Abstract

NVM Express (NVMe) is a data storage protocol that delivers the fastest response times for business-critical enterprise applications. However, NVMe is more than a storage specification; the broader NVMe over Fabrics protocol encompasses the entire data path, from server to network to storage system.

This technical report focuses on building modern SANs on NVMe over Fabrics, especially NVMe over Fibre Channel (NVMe/FC, also known as FC-NVMe).

This document describes the industry-first NVMe/FC implementation in NetApp® ONTAP®, includes guidelines for SAN design and implementation, and provides links to testing and qualification recommendations.

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.
Appendix I: Configuring and Connecting SuSE Enterprise Linux 15 to ONTAP NVMe/FC Namespaces ................................................................. 48
  Features and Limitations ........................................................................ 48
  Supported Configurations ......................................................................... 48
  No NVMe/FC Linux Host Utilities Support .................................................. 48
  Coexistent Host Configurations ................................................................. 48
  Configuration Requirements ...................................................................... 49
  Enabling NVMe/FC with ANA ................................................................. 49

Appendix J: Configuring and Connecting Red Hat Enterprise Linux to ONTAP NVMe/FC Namespaces ................................................................. 52
  Description .............................................................................................. 52
  Features and Limitations ......................................................................... 52
  Configuration Requirements ...................................................................... 52
  Fabric Switch ........................................................................................... 52
  Initiator .................................................................................................... 53
  Target ........................................................................................................ 53
  Enabling NVMe/FC on Red Hat Enterprise Linux 7.6 .................................. 53
  Broadcom FC Adapter ............................................................................. 53

Appendix K: Discovering and Connecting to an NVMe Namespace from a SuSE Linux Host Server ................................................................. 56
  List, Discover, and Connect to an NVMe Namespace from SuSE Linux Host (Summary) ................................................................. 56
  List, Discover, and Connect to an NVMe Namespace from SuSE Linux Host (Procedure) ................................................................. 56
  Command Options for NVMe Discover and Connect Commands ............... 57
  NVMe/FC Scalability and Limits ............................................................... 58
  Troubleshooting ....................................................................................... 58
  lpfc Verbose Logging for NVMe/FC .......................................................... 58
  Common nvme-cli Errors and Workarounds ............................................ 59

Where to Find Additional Information ......................................................... 60
  Standards Documents ............................................................................. 60
  SuSE Enterprise Linux Links ................................................................... 60
  Brocade Links ......................................................................................... 60
  Videos, Webcasts, and Blogs ................................................................. 61
  White Papers, Product Announcements, and Analysis ............................... 61
  NetApp Documentation, Technical Reports and other NVMe-related Collateral ....................................................................................... 61
LIST OF TABLES
Table 1) FCP and NVMe/FC terms. .................................................................................. 6
Table 2) OS Multipathing stacks ...................................................................................... 18
Table 3) ONTAP features supported with NVMe ................................................................. 20
Table 4) ONTAP coexisting features supported by NVMe .................................................. 20
Table 5) ONTAP Features Not Currently Supported by NVMe ........................................... 21
Table 6) AFF A700 4K random read NVMe/FC versus FCP ................................................ 26

LIST OF FIGURES
Figure 1) Why NVMe/FC is so fast ..................................................................................... 7
Figure 2) NetApp AFF A800 .............................................................................................. 9
Figure 3) NetApp end-to-end NVMe/FC ........................................................................... 10
Figure 4) AFF A800 FCP versus NVMe/FC performance comparison ................................. 10
Figure 5) ONTAP 9.4 delivers end-to-end NVMe flash with NVMe-FC and the new AFF A800. .............................................................. 11
Figure 6) NVMe can use multiple network transports ....................................................... 13
Figure 7) NVMe/FC versus FCP frames ........................................................................... 14
Figure 8) Adopt modern technology nondisruptively ....................................................... 15
Figure 9) TP 4004: ANA base proposal (ratified 3/18) ..................................................... 17
Figure 10) TP 4028: ANA path and transport (ratified 1/18) .............................................. 17
Figure 11) INCITS 540-201x, T11 FC-NVMe defines NVMe command and data transport using FC standards ............................................................ 17
Figure 12) NVMe/FC Storage Failover: ONTAP 9.5 introduces ANA ................................. 18
Figure 13) A comparison NVMe/FC with and without ANA .............................................. 19
Figure 14) NetApp IMT adds FC-NVMe protocol filter ..................................................... 22
Figure 15) NVMe/FC ultra-high-performance design ....................................................... 25
Figure 16) AFF A700 HA pair, ONTAP 9.4, 8K random read FCP versus NVMe/FC ......... 26
Figure 17) AFF A700 HA pair, ONTAP 9.4, 4k random read FCP versus NVMe/FC ......... 27
Figure 18) Performance improvements moving from FCP to NVMe/FC and also with each ONTAP upgrade ................................................................. 27
1 NVMe in the Modern Data Center

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 NVME 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 peripheral connect interface express (PCIe), a high-speed and high-bandwidth hardware protocol that is substantially faster than older standards like Small Computer Systems Interface (SCSI), Serial Attached SCSI (SAS), and Serial Advanced Technology Attachment (SATA).

SCSI was introduced almost 40 years ago and was designed for the storage technologies prevalent in that era: 8-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.

1.1 NVMe and FCP Naming

NVMe adds some new names for some common structures. Table 1 maps some common structures that have different names than those used in Fibre Channel Protocol (FCP).

An NVMe Qualified Name (NQN) identifies an endpoint and is similar to an iSCSI Qualified Name (IQN) in both format (domain registration date, domain registered, and something unique like a serial number). A namespace is analogous to a LUN (Logical Unit Number, a unique identifier for a logical or physical device); both represent an array of blocks presented to an initiator. A subsystem is analogous to an initiator group (igroup) (subsystems have considerably more functionality but for our purposes, we are focusing on how to map LUNs/namespaces), it is used to mask an initiator so that it can see and mount a LUN or namespace. Asymmetric Namespace Access (ANA) is a new protocol feature for monitoring and communicating path states to the host operating system’s Multipath I/O (MPIO) or multipath stack, which uses information communicated through ANA to select and manage multiple paths between the initiator and target. For more information about ANA, see 6.1 and 6.2.

<table>
<thead>
<tr>
<th>Fibre Channel Protocol (FCP)</th>
<th>NVMe/FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>World-wide Port Name (WWPN)</td>
<td>NVMe Qualified Name (NQN)</td>
</tr>
<tr>
<td>LUN</td>
<td>Namespace</td>
</tr>
<tr>
<td>Igroup, LUN mapping, and LUN masking</td>
<td>Subsystem</td>
</tr>
<tr>
<td>Asymmetric Logical Unit Access (ALUA)</td>
<td>Asymmetric Namespace Access (ANA)</td>
</tr>
</tbody>
</table>

1.2 Why NVMe Is So Fast

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

- **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 (Gbps). 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 vis-à-vis 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.

Figure 1) Why NVMe/FC is 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 16 and 32Gbps and soon 64Gbps.

• **RDMA Protocol.**
  - Data center fast Ethernet: Currently available in 25, 40, 50, and 100Gbps.
  - InfiniBand: currently available with speeds up to 100Gbps.

• **PCI Express 3.0.** Supports 8 gigatransfers per second (GT/s), which translates to approximately 6.4Gbps.

Throughput 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:

• 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.
2 NVMe and Modern SANs

SANs are the preferred architecture for many of the most important enterprise applications in use today. SANs are block storage. They present arrays of blocks to a host OS; the host OS then partitions and formats LUNs (the array of blocks) into a file system that the OS manages.

SANs based on FCP (SCSI and data inside FC frames) have the largest market share, with iSCSI in second place and the remainder being Fibre Channel over Ethernet (FCoE). All three of these block protocols are popular, efficient, and reliable transports.

The other primary form of networked storage is NAS, where the storage array owns the file system and presents files by using NFS, CIFS, or SMB. All block protocols are easily consumed by host operating systems and enterprise applications; they are fast, reliable, and invisible to the applications. SAN is the more prevalent network storage architecture: it has over 70% of the total networked storage market, and NAS is used in approximately 30% of the market. Both SAN and NAS enable organizations to have better control over their data, and to protect it while improving access and performance.

2.1 NVMe Use Cases

Like the data center flash revolution that preceded it, NVMe defines a new standard for performance. It’s especially significant because it enables low latency, which is crucial to the user experience.

Although virtually all applications benefit from low latency, its performance boost is especially valuable in business-critical enterprise database applications such as Oracle, Microsoft SQL Server, SAP HANA, and NoSQL databases (such as MongoDB).

In addition, NVMe accelerates many of today’s most important emergent business workloads:

1. Artificial intelligence (AI). Many AI workloads use NAS with file-based protocols (primarily NFS). However, AI file systems (such as GPFS) that work better with block protocols will see performance improvements from adopting NVMe/FC.

2. Machine learning (ML)/deep learning (DL). ML/DL workloads will gain significant performance improvements by using NVMe/FC, because it enables high data ingest rates and much higher concurrent performance. Another NVMe benefit in these environments would be the ability to support large numbers of initiators concurrently running I/O against centrally located data lakes.

3. Internet of Things (IoT). NVMe’s high ingest rates and ability to support huge numbers of client connections enable storage arrays that use NVMe to fulfill two roles. In one role, NVMe can act as both edge data collection points. In the other role, NVMe can act as centralized storage pools that can host the resultant data lakes created with all the sensor data.

In some cases, the increases in NVMe performance are significant enough that they enable activities that previously would not have been possible, or would have been cost-prohibitive. For instance:

A storage controller, such as the AFF A300 and single shelf of disks (5 rack units, or 5U), can be placed as an ultra-high data ingest edge sensor collection point. The size/cost performance ratio for the AFF A300 makes what would have once required multiple much larger (and more expensive) arrays to support the ultra-high ingest rate. Additionally, we can use NetApp® SnapMirror® replication technology to mirror the collected data back to a central data lake and possibly use storage located on NetApp Private Storage (NPS) that can store data on NetApp storage adjacent to cloud compute. By using NPS, organizations can host data next to cloud compute resources that can be rapidly brought online and scaled up to parse and manipulate the huge amounts of data located on NPS to extract value from that data in a highly flexible, almost unlimited scale. By using this architecture, organizations can rapidly increase or decrease cloud-sourced compute according to their requirements, while maintaining sovereignty over their data. For more information about NPS, see the NPS page.
3 NVMe as a Storage Attachment Architecture

NVMe is most commonly 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 serial-attached SCSI (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.

3.1 NetApp AFF A800

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

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 like 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 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, which enables organizations to achieve higher performance at lower latencies.
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 4 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 500µs compared with NVMe/FC, which has substantially more headroom (available performance capacity compared to FCP).
4 NVMe over Fabrics

NVMe isn’t just a storage specification. The NVMe over Fabrics (NVMe/FC) 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 Nonvolatile Memory Express 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 all the way 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.

4.1 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, InfiniBand, 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 protocols, such as NVMe/FC.

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.

Figure 5) ONTAP 9.4 delivers end-to-end NVMe flash with NVMe-FC and the new AFF A800.
NVMe-oF defines how NVMe can use existing transport technologies such as FC and Ethernet to transport the NVMe protocol over distances 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. It increases 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

One of the best characteristics of NVMe/FC is that it can coexist concurrently and use the same data network components that FC uses for SCSI access to storage. NVMe/FC thus can coexist on the same hosts, fabrics, and storage that are using FCP, enabling a seamless transition to new technology. This can be done as simply as having one or more FCP LIFs hosted on the same physical HBA port as one or more NVMe/FC LIFs.

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 32 Gbps; 8, 16 and 32 Gbps are the most commonly 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 section 5.
- **NVMe transport using RDMA**: There are several transports that support RDMA:
  - NVMe over InfiniBand (NVMe/IB). This solution uses InfiniBand, which can currently support 100Gbps, as an ultra-high-speed transport. Although it is incredibly fast, InfiniBand 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 100µs in a 2U platform. For more information, see the [EF570 datasheet](https://www.netapp.com/ef570-datasheet.pdf).
- **RDMA over Ethernet transports.**
  - 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 transports RDMA over Converged Ethernet; it offers lower latencies because it doesn’t require TCP. RoCE requires an Ethernet switch that supports Data Center Bridging (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 RDMA-capable network interface cards (NICs) installed. 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 (CNP) to acknowledge receipts.

