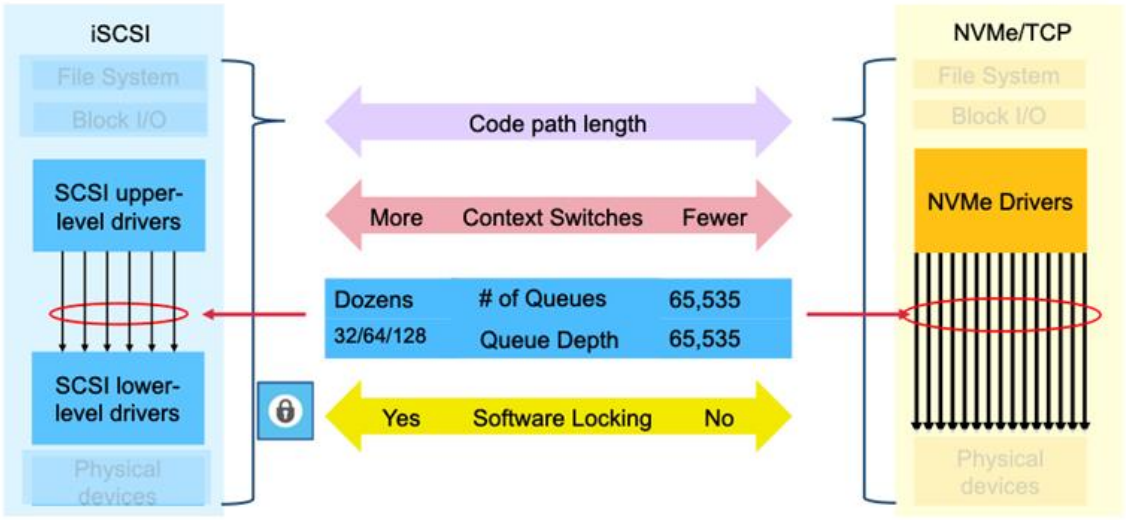
Performance improvements are a result of the massive parallelization possible with NVMe. This parallelization allows the protocol to distribute processing across multiple cores for the concurrent processing of multiple threads.

Latency improvements are a result of a combination of factors, including:

- High parallelism. I/O submission and completion queue pairs are aligned to host CPU cores. Each host/controller pair has an independent set of NVMe queues.
- Streamlining the NVMe command set
- A polling mode driver replacing hardware interrupts
- Elimination of software locks
- Removal of context switches

These factors work together to increase throughput and reduce latency, key metrics for an enterprise's business-critical applications.

Figure 3) Why is the NVMe protocol so efficient?



NVMe as a storage attachment architecture

NVMe is mostly used today for attaching disks and disk shelves. Many storage vendors and suppliers have introduced offerings based on using NVMe as a storage attachment architecture and standard. Technically, in most cases, NVMe is the protocol used to perform I/O, whereas the physical transport is primarily PCIe.

In this scenario, NVMe replaces the SCSI command set with the NVMe command set and frequently replaces SATA or SAS with PCIe to connect drives to the storage controller. NVMe relies on a physical attachment and transport. It uses PCIe as the transport.

NVMe-attached flash offers more bandwidth and reduced latencies because:

- It offers more and much deeper queues: 64k (65,535) queues, each with a queue depth of 64k.
- The NVMe command set is streamlined and, therefore, more efficient than legacy SCSI command sets.

Changing from a SAS 12GB back-end and the SCSI command set to PCIe-attached drives with the NVMe command set improves performance (throughput) and decreases latency for any back-end protocol. This improvement is due to disk access, which is more efficient, requires less processor power, and can be parallelized. Theoretically, performance improvements can improve throughput by approximately 10–15% and reduce latencies by 10–25%. Obviously, differences in workload protocols, other workloads running on the controller, and even the relative busyness of the hosts running I/O will cause these numbers to vary significantly.

Industry’s first end-to-end NVMe — NetApp AFF A800

The AFF A800 all-flash array was the first NetApp array that uses NVMe-attached solid-state drives (SSDs).

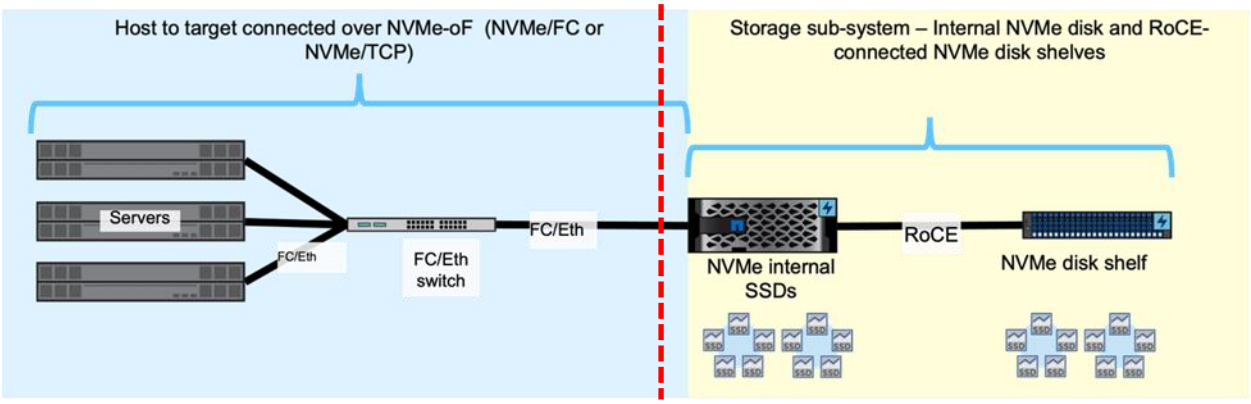
Figure 4) NetApp AFF A800.

If you were to start with a blank slate and set out to design an industry-leading, high-end all-flash array for 2018 and beyond, you would come up with a system such as the AFF A800. Here are just a few of the highlights:

- Industry-first, end-to-end NVMe/FC host-to-flash array over 32Gbps FC; the AFF A800 FC host bus adapters (HBAs) can auto negotiate down to 16 or 8Gbps (N-2)
- Industry-first 100GbE connectivity for front-end connections for NAS and iSCSI protocols
- 2.5PiB effective capacity in a 4U chassis with 15.3TB NVMe SSDs (coming soon)
- 100GbE NetApp MetroCluster IP (MCC IP) for peak performance

The internal NVMe SSDs with their high performance and low latency will speed up any application. Even better, when the AFF A800 is coupled with NVMe/FC, the result is end-to-end NVMe connectivity (Figure 5), which enables organizations to achieve higher performance at lower latencies.

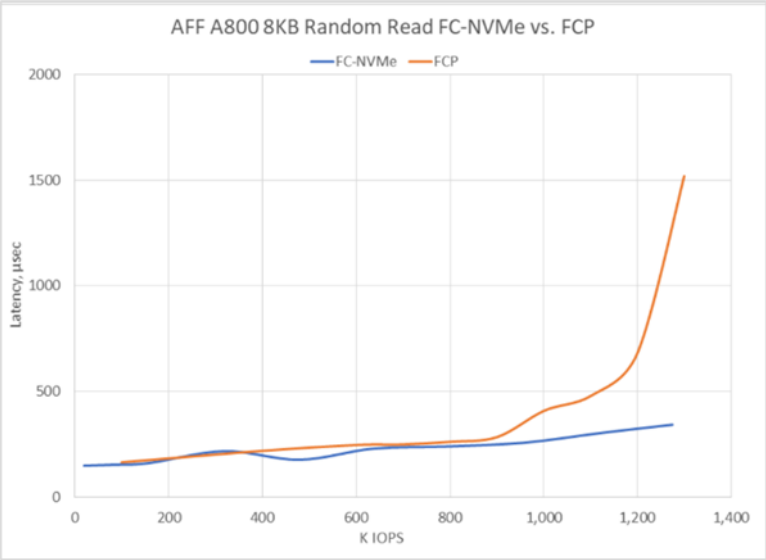
Figure 5) NetApp end-to-end NVMe.



Some customers have been waiting for a storage solution with support for 100GbE. And some might decide to enable 100GbE for NAS and iSCSI protocols along with 32Gbps FC (which NetApp released in 2016) for both FCP and NVMe/FC and end up with a system that delivers excellent application bandwidth.

The AFF A800 is optimal for either SAN-only, NAS-only environments, or a combination of both. For more information, see this short lightboard video. Figure 6 shows the performance improvements that organizations can expect by deploying FCP or NVMe/FC on an AFF A800. Notice that the blue NVMe/FC line stays flat compared with the FCP line, which indicates that NVMe/FC can provide substantially more performance capacity when compared to FCP on the AFF A800. This means you can expect higher throughput when using NVMe/FC on the AFF A800. The knee of the curve for FCP appears to be about 1.2M IOPS at a little over 500ms compared with NVMe/FC, which has substantially more headroom (available performance capacity compared to FCP).

Figure 6) AFF A800 FCP versus NVMe/FC performance comparison.



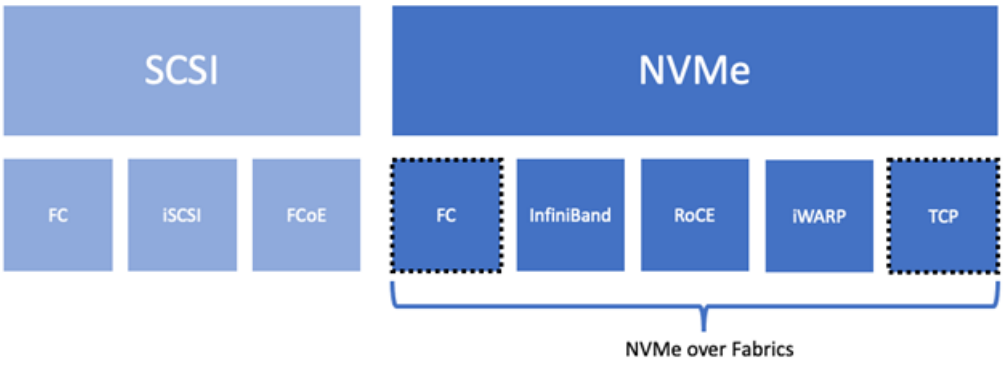
NVMe-oF

NVMe isn't just a storage specification. The NVMe-oF protocol extension encompasses the entire data path, from server to network to storage system. After the HDD bottleneck was removed by replacing HDDs with flash SSDs, another bottleneck appeared: the storage protocols being used to access data, both locally and over SANs.

The NVMe over Fabrics committee introduced NVMe to upgrade local storage access protocols. With the release of NVMe-oF, the committee has added specifications and architectures for using NVMe protocols and command sets over greater distances, using various network and fabric protocols. The result is large performance increases and reductions in latencies for workloads moved from FCP or iSCSI to NVMe-oF specifications such as NVMe/FC.

Implementing end-to-end NVMe requires not just NVMe-attached solid-state media, but also NVMe transport from the storage controller to the host server. The original NVMe specification designed the command set and architecture for the attachments but relies on PCIe (or another physical transport specification) for transport and transmission. It was primarily focused on attaching nonvolatile flash storage technologies to local servers. NVMe replaces SCSI commands and increases both the number of processing queues and queue depths for each of the processing queues. NVMe reduces context switches and is lockless. These enhancements dramatically improve access and response times of NVMe-attached disks, including those using a PCI bus. Figure 7 compares some SCSI and NVMe-oF protocol and transport combinations.

Figure 7) SCSI versus NVMe-oF transports.



NVMe and data fabrics

NVMe defines access protocols and architectures for connecting local nonvolatile storage to computers or servers. NVMe-oF enhances the original NVMe specifications by adding scaling and range improvements. NVMe-oF is the NVMe extension that effectively brings NVMe to the SAN marketplace, because it defines and creates specifications for how to transport NVMe over various network storage transports such as FC, Ethernet, IB, and others.

NVMe-oF ultimately adds NVMe as a new block storage protocol type. It generically specifies transport protocols and architectural specifications that vendors must follow if they develop specific NVMe-oF transports, such as NVMe/FC.

NVMe-oF defines how NVMe can use existing transport technologies such as FC and Ethernet to transport the NVMe protocol over distance and enable the use of networking technologies such as switches and routers. By supporting these transport protocols, NVMe-oF radically improves performance of large-scale storage arrays while increasing parallelization of storage protocols by replacing other protocols, such as:

- **FCP.** SCSI commands encapsulated inside FC frames.
- **iSCSI.** SCSI commands encapsulated in IP/Ethernet frames.
- **FCoE.** FC frames with encapsulated SCSI commands that are in turn encapsulated inside an Ethernet frame.
- With NetApp ONTAP 9.4, NetApp introduced its first NVMe-oF implementation that uses NVMe/FC. In fact, ONTAP 9.4 is the industry's first version of a complete end-to-end (host-to-storage) solution using NVMe/FC.
- ONTAP 9.10.1 added NVMe/TCP as an Ethernet-based NVMe-oF transport.
- Figure 5 illustrates ONTAP 9.4 delivers end-to-end NVMe flash with NVMe-FC and the new AFF A800.

NVMe-oF is primarily intended to extend the NVMe protocol to data networks and fabrics. It defines the access architectures and protocols used to attach compute to block-based storage. It is easiest to think of this as an update to current block protocols such as:

- **FCP.** FCP encapsulates SCSI Command Descriptor Blocks (CDBs) inside an FC frame. FC defines a transport method, whereas FCP specifically means using the FC protocol to encapsulate SCSI (CDBs). Currently, FCP is the most common SAN protocol. FCP fabric (network) speeds range from 1 to 32Gbps; 8, 16, and 32Gbps are the most encountered speeds.
- **iSCSI.** iSCSI was defined by the Internet Engineering Task Force (IETF) first in RFC 3270 Internet Small Computer Systems Interface. This document was superseded by RFC 7143 Internet Small

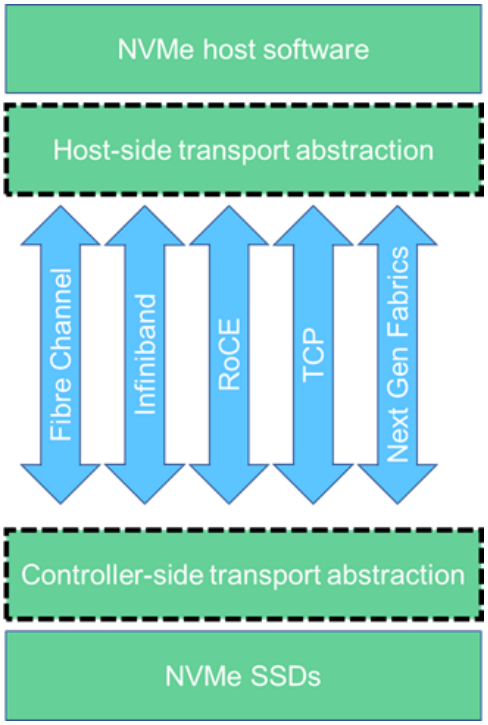
Computer System Interface (iSCSI) Protocol (Consolidated), which updated and modernized the original specification introduced in 2004.

