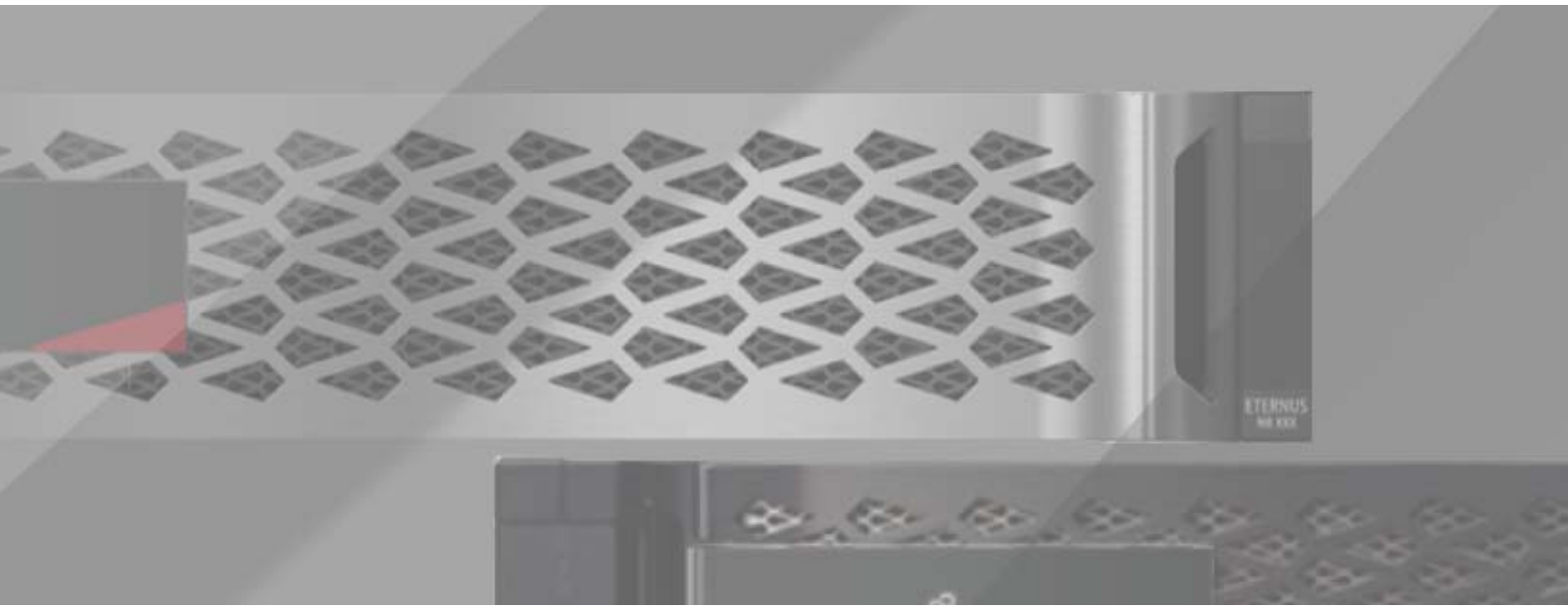
P3AG-6392-01ENZ0

FUJITSU Storage
ETERNUS AB series All-Flash Arrays,
ETERNUS HB series Hybrid Arrays

NVMe over Fabrics Support



Nonvolatile Memory Express over InfiniBand, RoCE, and Fibre Channel on ETERNUS AB/HB series Systems

FUJITSU

Table of Contents

List of Figures

List of Tables

Preface

The ETERNUS AB/HB series supports the new Nonvolatile Memory over Fabrics (NVMe-oF) protocol using either InfiniBand (IB), RDMA over Converged Ethernet (RoCE), or Fibre Channel (FC) connections. This document provides technical details for the implementation, benefits, and limitations of this system. It also compares SCSI and NVMe-oF structures.

First Edition
December 2021

Trademarks

Third-party trademark information related to this product is available at:
https://www.fujitsu.com/global/products/computing/storage/eternus/trademarks.html

Trademark symbols such as ™ and ® are omitted in this document.