- **iSCSI Extensions for RDMA (iSER).** An enhancement to the iSCSI protocol to support using RDMA.

Note: NVMe/FC support in ONTAP 9.4 is NetApp's first NVMe-oF offering; support for other transport protocols is planned for future releases.

With ONTAP 9.5, NetApp added support for SUSE Enterprise Linux 15 with Asymmetric Namespace Access (ANA) and Red Hat Enterprise Linux 7.6. For more information, see page 17.

5 **NVMe over Fibre Channel (NVMe/FC)**

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/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.

Figure 7) NVMe/FC versus FCP frames.

- FCP - SCSI-3 command set encapsulated in an FC frame

- FC-NVMe - NVMe command set encapsulated in an FC frame

NetApp has launched NVMe/FC first, because it is by far 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.
NetApp engineering took advantage of several open-source architectural specifications to rapidly develop the ONTAP NVMe/FC 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 International Committee for Information Technology Standards (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)

5.1 NetApp NVMe/FC Release Announcement

The [ONTAP 9.4 Release Notes](https://www.netapp.com) includes the following information about the support for NVMe over Fibre Channel (NVMe/FC):

ONTAP 9.4 introduces NVMe over Fibre Channel (NVMe/FC), an industry first.

NVMe/FC is a new block-access protocol that serves blocks to the host, similar to 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 IMT to verify the latest supported solution stack for ONTAP 9.4.

Note that ONTAP 9.4 implementation of NVMe/FC requires application-level high availability. If a controller loss or path failure occurs, application host needs to manage path failover to its (application) HA partner. This limitation exists because the NVMe multipathing specification called Asymmetric Namespace Access (ANA), analogous to ALUA in SCSI protocol, was still under development.

While implementing NVMe/FC, NetApp has 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.
6  NVMe/FC Features and Limitations

NetApp ONTAP 9.4 has several NVMe/FC protocol restrictions and limitations:

- The debut version of NVMe/FC introduced in ONTAP 9.4 does not support path failover. For details, see section 6.1, “ONTAP 9.4 NVMe/FC Target and Native HA Failover.”
- Storage virtual machines (SVMs), also known as Vservers, are only able to support either NVMe/FC or all the other ONTAP block and file protocols. The restriction was put in place for the ONTAP 9.4 rollout to speed and streamline development while limiting the possibility that the NVMe/FC target code could introduce a regression in any of the other ONTAP protocols. This restriction will be removed in a future ONTAP release.
- The NVMe service must be started before the LIF is created.
- NVMe LIFs and namespaces must be hosted on the same node.
- Only one NVMe LIF is permitted per SVM (Vserver). This restriction will be removed in an upcoming version of ONTAP.
- LUNs and namespaces cannot be mixed on the same volume. This restriction will be removed in an upcoming version of ONTAP.

6.1  ONTAP 9.4 NVMe/FC Target and Native HA Failover

NetApp’s first release with support for NVMe/FC is ONTAP 9.4, which does not have remote path failover and storage-managed high availability (HA). Thus, it needs to rely on application-based HA features, just like Oracle ASM, MongoDB, and other applications that manage failover at the application rather than storage networking layer.

Note: The lack of NVMe/FC HA failover does not affect the failover characteristics for any other ONTAP front-side protocol. ONTAP HA failover works normally for all other protocols: FCP, FCoE, iSCSI, NFS, and CIFS/SMB.

HA failover support is not available in ONTAP 9.4 NVMe/FC because the NVMe/FC protocol is under rapid development. While NetApp SAN target engineering was building the NVMe/FC target, they had to also engineer and create an HA path management and failover specification, because the NVMe/FC protocol lacked one.

In fact, NetApp’s NVM Express committee representative, Fred Knight, was the lead author submitting technical proposals TP 4004 and TP 4028, defining a functional HA error-reporting and failover protocol. The new protocol, ANA, was ratified in March of 2018.

6.2  ONTAP 9.5 NVMe/FC Features - Asymmetric Namespace Access

ONTAP 9.5 introduces ANA as part of the NVMe/FC target. 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 OS stack. ANA requires both the target and initiator to implement and support ANA to function. If either side is not available or implemented, ANA isn’t able to function, and NVMe/FC will fall back to not supporting storage HA. In those circumstances, applications will have to support HA for redundancy.

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 host OS path management such as MPIO or DM-Multipath provide path management and failover capabilities for NVMe/FC. Figure 9 and Figure 10 show the cover pages of both technical proposals submitted to NVM Express, Inc. for ratification. Figure 11 shows the International Committee for Information Technology Standards (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 9) TP 4004: ANA base proposal (ratified 3/18).

<table>
<thead>
<tr>
<th>NVM Express Technical Proposal for New Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Proposal ID</td>
</tr>
<tr>
<td>Change Date</td>
</tr>
<tr>
<td>Builds on Specification</td>
</tr>
</tbody>
</table>

Technical Proposal Author(s)

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred Knight</td>
<td>NetApp</td>
</tr>
<tr>
<td>David Black</td>
<td>Dell EMC</td>
</tr>
<tr>
<td>Curtis Ballard</td>
<td>HPE</td>
</tr>
<tr>
<td>Christoph Heig</td>
<td>WDC</td>
</tr>
</tbody>
</table>

Figure 10) TP 4028: ANA path and transport (ratified 1/18).

<table>
<thead>
<tr>
<th>NVM Express Technical Proposal for New Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Proposal ID</td>
</tr>
<tr>
<td>Change Date</td>
</tr>
<tr>
<td>Builds on Specification</td>
</tr>
</tbody>
</table>

Technical Proposal Author(s)

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred Knight</td>
<td>NetApp</td>
</tr>
<tr>
<td>David Black</td>
<td>Dell EMC</td>
</tr>
<tr>
<td>Sagi Grinberg</td>
<td>LightBlits Labs</td>
</tr>
</tbody>
</table>

Figure 11) INCITS 540-201x, T11 FC-NVMe defines NVMe command and data transport using FC standards.
Table 2) OS Multipathing stacks.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Multipath Software Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX</td>
<td>IBM AIX MPIO</td>
</tr>
<tr>
<td>HP Unix</td>
<td>LVM Alternate Links (PVlinks)</td>
</tr>
<tr>
<td>VMware ESXi</td>
<td>VMware Native Multipathing Plug-In (NMP)</td>
</tr>
<tr>
<td>Linux</td>
<td>dm-multipath or NVMe Multipathing - new MPIO implementation (an alternative to dm-multipath)</td>
</tr>
<tr>
<td>Oracle Solaris</td>
<td>Oracle Solaris Storage Multipathing MPxIO</td>
</tr>
<tr>
<td>Windows</td>
<td>MPIO</td>
</tr>
</tbody>
</table>

ANA has the following two components (as shown in Figure 12):

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

Figure 12) NVMe/FC Storage Failover: ONTAP 9.5 introduces ANA.

At the time of this writing, SuSE Enterprise Linux 15 (SLES 15) was the only host OS that had implemented ANA as part of its initiator stack. Therefore, it is currently the only OS that supports storage HA by having implemented ANA in its initiator stack. NetApp engineering currently has weekly engineering-level meetings to coordinate implementation and qualifications of the respective host OS stacks with ONTAP and its target-side implementation of ANA.

Unlike ALUA, which uses "active optimized" (preferred path) and "nonoptimized" (functional, but not the preferred path) paths, ANA categorizes paths as Active or Inactive. Active paths are both preferred and functional. Inactive paths are neither preferred nor functional. An Inactive path only becomes active in the event of a controller failover, which means there is no remote path support in the current ONTAP ANA implementation. This limitation will be removed in an upcoming ONTAP release to allow host access remote paths.
Note: The implementation of ANA in ONTAP 9.5 doesn’t support remote paths, which means that inactive paths are not usable until a failover event changes them from inactive to active paths.

Figure 13: A comparison NVMe/FC with and without ANA.

6.3 ONTAP 9.6 NVMe-oF Adds 512-Byte Blocks

ONTAP 9.6 introduced 512 byte blocks in addition to the default 4k block that ONTAP has always supported. NetApp added the 512 byte option to be able to more easily support VMware’s 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/ATS.

6.4 Early Releases of SuSE ANA

At the time of this writing (May 2019), SUSE had updated the default multipathing to round-robin so that it could take advantage of multiple active paths by using a round-robin load balancing of frames between the paths presented. If the copy of SUSE SLES15 doesn’t support round-robin, you should consider updating the installation, which should update the multipathing scheme to round-robin by default.

Note: Round-robin multipathing support was added to SLES15, which can now be used as the default multipathing method.

Early version of SLES 15 ANA implementation didn’t implement multipathing or load-balancing support even when multiple active paths to an NVMe namespace existed. NetApp target engineering and SUSE initiator engineering proposed that round-robin be added and made the default for multipathing support on SLES15.

NetApp workload and performance testing show a significant decrease in performance using SLES 15 with ANA support versus the performance seen using SLES 12 sp3 with dm-multipath (without ANA), because, unlike SLES 15, SLES 12SP3 already had load-balancing support.
A simple round-robin load balancer for NVMe multipath is currently being worked on in the upstream kernel to address this limitation. After it is reviewed and merged upstream, SuSE pulls in the corresponding fix to the SLES15 NVMe multipath implementation as well which would eventually show up in an upcoming SLES15 maintenance update (MU) kernel. The expectation is that this process should happen fairly soon after the ONTAP 9.5 RC release.

6.5 ONTAP Feature Support and Coexistence

Table 3 through Table 5 list the ONTAP tools and features that are currently supported, coexist with, or are not supported with NVMe/FC currently with ONTAP 9.5. All of 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/FC target continues to gain functionality and performance.

Table 3) ONTAP features supported with NVMe.

<table>
<thead>
<tr>
<th>ONTAP Tool or Feature</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management:</td>
<td>Thin provisioning:</td>
</tr>
<tr>
<td>• ONTAP System Manager (formerly known as OnCommand System Manager)</td>
<td>• Namespace thin provisioning</td>
</tr>
<tr>
<td>• Active IQ Unified Manager (formerly known as OnCommand Unified Manager)</td>
<td>• Volume thin provisioning</td>
</tr>
<tr>
<td></td>
<td>• Fractional reservations</td>
</tr>
<tr>
<td>Data protection:</td>
<td>Storage efficiency:</td>
</tr>
<tr>
<td>• SnapMirror (BRE and LRSE engines tested)</td>
<td>• Compression – secondary</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Cloning:</td>
<td>• Volume undelete</td>
</tr>
<tr>
<td>• FlexClone—volume clones</td>
<td></td>
</tr>
<tr>
<td>• FlexClone—NVMe namespace cloning</td>
<td></td>
</tr>
<tr>
<td>• Sub-NVMe namespace clone (ONTAP 9.5 +)</td>
<td></td>
</tr>
<tr>
<td>NetApp Snapshot™ copies</td>
<td>Tape (ndmpcopy)</td>
</tr>
<tr>
<td>FabricPools</td>
<td>Aggr Relocate (ONTAP 9.5 +)</td>
</tr>
<tr>
<td>NetApp Volume Encryption (NVE)</td>
<td>Volume rehost (partial - only with unmapped namespaces)</td>
</tr>
</tbody>
</table>

Table 4) ONTAP coexisting features supported by NVMe.