NVMe-oF provides specifications, architectural standards, and models that can be used to transport NVMe inside various current transports. NVMe-oF transports include:

- **NVMe/FC.** NVMe using FC as the transport. For details, see the section titled, “NVMe over Fibre Channel.”
- **NVMe/TCP NVMe encapsulated in a TCP datagram.** NVMe/TCP has the potential to be the most popular NVMe over Ethernet variant—and could ultimately become the logical replacement for iSCSI. Like iSCSI, NVMe/TCP uses standard NICs and Ethernet switches, which makes it an attractive option for an environment that would like to introduce NVMe-oF on Ethernet without having to use specialty building blocks like RDMA NICs (RNICs) and data center bridging (DCB) switches required to support RDMA over Converged Ethernet (RoCE).
- **NVMe transport using RDMA.** There are several transports that support RDMA:
  - **NVMe over InfiniBand (NVMe/IB).** This solution uses IB, which can currently support 100Gbps, as an ultra-high-speed transport. Although it is incredibly fast, IB is expensive and has both distance and scaling limitations. The first enterprise-class storage array to offer NVMe-oF (using an NVMe/IB target) is the NetApp EF570 array, which can deliver 1M IOPS and 21GBps, at less than 100ms in a 2U platform. For more information, see the NetApp EF570 All-Flash Array Datasheet.
  - **RDMA over Converged Ethernet (RoCE):**
    - Internet Wide-Area RDMA Protocol (iWARP) transports RDMA by using Direct Data Placement Protocol (DDP), which is transported by using either TCP or Secure TCP (STCP). DDP transmits data in streams and doesn’t segment it to fit into TCP protocol data units.
    - RoCE offers lower latencies because it doesn’t require TCP. RoCE requires an Ethernet switch that supports DCB and Priority Flow Control (PFC). DCB switches are not the same as standard Ethernet switches and tend to be more expensive. Hosts and storage controllers need to have RNICs installed. These requirements are likely to limit cloud adoption of RoCE. There are two variations of RoCE:
      - RoCE v1, the original RoCE specification, defines a data link layer protocol that allows communication between initiators and targets in the same subnet. RoCE is a link layer protocol that can’t be routed between subnets.
      - RoCE v2 is an internet layer protocol that uses User Datagram Protocol (UDP) over either IPv4 or IPv6. It is a layer 3 internet layer protocol that can be routed between subnets. Because UDP doesn’t enforce in-order delivery and the RoCE v2 specification doesn’t allow out-of-order packet delivery, the DCB network must deliver packets in the order they were sent. RoCE v2 also defines a flow-control mechanism that uses Explicit Congestion Notification (ECN) bits to mark frames and Congestion Notification Packets (CNPs) to acknowledge receipts.
  - **RDMA over iWarp.** An enhancement to support using RDMA.

Figure 8 illustrates the NVMe-oF stack and some of the networks and fabrics that can be used as transports.

Figure 8) NVMe can use multiple network transports.



NVMe can also refer to NVMe-oF, a standard enhancement that specifies how to encapsulate the NVMe command set and data payload inside of a wide variety of common networking and fabric protocols. Figure 9 illustrates an NVMe-oF stack and a few of these protocols/transports and Figure 9 illustrates how NVMe-oF can be used to extend the diameter of and number of objects that can be networked together with NVMe-oF.

Figure 9) NVMe over Fabrics (NVMe-oF).

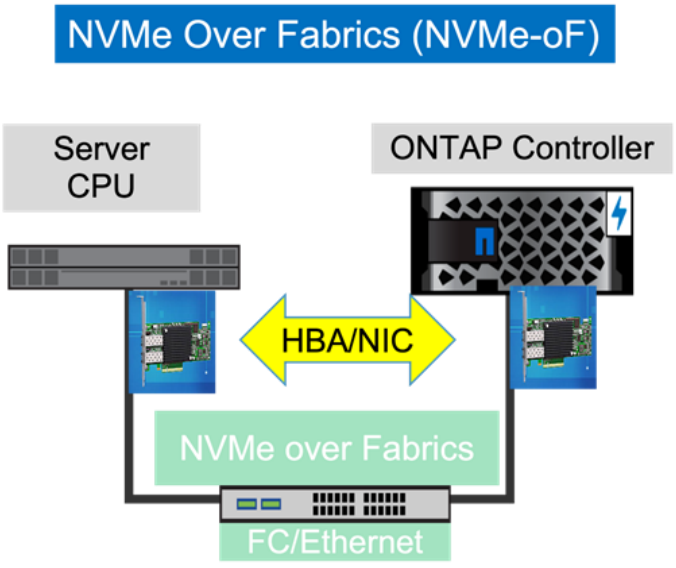


Figure 10 illustrates the differences between a native FC frame versus an NVMe encapsulated inside of an FC frame. As you can see, NVMe/FC simply replaces the SCSI-3 command descriptor block (CDB)

with a NVMe protocol command. This simple replacement allows NVMe protocols to be transported inside of an FC fabric, which increases both the diameter and number of objects that can be networked and provides a very reliable transport.

Figure 10) FCP versus an NVMe/FC frame.

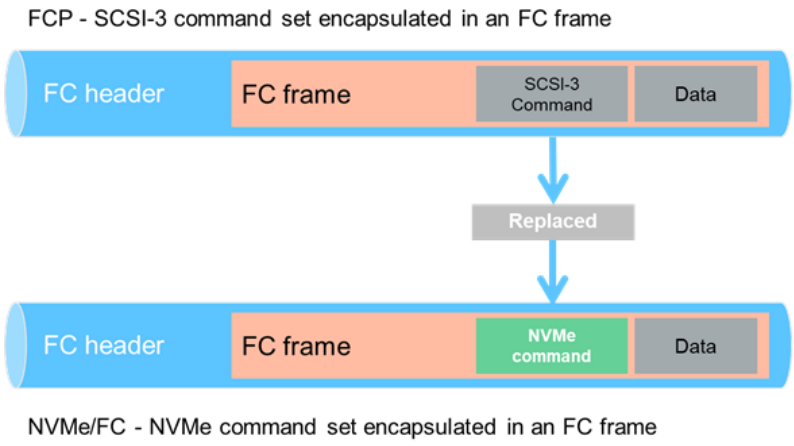


Figure 11 illustrates NVMe/TCP which is the NVMe command set encapsulated and transported inside of a TCP datagram.

Figure 11) iSCSI versus NVMe/TCP datagram.

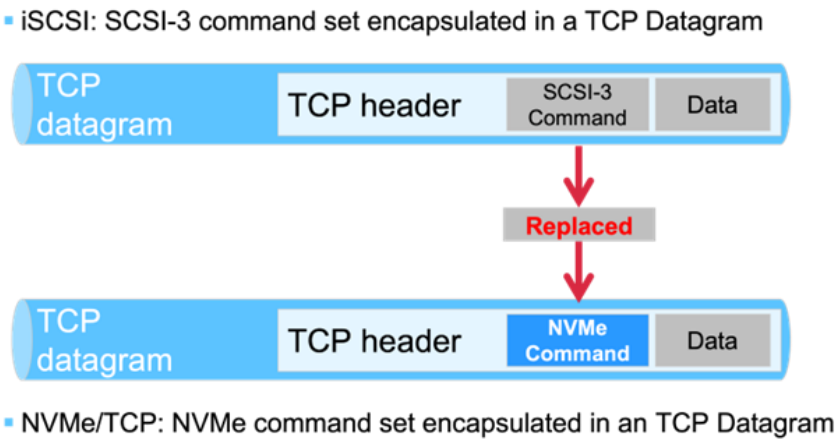


Figure 12 is another view of NVMe/TCP showing that both the NVMe protocol and payload are both encapsulated inside the data/payload portion of a TCP datagram. As you can see, NVMe-oF encapsulates NVMe commands and a data payload inside of a variety of existing networking or fabric protocols to extend NVMe and transport it over distance.

Figure 12) NVMe in a TCP datagram.

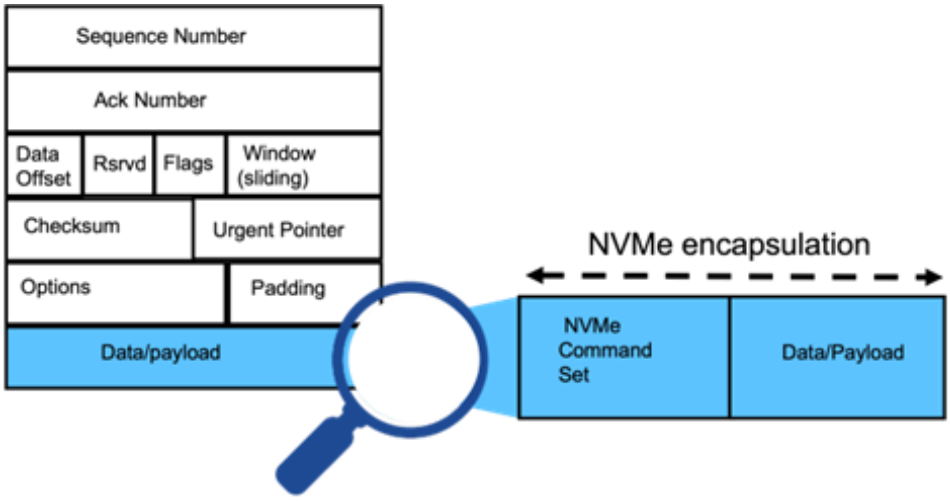
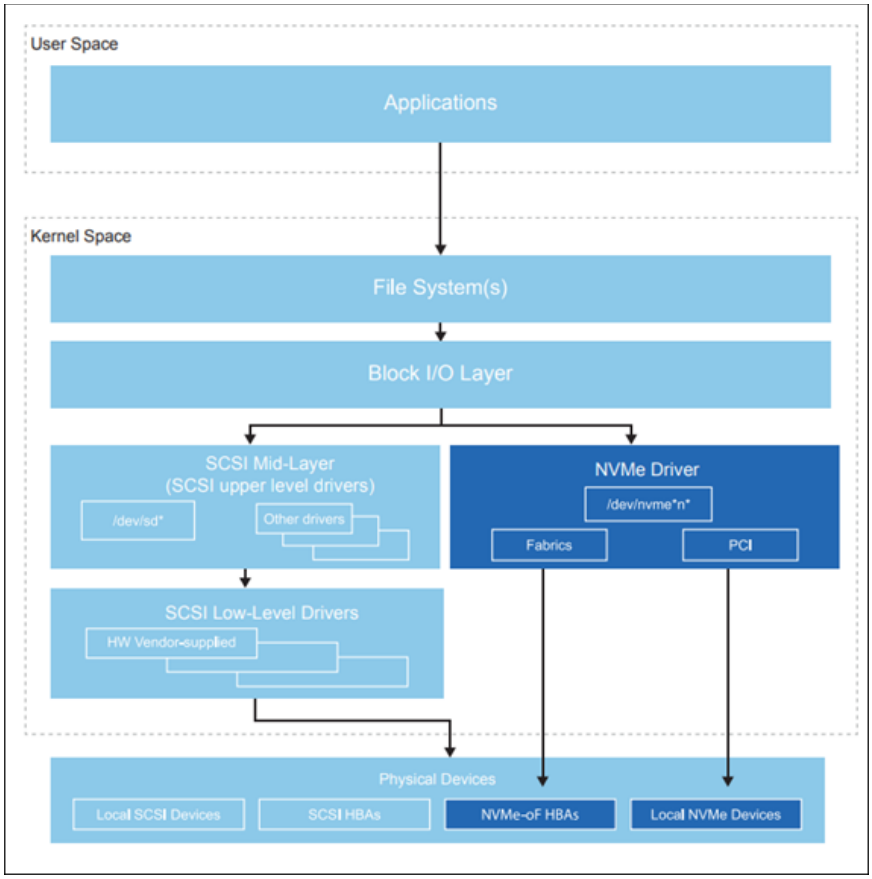


Figure 13 compares a SCSI and NVMe protocol stack. Notice how much shorter the NVMe stack is compared with the SCSI stack.

Figure 13) SCSI versus NVMe stack architecture.



Leveraging open-source frameworks

NetApp engineering took advantage of several open-source architectural specifications to rapidly develop the ONTAP NVMe-oF target and support the seamless interoperability with other NVMe hardware and software. While adopting and complying with these specifications, NetApp has also been active in several standards organizations including NVM Express, Inc., and the INCITS, where NetApp has donated designs, specifications, and guidance back to the open standards community. In addition to contributions of code and designs back to NVM Express and INCITS, NetApp has adopted and contributed to numerous reference designs and architectural standards, including:

- Data Plane Development Kit (DPDK)
- Storage Performance Development Kit (SPDK)

NVMe and high availability

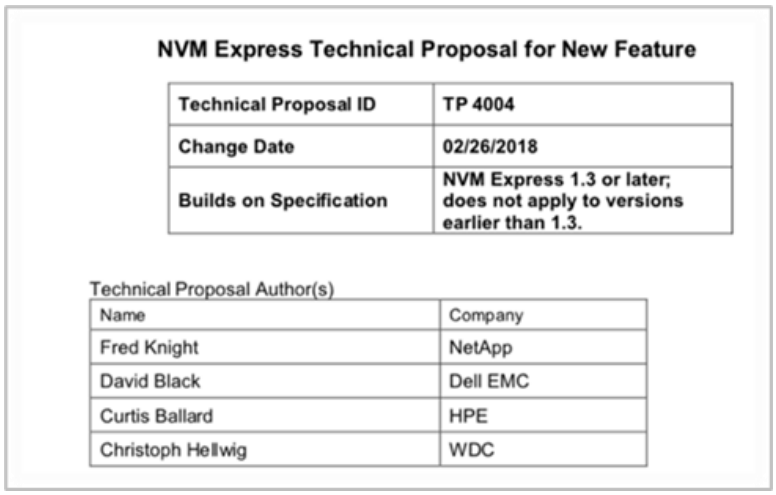
The NetApp NVM Express committee representative, Fred Knight, was the lead author who submitted technical proposals TP 4004 and TP 4028, which define a functional high-availability error-reporting and failover protocol. The new protocol, Asymmetric Namespace Access (ANA), was ratified in March of 2018.

ANA

Like ALUA, ANA requires both an initiator-side and target-side implementation for it to be able to provide all the path and path state information that the host-side multipathing implementation needs to work with the storage HA multipathing software used with each operating system stack. ANA requires both the target and initiator to implement and support ANA to function.

NVMe/FC relies on the ANA protocol to provide multipathing and path management necessary for both path and target failover. The ANA protocol defines how the NVMe subsystem communicates path and subsystem errors back to the host so that the host can manage paths and failover from one path to another. ANA fills the same role in NVMe/FC that ALUA does for both FCP and iSCSI protocols. ANA with a host operating system path management such as MPIO or Device Mapper Multipathing (DM-Multipath) provide path management and failover capabilities for NVMe/FC. Figure 14 and Figure 15 show the cover pages of both technical proposals submitted to NVM Express, Inc. for ratification. Figure 16 shows the INCITS T11 cover sheet for the T11-2017-00145-v004 FC-NVMe specification. The INCITS T11 committee is responsible for standards development in the areas of Intelligent Peripheral Interface (IPI), High-Performance Parallel Interface (HIPPI), and FC.

Figure 14) TP 4004: ANA base proposal (ratified 3/18).