About This Manual

Intended Audience

This manual is intended for system administrators who configure and manage operations of the ETERNUS AB/HB, or field engineers who perform maintenance. Refer to this manual as required.

Related Information and Documents

The latest information for the ETERNUS AB/HB is available at:
https://www.fujitsu.com/global/support/products/computing/storage/manuals-list.html

Document Conventions

Notice Symbols

The following notice symbols are used in this manual:

Caution	Indicates information that you need to observe when using the ETERNUS AB/HB. Make sure to read the information.
Note	Indicates information and suggestions that supplement the descriptions included in this manual.

1. Introduction

Nonvolatile Memory Express (NVMe) has become the industry standard interface for PCIe solid-state drives (SSDs). With a streamlined protocol and command set and fewer clock cycles per I/O, NVMe supports up to 64K queues and up to 64K commands per queue. These attributes make it more efficient than SCSI-based protocols like FC, SAS, and SATA.

The introduction of NVMe over Fabrics (NVMe-oF) makes NVMe more scalable without affecting the low latency and small overhead that are characteristic of the interface.

Some ETERNUS AB/HB series models support NVMe-oF, and are available with NVMe/IB, NVMe/RoCE, and NVMe/FC as the transport protocols. NVMe-oF is supported from the host to the front end of the AB5100 all-flash array or the HB5000 hybrid array, while the back end is still SCSI-based with SAS drives, as shown in Figure 1.

With the introduction of the AB3100/AB6100 all-flash array, ETERNUS AB/HB series now offers end-to-end support for NVMe from the front end all the way to the drives, as shown in Figure 2.

Figure 1 NVMe-oF front end on the AB5100/HB5000 systems

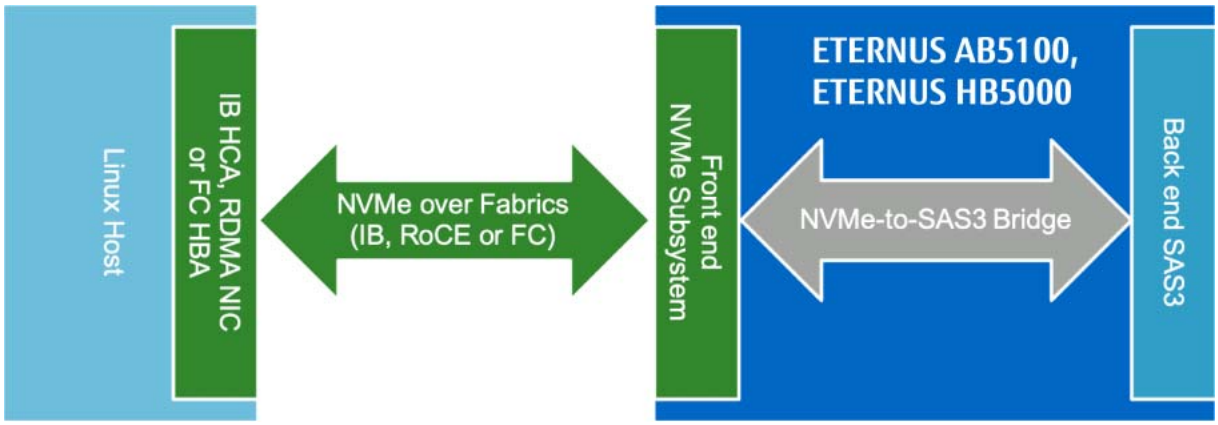
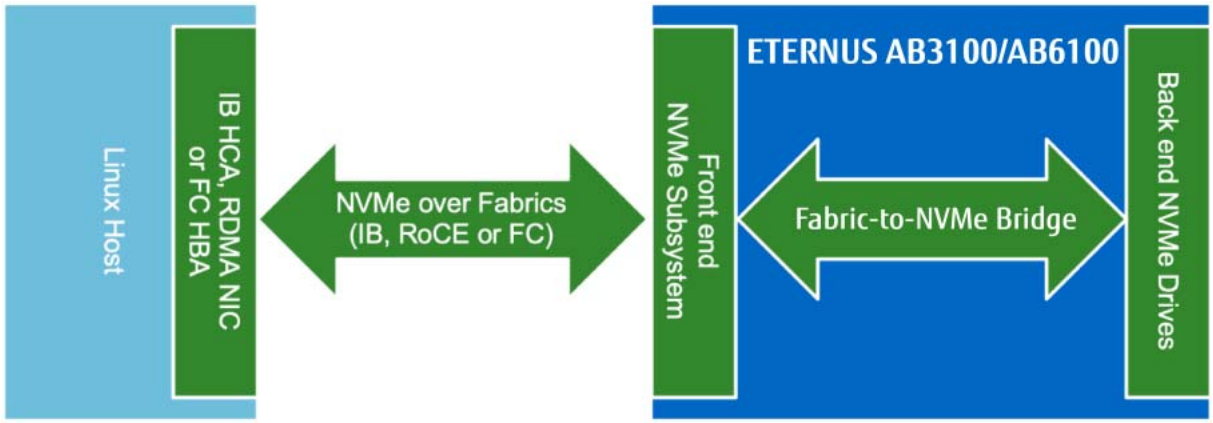