<table>
<thead>
<tr>
<th>Protocols:</th>
<th>VMware:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• CIFS/SMB (SMB 2.1/3.0)</td>
<td>• vStorage API (VAAI) support for NFS and SAN</td>
</tr>
<tr>
<td>• NFS</td>
<td>• VMware VVols</td>
</tr>
<tr>
<td>• FCP</td>
<td></td>
</tr>
<tr>
<td>• iSCSI</td>
<td></td>
</tr>
</tbody>
</table>
### QoS:
- Storage QoS
- Adaptive QoS

### Security:
- NetApp Storage Encryption (NSE)
- Onboard Key Manager (OKM)

<table>
<thead>
<tr>
<th>Volume move</th>
<th>MCC and MCC IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreign LUN Import (FLI)</td>
<td>Cluster peering</td>
</tr>
<tr>
<td>File auditing</td>
<td>IPspaces</td>
</tr>
<tr>
<td>NetApp Flash Cache™</td>
<td>IPv6</td>
</tr>
<tr>
<td>NetApp FPolicy™</td>
<td>Data compaction</td>
</tr>
<tr>
<td>NetApp ONTAP FlexGroup</td>
<td>ODX (MS copy offload)</td>
</tr>
</tbody>
</table>

Table 5) ONTAP Features Not Currently Supported by NVMe.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVMe namespaces move</td>
<td>Not allowed (enforced)</td>
</tr>
<tr>
<td>Volume move</td>
<td>Not allowed (enforced)</td>
</tr>
<tr>
<td>ONTAP Select</td>
<td>FC and FC-NVMe LIFs are not supported in ONTAP Select</td>
</tr>
<tr>
<td>FC Direct Connect</td>
<td>Not supported with FCP and FC-NVMe</td>
</tr>
<tr>
<td>Virtual Storage Console (VSC)</td>
<td>–</td>
</tr>
<tr>
<td>NVMe namespace move</td>
<td>Not allowed (enforced)</td>
</tr>
<tr>
<td>Namespaces (copy on demand)</td>
<td>Not supported for NVMe namespaces</td>
</tr>
<tr>
<td>7toC transition</td>
<td>Creating namespaces on a transitioned volume (SVM) isn’t recommended</td>
</tr>
<tr>
<td>Wafiiron</td>
<td>–</td>
</tr>
<tr>
<td>Sync SnapMirror</td>
<td>–</td>
</tr>
<tr>
<td>Application aware (AppDM/balanced placement)</td>
<td>–</td>
</tr>
</tbody>
</table>

### 7 Getting Started with NVMe/FC

NetApp forecasts that most NVMe/FC uptake that occurs with ONTAP 9.4 and 9.5 will be from organizations that are experimenting with testing and qualifying NVMe/FC 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 HA at the application rather than storage layer. As mentioned previously, some of these applications might include MongoDB or Oracle ASM.
### 8 Interoperability

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

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

#### 8.1 Interoperability Matrix Tool

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

- NetApp target
- Broadcom (Emulex) HBA driver, firmware
- SuSE Enterprise Linux’s NVMe/FC Linux initiator and control interfaces

Any departure from the IMT-listed configurations is likely to perform inconsistently and unexpectedly. The IMT has added a new protocol filter, FC-NVMe (shown in Figure 14), which can be used to check NVMe/FC qualified configurations.

Figure 14) NetApp IMT adds FC-NVMe protocol filter.
9 Best Practices for NVMe/FC

9.1 Currently Published Best Practices

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.

9.2 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. For more details on SAN Health, see:

- Technical Guide: Brocade SAN Health How-To Guide
- Video: SAN Discovery the Easy Way
- SAN Health download application download

The links above are for an internal NetApp site, please contact your NetApp or partner account team for copies of SAN Health and the video and guide.

9.3 NVMe/FC Best Practices: Pathing

To avoid any interface single points of failure, NetApp strongly recommends that you provision at least two paths per SVM, per node, per fabric.

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 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 host's multipathing stack, for instance dm-multipath, then manages path preferences and usage.

While ANA resembles ALUA in many respects, it also has some noteworthy differences. One significant difference is that indirect paths in ALUA are active nonoptimized. This means that these paths can be used but that they will use the cluster interconnect in the absence of a direct path. In ANA indirect paths are labeled Inactive, rather than Nonoptimized. This means that with ANA indirect paths aren’t used and won’t be used except after a takeover or giveback, which changes which paths are in direct vs. indirect. Because of this change of behavior, it is a strong best practice recommendation that storage admins configure at least two paths from each node to each fabric per SVM. This best practice is recommended because it remediates the single point of failure that a single active path represents given that all other paths through other controllers will be inactive and won’t be useable until and if there is a storage failover.
A common proof of concept reliability test is to pull an active link. If you were configured with a single interface per node, per SVM, per fabric and you then pull the active path, the inactive paths won't fail over unless or until there is an SFO. In fact, you might need to trigger a failover to cause the inactive paths to become active. This can be resolved by simply adding a second interface so that if one active path is impacted another will still be operational.

### 9.4 NVMe/FC Setup and Configuration

SLES 12 SP3 is the only host operating system that was qualified with the ONTAP NVMe/FC target for the ONTAP 9.4 release. With ONTAP 9.5, SuSE Enterprise Linux 15 (with ANA) and RHEL 7.6 were added to the list of host operating systems qualified to use NVMe/FC. For detailed implementation instructions, review the appendices at the end of this document.

#### Setup and Configuration Quick List

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

1. Verify that your configuration exactly matches a qualified configuration listed in the NetApp 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 Brocade SAN best practices. See following section called “Where to find Additional Information”
3. Enable N_Port ID virtualization (NPIV) on all fabric switches.
4. 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/FC objects (SVMs, volumes, namespaces, subsystems, and LIFs) by using ONTAP System Manager or the ONTAP command line. For details, see Appendix B: Using ONTAP System Manager to Create ONTAP NVMe/FC Objects and Appendix C: ONTAP NVMe/FC CLI Commands—Initial Setup and Discovery.
6. Use Active IQ Unified Manager to monitor the health and performance of newly created NVMe objects and create reporting thresholds and alerts.

#### 9.5 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 GUI), see Appendix B: Using ONTAP System Manager to Create ONTAP NVMe/FC Objects.
- To use the CLI to set up and configure NVMe/FC in ONTAP, see Appendix C: ONTAP NVMe/FC CLI Commands—Initial Setup and Discovery.
- To set up SuSE 12.3, SuSE 15 or Red Hat Enterprise Linux 7.6 and configure, discover, and connect to an ONTAP NVMe/FC target, see appendices G, H, and I.
- To display NVMe objects and run I/O against them, see Appendix D: CLI—Display Objects and Run I/O.
10 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 15 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 15) NVMe/FC ultra-high-performance design.

Table 6 Figure 16, and Figure 17 display the results of some of the internal testing by NetApp.

Table 2 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. Figure 16 and Figure 17 show 8k and 4k random read performance from the testing in NetApp’s performance characterization labs. The testing was done as a tuning work set before building,
testing, and documenting reference architectures verified by NetApp, Brocade, and Broadcom for several popular critical enterprise applications.

Table 6) AFF A700 4K random read NVMe/FC versus FCP.

<table>
<thead>
<tr>
<th></th>
<th>NVMe/FC</th>
<th>Delta Versus FCP (Percentages)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single port, IOPS</td>
<td>619K</td>
<td>207%</td>
</tr>
<tr>
<td>Single namespace/LUN, IOPS</td>
<td>540K</td>
<td>880%</td>
</tr>
<tr>
<td>Peak IOPS</td>
<td>865K</td>
<td>+51%</td>
</tr>
</tbody>
</table>

Figure 16, Figure 17, and Figure 18 compare 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. Additionally Figure 18 also shows performance improvements when upgrading from ONTAP 9.4 to ONTAP 9.5.

Figure 16) AFF A700 HA pair, ONTAP 9.4, 8K random read FCP versus NVMe/FC.
10.1 Supporting Documentation

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


Additional NVAs 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](#)

**Appendix A: 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. The NVMe standard covers the software stack through to device access for modern storage devices.

NVM Express published a complementary specification, the NVMe Management Interface (NVMe-MI), in November 2015. NVMe-MI is focused primarily on 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.

**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 INCITS T11 Committee FC-NVMe standard, [T11-2017-00145-v004 FC-NVMe](#).

**Appendix B: Using ONTAP System Manager to Create ONTAP NVMe/FC Objects**

Use ONTAP System Manager to create NVMe objects.

1. Create an SVM with NVMe support. NVMe cannot be selected with other protocols in ONTAP System Manager. This is a temporary restriction for the ONTAP 9.4 release. 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 (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:

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

   To display the NQN of a subsystem’s NQN, run the following command:

   ```bash
   vserver nvme subsystem show -vserver <vserver_name>
   ```
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.

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.
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.
4. To go back to the SVM dashboard page, click the Back button 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. Review the NVMe creation summary page. Select either Email Configuration, Copy Configuration CSV, and/or Export Configuration to receive a listing of the configuration.

7. Click Close.

8. To view performance information for a namespace, select the required namespace. Then, click the Performance tab to view the near real-time performance data (latency, throughput, IOPS) of the mapped namespace.
Appendix C: ONTAP NVMe/FC CLI Commands—Initial Setup and Discovery

On the ONTAP Controller

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

```
AFF::> fcp adapter show -data-protocols-supported fc-nvme
    (network fcp adapter show)
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. Create a SVM to host NVMe traffic.

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

3. Display the SVM created.

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

4. Remove all other protocols from the SVM.

```
AFF::> vserver remove-protocols -vserver nvme1 -protocols iscsi,fcp,nfs,cifs,ndmp
```

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.
7. Display the allowed protocols on the SVM.

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

8. Create the NVMe service.

AFF::> vserver nvme create -vserv nvme1

9. Display NVMe service status.

AFF::> vserver nvme show -vserv nvme1

Vserver Name: nvme1
Administrative Status: up

10. Create an NVMe/FC LIF.

AFF::> network interface create -vserv nvme1 -lif fcnvme-node1-la -role data -data-protocol fc

11. Display the newly created LIF.

AFF::> net interface show -vserv nvme1
(network interface show)
Logical Status Network Current Current Is
Vserver Interface Admin/Oper Address/Mask Node Port Home
---------- ---------- ---------- ---------- ---------- ----
nvme1 fcnvme-node1-la
   up/up 20:60:00:a0:98:b3:f7:a7
          AFF_1 1a true

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

AFF::> vol create -vserv 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

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.

13. Create a namespace.

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

14. Create a subsystem.

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

15. Display the newly created subsystem.

AFF::> vserver nvme subsystem show -vserv 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 my subsystem -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 my subsystem -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-ae9-d093e03c4f44
  Restore Inaccessible: false
  Node Hosting the Namespace: AFF_1
  Volume Name: nsvol1
  Qtree Name:
  Attached Subsystem: my subsystem
  Namespace ID: 1
  Vserver ID: 89
```

**Appendix D: CLI—Display Objects and Run I/O**

**On the Controller**

1. **Display FCP initiators (you should see the SLES initiator logged in now).**

```
>fcp initiator show
(vserver fcp initiator show)

Logical Port  Initiator   Initiator
Vserver  Interface     Address WWNN     WWPN     Igroup
--------- ----------------- --------- -------- --------
LNX_nvme LNX_nvme     10200  20:00:00:10:9b:1c:0f:8f
           10:00:00:10:9b:1c:0f:8f
           NVMe_demo AFF_1_lif 10200  20:00:00:10:9b:1c:0f:8f
           10:00:00:10:9b:1c:0f:8f
           NVMe_test AFF_1_lif 10200  20:00:00:10:9b:1c:0f:8f
           10:00:00:10:9b:1c:0f:8f

3 entries were displayed.
```
2. Display the SVM. Note the allowed (NVMe) and disallowed protocols (NFS, CIFS, FCP, iSCSI, and NDMP).