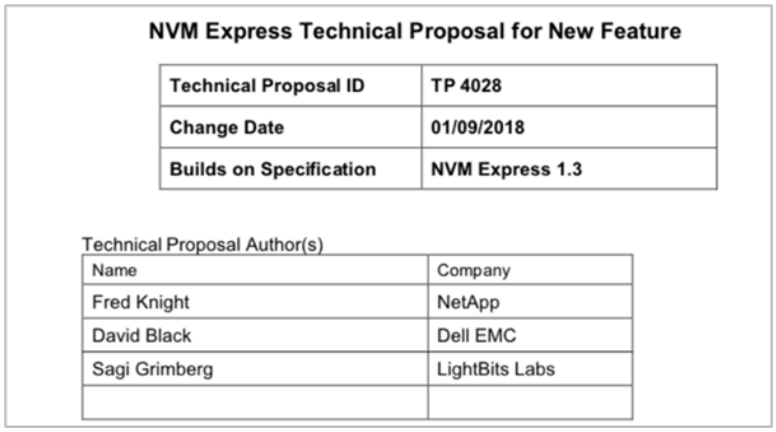
NVM Express Technical Proposal for New Feature

Technical Proposal ID	**TP 4004**
Change Date	**02/26/2018**
Builds on Specification	**NVM Express 1.3 or later; does not apply to versions earlier than 1.3.**

Technical Proposal Author(s)

Name	Company
Fred Knight	NetApp
David Black	Dell EMC
Curtis Ballard	HPE
Christoph Hellwig	WDC

Figure 15) TP 4028: ANA path and transport (ratified 1/18).



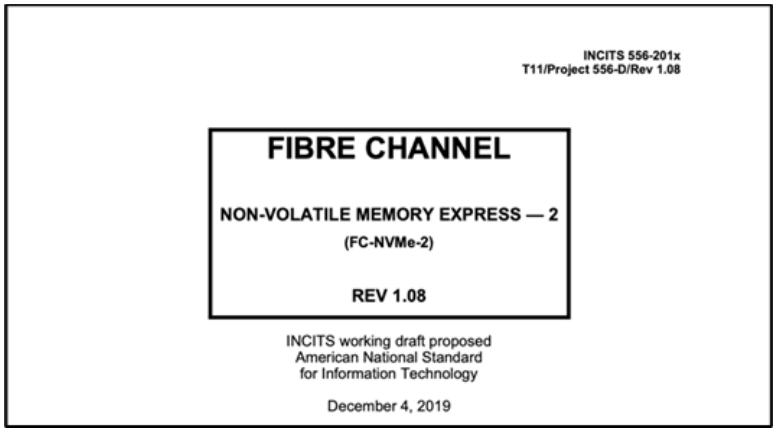
NVM Express Technical Proposal for New Feature

Technical Proposal ID	**TP 4028**
Change Date	**01/09/2018**
Builds on Specification	**NVM Express 1.3**

Technical Proposal Author(s)

Name	Company
Fred Knight	NetApp
David Black	Dell EMC
Sagi Grimberg	LightBits Labs

Figure 16) INCITS FC-NVMe-2 Rev 1.08 T11-2019-00210-v004 defines NVMe command and data transport using FC standards.



INCITS 556-201x
T11/Project 556-D/Rev 1.08

FIBRE CHANNEL

NON-VOLATILE MEMORY EXPRESS — 2
(FC-NVMe-2)

REV 1.08

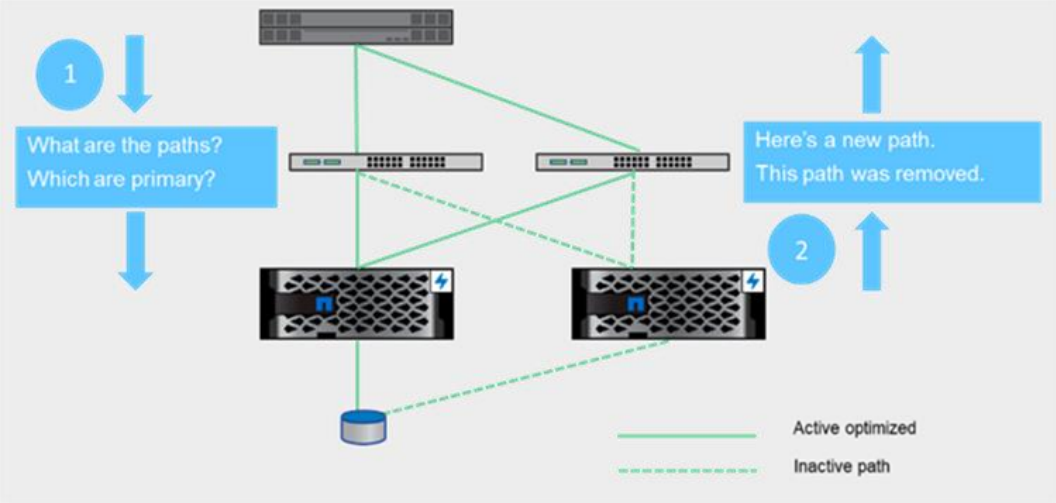
INCITS working draft proposed
American National Standard
for Information Technology

December 4, 2019

ANA has the following two components:

- The initiator-side ANA queries the target for path attributes, such as primary versus secondary. This data is used by initiator MPIO stack to optimize paths.
- The target-side ANA communicates path state changes.

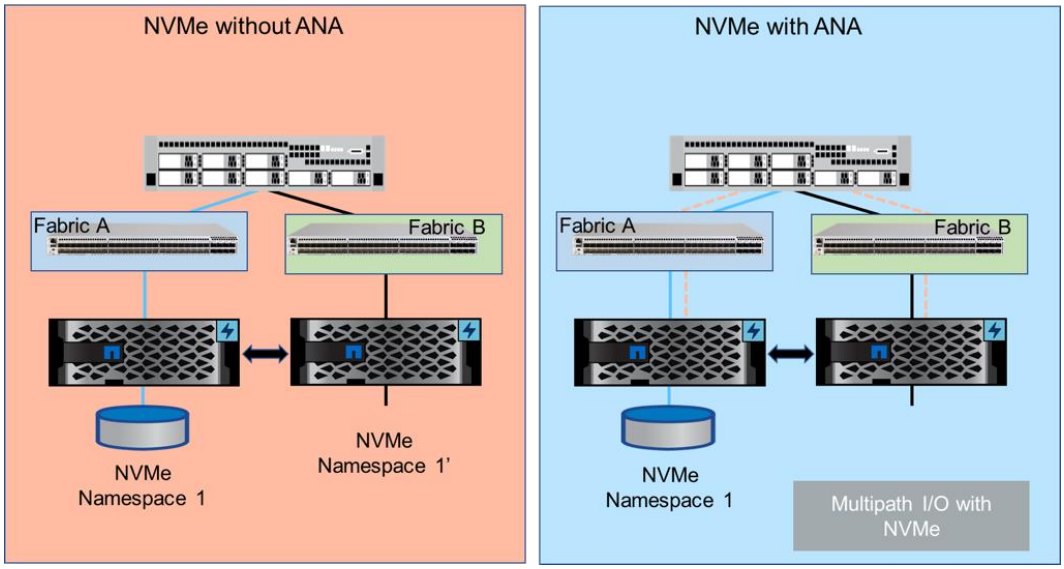
Figure 17) NVMe/FC storage failover: ONTAP 9.5 introduces ANA.



ONTAP offers remote I/O support, which looks like our ALUA implementation.

Figure 18 shows the comparison of NVMe/FC with and without ANA.

Figure 18) A comparison of NVMe/FC with and without ANA.



NVMe over Fibre Channel

NetApp customers build and manage some of the biggest data centers in the world. Over the next few years, they will all upgrade their data storage systems to NVMe. But which NVMe-oF transport will they choose?

Although RDMA transports are important, it's likely that NVMe over Fibre Channel (NVMe/FC) will initially be the dominant transport in data center fabrics. Using FC as a transport provides these benefits:

- Almost all high-performance block workloads are currently running on FCP today.
- Almost all (~70%) of these organizations are currently using a SAN with FCP.

- Most performance-focused workloads currently have either Gen 5 or 6 (16Gbps or 32Gbps) switches already in their fabrics.
- There is a small footprint of 25/50/100Gbps Ethernet switches currently installed in data centers that would form the backbone infrastructure for any RDMA over IP, TCP, RoCE, or other similar transports.
- Both FCP and NVMe/FC can use the same physical components to transport SCSI-3 and NVMe concurrently.
- Many NetApp customers already own all the hardware necessary to run NVMe/FC now and will be able to start using NVMe/FC with a simple software upgrade to NetApp ONTAP 9.4 or later.

Both FCP and NVMe can share all the common hardware and fabric components and can coexist on the same wires (technically optical fibers), ports, switches, and storage controllers. Thus, an organization can easily transition to NVMe, because they can do so at their own pace. In fact, if they have recently upgraded switches and directors (that is, Gen 5/6), they will be able to upgrade nondisruptively to ONTAP 9.4 or later.

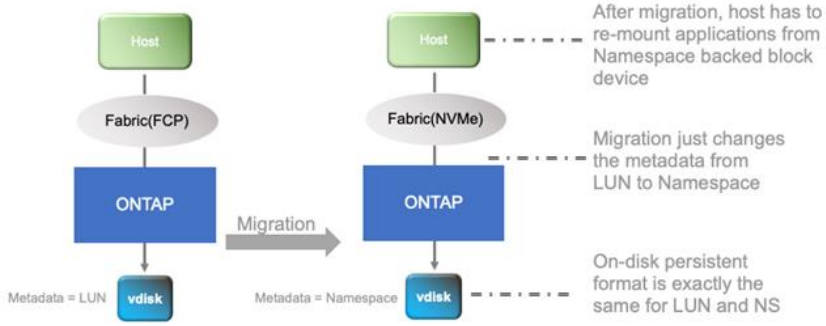
NVMe/FC looks very much like FCP, which is defined as encapsulating SCSI-3 CDB inside FC frames. The reason both look so similar is that NVMe/FC swaps out the older SCSI-3 CDB for the new, streamlined NVMe command set. With this simple replacement, NVMe/FC offers substantial improvements to throughput and latency.

NetApp launched NVMe/FC first, because it is the dominant SAN protocol, with about three times the adoption of the next-largest protocol—iSCSI. This means organizations already have significant investments in FC infrastructure and skill sets. Furthermore, when performance is the primary concern, FC SAN is almost always the transport of choice.

Because NVMe/FC simply swaps command sets from SCSI to NVMe, it is an easy transition to make. NVMe/FC uses the same FC transport and therefore the same hardware from the host, through the switch and all the way to the NVMe/FC target port on the storage array. Thus, NVMe/FC implementations can use existing FC infrastructure, including HBAs, switches, zones, targets, and cabling.

Although ONTAP uses a NVMe/FC LIF that is separate from the FCP LIFs, both LIFs can be hosted on the same physical HBA port at both the host initiator and storage target. NVMe/FC and FCP can share the same physical infrastructure concurrently, so the same physical port, cable, switch port, and target port can simultaneously host and transmit both FCP and NVMe/FC frames. The two protocols are separated at the logical rather than physical layers, making adoption and transition from FCP to NVMe/FC simple and seamless. You can migrate workloads from FCP to NVMe/FC at your own pace without having to disrupt your production operations or run multiple parallel infrastructures. Because NVMe/FC and FCP use the same physical infrastructure, NetApp customers can nondisruptively implement a new technology to improve performance, introduce new workflows, and improve the performance of existing workflows that are transitioned to it, as shown in Figure 19.

Figure 19) Adopt modern technology nondisruptively.



NetApp NVMe/FC release announcement

The ONTAP 9.4 Release Notes includes the following information about the support for NVMe/FC:

ONTAP 9.4 introduces NVMe/FC, an industry first.

NVMe/FC is a new block-access protocol that serves blocks to the host, such as FCP and iSCSI, using the NVMe command set instead of SCSI. The NVMe architecture constructs a lean command set and scalable sessions, which enable significant reductions in latency and increases in parallelism, making it well-suited to low-latency and high-throughput applications such as in-memory databases, analytics, and more.

NVMe/FC can be provisioned and configured through on-box NetApp OnCommand® System Manager software (point a web browser at the IP address of the cluster management or any of the node management ports) or the CLI.

End-to-end NVMe/FC connectivity from the host through SAN fabric to NetApp AFF controllers is necessary to get the maximum performance using this new protocol. Consult the NetApp Interoperability Matrix Tool (IMT) to verify the latest supported solution stack for ONTAP.

Note: The ONTAP implementation of NVMe/FC requires application-level high availability. If a controller loss or path failure occurs, the application host must manage path failover to its (application) HA partner. This limitation exists because the NVMe multipathing specification called ANA, analogous to ALUA in SCSI protocol, was still under development.

While implementing NVMe/FC, NetApp helped design the ANA protocol in the NVMe forum, where it was recently ratified. A future release of ONTAP will offer support for this enhancement.

Note: Some NetApp documentation and UI might refer to NVMe over Fibre Channel as FC-NVMe, rather than the now standard and trademarked term NVMe/FC. FC-NVMe and NVMe/FC are interchangeable abbreviations that both refer to NVMe over Fibre Channel.

NVMe over TCP (NVMe/TCP)

ONTAP 9.10.1 added ONTAP's first NVMe-oF Ethernet-based protocol NVMe/TCP. NVMe/TCP joins NVMe/FC as ONTAP's second supported NVMe block protocol. NVMe/TCP uses Transmission Control Protocol (TCP) over Ethernet. The combination of TCP and Ethernet for transport means that NVMe/TCP can be deployed anywhere TCP and Ethernet exists. Unlike FC, which is almost always confined to owned data centers, TCP and Ethernet are ubiquitous. Furthermore, NVMe/TCP doesn't have any real networking hardware limitations. It is supported on virtually any Ethernet networking equipment. For instance, you could run NVMe/TCP over 1Mbps network interface cards (NICs) and a 1Mbps switch and/or routers. The solution would work, although you would probably be dissatisfied with performance. The point is that because there are no real hardware requirements and the nearly universal nature of Ethernet and TCP, NVMe/TCP can be run almost anywhere. This is particularly important with the rapid growth of the cloud. NVMe/TCP joins iSCSI as SAN or block protocols that can connect between corporate data centers, third-party hosting, and a variety of cloud end points. NVMe/TCP's biggest strengths include all the efficiency gains seen when moving from a SCSI-based protocol such as FC or iSCSI to an NVMe-based protocol such as NVMe/FC or NVMe/TCP, and the flexibility and portability of TCP and Ethernet. As of this writing (January 2023, Red Hat Enterprise Linux, SUSE Enterprise Linux, Oracle Linux, and VMware ESXi support NVMe/TCP. Other operating systems are expected to add support going forward, therefore you should always check the NetApp IMT for a listing of OSs that currently support NVMe/TCP.

Note: Like SCSI, NVMe protocols are transport agnostic. That means that you can access an NVMe namespace by using one or more NVMe-oF protocols, either separately or concurrently.

Getting started with NVMe

NetApp forecasts that most NVMe-oF initial uptakes will be from organizations that are experimenting with testing and qualifying NVMe/FC and/or NVMe/TCP in their organizations before migrating production workloads to it. Most enterprise storage teams are risk averse and want to run thorough qualification and testing before putting a new protocol into production. Additionally, NetApp expects that most early adopters will wait until ANA is added to the ONTAP NVMe/FC target and the desired host OS's support of ANA as part of their NVMe/FC support. The only adopters likely to be unconcerned about the lack of ANA are those who have applications that manage high availability at the application layer rather than storage layer. As mentioned previously, some of these applications might include MongoDB or Oracle ASM.

When should I choose to deploy NVMe/FC versus NVMe/TCP

In most circumstances, new NVMe-oF deployments are rarely greenfield deployments. Most are additions to existing data centers operations. That means there is likely to be some existing infrastructure that might militate in favor of one transport over another. Of course, it is also true that there is no requirement that you can't mix, and match transports based on the requirements of various workflows and infrastructures already installed.