   ```
   > vserver show -vserver NVMe_test
   
   Vserver: NVMe_test
   Vserver Type: data
   Vserver Subtype: default
   Vserver UUID: 87b3c578-fee8-11e7-94f2-00a098b3f653
   Root Volume: NVMe_test_root
   Aggregate: AFF_1_SSD_1
   NIS Domain: -
   Root Volume Security Style: unix
   LDAP Client: -
   Default Volume Language Code: C.UTF-8
   Snapshot Policy: default
   Comment: 
   Quota Policy: default
   Limit on Maximum Number of Volumes allowed: unlimited
   Vserver Admin State: running
   Vserver Operational State: running
   Vserver Operational State Stopped Reason: -
   Allowed Protocols: nvme
   Disallowed Protocols: nfs, cifs, fcp, iscsi, ndmp
   Is Vserver with Infinite Volume: false
   QoS Policy Group: -
   Caching Policy Name: -
   Config Lock: false
   Volume Delete Retention Period: 12
   IPspace Name: Default
   Foreground Process: -
   Is Mailbox Preserved for DR: false
   Is this Vserver a source or destination of migrate: -
   Force start required to start Destination in multiple IDP fan-out case: false
   
   3. Display the NVMe service on the SVM.

   ```
   > vserver nvme show -vserver NVMe_test
   
   Vserver Name: NVMe_test
   Administrative Status: up
   Vserver UUID: 87b3c578-fee8-11e7-94f2-00a098b3f653
   Vifowner Id: NVMe_test
   
   4. Display the new LIF.

   ```
   > vserver nvme show-interface -vserver NVMe_test
   
   Vserver Logical Interface  Home Node:Port   Transport Protocols
   ------- ----------------- --------------------------
   NVMe_test  AFF_1_lif      AFF_1:1a      fc-nvme
   Transport Address: nn-0x203300a098b3f7a7:pn-0x203500a098b3f7a7
   
   5. Display the new namespace.

   ```
   > vserver nvme namespace show -vserver NVMe_test
   
   Vserver Path State  Size  Subsystem NSID
   ------- ------ ---------- ---------- ----------
   NVMe_test 
   /vol/NVMe_test_volume_1/NVMe_test_namespace_322
   online 120GB NVMe_test_subsystem
   /vol/NVMe_test_volume_1/NVMe_test_namespace_1
   online 150GB NVMe_test_subsystem
   2 entries were displayed.
   
   6. Display subsystems on the SVM named NVMe_test. This is a good method to list the NQN for the SVM.
   ```
>vserver nvme subsystem show -vserver NVMe_test
Vserver Subsystem Target NQN
--------- ----------------------------------------
NVMe_test
  NVMe_test_subsystem
nqn.1992-08.com.netapp:sn.87b3c578feeb11e794f200a098b3f653:subsystem.NVMe_test_subsystem

On the Host

1. Display the block devices.

```
SLES_host:~ # lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda    8:0  0 279.4G 0 disk
├─sda1  8:1  0  156M 0 part /boot/efi
├─sda2  8:2  0  2G 0 part [SWAP]
├─sda3  8:3  0  40G 0 part /var/lib/mariadb
└─sda4  8:4  0 237.2G 0 part /home
sr0   11:0  1  3.6G 0 rom
nvme0n1 259:0  0  200G 0 disk
```

2. Display the HBAs installed in the host.

```
SLES_host:~ #./lpfc_nvme_list.sh
Available HBAs for NVME Initiator
lpfc0 WWPN x100000109b1c0f8f WWNN x200000109b1c0f8f DID x010200 ONLINE
WWPN x202800a98b3f7a7 WWNN x203000a98b3f7a7 DID x010507 TARGET DISCSRVC ONLINE
WWPN x2040000a98b3f7a7 WWNN x203000a98b3f7a7 DID x010507 TARGET DISCSRVC ONLINE
WWPN x203500a98b3f7a7 WWNN x203300a98b3f7a7 DID x01050a TARGET DISCSRVC ONLINE
lpfc1 WWPN x100000109b1c0f90 WWNN x200000109b1c0f90 DID x000000 UNKNOWN
```

On the Controller

1. Display NVMe I/O (to display I/O, you generate from the host).

```
>run local -command sysstat -n
CPU  NFS  FCP  NVMe  Net  kB/s Disk  kB/s NVMe  kB/s Cache
   in  out  read  write  in  out  read  write  age
1%  0   0   0    2    16  0   0   0   >60
1%  0   0   0    5    16  0   0   0   >60
1%  0   0   0    1    24  0   0   0   >60
1%  0   0   0   120   5  16  0   0   >60
```

On the Host

1. Run I/O on the host server (use "-" per argument to add arguments to the fio command).

```
SLES_host:~ #fio --filename=/dev/nvme0n1 --direct=1 --rw=randrw --refill_buffers --norandommap --randrepeat=0 --bs=4k --rwmixread=50 --iodepth=16 --numjobs=16 --runtime=15 --group_reporting --name=512b_read_write_test --output=NVMe_read_write_stats
Jobs: 16 (f=16):  [m(16)] [20.0%] [r=175MiB/s,w=176MiB/s] [r=44.8k,w=45.1k IOPS] [etaJobs: 16 (f=16): [m(16)] [26.7%] [r=185MiB/s,w=186MiB/s] [r=47.4k,w=47.3k IOPS] [etaJobs: 16 (f=16): [m(16)] [33.3%] [r=186MiB/s,w=186MiB/s] [r=47.6k,w=47.7k IOPS] [etaJobs: 16 (f=16): [m(16)] [40.0%] [r=187MiB/s,w=186MiB/s] [r=47.8k,w=47.7k IOPS] [etaJobs: 16 (f=16): [m(16)] [46.7%] [r=180MiB/s,w=181MiB/s] [r=46.0k,w=46.4k IOPS] [etaJobs: 16 (f=16): [m(16)] [53.3%] [r=185MiB/s,w=185MiB/s] [r=48.0k,w=40.5k IOPS] [etaJobs: 16 (f=16): [m(16)] [60.0%] [r=194MiB/s,w=195MiB/s] [r=49.8k,w=49.0k IOPS] [etaJobs: 16 (f=16): [m(16)] [66.7%] [r=196MiB/s,w=197MiB/s] [r=50.2k,w=50.4k IOPS] [etaCSLES_host:~
```

Appendix E: Troubleshooting

Before you troubleshoot any of the NVMe/FC failures, always make sure that you are running a configuration that is compliant with the NetApp 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:

```c
#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 initramfs 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.

```bash
# 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):** 0x00000083

**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:

```bash
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
NVME LPORT lpfc0 WWPN x10000090fae0ec9d WWNN x20000090fae0ec9d DID x012000 ONLINE
NVME REPORT WWPN x200a00a098c80f09 WWNN x200a00a098c80f09 DID x010601 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000000 Cmpl 0000000000000000
FCP: Rd 0000000000000000 Wr 0000000000000000
Cmpl 0000000000000000
NVME Initiator Enabled
NVME LPORT lpfc1 WWPN x10000090fae0ec9e WWNN x20000090fae0ec9e DID x012400 ONLINE
NVME REPORT WWPN x200900a098c80f09 WWNN x200800a098c80f09 DID x010301 TARGET DISCSRVC ONLINE
```
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 F: NVMe/FC Scalability and Limits**

Scalability and limits for NVMe/FC at the time of this writing can be found in the [SAN Configuration Guide](#). The current limits aren’t high, because they were chosen to speed NVMe/FC development. The limits should be sufficient to allow robust testing and qualification and will be increased in future releases of ONTAP.

**Appendix G: Configuring and Connecting SuSE Enterprise Linux to ONTAP NVMe/FC Namespaces**

This appendix describes how to configure SLES12 SP3 as an initiator connecting to the ONTAP 9.4 (and later) NVMe/FC target. All specific hardware and software, including firmware and service pack levels, were and are the currently qualified levels. However, because NVMe/FC is under rapid development, versions can change, and the number of items qualified will continue to expand rapidly. For this reason,
and because it is always a best practice, review the NetApp IMT and adjust the versions and hardware specifications to those listed in the tool.

**Features and Limitations**

The supported configurations and limitations are based on the NVMe-oF and NVMe/FC testing and qualifications at the time of writing. As previously stated, review the NetApp IMT and the release notes for any updates to the following items and any new features or limitations that are introduced going forward.

**Supported Configurations**

- NVMe/FC is currently supported only on 16GB and 32Gb fabrics with 32Gbps HBAs with ONTAP 9.4 and SLES12 SP3 as the initiator.
- Only Broadcom 32Gbps FC adapters (such as LPe3200X) can be used for NVMe/FC support (host and target).

**Single LIF per SVM limitation**

ONTAP 9.4 has a single LIF per NVMe/FC SVM limitation that will be lifted in future releases.

**No NVMe/FC Path Failover support**

Due to the single LIF per NVMe/FC SVM limitation, there is no NVMe/FC path failover support in ONTAP 9.4.

*Note:* ANA, which formalizes NVMe multipathing and path failover support, was ratified in March of 2018. It will be adopted by host OS vendors and added to an upcoming version of ONTAP.

**No NVMe/FC Linux Host Utilities Support**

There is no SAN LUN support for NVMe/FC and no LUHU support for NVMe/FC on SLES15 (just as in SLES12 SP3); therefore, there is no SAN LUN support either. The NetApp SAN engineering team is bolstering the nvme-cli host utility to display ONTAP specific details.

**Protocol Coexistent Host Configurations**

The SLES12 SP3 initiator host can serve both NVMe/FC and FC-SCSI traffic through the same 32Gbps FC adapter ports. In fact, that is expected to be the commonly deployed host configuration for end users. For FCP (FC-SCSI), you can configure dm-multipath as usual for the SCSI LUNs resulting in multipath devices. This configuration allows for normal multipath operation for those FCP LUNs without impacting the NVMe/FC namespace devices (such as /dev/nvmeXnY) on the SLES12SP3 host, which will continue to be single pathed and won’t support path failover.

**Configuration Requirements**

**Host Bus Adapter**

- 32Gbps FC Broadcom/Emulex LPe3200X adapters used as NVMe/FC initiators and targets
- LPe3200X firmware v11.4.367.4 available on the Broadcom site here
- LPe3200X adapter firmware v11.4.204.32 available on the Broadcom site here

**Fabric Switch**

- 32Gbps FC Brocade or Cisco switch with supported FOS versions (refer to the Switch IMT matrix for switch models and switch OS versions)
- Single-initiator zoning enabled for the initiator and target ports
Initiator
- SLES12 SP3 NVMe/FC host running:
  - kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704 (available from SuSE)
  - nvme-cli-1.2-6.15.1 (available at SuSE's Patch Finder)

Target
- ONTAP 9.4 or later
- Controller models:
  - NetApp AFF A800
  - NetApp AFF A700
  - NetApp AFF A700s
  - NetApp AFF A300
- Any of the NetApp controllers listed above require one or more 32Gbps target HBAs.
  For specific options available for your controller, see the NetApp Hardware Universe.

Enabling NVMe/FC
Configuring SLES12 SP3 Initiator
To enable NVMe/FC on the SLES12 SP3 initiator host, complete the following steps:

1. Separately install the SLES12 SP3 PTF Kernel MU kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704 before you set up the initiator.
   ```
   # rpm -ivh kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704.x86_64.rpm
   kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704.noarch.rpm
   kernel-devel-4.4.126-94.22.2.15326.0.PTF.1091704.noarch.rpm
   kernel-firmware-20170530-21.7.1.noarch
   kernel-devel-4.4.126-94.22.2.15326.0.PTF.1091704.noarch
   kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704.x86_64
   kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704.x86_64
   #
   # Reboot into this MU kernel and then set up the initiator.
   
   # rpm -Uvh nvme-cli-1.2-6.15.1.x86_64.rpm
   # rpm -qa|grep nvme-cli
   nvme-cli-1.2-6.15.1.x86_64
   
   4. Separately install the SLES12 SP3 nvme-cli-1.2-6.15.1 MU package.
   ```

2. Reboot into this MU kernel and then set up the initiator.
3. Verify that you are running with the specified SLES12 SP3 kernel.
   ```
   # uname -r
   4.4.126-94.22.2.15326.0.PTF.1091704-default
   # rpm -qa|grep kernel
   kernel-source-4.4.126-94.22.2.15326.0.PTF.1091704.noarch
   kernel-macros-4.4.126-94.22.2.15326.0.PTF.1091704.noarch
   kernel-firmware-20170530-21.7.1.noarch
   kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704.noarch
   kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704.x86_64
   kernel-default-devel-4.4.126-94.22.2.15326.0.PTF.1091704.x86_64
   #
   5. Install the recommended lpfc off box driver and associated NVMe/FC autoconnect scripts by first downloading the respective lpfc tar ball, untarring it, and then cd into it. And then run:
   ```
   # ./elx_lpfc_install.sh -i -n
   ```

6. Next, upgrade the adapter firmware to the recommended version, Check the ONTAP Interoperability Matrix Tool (IMT) for current qualified versions, at the time of this writing - 11.4.204.32. And then reboot the host.
7. After rebooting, verify that you are using the specified LPe3200X adapter and recommended lpfc firmware and driver versions and NVMe/FC autoconnect package on the host:
8. To enable SLES12 SP3 as an NVMe/FC initiator, you need to set `lpfc_enable_fc4_type` to 3 (which ensures the lpfc driver is enabled to support both NVMe/FC and FC-SCSI). This can be done by specifying these settings in the modprobe.d, and then reloading the lpfc driver:

```bash
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_enable_fc4_type=3
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

9. Rebuild the initramfs by using `dracut --force` for NVMe to be enabled during bootup. And then reboot the host.

   **Note:** After rebooting, verify that the initiator ports are up and running, as well as all the recommended NVMe/FC modules are loaded.

```bash
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec9d
0x10000090fae0ec9e

# cat /sys/class/fc_host/host*/port_state
Online
Online
```

10. Verify that the NVMe/FC initiator ports are enabled and able to see the target LIFs, and that all are up and running.

```bash
# 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 DISCSRV C ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 0000000000000006 Wr 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 DISCSRV C ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 0000000000000006 Wr 00000000000000073 Wr 0000000000000005 IO 0000000000000031
```
11. Verify the `hostnqn` string at `/etc/nvme/hostnqn` and make sure that it properly matches the `hostnqn` string for the corresponding subsystem on the NetApp array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:a9d5d070-2c77-400e-96c4-b6d4aa220c71
```

```
*> vserver nvme subsystem host show -vserver vs_nvme_1
Vserver Subsystem Host NQN
-------- -----------------------------------------------------------
vs_nvme_1 sles_115_nvme_ss_1
nqn.2014-08.org.nvmexpress:uuid:a9d5d070-2c77-400e-96c4-b6d4aa220c71
```

12. Discover and connect to these target ports by using the `nvme-cli` utility, which results in the creation of NVMe/FC namespace devices. The `--traddr` refers to the target port WWNNs/WWPNs, whereas the `--host-traddr` refers to the initiator port WWNNs/WWPNs. On the target, we created two SVMs with two subsystems, with one namespace each. You should see two NVMe/FC namespace devices on the initiator host, with one underlying path each.

```
# nvme connect --all --transport=fc --traddr=nn-0x200a00a098c80f09:pn-0x200b00a098c80f09 --host-traddr=nn-0x200a00a098c80f09:pn-0x200b00a098c80f09
# nvme connect --all --transport=fc --traddr=nn-0x200800a098c80f09:pn-0x200900a098c80f09 --host-traddr=nn-0x200800a098c80f09:pn-0x200900a098c80f09
# nvme list
```

```
Node SN Model Namespace Usage Format FW Rev
-------- -----------------------------------------------------------
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 128.85 GB / 128.85 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme1n1 80BADBKnB/JsAAAAAAAC NetApp ONTAP Controller 1 118.11 GB / 118.11 GB 4 KiB + 0 B FFFFFFFF
```

```
# cat /sys/block/nvme0n1/uuid
71bcfd72-dbdb-4bdf-8ad3-bf5866e1e28c
# cat /sys/block/nvme1n1/uuid
d856ab9e-6489-475b-9e46-850f7447cb6e
```

13. Verify that the UUID strings in step 12 match the respective namespace UUIDs on the NetApp array.

```
*> nvme namespace show -vserver vs_nvme_1 -path * -fields uuid (vserver nvme namespace show)
vs_nvme_1 /vol/sles_115_vol_1/sles_115_ns_1 71bcfd72-dbdb-4bdf-8ad3-bf5866e1e28c
vs_nvme_1 /vol/sles_115_vol_1/sles_115_ns_1_1 d856ab9e-6489-475b-9e46-850f7447cb6e
```

```
# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 278.9G 0 disk
├─sda1 8:1 0 2G 0 part [SWAP]
├─sda2 8:2 0 130G 0 part /var/lib/pgsql
├─sda3 8:3 0 146.9G 0 part /home
sr0 11:0 1 1024M 0 rom
nvme0n1 259:0 0 120G 0 disk
nvme1n1 259:1 0 110G 0 disk
```

14. Since the `nvmefc-connect` package is already installed, which would take care of configuring the devices during bootup as well as auto connecting during link bounce events (LIF/Port toggles), reboot the host and verify that the NVMe devices in step 13 are discovered during bootup as well.
Appendix H: Discovering and Connecting to an NVMe Namespace from a SuSE Linux 12 sp3 Host Server

In order to use NVMe/FC, your host server must find and connect to one or more NVMe namespaces. Currently, the Broadcom (Emulex) supplied firmware and drivers in the SuSE host use several scripts and commands to list, discover, and connect to NVMe namespaces, those include:

- **Lsblk.** Lists all block devices seen by the host server.
- **Lpfc_nvme_list.sh** (/usr/sbin/lpfc_nvme_list.sh). The list command is used to show the initiators and target WWNNs and WWPNs that can be seen on the host.
- **Nvme discovery.** Run the nvme discovery command to discover and list any NVMe devices that the host can see.
- **Nvme connect.** Run the connect command to connect to one or more of the NVMe subsystems that the host server can see.

**List, Discover, and Connect to an NVMe Namespace from SuSE Linux Host (Summary)**

To list, discover, and connect to an NVMe namespace from SuSE Linux, you will complete the following tasks:

1. List all of the block devices that the host server can see.
2. List all of the NVMe WWNs that the host server can see.
3. Create the **discover** and **connect** commands with the correct transport and NQNs. Then issue the commands.
4. List all of the block devices that the host server can see.

**List, Discover, and Connect to an NVMe Namespace from SuSE Linux Host (Procedure)**

To list, discover, and connect to an NVMe namespace from SuSE Linux, complete the following steps:

1. List all the block devices that the host server can see.

```bash
# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda     8:0  0 279.4G  0 disk
├─sda1  8:1  0  156M  0 part /boot/efi
├─sda2  8:2  0  2G   0 part [SWAP]
├─sda3  8:3  0  40G   0 part /var/lib/mariadb
└─sda4  8:4  0 237.2G  0 part /home
sr0    11:0  1  3.6G   0 rom
```

2. List all the NVMe WWNs that the host server can see.

```bash
# ./lpfc_nvme_list.sh
Available HBAs for NVME Initiator
lpfc0 WWPN x100000109b1c0f8f VMware x200000109b1c0f8f DID x010200 ONLINE <<Init WWNs
WWPN x202800a98b3f7a7 WWNN x203000a98b3f7a7 DID x010501 TARGET DISCSRVC ONLINE <<Target
WWPN x204000a98b3f7a7 WWNN x203f000a98b3f7a7 DID x010504 TARGET DISCSRVC ONLINE LIF WWNs
seen from lpfc0 (2x LIFS found), in this example we use the 2nd LIF's WWNs.
lpfc1 WWPN x100000109b1c0f90 WWNN x200000109b1c0f90 DID x000000 UNKNOWNN
```

3. Create the **discover** and **connect** commands with the correct transport and NQNs.
4. Use the outputs from the commands in step 2 to construct the `nvme discover` command.

**Note:** The examples are color coded to ensure that the correct variables are entered as the correct arguments. First is the command format followed by a full example of the command and command output.
Here's an example discovery command using the WWNs listed above. We will use the NQNs from its output to run `nvme connect`.

```bash
# nvme discover --transport=fc -traddr=nn-0x203f00a098b3f7a7:pn-0x204000a098b3f7a7 --host-traddr=nn-0x200000109b1c0f8f:pn-0x1000000109b1c0f8f
```

Discovery Log Number of Records: 1, Generation counter: 17

=====Discovery Log Entry 0======
trtype: fibre-channel
adrfam: fibre-channel
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: none
subnqn: nqn.1992-08.com.netapp:sn.5b584a86fa6411e794f200a098b3f653:subsystem.NVMe_demo_subsystem
traddr: nn-0x203f00a098b3f7a7:pn-0x204000a098b3f7a7

After you issue the discover command, you have the required Subsystem NQN to create the `nvme connect` command. As in the previous example, first the command structure and then an example of the command. There is no output.

```bash
nvme connect --transport=fc -traddr=nn-0x200000109b1c0f8f:pn-0x1000000109b1c0f8f --host-traddr=nn-0x200000109b1c0f8f:pn-0x1000000109b1c0f8f -n subsystem_NQN
```

The following is an `nvme connect` command using the outputs from `list` and `discover` command above:

```bash
nvme connect --transport=fc -traddr=nn-0x203f00a098b3f7a7:pn-0x204000a098b3f7a7 --host-traddr=nn-0x200000109b1c0f8f:pn-0x1000000109b1c0f8f -n nqn.1992-08.com.netapp:sn.5b584a86fa6411e794f200a098b3f653:subsystem.NVMe_demo_subsystem
```

List all block devices the host server can see.

```bash
# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda     8:0    0 279.4G  0 disk
├─sda1   8:1    0   156M  0 part /boot/efi
├─sda2   8:2    0   2G   0 part [SWAP]
├─sda3   8:3    0   40G   0 part /var/lib/mariadb
└─sda4   8:4    0 237.2G  0 part /home
sr0     11:0    1   3.6G  0 rom
nvme0n1 259:0    0  200G  0 disk <<< New 200GB NVMe namespace
```

Command Options for NVMe Discover and Connect Commands

```bash
# nvme connect -t
Usage: nvme connect <device> [OPTIONS]
Connect to NVMe-oF subsystem

Options:
  [ --transport=[LIST], -t [LIST] ] --- transport type
  [ --nqn=[LIST], -n [LIST] ] --- nqn name
  [ --traddr=[LIST], -a [LIST] ] --- transport address
  [ --trsvcid=[LIST], -s [LIST] ] --- transport service id (e.g. IP port)
  [ --host-traddr=[LIST], -w [LIST] ] --- host traddr (e.g. FC WWN's)
  [ --hostnqn=[LIST], -q [LIST] ] --- user-defined hostnqn
  [ --hostid=[LIST], -i [LIST] ] --- user-defined hostid (if default not used)
  [ --nr-io-queues=[LIST], -i [LIST] ] --- number of io queues to use (default is core count)
  [ --queue-size=[LIST], -Q [LIST] ] --- number of io queue elements to use (default 128)
  [ --keep-alive-tmo=[LIST], -k [LIST] ] --- keep alive timeout period in seconds
  [ --reconnect-delay=[LIST], -c [LIST] ] --- reconnect timeout period in Seconds
```
NVMe/FC Scalability and Limits

At the time of this writing, the scalability and limits for NVMe/FC were described in the SAN Configuration Guide. The current limits aren’t particularly high because they were chosen to speed NVMe/FC development. The limits should be sufficient to allow for robust testing and qualification and will be increased in the future releases of ONTAP.

Troubleshooting

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

lpfc Verbose Logging for NVMe/FC

The following is the list of lpfc driver logging bitmasks that are available for NVMe/FC, as seen at drivers/scsi/lpfc/lpfc_logmsg.h:

```c
#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 initramfs 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.

```bash
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_enable_fd4_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): 0x00080000
Common nvme-cli Errors and Workarounds

- **Error message:** The nvme-cli utility displays “Failed to Write to /dev/nvme-fabrics: Invalid Argument” error during NVMe discover, NVMe connect, or NVMe connect-all.
  
  **Probable cause:** This error message is generally displayed if the syntax is wrong.
  
  **Workaround:** Make sure that you are using the correct syntax for the above NVMe commands.

- **Error message:** The nvme-cli utility displays “Failed to Write to /dev/nvme-fabrics: No Such File or Directory” during nvme discover, nvme connect or nvme connect-all.
  
  **Probable cause:** Multiple issues could trigger this message. For example, you passed wrong arguments to the above NVMe commands.
  
  **Workaround:** Make sure that you passed the proper arguments (such as the proper WWNN string, WWPN string, and so on) for the above commands.

If the arguments are correct, and you are still seeing this error, check to see if the 

```
# cat /sys/class/scsi_host/host*/nvme_info
```

output is correct with the NVMe initiator showing as “Enabled” and NVMe/FC target LIFs properly showing up here under the remote ports sections.