The first thing to understand is that both NVMe-oF transports coexist quite well together and with SCSI-based block protocols such as iSCSI and FCP. Furthermore, NVMe/FC can use the exact same components concurrently with FCP, the same is also true of the two Ethernet protocols, iSCSI, and NVMe/TCP. In fact, you could very easily use the same host to access a dozen LUNs using FCP and/or iSCSI and another dozen namespaces using either or both NVMe/FC and NVMe/TCP using the exact same physical HBA and NIC ports, cables, and switches and ONTAP controllers. This makes both coexistence and migrating among the four block protocols quite easy.

When considering NVMe-oF transports to deploy with a specific workflow, the question is what you so want to optimize? If you are looking for sheer performance within an owned data center, NVMe/FC will probably be your first choice. When trying to optimize flexibility and portability, especially where FC fabric doesn't exist or there is a requirement to connect between an owned data center and a cloud endpoint, NVMe/TCP will likely be first your choice.

ONTAP feature support and coexistence

Table 2 and Table 3 list the ONTAP tools and features that are currently supported, coexist with, or are not supported with NVMe-oF. All the items in these tables were accurate at the time of this writing; however, over time, more of the features in the unsupported table are likely to be moved to supported as the ONTAP NVMe-oF targets continue to gain functionality and performance.

Table 1) SCSI and NVMe terms.

FCP/iSCSI	NVMe-oF remarks	Remarks
FCP - Worldwide Port Name (WWPN) iSCSI – iSCSI Qualified Name (IQN)	NVMe Qualified Name (NQN)	Unique identifier
SCSI target	NVMe subsystem	Entity with storage objects and access points
Port	Port	Access point for communication
I_T nexus	NVMe controller (with multiple queue pairs)	Session between initiator and target
LUN	Namespace	Storage object

FCP/iSCSI	NVMe-oF remarks	Remarks
Asymmetric logical unit access (ALUA)	ANA	Asymmetric access characteristics

Table 2) ONTAP features that are either supported by NVMe or can coexist with it.

ONTAP release	NVMe-oF Feature
9.4	• NVMe/FC • Single node only (such as no high availability) • 4K Block Size
9.5	• 2-node HA with Asymmetric Namespace Access (ANA) • NVMe-oF Licensing
9.6	• Copy and Write (CAW) • 512b Block Size • Read-only Namespace • Additional log pages like Changed NS List & Vendor Specific Log pages
9.7	• NVMe-oF Sync SnapMirror
9.8	• LUN/Namespace coexistence in the same SVM
9.9	• NVMe/FC in ASA • Large Namespace (extending up to 128TB) in ASA • NVMe/FC VMware vSphere Virtual Volumes (vVols) • NVMe-oF Abort • NVMe/FC on 4-node array
9.10.1	• NVMe/TCP • Namespace Resize • NVMe-oF Cancel • NVMe-oF on FAS • IPSec on NVMe/TCP
9.11.1	• NVMe/TCP Perf enhancements • LUN <-> Namespace bidirectional in-place conversion • NVMe-oF Scale enhancements (up to 12-node support, and so on) **Note:** For more information, see the NetApp HWU.
9.12.0	• NVMe/TCP in Cloud Volumes ONTAP (Amazon Web Services [AWS] and Azure) and FSx
9.12.1	• NVMe/TCP in GCP Cloud Volumes ONTAP • NVMe-oF in-band Authentication • NVMe/FC on MCC IP • NVMe-oF Sync SM with NDO

Table 3) ONTAP features not currently supported by NVMe.

Feature	Notes
NVMe-oF Symmetric Active/Active on NetApp All SAN Array (ASA)	Only FCP and iSCSI is A/A in ASA, and not NVMe-oF
Namespace quality of service (QoS)	QoS only supported at Vol and SVM level, and not on a NS level

Feature	Notes
Namespace move	Not allowed (enforced)
NVMe/TCP vVols	Not supported (only NVMe/FC vVol currently supported)
NVMe/TCP on MCC IP	Not supported (NVMe/FC currently supported on MCC IP 4-pack)
TLS support in NVMe/TCP	Not supported
NVMe-oF in NetApp SnapMirror Business Continuity	Only FCP and iSCSI supported in SnapMirror Business Continuity currently
SVM Migrate	No block support
Foreign LUN Import (FLI)	FLI performs all migrations using FCP

Note: Imported LUNs can be converted to namespaces after the import finishes using the built-in bidirectional SCSI LUN to NVMe namespace in-place conversion utility. The utility is very fast because it is only changing metadata about the LUN or namespace.

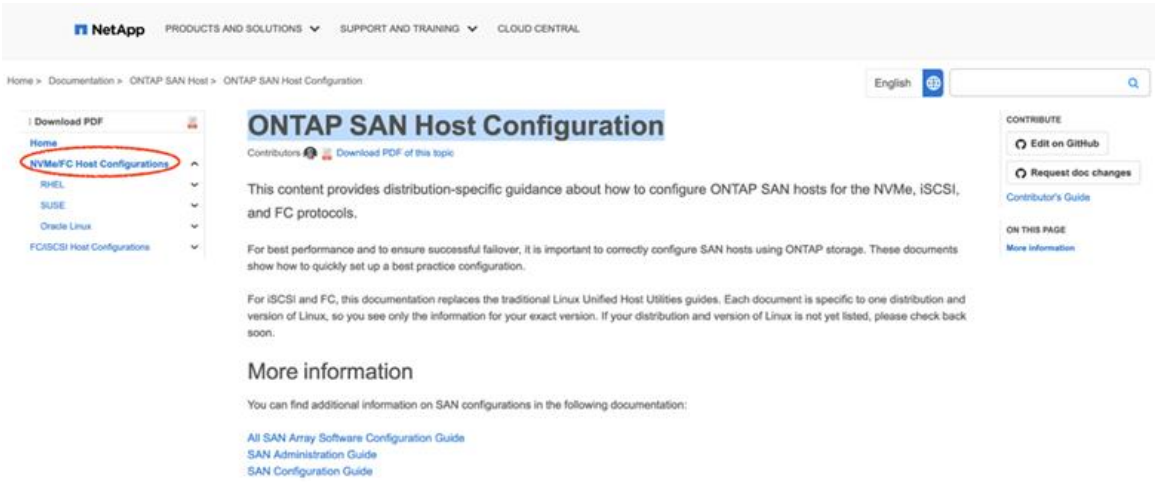
Interoperability

In partnership with SUSE and Broadcom, NetApp was the first to market with a production-ready NVMe/FC offering. At the time of the NetApp ONTAP 9.4 release, the following components were qualified from four vendors who partnered to introduce NVMe/FC to the market:

- SUSE Enterprise Linux version 12 SP3
- Broadcom/Emulex LPe3200X HBA
- Broadcom Brocade Generation 5 or 6 (16Gbps or 32Gbps switches) running Fabric OS 8.1.0a or later
- NetApp AFF A300, AFF A700, AFF A700s, or AFF A800 systems and ONTAP 9.4 with at least one 32Gbps target adapter in each node

Since then, multiple new ONTAP releases and the number of host OS vendors have added NVMe/FC and NVMe/TCP support on new releases of their OSs. For a current list of supported configurations, see the IMT. OS configuration procedures can be found on the ONTAP SAN Host Configuration page (under the NVMe/FC Host Configuration section). An example of the NVMe/FC Host Configurations page is illustrated in Figure 20.

Figure 20) NVMe/FC Host Configurations page.



Note: For specific versions of software firmware and drivers, see the NetApp IMT. More items will be added to the IMT as components are tested and qualified. Make sure to always refer to the NetApp IMT for the latest interoperability details.

NetApp Interoperability Matrix Tool

Verify that your entire planned or current configuration matches one of the configurations in the NetApp IMT. This is important because NVMe-oF is undergoing rapid development on the following three different axes:

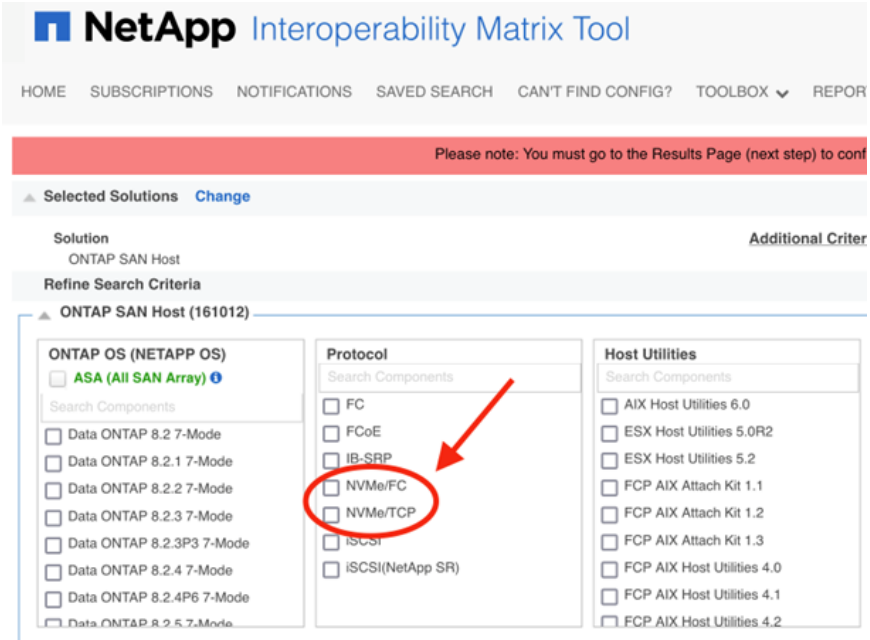
- NetApp NVMe-oF targets
- Any associated switches, HBAs, and so on
- Current host OS support for NVMe-oF include:
  - Linux (RHEL, Oracle Linux, and SLES)
  - VMware ESXi
  - Windows

NVMe/FC-only

- HBA (>32G – Gen6 or later) match driver, firmware:
  - Broadcom (Emulex)
  - Marvel (Qlogic)
- FC switches (>16G Gen 5 or later) switch OS:
  - Broadcom (Brocade)
  - Cisco

Any departure from the IMT-listed configurations is likely to perform inconsistently and unexpectedly. The IMT has added a new protocol filter, NVMe/FC (shown in Figure 21), which can be used to check NVMe/FC qualified configurations.

Figure 21) New IMT NVMe/FC and NVMe/TCP protocol filters.



Converting LUNs to namespaces or namespaces to LUNs

ONTAP 9.11.1 adds a built-in in-place bidirectional LUN to namespace conversion utility. Conversions between LUNs and namespaces are both very fast (only metadata is modified) and very easy. This will ease adoption of NVMe and will make migrating between SCSI and NVMe protocols more seamless.

Feature highlights

- In-place LUN <-> namespace conversion within the same ONTAP volume
- No data copy involved; actual user data remains untouched and only requires metadata to be updated
- Permits data management and protection continuity since access to existing Snapshot copies (created pretransition) is not lost
- Preserves block size as well as identifiers (such as serial number and UUID) during conversion
- No performance regression after the conversion (compared to when newly created on a similar platform configurations)
- Host remediation required to complete the end-to-end LUN <-> Namespace conversion process

Feature restrictions

- Bulk conversion of LUNs or namespaces in a single command is not permitted
- Mapped LUN cannot be converted to a namespace (and vice versa)
- LUN in an FLI relationship cannot be converted to a namespace
- LUN in a SnapMirror Business Continuity relationship cannot be converted to a namespace
- LUN in a MetroCluster configuration cannot be converted to a namespace
- LUN that has vVol bindings or acting as a PE cannot be converted to a namespace
- LUN having a non-zero prefix and/or suffix stream cannot be converted to a namespace (because ONTAP doesn't support prefix/suffix stream for a namespace)
- A namespace with 4K block size cannot be converted to a LUN (because ONTAP doesn't support 4K LUNs)
- Only LUNs with valid and supported `os_types` for namespace can be converted

Table 4) LUN <-> namespace conversion utility feature support.

LUN os_type	Prefix size	Suffix size	LUN <-> NS permitted?
vmware	0	0	Yes
hyper_v	0	0	Yes
windows_2008	0	0	Yes (to 'windows')
windows_gpt	17 (pre ONTAP 9.8) 0 (ONTAP 9.8 onwards)	0	Yes (only if prefix is 0, to 'windows')
windows	31.5 (pre ONTAP 9.8) 0 (ONTAP 9.8 onwards)	0	Yes (only if prefix is 0)
linux	0	0	Yes
xen	0	0	Yes
solaris	1 * cylinder_size (ONTAP 9.6 and earlier) 0 (ONTAP 9.6 and later)	2 * cylinder_size (pre ONTAP 9.6) 0 (ONTAP 9.6 onwards)	No (unsupported NVMe os_type)
solaris_efi	17 (pre ONTAP 9.8) 0 (ONTAP 9.8 and later)	0	No (unsupported NVMe os_type)

LUN os_type	Prefix size	Suffix size	LUN <-> NS permitted?
hpux	0	0	No (unsupported NVMe os_type)
aix	0	0	No (unsupported NVMe os_type)
netware	0	0	No (unsupported NVMe os_type)
openvms	0	0	No (unsupported NVMe os_type)
image (default)	0	0	No (unsupported NVMe os_type)

For more information about converting between LUNs and namespaces, see Appendix D: Converting between LUNs and namespaces.

Best practices for NVMe-oF

Best practices for NVMe/FC

Regardless of whether an organization chooses to test and qualify or use NVMe/FC workloads in production, all teams should follow general FCP SAN best practices. These best practices apply because NVMe/FC uses FC as a transport. NetApp SAN best practices can be found in TR-4080: Best Practices for Scalable SAN.

Fabric and switch configuration and operational best practices

NVMe/FC doesn't require any special configurations or best practices that differ from the general Brocade or Cisco FC switch and fabric best practices. Single-initiator zoning is a best practice. Another best practice is to use WWPNs to assign zone memberships (instead of switch-port-based zone memberships or hard zoning).

Best practice

Run Brocade's SAN Health tool to collect fabric configuration details. SAN Health verifies that your fabrics are healthy and error-free. SAN Health also automatically documents and visualizes your fabrics. SAN Health can be used on any fabric regardless of whether any Brocade switches are present.

Contact your NetApp or partner account team for copies of SAN Health and the video and guide.

NVMe-oF best practices: Pathing

To avoid any interface single points of failure, NetApp strongly recommends that you provision two paths per SVM, per node, per fabric. ONTAP will not allow storage administrators to create more than two LIFs per node/per SVM. This was a conscious choice by NetApp SAN target engineers to limit LIF creation to reduce the number of paths presented to an initiator. By creating this limit, NetApp can ensure sufficient redundancy to eliminate single points of failure while also reducing or eliminating the need for features such as Selective LUN Map (SLM), which is required in SCSI-based configurations to limit the number of paths presented to the initiator. If additional LIFs are needed for a given node, an additional SVM can be created where an additional two LIFs can be created per node to balance I/O to and from namespaces.

NVMe added the ANA protocol to manage communicating, alerting, and managing pathing and path state changes. ANA consists of two components:

- A host-side implementation that is responsible for querying the target (ONTAP node) for the current path state information.
- A storage node implementation that is responsible for alerting when there is a path state change and answering initiator-side queries for enumerations of all available paths.

The host-side ANA implementation is responsible for passing all pathing information it receives to the host's multipathing stack. The hosts multipathing stack, for instance dm-multipath, then manages path preferences and usage.

Note: ONTAP 9.9.1 adds NVMe-oF remote I/O support that changes NVMe-oF pathing to advertise paths as either AO or ANO, because they are with SCSI-based protocols on unified systems (AFF/FAS).

Note: Unlike the SCSI protocols iSCSI and FC, which are symmetric on ASA, NVMe-oF protocols have the same asymmetric AO/ANO characteristics on ASA as they do on non-ASA. This is due to NVMe-oF protocol differences regarding local versus remote path operations.

NetApp recommends using Linux NVMe multipath exclusively for ONTAP namespaces.

Multipathing recommendations

NetApp recommends using Linux dm-multipath exclusively for ONTAP LUNs and using NVMe multipath for ONTAP namespaces.

Previous SUSE 15 (SLES15) versions did not have round-robin load balancing available in NVMe multipath, which caused performance degradations. This limitation is now addressed with the later SLES15 kernel/nvme-cli versions. For more information, see the ONTAP IMT.

NVMe-oF setup and configuration

Setup and configuration quick list

Before setting up NVMe/FC and/or NVMe/TCP, make sure that the following requirements are in place:

1. Verify that your configuration exactly matches a qualified configuration listed in the IMT. Failure to do so is likely to lead to suboptimal, poorly configured storage implementation.
2. Deploy, cable, and configure your physical infrastructure to adhere to the NetApp and switch vendor SAN best practices. See the section titled "Where to find additional information."
3. (NVMe/FC only) Enable N_Port ID virtualization (NPIV) on all fabric switches.
4. (NVMe/FC only) Use single-initiator zoning and use WWPNs to specify zone membership. Do not use switch port connectivity to denote zone membership or hard zoning.
5. Create NVMe-oF objects (SVMs, volumes, namespaces, subsystems, and LIFs) by using ONTAP System Manager or the ONTAP CLI. For details, see Appendix A: Using ONTAP System Manager to create ONTAP NVMe/FC and NVMe/TCP objects and Appendix B: ONTAP NVMe/FC and NVMe/TCP CLI commands—Initial setup and discovery.
6. Use NetApp Active IQ Unified Manager to monitor the health and performance of newly created NVMe objects and create reporting thresholds and alerts.

Detailed setup and configuration procedure references

For more information about performing NVMe setup and configuration tasks, see the following references:

- To set up and configure the NVMe/FC objects in ONTAP by using ONTAP System Manager (the on-box UI), see Appendix A: Using ONTAP System Manager to create ONTAP NVMe/FC and NVMe/TCP objects.
- To use the CLI to set up and configure NVMe/FC in ONTAP, see Appendix B: ONTAP NVMe/FC and NVMe/TCP CLI commands—Initial setup and discovery.
- For configuration information for supported host operating systems, review the ONTAP SAN Host Configuration Guides.
- To display NVMe objects and run I/O against them, see Appendix B: ONTAP NVMe/FC and NVMe/TCP CLI commands—Initial setup and discovery

Performance

NVM Express, Inc. initially developed the NVMe specifications and architectures to address the bottleneck caused by flash. When flash replaced hard drives, it removed the principal storage bottleneck that the hard drives created but cause another bottleneck: the command set and control plane. The NVMe specifications and architecture replace the SCSI command set with a new, leaner command set that:

- Streamlines commands (SCSI commands were backward compatible to the original standard first authored almost 40 years ago)
- Uses a polling mode rather than hardware interrupts
- Reduces context switches
- Is lockless
- Increases queues to 64k (65,535), each with a queue depth of 64k

Figure 22 illustrates how each of the previous points affects throughput and latency. The lean command set (I/O path) is more efficient, enabling more I/O in the same amount of time with the same working set and infrastructure. By reducing context switches and using polling mode rather than hardware interrupts, NVMe significantly reduces forced processor idle time. The removal of software locks also reduces the processor time spent at idle. The largest performance increases can be attributed to the huge number of queues and their associated queue depths, which allow each of those queues to use a separate processor core to concurrently process I/O.

Together, these enhancements create large increases in performance. These increases can most readily be seen in increases in throughput or IOPS and decreases in latency. During the initial testing, NetApp workload engineering and performance teams observed performance improvements that were often higher than 50%, measuring IOPS while reducing latencies by 80–100µs.

Figure 22) NVMe/FC ultra-high-performance design.

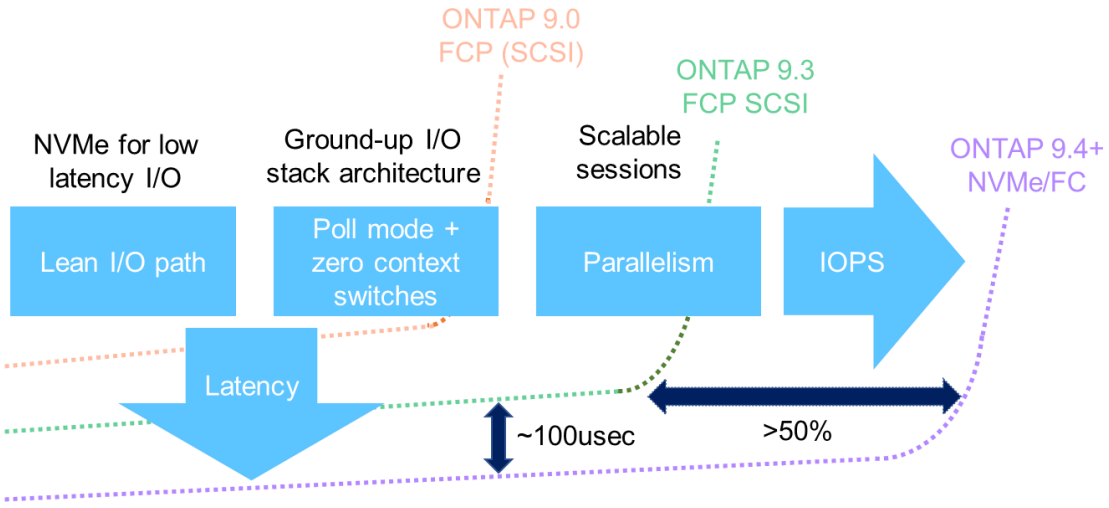


Table 5 displays the results of some of the internal testing by NetApp. It shows huge increases comparing single LUN versus single namespace access. This comparison highlights how much the increases in parallelization can affect performance. NetApp doesn't recommend using single LUNs when performance is required, because it limits I/O processing for that stream to a single CPU core.

Table 5) AFF A700 4K random read NVMe/FC versus FCP.

	NVMe/FC	Delta versus FCP (percentages)
Single port, IOPS	619K	207%
Single namespace/LUN, IOPS	540K	880%
Peak IOPS	865K	+51%

Figure 23 and Figure 24 show 8k and 4k random read performance from the testing in NetApp's performance characterization labs. The testing was done by defining, building, testing, and documenting reference architectures verified by NetApp, Brocade, and Broadcom for several popular critical enterprise applications. Links to the resulting documents can be found in the NetApp Verified Architectures section of this document.

Figure 23) AFF A700 high-availability (HA) pair, ONTAP 9.4, 8K random read FCP versus NVMe/FC.

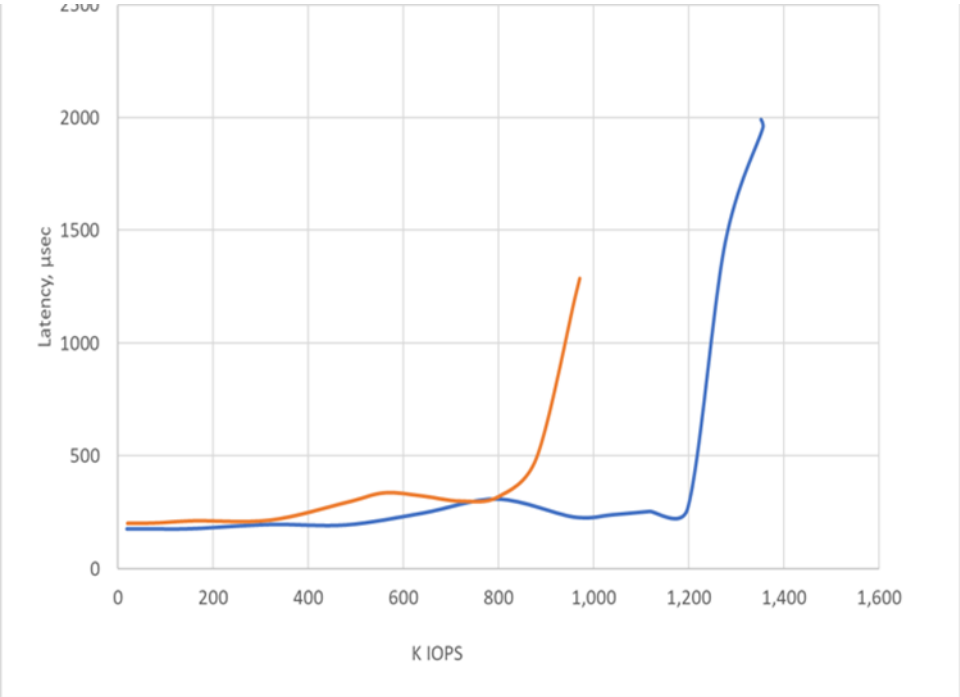


Figure 24) AFF A700 HA pair, ONTAP 9.4, 4K random read FCP versus NVMe/FC.

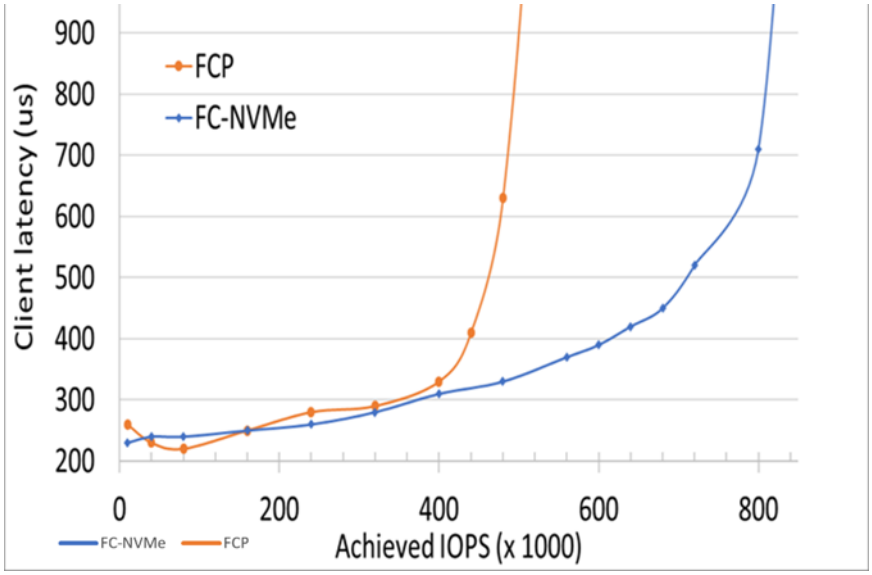
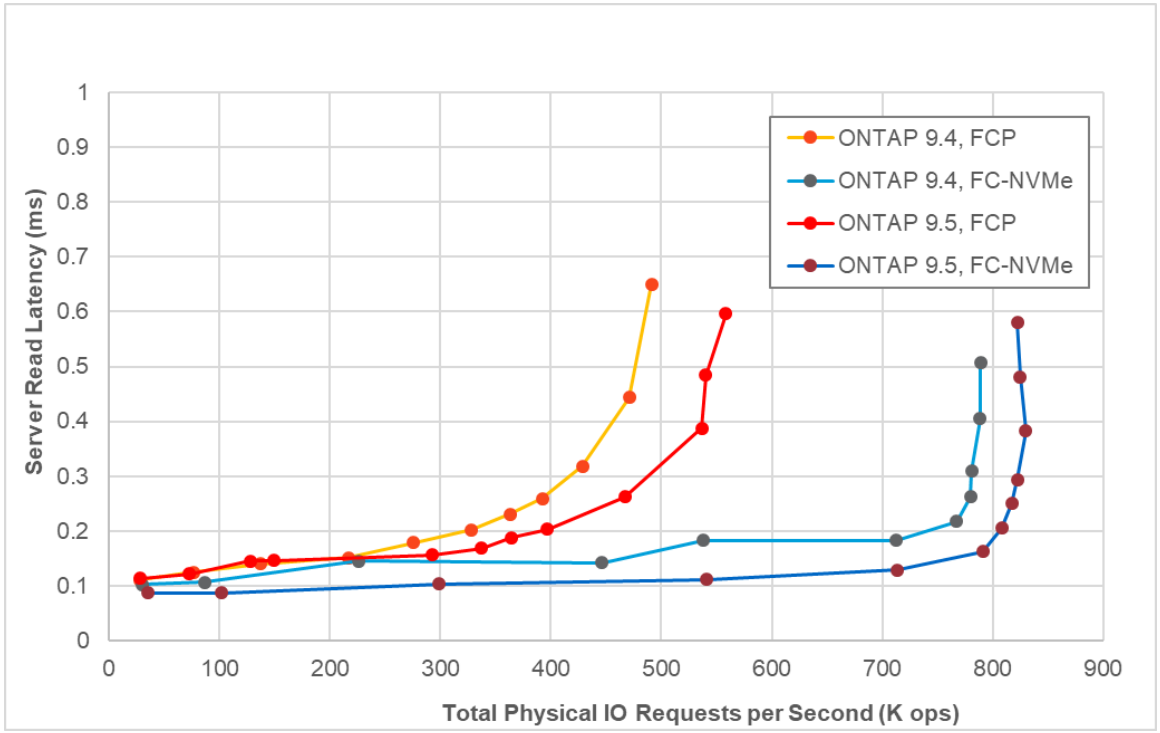


Figure 25 compares the number of IOPS per latency for both FCP and NVMe. They show a substantial increase in the number of I/Os that the controller can complete within a certain period. Interesting, NVMe/FC increases the number of IOPS achieved and reduces the time needed to complete those IOPS.

Figure 25) Performance improvements moving from FCP to NVMe/FC and with each ONTAP upgrade.



For configuration information for supported host operating systems, review the ONTAP SAN Host Configuration Guides.

Best practices for NVMe/TCP

NVMe/TCP uses Ethernet and TCP/IP, therefore anything that improves network performance is likely to have a positive effect on NVMe/TCP performance, much as it would for iSCSI. Generally, we recommend adhering to your networking vendors best practices for their switches and routers.

NVMe-oF enhancements

ONTAP 9.6

512-byte blocks

The block size for NVMe is 4096 or 4k for all OSs. NetApp introduced a 512-byte default block size for ESXi only in ONTAP 9.6 to more easily support VMware's Virtual Machine File System (VMFS). By reducing the block size to 512-byte blocks, NetApp more easily interoperates with ESXi by offering a common block size instead of having to aggregate multiple ESX's 512b into 4k ONTAP blocks. The 512-byte block support also enhances the ability of ONTAP to support ESXi copy and write/Atomic Test and Set (ATS).

ONTAP 9.9.1

NVMe-oF remote I/O support

NVMe-oF adds remote I/O support. This changes NVMe-oF pathing from an active/inactive model to the AO/active nonoptimized (ANO) model that all other ONTAP block protocols use.

Figure 26 shows NVMe-oF without remote I/O support and Figure 27 shows NVMe-oF with remote I/O support.

Figure 26) NVMe-oF without remote I/O support.