For example:

```
NVME Initiator Enabled
NVME LPORT lpfc0 WWPN x10000090fae0ec9d WWNN x20000090fae0ec9d DID x012000 ONLINE
NVME RPORT WWPN x200b00a098c80f09 WWNN x200a00a098c80f09 DID x010601 TARGET DISCSRVVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 0000000000000003 Wr 0000000000000003 IO 00000000000000031
Cmpl 0000000000000006 Outstanding 0000000000000001
NVME Initiator Enabled
NVME LPORT lpfc1 WWPN x10000090fae0ec9e WWNN x20000090fae0ec9e DID x012400 ONLINE
NVME RPORT WWPN x200900a098c80f09 WWNN x200800a098c80f09 DID x010301 TARGET DISCSRVVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 0000000000000073 Wr 0000000000000073 IO 0000000000000073
Cmpl 0000000000000006 Outstanding 0000000000000001
```

**Workaround:** If the target LIFs don’t show up as above in the nvme_info output, verify the 

```
/va... 
```

for any suspicious NVMe/FC failures, and report or fix them accordingly.

- **Error message:** The nvme-cli utility displays “No Discovery Log Entries to Fetch” during the NVMe discover, NVMe connect, or NVMe connect-all.
  
  **Probable cause:** This error message is generally displayed if the /etc/nvme/hostnqn string was not added to the corresponding subsystem on the NetApp array. Or an incorrect hostnqn string was added to the respective subsystem.
  
  **Workaround:** Make sure that the exact /etc/nvme/hostnqn string is added to the corresponding subsystem on the NetApp array (verify through the ‘vserver nvme subsystem host show’).

- **Error message:** The nvme-cli utility displays “Failed to Write to /dev/nvme-fabrics: Operation Already in Progress” during the NVMe discover, NVMe connect, or NVMe connect-all.
  
  **Probable cause:** This error message is displayed if the controller associations or specified operation is already created or in the process of being created. This error could happen as part of the auto-connect scripts that were previously installed.

**Workaround:** There is no workaround. For NVMe discover, run this command after some time. For the NVMe connect and connect-all, run an NVMe list to verify that the namespace devices are already created and displayed on the host.
Files and Command Outputs Required for Debugging

If you are still facing issues, collect the following files and command outputs and open a support case with NetApp support 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-0x200a00a098c80f09 --host-traddr=nn-0x20000090fae0ec9d:pn-0x10000090fae0ec9d
nvme list
```

Appendix I: Configuring and Connecting SuSE Enterprise Linux 15 to ONTAP NVMe/FC Namespaces

At the time of this writing, SLES15 was the only enterprise host OS supporting NVMe/FC with ANA required for storage high availability on the NetApp array running ONTAP 9.5. ANA performs the same function and works similarly to ALUA in FCP and is currently implemented with NVMe multipath (and not dm-multipath) on SLES15. This appendix details the host integration steps for enabling NVMe/FC ANA support on SLES15 and ONTAP 9.5 as the target. All specific hardware and software, including firmware and service pack levels, were and are the currently qualified levels. However, because NVMe/FC is under rapid development, versions can change, and the number of items qualified will continue to expand rapidly. For this reason, and because it is always a best practice, check the NetApp IMT and adjust versions and hardware specifications to those listed in the tool.

Features and Limitations

The supported configurations and limitations are based on the NVMe-oF and NVMe/FC testing and qualifications at the time of writing. As mentioned above, review the NetApp IMT and release notes for any updates to the items listed below and any new features or limitations that are introduced going forward.

Supported Configurations

- NVMe/FC is currently supported only on 16Gb and 32Gb fabrics with 32Gbps HBAs with ONTAP 9.4 and SLES12 SP3 as the initiator.
- Only Broadcom 32Gbps FC adapters (such as LPe3200X) can be used for NVMe/FC support (host and target).
  
  Note: ANA is supported by NVMe multipath but not dm-multipath in SLES15.

No NVMe/FC Linux Host Utilities Support

There is no SANLUN support for NVMe/FC and there is no LUHU support for NVMe/FC on SLES15 (just as in SLES12 SP3); therefore, there is no SANLUN support. The NetApp SAN engineering team is bolstering the nvme-cli host utility to display ONTAP specific details.

Coexistent Host Configurations

The SLES15 initiator host can serve both NVMe/FC and FC-SCSI traffic through the same 32Gbps FC adapter ports. In fact, this is expected to be the commonly deployed host config for end users. For FCP (FC-SCSI), you can configure dm-multipath as usual for the SCSI LUNs resulting in mpath devices, this allows for normal multipath operation for those FCP LUNs without impacting the NVMe/FC namespace devices (such as /dev/nvmeXnY) on the SLES12SP3 host, which continues to be single-pathed and won’t support path failover.
Configuration Requirements

Host Bus Adapter

Broadcom/Emulex
- 32Gbps FC LPe32002 adapters used as NVMe/FC initiators on the SLES15 NVMe/FC ANA host
- LPe32002 firmware v12.0.193.13 (available on the Broadcom site)
- Outbox driver v12.0.288.10 (can be downloaded from the Broadcom/Emulex site)

Fabric Switch
- Gen 5 or Gen 6 FC switches from either Brocade or Cisco that are qualified and listed in the NetApp IMT.
- All zoning must use single-initiator zoning enabled for the initiator ports and target LIFs.

Initiator
During the operating system installation, NetApp recommends that you select all packages included in the base system, base development, and Linux kernel development under the Software section. Alternately, you can install the requisite packages by using the Zypper utility post the SLES15 OS install, after you mounted the Installer-1 & Packages-1 ISOs using the Virtual Media Wizard on your server.

Target
- ONTAP 9.5 or later
- Controller models:
  - NetApp AFF A800
  - NetApp AFF A700
  - NetApp AFF A700s
  - NetApp AFF A300
- Any of these NetApp controllers requires one or more 32Gbps target HBAs. For specific options that are available for your controller, see the NetApp Hardware Universe.

Enabling NVMe/FC with ANA

Configuring ONTAP ANA Target
The ANA target implementation in ONTAP 9.5 doesn’t require any user intervention and doesn’t have any tuning options. Both ANA and non-ANA OSs are supported concurrently by ONTAP 9.5, with no user intervention or adjustments required:
- When using an OS with ANA support, ONTAP 9.5 or later, they support storage HA and use ANA.
- When non-ANA OSs are connected to ONTAP 9.5, they are supported without storage HA support.

Configuring SLES15 ANA Initiator
1. In SLES15, NVMe multipath is enabled by default. NetApp recommends using the MPIO configuration to enable ANA. You can verify that NVMe multipath is enabled by running the following command:

```bash
# cat /sys/module/nvme_core/parameters/multipath
Y
```

2. If you want to turn off NVMe multipath and enable dm-multipath instead for NVMe namespace devices, then pass `nvme-core.multipath=N` in the kernel command line, either in the grub boot
loader or during bootup. After bootup, you can verify that this was applied by running the same `cat` command:

```bash
# cat /sys/module/nvme_core/parameters/multipath
```

3. After SLES15 is installed, install the SLES15 `nvme-cli-1.5-7.5.1` MU package by running the following commands:

```bash
# rpm -Uvh nvme-cli-1.5-7.5.1.x86_64.rpm
# rpm -qa | grep nvme-cli
```

4. Reboot the host and make sure you boot into the newly installed KOTD kernel. After booting, verify that you are running with the recommended packages here.

```bash
# uname -r
4.12.14-172.gb6e2ad9-default
# rpm -qa | grep kernel
```

5. Make sure that the respective NVMe/FC modules have been loaded.

```bash
# lsmod | grep nvme_
nvme_fc 36864 1 lpfc
nvme_fabrics 24576 1 nvme_fc
nvme_core 94208 2 nvme_fc,nvme_fabrics
```

6. Depending on the FC adapter that you are using, proceed to the following sections.

**Broadcom FC Adapter**

1. Verify that you are running with the recommended adapter and firmware versions.

   For example:

   ```bash
   # cat /sys/class/scsi_host/host*/modelname
   LPe32002-M2
   LPe32002-M2
   # cat /sys/class/scsi_host/host*/modeldesc
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   # cat /sys/class/scsi_host/host*/fwrev
   12.0.193.13, sli-4:2:c
   12.0.193.13, sli-4:2:c
   ```

2. To enable SLES15 lpfc as an NVMe/FC initiator, set `lpfc_enable_fc4_type` to 3. This configuration ensures that the lpfc driver is enabled to support both NVMe/FC and FC-SCSI. You can set this configuration by specifying the lpfc setting in the `modprobe.d`.

   ```bash
   # cat /etc/modprobe.d/lpfc.conf
   options lpfc lpfc_enable_fc4_type=3
   ```

3. Install the recommended Broadcom outbox driver and the autoconnect scripts.

   **Note:** You can download the driver and script from the [Broadcom/Emulex site](https://www.broadcom.com)

   ```bash
   # tar -xvzf elx-lpfc-dd-sles15sp-12.0.288.10-ds-1.tar.gz
   # cd elx-lpfc-dd-sles15sp-12.0.288.10-ds
   # ./elx_lpfc_install.sh -i -n
4. Reboot the host and verify that the recommended lpfc outbox driver and autoconnect package are installed after the bootup.

```
# cat /sys/module/lpfc/version
0:12.0.288.10
# rpm -qa | grep nvmefc
nvmefc-connect-12.0.288.10-1.sles15sp0.x86_64
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

5. Verify that the initiator ports are up and running.

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec61
0x10000090fae0ec62
# cat /sys/class/fc_host/host*/port_state
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and able to see the target ports, and all are up and running. In this example, only one initiator port is enabled and connected with two target LIFs, as shown in the following output:

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2947 ELS 250
NVME LPF0 lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201 TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000000001a680 Issue 000000000001a682 OutIO 0000000000000002
abort 00000000 noxri 00000000 nondip 00000000 gdepth 00000000 wqerr 00000000
Err 00000000
FCP Cmpl: xb 00000000
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 NVME 2947 SCSI 2947 ELS 250
NVME LPF0 lpfc1 WWPN x10000090fae0ec61 WWNN x20000090fae0ec62 DID x012400 ONLINE
```

7. In this example, one namespace is mapped to the above SLES15 ANA host and visible through two target LIFs: one local node LIF and the other partner/remote node LIF. This in turn is reflected as one ANA active path and one ANA inactive path for the namespace device on the host. Reboot the SLES15 host and validate the same:

```
# nvme list
```
Appendix J: Configuring and Connecting Red Hat Enterprise Linux to ONTAP NVMe/FC Namespaces

Red Hat Enterprise Linux version 7.6 is the first version of Red Hat Enterprise Linux to support NVMe/FC using Broadcom adapters, which will be supported with ONTAP 9.5 NVMe/FC. This page captures the details for enabling NVMe/FC support using Red Hat Enterprise Linux 7.6 initiator.

Description

At the time of this writing, the versions listed here were the current supported versions. However, because NVMe/FC is under rapid development, versions can change, and the number of items qualified will continue to expand rapidly. For this reason, and because it is always a best practice, check the NetApp IMT and adjust versions and hardware specifications to those listed in the tool.

Features and Limitations

Supported Configs

- NVMe/FC is currently supported only on 16GB or 32GB FC fabrics using 32GB HBAs with ONTAP 9.4 or 9.5 and SLES12 SP3, SLES15, and Red Hat Enterprise Linux 7.6 as the initiator.
- Only Broadcom 32Gbps FC adapters (such as LPe3200X) can be used for NVMe/FC support (host and target).

Single LIF per SVM Limitation

- Red Hat Enterprise Linux 7.6 has a single LIF per NVMe/FC SVM limitation that will be lifted in future releases. No NVMe/FC Path Failover support.
- Red Hat Enterprise Linux 7.6 support is limited to a single path/namespace without multipath.

No NVMe/FC Linux Host Utilities Support

There is no SANLUN support for FC-NVMe. And hence no Linux Host Utilities (LUHU) support for NVMe/FC on Red Hat Enterprise Linux 7.6.