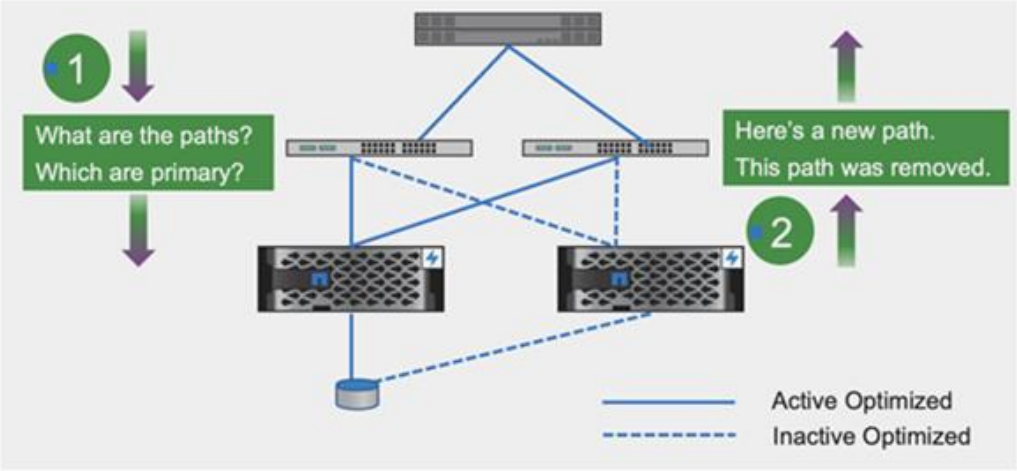
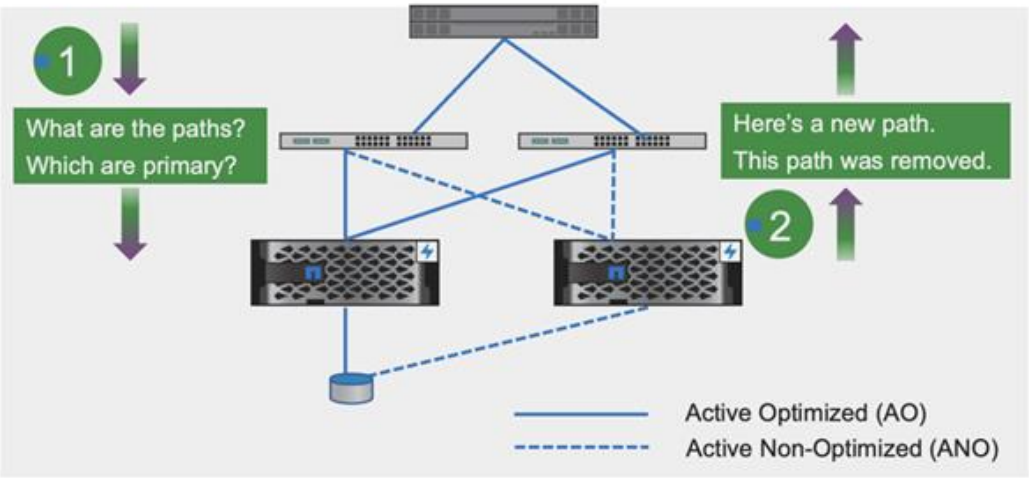


Figure 27) NVMe-oF with remote I/O.



Comparing Figure 26 and Figure 27 , it doesn't seem like much of a difference, but there is a subtle one. With remote I/O, support in NVMe-oF all paths are active, which means that I/O sent down any of those paths will be acknowledged and responded to or answered. Previously, without remote I/O, inactive paths were no available and could not be used.

```
All SAN Array adds NVMe/FC support
ONTAP 9.9.1 adds NVMe/FC as an additional block protocol. Unlike either FC or iSCSI, NVMe/FC on
All SAN Array (ASA) will continue to be asymmetric (AO/ANO). This assignment is due to
differences in how NVMe-oF works with remote versus local paths.
```

ONTAP 9.10.1

NVMe/TCP was introduced

ONTAP's first Ethernet-based and second NVMe-oF protocol transport.

Namespace resize

Introduced the ability to resize a namespace.

Large namespaces

Increased the maximum size of a namespace to 128TB on ASA. Large namespace support will be added to FAS and AFF controllers with a public preview in ONTAP 9.12.1 and will become a generally available feature in an upcoming release of ONTAP.

Asynchronous Event Request (AER, opcode OC) support on discovery controllers

Asynchronous Event Requests (AERs) are triggered when new maps or a subsystem with maps are added. AERs are issued to all active sessions that are part of the subsystem the map is added. Unmap support will be added in a future release of ONTAP. AERs are processed like any other admin queue request. AERs can take a long time to complete.

ONTAP 9.11.1

NVMe/TCP performance enhancements

ONTAP 9.11.1 introduced the following NVMe/TCP performance enhancements:

- First-burst such as in-capsule data for writes
- Read-response collapse for reads

These enhancements have significantly improved NVMe/TCP performance and brought it on par and even exceeded iSCSI for multiple I/O workloads.

LUN to namespace bi-directional conversion utility

An easy-to-use and very fast LUN-to-namespace and namespace-to-LUN conversion utility. The utility is built into ONTAP. Conversions are very fast because they are in-place conversions that only change metadata about the LUN/namespace.

For more information about converting between LUNs and namespaces, see Appendix D: Converting between LUNs and namespaces.

ONTAP 9.12.0

NVMe/TCP support introduced in AWS FSx and Cloud Volumes ONTAP

- NVMe/TCP support added to FSx HA
- NVMe/TCP support added to Cloud Volumes ONTAP AWS HA
- NVMe/TCP support added to Cloud Volumes ONTAP Azure HA

ONTAP 9.12.1

NVMe/TCP cloud support for additional NetApp cloud offerings

- NVMe/TCP support added to FSx Single Node
- NVMe/TCP support added to Cloud Volumes ONTAP AWS Single Node
- NVMe/TCP support added to Cloud Volumes ONTAP Azure Single Node
- Cloud Volumes ONTAP GCP Single Node and HA

Bidirectional in-band authentication to NVMe/TCP

Beginning with ONTAP 9.12.1, secure, bidirectional authentication between an NVMe host and controller is supported over NVME-TCP using the DH-HMAC-CHAP authentication protocol.

Each host or controller must be associated with a DH-HMAC-CHAP key which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator for an NVMe host or controller to authenticate its peer, it must know the key associated with the peer. SHA-256 is the default hash function.

Things to verify before you revert

If you are running the NVMe/TCP protocol and you have established secure authentication using DH-HMAC-CHAP, you must remove any host using DH-HMAC-CHAP from the NVMe subsystem before you revert. If the hosts are not removed, the revert will fail.

NVMe/FC support to MCC IP

ONTAP 9.12.1 introduces NVMe/FC support to MCC IP. For more information, see Appendix I: Configuration and Setup for NVMe/FC on MetroCluster IP.

Appendix A: Using ONTAP System Manager to create ONTAP NVMe/FC and NVMe/TCP objects

Use ONTAP System Manager to create NVMe objects by completing the following steps:

1. Create an SVM with NVMe support.

   Note: This step creates a SVM that contains all the NVMe storage objects created in the rest of this workflow.

   a. In ONTAP System Manager, navigate to Storage > SVMs. Click Create.

   b. Selecting NVMe triggers a prompt to create and define subsystem, NVMe Qualified Name (NQN) and namespace information to configure NVMe as part of the SVM Setup dialog box. Click Submit & Continue.

   Note: To display the host NQN, run the following command Linux command:

```
# cat /etc/nvme/hostnqn
```

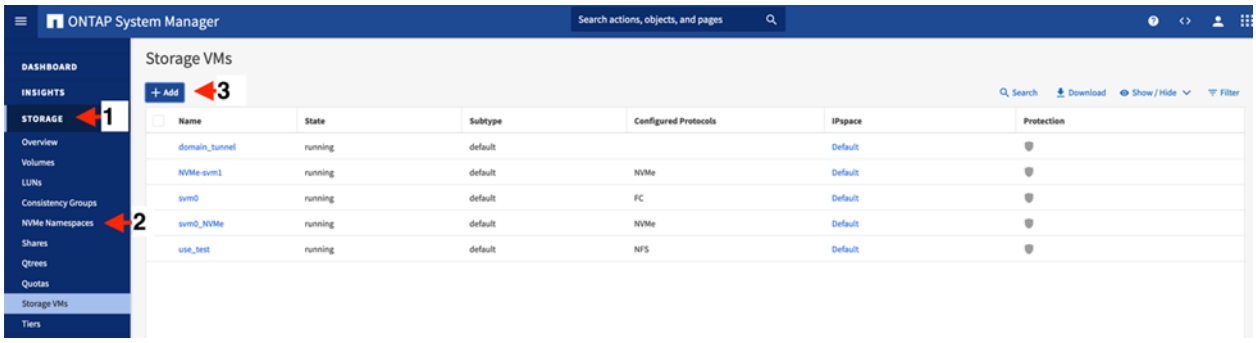
   Note: To display the SVM's NQN, run the following command:

```
vserver nvme show -vserver <vserver_name>
tme-a800::> nvme show -vserver NVMe-svm1
  (vserver nvme show)

           Vserver Name: NVMe-svm1
  Administrative Status: up
Discovery Subsystem NQN: nqn.1992-08.com.netapp:sn.70c0b1366f3611edacf100a098e22473:discovery
```

   Note: You can also view the subsystem's NQN in ONTAP System Manager by navigating to Storage > NVMe > NVMe Namespaces. Then, click the namespace link whose NQN you want to display.

Figure 28) OnCommand System Manager – Create SVM.



   c. Do one of the following:

   – Configure the SVM administrator details in the SVM administrator dialog box.

   – Click Skip to bypass adding a specific SVM administration account.

Figure 29) OnCommand System Manager – Create SVM: Configure NVMe transports – NVMe/FC and NVMe/TCP.

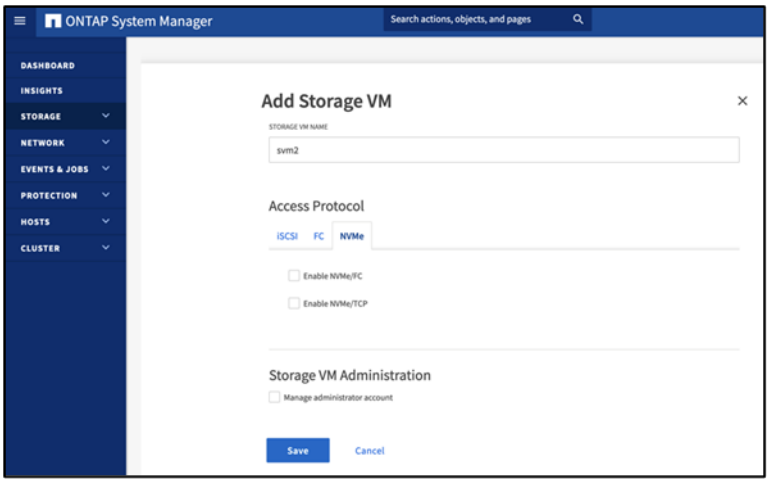


Figure 30) OnCommand System Manager – Create SVM: Configure NVMe/FC.

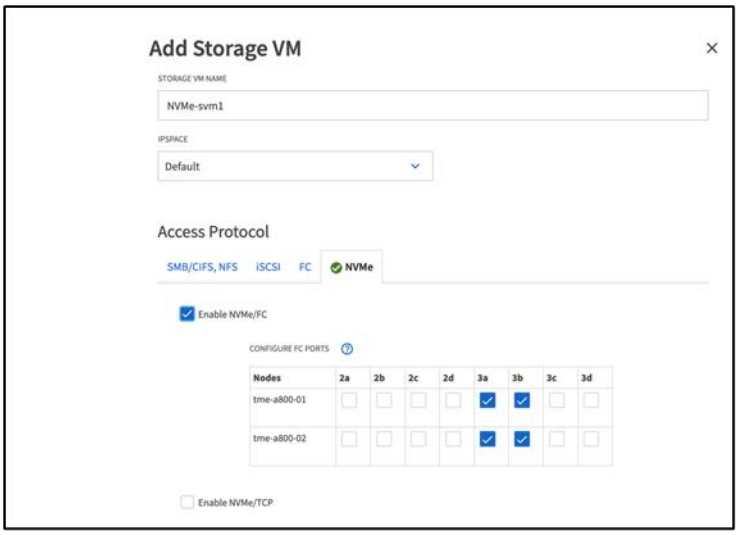


Figure 31) OnCommand System Manager – Create SVM: Configure NVMe/TCP.

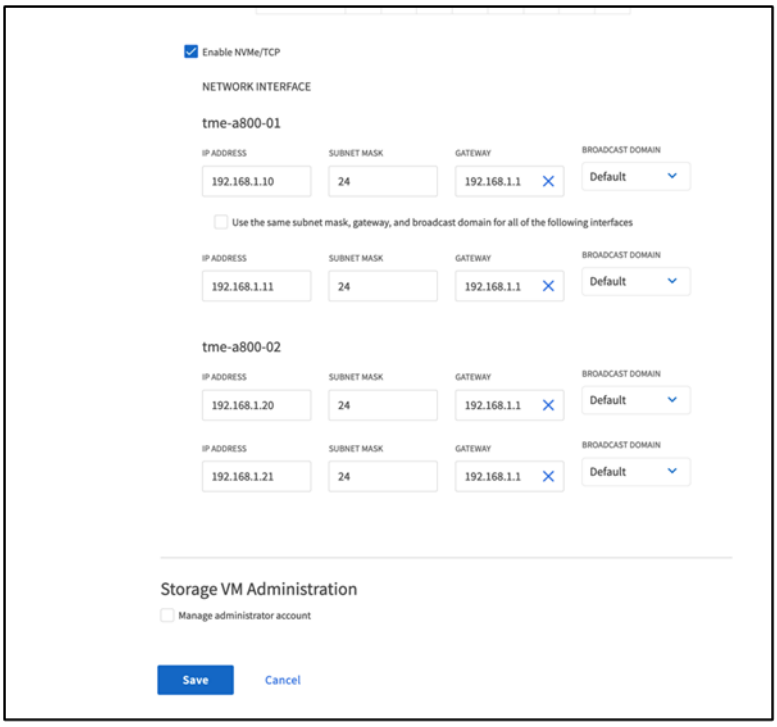
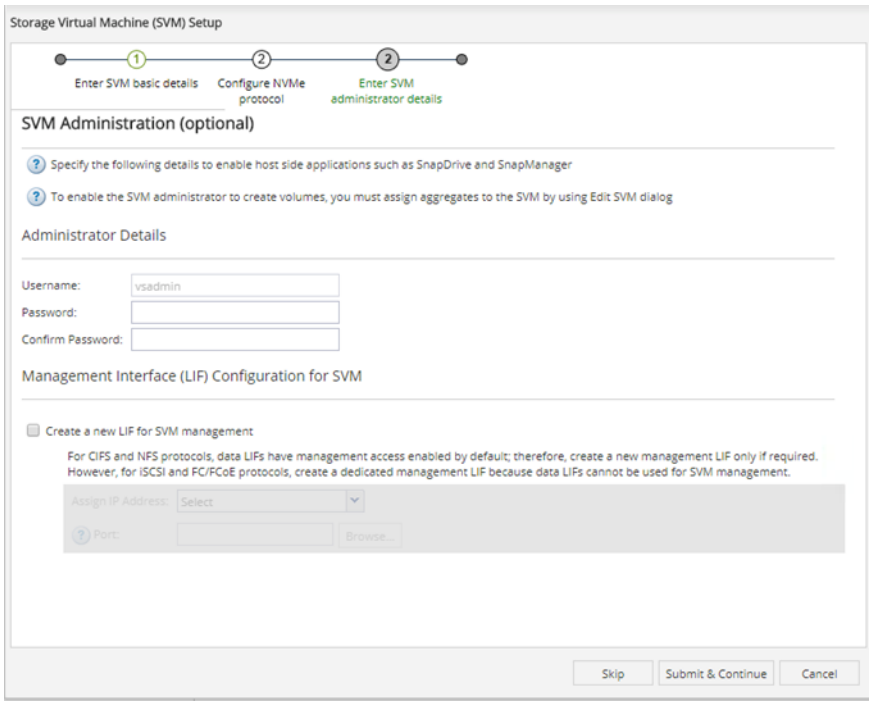
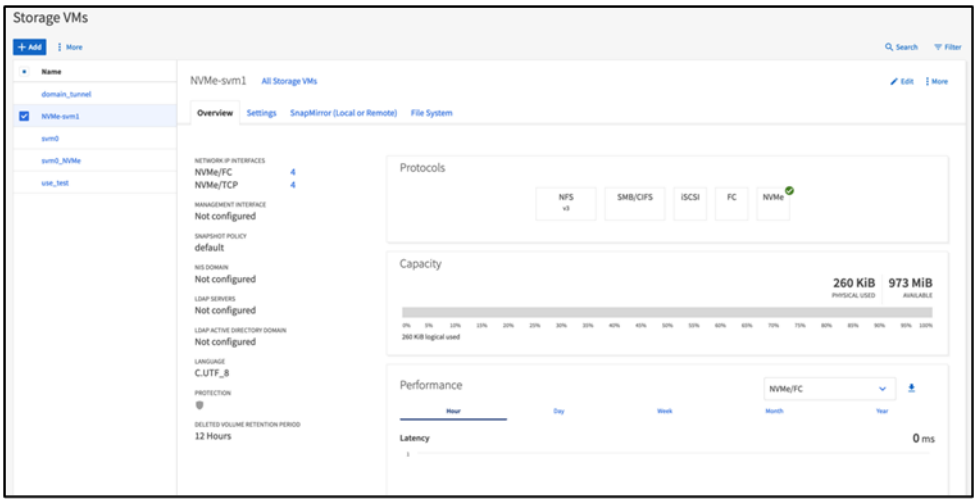


Figure 32) OnCommand System Manager – Create SVM: Configure admin details.



2. Review the summary of the SVM created and then click OK.
3. Select the newly created SVM. To review all the protocol settings and service status, click the SVM Settings from the top menu.

Figure 33) View newly created SVM.



4. To go back to the SVM dashboard page, click Back at the top-right corner of the SVM settings page. The SVM Dashboard page displays the NVMe status in green.
5. Launch the namespace management window that shows details for all the namespaces in the cluster. In the left menu pane, navigate to Storage > NVMe > NVMe Namespaces. Create a namespace as follows:
   a. Click Create.
   b. Select the SVM created.
   c. Use Advanced options to create a naming pattern to prefix all the namespace names.
   d. Enter the relevant details the Naming Pattern dialog box.
   e. Click Apply.
   f. Click Submit to create a namespace.
6. Click Close.

Figure 34) OnCommand System Manager – Create new NVMe namespace.

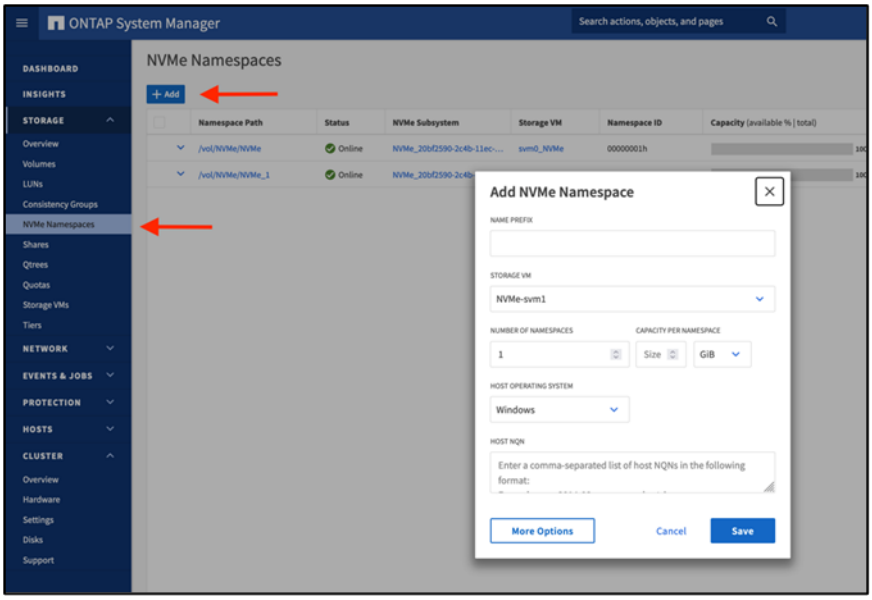


Figure 35) OnCommand System Manager – display newly created NVMe namespace.

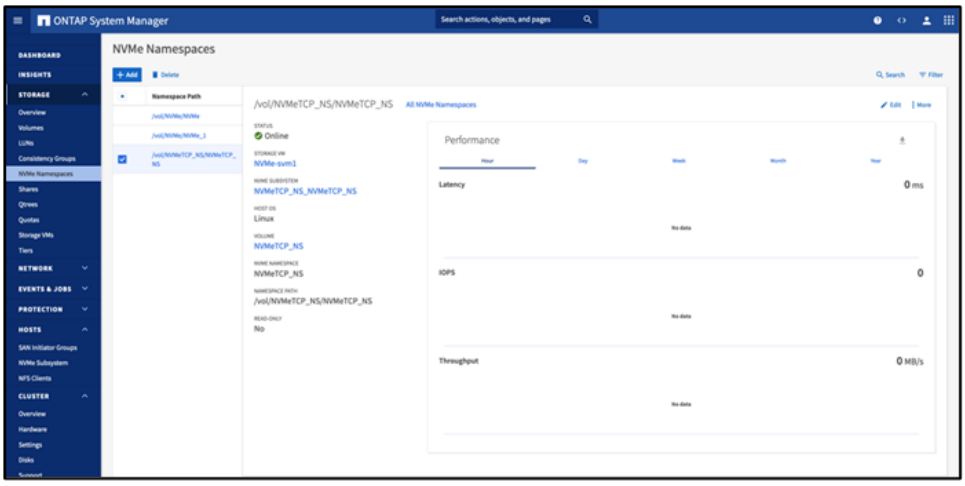
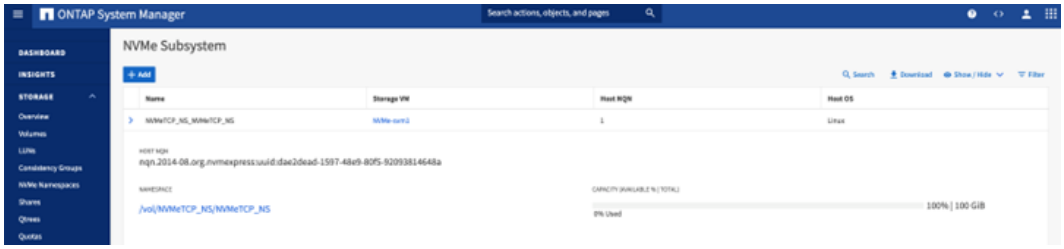


Figure 36) View newly created NVMe subsystem.



Appendix B: ONTAP NVMe/FC and NVMe/TCP CLI commands—Initial setup and discovery

On the ONTAP controller

1. Verify that there are NVMe/FC-capable adapters installed in the cluster.

   Note: These are needed for NVMe/FC, but unnecessary for NVMe/TCP.

```
AFF::> fcp adapter show -data-protocols-supported fc-nvme
 (network fcp adapter show)
           Connection Port   Admin  Operational
Node     Adapter Established Address  Status Status
------------ ------- ----------- --------- -------- -----------------------
AFF_1    1a    true    10100   up    online
AFF_1    1b    true    10200   up    online
2 entries were displayed.
```

2. Display Ethernet ports that host NVMe/TCP LIFs.

```
AFF::> net port show
  (network port show)

Node: tme-a800-01
                                                       Speed(Mbps)  Health
Port      IPspace      Broadcast Domain  Link MTU  Admin/Oper  Status
--------- ------------ ----------------  ---- ---- ----------- --------
e0M       Default      Default           up   1500  auto/1000  healthy
e0a       Cluster      Cluster           up   9000  auto/100000
                                                              healthy
```

```
e1a       Cluster      Cluster          up   9000  auto/100000
                                                   healthy
e4a       Default      Default          down 1500  auto/-     -
e4b       Default      Default          down 1500  auto/-     -
e4c       Default      Default          down 1500  auto/-     -
e4d       Default      Default          down 1500  auto/-     -
<Output omitted>
e5b       Default      NVMeT            up   9000  auto/100000
                                                   healthy
19 entries were displayed.
```

3. Create an SVM to host NVMe traffic.

```
AFF::> vserver create -vserver nvme1
[Job 2831] Job succeeded:
Vserver creation completed.
```

4. Display the SVM that was created.

```
AFF::> vserver show
                Admin   Operational Root
Vserver   Type  Subtype  State   State    Volume   Aggregate
----------- ------- ---------- ---------- ----------- ---------- ----------
AFF     admin  -     -     -      -     -
AFF_1    node  -     -     -      -     -
AFF_2    node  -     -     -      -     -
nvme1    data  default  running  running   svm_root  AFF_2_SSD_1

4 entries were displayed.
```

5. Display the allowed protocols on the SVM.

```
AFF::> vserver show -vserver nvme1 -fields allowed-protocols

vserver allowed-protocols
------- -----------------
nvme1  -
```

6. Add the NVMe protocol.

```
AFF::> vserver add-protocols -vserver nvme1 -protocols nvme
```

7. Display the allowed protocols on the SVM.

```
AFF::> vserver show -vserver nvme1 -fields allowed-protocols
vserver allowed-protocols
------- -----------------
nvme1  nvme
```

8. Create the NVMe service.

```
AFF::> vserver nvme create -vserver nvme1
```

9. Display NVMe service status.

```
AFF::> vserver nvme show -vserver nvme1

     Vserver Name: nvme1
Administrative Status: up
```

10. Create an NVMe/FC LIF.

```
AFF::> network interface create -vserver nvme1 -lif fcnvme-node1-1a -role data -data-protocol fc-
nvme -home-node node1 -home-port 1a
```

11. Create an NVMe/TCP LIF.

```
network interface create -vserver nvme1 -lif lif_nvme1_182 -role data -data-protocol nvme-tcp -
home-node tme-a800-02 -home-port e5b -address 192.168.1.20 -netmask 255.255.255.0
```

12. Display the newly created LIF.

```
AFF::> net interface show -vserver nvme1
```

```
 (network interface show)
      Logical  Status   Network      Current    Current Is
Vserver   Interface Admin/Oper Address/Mask    Node     Port  Home
----------- ---------- ---------- ------------------ ------------- ------- ----
nvme1
      fcnvme-node1-1a
             up/up  20:60:00:a0:98:b3:f7:a7
                           AFF_1     1a   true
      lif_nvme1_182
             up/up  192.168.1.20/24
                           AFF_1     e5b   true
```

13. Create a volume on the same node as the LIF.

```
AFF::> vol create -vserver nvme1 -volume nsvol1 -aggregate AFF_2_SSD_1 -size 50gb

Warning: You are creating a volume that, based on the hardware configuration, would normally have
the "auto" efficiency policy enabled. Because the effective cluster version is not 9.3.0 or
later, the volume will be created without the "auto" efficiency policy. After upgrading, the
"auto" efficiency policy can be enabled by using the "volume efficiency modify" command.
[Job 2832] Job succeeded: Successful   l
```

Note: You can safely ignore this warning. It explains that you must add “auto” efficiencies to the volume you are creating using the `volume efficiency modify` command.

14. Create a namespace.

```
AFF::> vserver nvme namespace create -vserver nvme1 -path /vol/nsvol1/ns1 -size 1GB -ostype linux
Created a namespace of size 1GB (1073741824).
```

15. Create a subsystem.

```
cluster1::> vserver nvme subsystem create -vserver nvme1 -subsystem mysubsystem -ostype linux
```

16. Display the newly created subsystem.

```
AFF::> vserver nvme subsystem show -vserver nvme1
Vserver Subsystem  Target NQN
------- ------------ ------------------------------------------------------------
nvme1
    mysubsystem nqn.1992-08.com.netapp:sn.a6f7f76d40d511e8b3c900a098b3f7a7:subsystem.mysubsystem
```

On the host

1. Get the NQN from the host.

Note: The `hostnqn` string is automatically populated at `/etc/nvme/hostnqn` during the nvme-cli package installation itself, and it is persistent. That string is already unique. So, there’s no need to separately generate the `hostnqn` string by using the Linux `nvme gen-hostnqn` command. If the host NQN is deleted, it can be generated with the Linux `nvme get-hostnqn` utility. To make the Linux host NQN persistent, add it to the `/etc/nvme/hostnqn` file.

2. Display the host NQN.

```
SLES_host:~ # cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:fc_lif:uuid:2cd61a74-17f9-4c22-b350-3020020c458d
```

On the ONTAP controller

1. Add the `hostnqn` string to the subsystem.

```
AFF::> vserver nvme subsystem host add -vserver nvme1 -subsystem mysubsystem -host-nqn nqn.1992-
08.com.netapp:sn.a6f7f76d40d511e8b3c900a098b3f7a7:subsystem.mysubsystem
```

2. Map the namespace to the subsystem.

```
AFF::> vserver nvme subsystem map add -vserver nvme1 -subsystem mysubsystem -path /vol/nsvol1/ns1
```

```
AFF::> vserver nvme namespace show -vserver nvme1 -instance

       Vserver Name: nvme1
      Namespace Path: /vol/nsvol1/ns1
           Size: 1GB
        Block Size: 4KB
         Size Used: 0B
          OS Type: linux
          Comment:
           State: online
       Is Read Only: false
       Creation Time: 4/15/2018 18:09:09
      Namespace UUID: 567fb229-a05e-4a57-aec9-d093e03cdf44
   Restore Inaccessible: false
Node Hosting the Namespace: AFF_1
        Volume Name: nsvol1
        Qtree Name:
    Attached Subsystem: mysubsystem
       Namespace ID: 1
        Vserver ID: 89
```

Appendix C: Host configuration information

For host configuration instructions, see https://docs.netapp.com/us-en/ontap-sanhost/.

Appendix D: Converting between LUNs and namespaces

Converting a LUN to a namespace

1. Display the LUNs.

```
tme-a700s-clus::> lun show
Vserver   Path                            State   Mapped   Type        Size
--------- ------------------------------ ------- -------- -------- --------
svm0     /vol/testLUN/testLUN                  online  mapped   linux        1GB
```

2. Unmap the LUN to be converted.

```
tme-a700s-clus::> lun unmap -vserver svm0 -path /vol/testLUN/testLUN -igroup
new_15Mar21_tif5_igroup
```

3. Convert the LUN.

```
tme-a700s-clus::> vserver nvme namespace convert-from-lun -vserver svm0 -lun-path
/vol/testLUN/testLUN
```

4. Map the namespace to the NVMe subsystem.

```
vserver nvme subsystem map add -vserver svm0 -subsystem svm0_subsystem_909 -path
/vol/testLUN/testLUN
```

5. Display the new namespace.

```
tme-a700s-clus::> vserver nvme namespace show
Vserver Path                            State      Size Subsystem       NSID
------- ------------------------------- ------- ------- ---------- ---------
Svm0    /vol/testLUN/testLUN            online      1GB svm0_subsystem_909 00000001h
```

Converting a namespace to a LUN

1. Display the new namespace.

```
tme-a700s-clus::> vserver nvme namespace show
Vserver Path                            State      Size Subsystem       NSID
------- ------------------------------- ------- ------- ---------- ---------
```

```
Svm0    /vol/testLUN/testLUN             online      1GB svm0_subsystem_909 00000001h
```

2. Unmap the namespace.

```
vserver nvme subsystem map remove -vserver svm0 -subsystem svm0_subsystem_909 -path
/vol/testLUN/testLUN
```

3. Convert the namespace to a LUN.

```
lun convert-from-namespace -vserver svm0 -namespace-path /vol/testLUN/testLUN
```

4. Map the LUN to an igroup.

```
lun map -vserver svm0 -path /vol/testLUN/testLUN -igroup new_15Mar21_tif5_igroup -lun-id 20
```

5. Display the LUNs.

```
tme-a700s-clus::> lun show -vserver svm0
Vserver   Path                            State   Mapped   Type        Size
--------- ------------------------------ ------- -------- -------- --------
svm0      /vol/testLUN/testLUN            online  mapped   linux         1GB
```

Appendix E: NVMe/FC scalability and limits

At the time of this writing, scalability and limits for NVMe/FC were located here: https://hwu.netapp.com/.