Protocol Coexistent Host Configs

Red Hat Enterprise Linux 7.6 initiator host can serve both NVMe/FC and FC-SCSI traffic through the same 32Gbps FC adapter ports. In fact, that is expected to be the commonly deployed host config for customers. So, for FC-SCSI, one may configure dm-multipath as usual for the SCSI LUNs resulting in mpath devices, along with NVMe/FC namespace devices (such as /dev/nvmeXnY) on the Red Hat Enterprise Linux 7.6 initiator host.

Configuration Requirements

Host Bus Adapter

Broadcom

- 32Gbps FC LPe32002 adapters used as NVMe/FC initiators on the Red Hat Enterprise Linux 7.6 NVMe/FC host
- LPe32002 firmware v12.0.193.13 (available on the Broadcom site here)

Fabric Switch

- 32Gbps FC Brocade or Cisco switch with supported FOS versions (refer to the Switch IMT matrix for switch models and switch OS versions)
• Single-initiator zoning enabled for the initiator and target ports

Initiator
• Red Hat Enterprise Linux 7.6 NVMe/FC host running:
  – kernel-default-4.4.126-94.22.2.15326.0.PTF.1091704 (available from SuSE)
  – nvme-cli-1.2-6.15.1 (available at SuSE’s Patch Finder)

Target
• ONTAP 9.4 or later
• Controller models:
  – NetApp AFF A800
  – NetApp AFF A700
  – NetApp AFF A700s
  – NetApp AFF A300
  
  Any of the NetApp controllers listed above requires one or more 32Gbps target HBAs. For specific options available for your controller, see the NetApp Hardware Universe.

Enabling NVMe/FC on Red Hat Enterprise Linux 7.6
To enable NVMe/FC on the Red Hat Enterprise Linux 7.6 initiator host, complete the following steps:

1. Install Red Hat Enterprise Linux 7.6 on the server.

2. After the installation is complete, verify that you are running with the specified Red Hat Enterprise Linux 7.6 kernel.

   ```
   # uname -r
   3.10.0-944.el7.x86_64
   ```

3. Verify that the nvme-cli package from Red Hat Enterprise Linux 7.6 ISO is installed.

   ```
   # rpm -qa|grep nvme-cli
   nvme-cli-1.6-1.el7.x86_64
   ```

4. After the nvme-cli utility package is installed, verify that the hostnqn file has been created.

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

5. If the /etc/nvme/hostnqn file was not created, run nvme gen-hostnqn and save the nqn string that was generated in the file /etc/nvme/hostnqn.

   ```
   # nvme gen-hostnqn
   ```

6. Make sure to save the nqn string generated in the file /etc/nvme/hostnqn. Verify the string.

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

Broadcom FC Adapter
1. Copy the nvmefc autoconnect scripts from the following example and install them.

   ```
   # rpm -Uvh nvmefc-connect-12.0.288.10-1.rhel7u6.x86_64.rpm
   ```

   Autoconnect scripts are included with the inbox drivers (drivers that are included with the OS release). Alternatively, if you download the off-box drivers (drivers that are not included with the OS release), an autoconnect script is included in the download and should be installed as part of the driver and firmware installation process.

2. Verify that you are using the specified adapter and recommended lpfc firmware and inbox driver versions and nvmefc autoconnect package on the host.
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2

# cat /sys/class/scsi_host/host*/modeldesc
LPe32002-M2 Emulex LightPulse LPe32002-M2-NA 2-Port 32Gb Fibre Channel Adapter
LPe32002-M2 Emulex LightPulse LPe32002-M2-NA 2-Port 32Gb Fibre Channel Adapter

# cat /sys/class/scsi_host/host*/fwrev
12.0.193.13, sli-4:2:c
12.0.193.13, sli-4:2:c

# cat /sys/module/lpfc/version
0:12.0.0.5

# rpm -qa | grep nvmefc
nvmefc-connect-12.0.288.10-1.rhel7u6.x86_64

To enable Red Hat Enterprise Linux 7.6 as an NVMe/FC initiator, set `lpfc_enable_fc4_type` to 3 to ensure that the lpfc driver is enabled to support both NVMe/FC and FC-SCSI. You can do this by specifying these settings in the `modprobe.d`, and then reloading the lpfc driver.

```bash
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_enable_fc4_type=3
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. After adding the above entry to `lpfc.conf`, rebuild the `initramfs` by using `dracut --force` for NVMe to be enabled during bootup. Reboot the host.

5. Verify that the initiator ports are up and running and that the recommended NVMe/FC modules are loaded.

```bash
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec9d
0x10000090fae0ec9e
# cat /sys/class/fc_host/host*/port_state
Online
Online
# lsmod | grep nvme
nvmet_fc 27790 1 lpfc
nvme 51048 1 nvmet_fc
nvme_fabrics 19997 1 nvme_fc
nvme_core 58852 2 nvme_fabrics,nvme_fc
```

6. Verify that the NVMe/FC initiator ports are enabled and able to see the target LIFs and that they are up and running.

```bash
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
NVME LPORT lpfc0 WWPN x10000090fae0ec9d WWNN x200000090fae0ec9d DID x012000 ONLINE
NVME RPORT WWPN x200b00a098c80f09 WWNN x200a00a098c80f09 DID x010601 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 00000000000000071 Wr 0000000000000005 IO 0000000000000031
Cmpl 0000000000000000a6 Outstanding 000000000000000001
NVME Initiator Enabled
NVME LPORT lpfc1 WWPN x10000090fae0ec9e WWNN x200000090fae0ec9e DID x012400 ONLINE
NVME RPORT WWPN x200900a098c80f09 WWNN x200800a098c80f09 DID x010301 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 00000000000000073 Wr 0000000000000005 IO 0000000000000031
Cmpl 0000000000000000a8 Outstanding 000000000000000001
```

7. Check the `hostnqn` string at `/etc/nvme/hostnqn` and make sure that it properly matches with the `hostnqn` string for the corresponding subsystem on the NetApp array.
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmeexpress:uuid:a9d5d070-2c77-400e-96c4-b6d4aa220c71

8. On the ONTAP controller, display the NQN for the SVM.

```shell
*> vserver nvme subsystem host show -vserver vs_nvme_1
Vserver Subsystem Host NQN
----------- -------------------------------------
vs_nvme_1
rhel_133_nvme_ss_1
nqn.2014-08.org.nvmeexpress:uuid:a9d5d070-2c77-400e-96c4-b6d4aa220c71
```

9. Discover and connect to these target ports by using the nvme-cli utility, which results in the NVMe/FC namespace devices getting created. The --traddr refers to the target port WWNNs/WWPNs, whereas the --host-traddr refers to the initiator port WWNNs/WWPNs. On the target, we created two SVMs with two subsystems, with one namespace each. You should see two NVMe/FC namespace devices on the initiator host, with one underlying path each.

```shell
# nvme connect -v all --transport=f --traddr=nn-0x200a00a098c80f09:pn-0x200b00a098c80f09 --host-traddr=nn-0x20000090fae0ec9d:pn-0x10000090fae0ec9d

# nvme connect -v all --transport=f --traddr=nn-0x200800a098c80f09:pn-0x200900a098c80f09 --host-traddr=nn-0x20000090fae0ec9e:pn-0x10000090fae0ec9e
```

```shell
# nvme list
Node SN Model Namespace Usage Format FW Rev
-------- ----------------- ------- --------------- ---------- -------
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 128.85 GB / 128.85 GB 4 KiB + 0 B 
/dev/nvme1n1 80BADBKnB/JsAAAAAAAC NetApp ONTAP Controller 2 118.11 GB / 118.11 GB 4 KiB + 0 B FFFFFFFF
# cat /sys/block/nvme0n1/uuid
71bcfd72-dbdb-4bdf-8ad3-bf5866e1e28c

# cat /sys/block/nvme1n1/uuid
d856ab9e-6489-475b-9e46-850f7447cb6e
```

10. Verify the above UUID strings match with the respective namespace UUIDs on the NetApp array.

```shell
*> nvme namespace show -vserver vs_nvme_1 -path * -fields uuid
/vserver nvme namespace show)
vserver path uuid
----------- -----------
vs_nvme_1 /vol/rhel_133_vol_1/rhel_133_ns_1 71bcfd72-dbdb-4bdf-8ad3-bf5866e1e28c
vs_nvme_1 /vol/rhel_133_vol_1/rhel_133_ns_2 d856ab9e-6489-475b-9e46-850f7447cb6e
```

```shell
# lsblk
NAME MAJ:MIN  RM SIZE RO TYPE MOUNTPOINT
sda 8:0 278.9G 0 disk
├─sda1 8:1 0 1G 0 part /boot
└─sda2 8:2 0 277.9G 0 part
   └─rhel_chat--16--133-root 253:0 0 100G 0 lvm /
   └─rhel_chat--16--133-swap 253:1 0 27.9G 0 lvm [SWAP]
   └─rhel_chat--16--133-home 253:22 0 150G 0 lvm /home
```
11. Since the nvmeofc-connect package was installed, which takes care of configuring the devices during bootup and auto connecting during link bounce events (LIF/port toggles), reboot the host and verify that the NVMe devices are discovered during bootup.

Appendix K: Discovering and Connecting to an NVMe Namespace from a SuSE Linux Host Server

In order to use NVMe/FC, your host server must find and connect to one or more NVMe namespaces. Currently, the Broadcom (Emulex) supplied firmware and drivers in the SuSE host uses several scripts and commands to list, discover and connect to NVMe namespaces, those include:

- **Lsblk.** Lists all block devices seen by the host server.
- **Lpfc_nvme_list.sh (/usr/sbin/lpfc_nvme_list.sh).** The list command is used to show the initiators and target WWNNs and WWPNs that can be seen on the host.
- **Nvme discovery.** Run the `nvme discovery` command to discover and list any NVMe devices that the host can see.
- **Nvme connect.** Run the `connect` command to connect to one or more of the NVMe subsystems that the host server can see.

**List, Discover, and Connect to an NVMe Namespace from SuSE Linux Host (Summary)**

1. List all block devices the host server can see.
2. List all of the NVMe WWNs the host server can see.
3. Create the discover and connect commands with the correct transport and NQNs. Then issue the commands.
4. List all block devices the host server can see.

**List, Discover, and Connect to an NVMe Namespace from SuSE Linux Host (Procedure)**

1. List all block devices the host server can see.

   ```
   # lsblk
   NAME   MAJ:MIN  RM  SIZE   RO  TYPE MOUNTPOINT
   sda     8:0      0 279.4G  0  disk
   |-----sda1 8:1      0 156M   0  part /boot/efi
   |-----sda2 8:2      0   2G   0  part [SWAP]
   |-----sda3 8:3      0   40G   0  part /var/lib/mariadb
   |-----sda4 8:4      0 237.2G  0  part /home
   sr0    11:0      1  3.6G   0  rom
   ```

   List all of the NVMe WWNs the host server can see.

   ```
   # ./lpfc_nvme_list.sh
   Available HBAs for NVME Initiator
   ipfc0 WWPN x1000000109b1c0f8f WWNN x2000000109b1c0f8f DID x010200 ONLINE <<Init WWNs
   WWPN x204000a096b3f7a7 WWNN x203000a096b3f7a7 DID x010501 TARGET DISCSRVC ONLINE <<Target WWPN x204000a096b3f7a7 WWNN x203000a096b3f7a7 DID x010504 TARGET DISCSRVC ONLINE LIF WWNs seen from ipfc0 (2x LIFS found), in this example we use the 2nd LIF's WWNs.
   ipfc1 WWPN x1000000109b1c0f90 WWNN x2000000109b1c0f90 DID x000000 UNKNOWN
   ```

2. Create the discover and connect commands with the correct transport and NQNs.
3. Use the outputs from the commands above to construct the `nvme discover` command.
The following examples are color coded to ensure that the correct variables are entered as the correct arguments. First is the command format and then a full example of the command and command output.

```
nvme discover --transport=fc -traddr-nn=0x200000a098b3f7a7 --host-traddr-nn=0x204000a098b3f7a7
```

The following example discover command uses the WWNs listed above. Use the NQNs from its output to run nvme connect.