Appendix F: Troubleshooting

Before you troubleshoot any of the NVMe/FC failures, always make sure that you are running a configuration that is compliant with the IMT specifications. Then proceed to the steps in the following section to debug any host-side issues here.

lpfc verbose logging for NVMe/FC

```
Here is a list of lpfc driver logging bitmasks available for NVMe/FC, as seen in
drivers/scsi/lpfc/lpfc_logmsg.h:
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
#define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
```

Set the `lpfc_log_verbose` driver setting (appended to the `lpfc` line in `/etc/modprobe.d/lpfc.conf`) to any of the previous values for logging NVMe/FC events from an `lpfc` driver perspective. Then, recreate the `initiramfs` by running `dracut -f` and reboot the host. After you reboot, verify that the verbose logging is applied by verifying the following output using the previous `LOG_NVME_DISC` bitmask as an example.

```
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_enable_fc4_type=3 lpfc_log_verbose=0x00200000
# cat /sys/module/lpfc/parameters/lpfc_log_verbose
2097152
```

NetApp recommends the following lpfc logging bitmask values for common issues:

- General NVMe discovery/connect events: 0x00200000
- lpfc driver events related to FC-LS discovery issues during link bounces (such as LIF/Port toggle events): 0xf00083

Common nvme-cli errors and their workarounds

This section describes some of the error messages that nvme-cli utility displays during the `nvme discover`, `nvme connect`, and `nvme connect-all` operations. It describes the possible causes of these errors and their workarounds.

Error message

```
Failed to write to /dev/nvme-fabrics: Invalid argument.
```

- **Probable cause**: This error message generally appears if the syntax is wrong.
- **Workaround**: Make sure to use the correct syntax for the previous NVMe commands.

Error message

```
Failed to write to /dev/nvme-fabrics: No such file or directory.
```

- **Probable cause**: Several issues that could cause this error. Some of the common causes are:

  Wrong arguments were passed to the previous NVMe commands.
- **Workaround**: Make sure that you have passed the proper arguments (correct WWNN string, WWPN string, and so on) for the previous commands.

  If the arguments are correct, but the error still appears, see if the `/sys/class/scsi_host/host*/nvme_info` output is correct, with the NVMe initiator appearing as `Enabled` and NVMe/FC target LIFs correctly appearing under the remote ports sections.

  For example:

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
NVME LPORT lpfc0 WWPN x10000090fae0ec9d WWNN x20000090fae0ec9d DID x012000 ONLINE
NVME RPORT WWPN x200b00a098c80f09 WWNN x200a00a098c80f09 DID x010601 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 0000000000000071 Wr 0000000000000005 IO 0000000000000031
Cmpl 00000000000000a6 Outstanding 0000000000000001
NVME Initiator Enabled
NVME LPORT lpfc1 WWPN x10000090fae0ec9e WWNN x20000090fae0ec9e DID x012400 ONLINE
NVME RPORT WWPN x200900a098c80f09 WWNN x200800a098c80f09 DID x010301 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 0000000000000073 Wr 0000000000000005 IO 0000000000000031
Cmpl 00000000000000a8 Outstanding 0000000000000001
```

  Workaround: If the target LIFs don't appear as above in the `nvme_info` output, check the `/var/log/messages` and `dmesg` output for any suspicious NVMe/FC failures, and report or fix accordingly.

Error message

```
Failed to write to /dev/nvme-fabrics: Operation already in progress
```

Probable cause: This error message is seen if the controller associations or specified operation is already created or in the process of being created. This could happen as part of the autoconnect scripts installed.

Workaround: None. For `nvme discover`, try running this command after some time. And maybe for `nvme connect` and `connect-all`, run `nvme list` to verify that the namespace devices are already created and displayed on the host.

Error message

```
No discovery log entries to fetch
```

Probable cause: This error message generally appears if the `/etc/nvme/hostnqn` string has not been added to the corresponding subsystem on the NetApp array. This error can also appear if an incorrect hostnqn string has been added to the respective subsystem.

Workaround: Ensure that the exact `/etc/nvme/hostnqn` string is added to the corresponding subsystem on the NetApp array. Verify by running the `vserver nvme subsystem host show` command.

Files and command outputs required for debugging

If you continue to have issues, collect the following files and command outputs and send them to NetApp for further triage.

```
 cat /sys/class/scsi_host/host*/nvme_info
/var/log/messages
dmesg
```

`nvme discover` output, as in:

```
nvme discover --transport=fc --traddr=nn-0x200a00a098c80f09:pn-0x200b00a098c80f09 --host-
traddr=nn-0x20000090fae0ec9d:pn-0x10000090fae0ec9d
nvme list
```

Appendix G: Configuration and Setup for NVMe/FC on MCC IP

ONTAP 9.12.1 added MCC IP support for NVMe/FC on MCC IP 4-pack clusters.

MCC NVMe Host timeout settings

During an ONTAP MCIP Switchover, an All Paths Down (APD) window is encountered while the Primary Cluster site switches over to its Secondary/Peer Cluster site. During this APD window, the host/client loses access to all NVMe namespace devices since all paths are down here. If this APD window exceeds a specified interval, the Linux NVMe/FC host gives up resulting in an I/O error to the overlying applications. This behavior is caused by the Linux NVMe/FC host behavior during a link loss is governed by a transport layer parameter called the `NVMe/FC dev_loss_tmo`.

Review the following knowledge base article to adjust the NVMe/FC dev_loss_tmo:

Linux Host Recommendations for NVMe/FC protocol support with NetApp MetroCluster IP.

NVMe Platform support and configuration limitations

Support for NVMe-oF protocol varies by platform and configuration based upon your version of ONTAP.

NVMe FC LIF restriction

FC ports on both clusters should be connected to the same fabric and soft zoning should be used. This will ensure that LIFs are placed on connected ports and hosts can login to the LIFs after a switchover.

For more information, see the following MetroCluster technical reports:

- TR-4689: MetroCluster IP - Solution Architecture and Design
  https://www.netapp.com/pdf.html?item=/media/13481-tr4689.pdf
- TR-4705: NetApp MetroCluster Solution Architecture and Design
  https://www.netapp.com/media/13480-tr4705.pdf

Appendix H: Set up secure authentication over NVMe/TCP

Steps

1. Add DH-HMAC-CHAP authentication to your NVMe subsystem.

```
vserver nvme subsystem host add- vserver svm_name
…
-subsystem subsystem
-host-nqn host_nqn
-dhchap-host-secret authentication_host_secret
-dhchap-controller-secret authentication_controller_secret
-dhchap-hash-function{sha-256|sha-512}
-dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-bit|8192-bit}
```

2. Verify that the DH-HMAC CHAP authentication protocol is added to your host.

```
vserver nvme subsystem host show
…
 [ -dhchap-hash-function{sha-256|sha-512} ]  Authentication Hash Function
 [ -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-bit|8192-bit} ]
                Authentication Diffie-Hellman
                Group
 [ -dhchap-mode {none|unidirectional|bidirectional} ]
                Authentication Mode
```

3. Verify that the DH-HMAC CHAP authentication protocol is added to your controller.

```
vserver nvme subsystem controller show
…
 [ -dhchap-hash-function{sha-256|sha-512} ]  Authentication Hash Function
 [ -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-bit|8192-bit} ]
                Authentication Diffie-Hellman
                Group
 [ -dhchap-mode {none|unidirectional|bidirectional} ]
                Authentication Mode
```

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

Host OS setup and configuration

- Host setup and configuration information
  https://docs.netapp.com/us-en/ontap-sanhost/

Standards documents

- Proposed T10 SCSI block commands
  http://t10.org/ftp/t10/document.05/05-344r0.pdf
- RFC 3270 Internet Small Computer Systems Interface (iSCSI)
  https://tools.ietf.org/html/rfc3720
- RFC 7143 Internet Small Computer System Interface (iSCSI) Protocol (Consolidated)
  https://tools.ietf.org/html/rfc7143
- RFC 5041 Direct Data Placement over Reliable Transports
  https://tools.ietf.org/html/rfc5041
- INCITS T11 - T11-2017-00145-v004 FC-NVMe specification
  https://standards.incits.org/apps/group_public/document.php?document_id=92164

SUSE Enterprise Linux links

- SUSE downloads
  https://www.suse.com/download-linux/
- SUSE Linux documentation
  https://www.suse.com/documentation/

Brocade links

- SAN Health:
  https://www.broadcom.com/support/fibre-channel-networking/tools/san-health/diagnostics-capture
- Support Download SAN Health Diagnostics Capture
  https://www.broadcom.com/support/fibre-channel-networking/tools/san-health/diagnostics-capture
- Unleash the Power of NVMe with Fibre Channel
  https://docs.broadcom.com/docs/12395453
- NVMe over Fibre Channel for Dummies Book (3rd edition):
  https://docs.broadcom.com/docs/nvme-over-fibre-channel-for-dummies-book
- Planning for the Transition of Productions-Ready NVMe over Fabrics Deployments in the Enterprise:
  https://docs.broadcom.com/docs/planning-for-the-transition-to-production-ready-nvme
- Broadcom NVMe over Fibre Channel with VMware vSphere 7.0 Support
  https://docs.broadcom.com/docs/12399210
- IDC: Native NVMe/FC Support Provides a Performance Growth Path for Virtual Infrastructure
  https://docs.broadcom.com/docs/12398958
- Tolly Test Report- Emulex Gen 7 LPe35002 VMware ESXi 7.0, NVMe/FC vs. SCSI
  https://docs.broadcom.com/docs/Emulex-Gen-7-LPe35002-VMware-ESXi-7.0-NVMe-FC-SCSI

Videos, webcasts, and blogs

- FCIA: Introducing Fibre Channel NVMe (Bright TALK Webcast)
  https://www.brighttalk.com/webcast/8615/242341?utm_campaign=webcasts-search-results-feed&utm_content=FCIA&utm_source=brighttalk-portal&utm_medium=web
- FCIA: Dive Deep into NVMe over Fibre Channel (FC-NVMe) (Bright TALK Webcast)
  https://www.brighttalk.com/webcast/14967/265459?utm_campaign=webcasts-search-results-feed&utm_content=FCIA&utm_source=brighttalk-portal&utm_medium=web

White papers, product announcements, and analysis

- RoCE Versus iWARP Competitive Analysis
  http://www.mellanox.com/related-docs/whitepapers/WP_RoCE_vs_iWARP.pdf
- Ultra-Low Latency with Samsung Z-NAND SSD
  https://www.samsung.com/us/labs/pdfs/collateral/Samsung_Z-NAND_Technology_Brief_v5.pdf

NetApp documentation, technical reports, and other NVMe-related collateral

- Licensing Information for NVMe Protocol on ONTAP
  https://kb.netapp.com/Advice_and_Troubleshooting/Data_Storage_Software/ONTAP_OS/Licensing_information_for_NVMe_protocol_on_ONTAP
- TR-4080: Best Practices for Scalable SAN
  https://www.netapp.com/pdf.html?item=/media/10680-tr4080pdf.pdf
- NetApp NVMe solutions: Customer-focused technology leadership
  www.netapp.com/us/info/nvme.aspx
- A nice blog discussing NVMeCLI – NVMe command set
  https://nvmexpress.org/open-source-nvme-management-utility-nvme-command-line-interface-nvme-cli/

- Tech ONTAP Podcast Episode 72: Demystifying NVMe
  https://soundcloud.com/techontap_podcast/episode-72-demystifying-nvme?utm_source=clipboard&utm_medium=text&utm_campaign=social_sharing
- Datasheet: NetApp EF570 All-Flash Array
  https://www.netapp.com/pdf.html?item=/media/8117-ds-3893.pdf
- NetApp documentation
  https://mysupport.netapp.com/documentation/productsatoz/index.html#O
- NetApp software downloads
  https://mysupport.netapp.com/NOW/cgi-bin/software/

ONTAP Cloud

- What is Amazon FSx for NetApp ONTAP?
  https://docs.aws.amazon.com/fsx/latest/ONTAPGuide/what-is-fsx-ontap.html
- Azure NetApp Files documentation
  https://learn.microsoft.com/en-us/azure/azure-netapp-files/
- Cloud Volumes ONTAP documentation
  https://docs.netapp.com/us-en/cloud-manager-cloud-volumes-ontap/

NetApp Verified Architectures

You might also want to review one or more of the NetApp Verified Architecture papers that document a best practice reference architecture, test, and tested configurations, and expected performance results. At the time of this writing, there were two NetApp Verified Architectures that cover solutions with NVMe/FC:

- NVA-1126-Design: NetApp and Broadcom Modern SAN Cloud-Connected Flash Solution Oracle and SUSE NetApp Verified Architecture Design Edition
- NVA-1127-Design: NetApp and Broadcom Modern SAN Cloud-Connected Flash Solution MongoDB and SUSE NetApp Verified Architecture Design Edition
- NVA-1145-Design: Modern SAN Cloud-Connected Flash Solution NetApp, VMware, and Broadcom Verified Architecture Design Edition: With MS Windows Server 2019 and MS SQL Server 2017 Workloads
- NVA-1147-Design: SAP HANA on NetApp All SAN Array Modern SAN, Data Protection, and Disaster Recovery
- NVA-1159-Design: Epic on modern SAN NetApp Broadcom healthcare solution

Additional NetApp Verified Architectures are being written and planned.

Other NVMe documentation

For more information about how NVMe and NVMe/FC improve performance and reduce latency, see:

- NVMe Modern SAN Primer
- Demartek Evaluation: Performance Benefits of NVMe over Fibre Channel – A New, Parallel, Efficient Protocol

Version history

Version	Date	Document version history
Version 1.0	June 2018	Initial release.
Version 2.0	November 2018	Updated for ONTAP 9.5, ANA, added host install and configuration appendixes.
Version 3.0	April 2019	Minor updates and errata.

Version	Date	Document version history
Version 4.0	June 2019	Updated for ONTAP 9.6, ANA, round-robin, and multipathing recommendations.
Version 4.1	April 2020	Updated to include ESXi 7.0 NVMe configuration.
Version 5.0	December 2020	Updated to include OS config links and other minor updates.
Version 6.0	June 2021	Updated to include ONTAP 9.9.1 new features and recommendations.
Version 7.0	February 2023	Updated to add NVMe/TCP transports and other ONTAP updates. Added a section on optimizing host and target-side configuration recommendations.

Contact us

Let us know how we can improve this technical report. Contact us at decom-ng-doccomments@netapp.com, and include Technical Report 4684 in the subject line.

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

TR-4684-0223

NetApp