```
# nvme discover --transport=fc -traddr-nn=0x200000a098b3f7a7 --host-traddr-nn=0x100000109b1c0f8f

Discovery Log Number of Records 1, Generation counter 17
Discovery Log Entry D------
trtype: fibre-channel
adrfam: fibre-channel
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: none
subnqn: nqn.1992-08.com.netapp:en.5b584a86fa6411e794f200a098b3f653:subsystem.NVMe_demo_subsystem
traddr: nn=0x203f00a098b3f7a7
```

4. After you run the discover command, you will have the Subsystem NQN that you need to create the nvme connect command. As in the previous example, first the command structure and then an example of the command. There is no output.

```
nvme connect --transport=fc -traddr-nn=0x200000a098b3f7a7 --host-traddr-nn=0x204000a098b3f7a7
```

5. The following nvme connect command uses the outputs from list and discover commands shown in step 4.

```
nvme connect --transport=fc -traddr-nn=0x203f00a098b3f7a7 --host-traddr-nn=0x100000109b1c0f8f
```

6. List all of the block devices that the host server can see.

```
# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda  8:0  0  279.4G  0 disk
└─sda1  8:1  0 156M  0 part /boot/efi
└─sda2  8:2  0  2G  0 part [SWAP]
└─sda3  8:3  0  40G  0 part /var/lib/mariadb
└─sda4  8:4  0 237.2G  0 part /home
sr0  11:0  1  3.6G  0 rom
nvme0n1 259:0  0  200G  0 disk <<< New 200GB NVMe namespace
```

**Command Options for NVMe Discover and Connect Commands**

```
# nvme connect -t
Usage: nvme connect <device> [OPTIONS]

Connect to NVMe-oF subsystem

Options:
[  --transport=|LIST>, -t |LIST> ]  --- transport type
[  --nqn=|LIST>, -n |LIST> ]  --- nqn name
[  --traddr=|LIST>, -a |LIST> ]  --- transport address
[  --trsvcid=|LIST>, -s |LIST> ]  --- transport service id (e.g. IP port)
[  --host-traddr=|LIST>, -w |LIST> ]  --- host traddr (e.g. FC WNN's)
[  --hostnqn=|LIST>, -q |LIST> ]  --- user-defined hostnqn
[  --hostid=|LIST>, -I |LIST> ]  --- user-defined hostid (if default not used)
[  --nr-io-queues=|LIST>, -i |LIST> ]  --- number of io queues to use (default is core count)
[  --queue-size=|LIST>, -Q |LIST> ]  --- number of io queue elements to
use (default 128)

[ --keep-alive-tmo=<LIST>, -k <LIST> ] --- keep alive timeout period in seconds
[ --reconnect-delay=<LIST>, -c <LIST> ] --- reconnect timeout period in seconds

# nvme discover -t
Usage: nvme discover <device> [OPTIONS]

Send Get Log Page request to Discovery Controller.

Options:
[ --transport=<LIST>, -t <LIST> ] --- transport type
[ --traddr=<LIST>, -a <LIST> ] --- transport address
[ --trsvcid=<LIST>, -s <LIST> ] --- transport service id (e.g. IP port)
[ --host-traddr=<LIST>, -w <LIST> ] --- host addr (e.g. FC WWN's)
[ --hostnqn=<LIST>, -q <LIST> ] --- user-defined hostnqn (if default not used)
[ --hostid=<LIST>, -I <LIST> ] --- user-defined hostid (if default not used)
[ --queue-size=<LIST>, -Q <LIST> ] --- number of io queue elements to use (default 128)
[ --nr-io-queues=<LIST>, -I <LIST> ] --- number of io queues to use (default is core count)
[ --raw=<LIST>, -r <LIST> ] --- raw output file

No discover params given and no /etc/nvme/discovery.conf

NVMe/FC Scalability and Limits

At the time of this writing, the scalability and limits for NVMe/FC were described in the SAN Configuration Guide. The current limits aren’t particularly high because they were chosen to speed NVMe/FC development. The limits should be sufficient to allow for robust testing and qualification and will be increased in the future releases of ONTAP.

Troubleshooting

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

lpfc Verbose Logging for NVMe/FC

The following lpfc driver logging bitmasks are available for NVMe/FC, as seen at drivers/scsi/lpfc/lpfc_logmsg.h:

```c
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS 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 initramfs 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.

```bash
# 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
- lpcf driver events related to FC-LS discovery issues during link bounces (such as LIF/Port toggle events): 0xf00083

**Common nvme-cli Errors and Workarounds**

- **Symptom:** The nvme-cli utility displays "Failed to Write to /dev/nvme-fabrics: Invalid Argument Error" during NVMe discover, NVMe connect, or NVMe connect-all.
  - **Probable cause:** This error message is generally displayed if the syntax is wrong.
  - **Workaround:** Make sure that you are using the correct syntax for the above NVMe commands.

- **Symptom:** The nvme-cli utility displays "Failed to Write to /dev/nvme-fabrics: No Such File or Directory During NVMe discover, NVMe connect or NVMe connect-all.
  - **Probable cause:** Multiple issues could trigger this error. For example, you passed erroneous arguments to the above NVMe commands.
  - **Workaround:** Make sure that you passed the proper arguments (such as the proper WWNN string, WWPN string, and so on) for the above commands.

If the arguments are correct, but you still see this error, check to see the /sys/class/scsi_host/host*/nvme_info output is correct with the NVMe initiator showing as Enabled and NVMe/FC target LIFs properly showing under the remote ports sections.

For example:

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
NVME LPOR1 lpfc0 WWPN x10000090fae0ec9d WWNN x20000090fae0ec9d DID x012000 ONLINE
NVME RPOR1 WWPN x200800a098c80f09 WWNN x200a00a098c80f09 DID x010601 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 0000000000000071 Wr 0000000000000005 IO 0000000000000031
Cmpl 000000000000000a Outstanding 0000000000000001
NVME Initiator Enabled
NVME LPOR1 lpfc1 WWPN x10000090fae0ec9e WWNN x20000090fae0ec9e DID x012400 ONLINE
NVME RPOR1 WWPN x200900a098c80f09 WWNN x200800a098c80f09 DID x010301 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl 0000000000000006
FCP: Rd 0000000000000073 Wr 0000000000000005 IO 0000000000000031
Cmpl 000000000000000a Outstanding 0000000000000001
```

- **Workaround:** If the target LIFs don't show up as above in the nvme_info output, verify the /var/log/messages and dmesg output for any suspicious NVMe/FC failures, and report/fix accordingly.

- **Symptom:** The nvme-cli utility displays "No Discovery Log Entries to Fetch" during NVMe discover, NVMe connect, or NVMe connect-all.
  - **Probable cause:** This error message is generally seen if the /etc/nvme/hostnqn string has not been added to the corresponding subsystem on the NetApp array. Or an incorrect hostnqn string has been added to the respective subsystem.
  - **Workaround:** Make sure that the exact /etc/nvme/hostnqn string is added to the corresponding subsystem on the NetApp array (verify through the vserver nvme subsystem host show).

- **Symptom:** The nvme-cli utility displays "Failed to Write to /dev/nvme-fabrics: Operation Already in Progress" during NVMe discover, NVMe connect, or NVMe connect-all.
  - **Probable cause:** This error message is displayed if the controller associations or specified operation is already created or in the process of being created. This error could happen as part of the auto-connect scripts previously installed.
  - **Workaround:** There is not workaround. For NVMe discover, run this command after some time. For NVMe connect and connect-all, run an NVMe list to verify that the namespace devices are already created and displayed on the host.
Where to Find Additional Information

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

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)
- RFC 7143 Internet Small Computer System Interface (iSCSI) Protocol (Consolidated)
- RFC 5041 Direct Data Placement over Reliable Transports
- INCITS T11 - T11-2017-00145-v004 FC-NVMe specification

SuSE Enterprise Linux Links

- SuSE downloads
  https://www.suse.com/download-linux/
- Use NVMe over Fabrics with SuSE Linux Enterprise 12 to Speed Up Remote Storage
  https://www.suse.com/c/use-nvme-fabrics-suse-linux-enterprise-12-speed-remote-storage/
- SuSE Linux documentation
  https://www.suse.com/documentation/

Brocade Links

- SAN Health
  https://community.brocade.com/t5/SAN-Health-Utility/tkb-p/SANHealth
- Webinar: Tech Refresh: Get Ready for NVMe over Fibre Channel
- NVMe over Fibre Channel
  https://docs.broadcom.com/docs/12379957
- Why Fibre Channel is the NVMe Fabric of Choice
- Networking the Next Generation of Enterprise Storage: NVMe Fabric
  https://docs.broadcom.com/docs/12380132
- Top 5 Reasons Why NVMe Over FC
  https://docs.broadcom.com/docs/12380041?eula=true
- Brocade Flow Vision Configuration Guide, 8.2.0
- NVMe over Fibre Channel for Dummies
  http://media.wiley.com/assets/7359/40/9781119399711.pdf
- Brocade Fabric OS Administration Guide, 8.2.0a
  https://docs.broadcom.com/docs/53-1005237-05?eula=true
• SAN Design and Best Practices
  https://docs.broadcom.com/docs/12379731
• SAN Fabric Administration Best Practices Guide
  https://docs.broadcom.com/docs/12379730
• Brocade SAN Architectural Brief
  https://docs.broadcom.com/docs/12379732
• SAN Fabric Resiliency Best Practices
  https://docs.broadcom.com/docs/12379740

Videos, Webcasts, and Blogs
• FCIA: Introducing Fibre Channel NVMe (Bright TALK Webcast)
• FCIA: Dive Deep into NVMe over Fibre Channel (FC-NVMe) (Bright TALK Webcast)
• Storage Class Memory: What’s Next in Enterprise Storage (NetApp Blog)
  https://blog.netapp.com/storage-class-memory-what-is-next-in-enterprise-storage/
• Support Download SAN Health Diagnostics Capture
  https://www.broadcom.com/support/fibre-channel-networking/tools/san-health/diagnostics-capture
• SAN Health
  https://www.broadcom.com/support/fibre-channel-networking/tools/san-health

White Papers, Product Announcements, and Analysis
• RoCE Versus iWARP Competitive Analysis
  http://www.mellanox.com/related-docs/whitepapers/WP_RoCE_vs_iWARP.pdf
• 3D Point Technology
  https://www.micron.com/products/advanced-solutions/3d-xpoint-technology
• Ultra-Low Latency with Samsung Z-NAND SSD

NetApp Documentation, Technical Reports and other NVMe-related Collateral
• Licensing Information for NVMe Protocol on ONTAP
  https://kb.netapp.com/app/answers/answer_view/a_id/1076722
• Configuring and Connecting SuSE Enterprise Linux to ONTAP NVMe/FC Namespaces
  https://kb.netapp.com/app/answers/answer_view/a_id/1076721
• NVMe Modern SAN Primer
• Demartek Evaluation: Performance Benefits of NVMe over Fibre Channel – A New, Parallel, Efficient Protocol
• TR-4080: Best Practices for Scalable SAN
• NetApp NVMe: Leading the Future of Flash
• NetApp’s Visions for NVMe over Fabric and Storage Class Memory  
• Tech ONTAP Podcast Episode 72: Demystifying NVMe  
• Popup TechTalks: NVMe and the Future of Flash  
• Datasheet: NetApp EF570 All-Flash Array  
• NetApp documentation  
  https://mysupport.netapp.com/documentation/productsatoz/index.html#O
• NetApp software downloads  
  https://mysupport.netapp.com/NOW/cgi-bin/software/

**Version History**

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Document Version History</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.0</td>
<td>June 2018</td>
<td>Initial release.</td>
</tr>
<tr>
<td>Version 2.0</td>
<td>November 2018</td>
<td>Updated for ONTAP 9.5, ANA, added host install and configuration appendixes.</td>
</tr>
<tr>
<td>Version 3.0</td>
<td>April 2019</td>
<td>Minor updates and errata.</td>
</tr>
<tr>
<td>Version 4.0</td>
<td>June 2019</td>
<td>Updated for ONTAP 9.6, ANA, round-robin, and multipathing recommendations.</td>
</tr>
</tbody>
</table>
Contact Us

Let us know how we can improve this technical report. Contact us at docfeedback@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

Copyright © 2018 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

TR-4684-0619