Figure 2 NVMe end-to-end on the AB3100/AB6100 system



2. Transport Layers: InfiniBand, RoCE, and FC

The NVMexpress.org specifications outline support for NVMe-oF over remote direct memory access (RDMA) and FC. The RDMA-based protocols can be either IB or RDMA over Converged Ethernet version 2 (RoCE v2). Throughout this document, whenever NVMe/RoCE is mentioned, version 2 is implied whether it's explicitly noted or not.

The ETERNUS AB/HB series implementation supports NVMe/IB, NVMe/RoCE v2, and NVMe/FC with benefits including but not limited to the following capabilities:

- The ETERNUS AB/HB series storage already supports FC as a transport layer for SCSI protocol commands. NVMe/FC adds a new protocol over such well-established transport layer.
- The same hardware on the ETERNUS AB/HB series that runs FC can run NVMe/FC (although not at the same time).
- Both protocols (FC and NVMe/FC) can coexist on the same fabric and even on the same FC host bus adapter (HBA) port on the host side. This capability allows customers with existing fabrics running FC to connect the ETERNUS AB/HB series running NVMe/FC to the same fabric.
- All FC components in the fabric (ETERNUS AB/HB series, switches, and HBAs) can negotiate the speed down as needed: 32Gbps; 16Gbps; and 8Gbps. A lower speed makes it easier to connect to legacy components.
- IB and RoCE have RDMA built into them.
- ETERNUS AB/HB series storage supports other protocols over RDMA (SCSI-based), such as iSCSI Extensions for RDMA (iSER) and SCSI RDMA Protocol (SRP).
- The same host interface card on the AB5100 and HB5000 can run iSER, SRP, NVMe/IB or NVMe/RoCE, although not at the same time.
- The same host interface card on the AB3100/AB6100 can run NVMe/IB or NVMe/RoCE, although not at the same time.
- All three protocols (iSER, SRP, and NVMe/IB) can coexist on the same fabric and even on the same InfiniBand host channel adapter (HCA) port on the host side. This capability allows customers with existing fabrics running iSER and/or SRP to connect the ETERNUS AB/HB series running NVMe/IB to the same fabric.
- Both iSCSI and NVMe/RoCE can coexist on the same fabric on the host side.
- The ETERNUS AB/HB series supports 100Gbps, 50Gbps, 40Gbps, 25Gbps, and 10Gbps speeds for NVMe/RoCE.
- The ETERNUS AB/HB series supports NVMe/RoCE v2 (which is routable), and they are also backward compatible with RoCE v1.
- All IB components in the fabric (ETERNUS AB/HB series, switches, and HCAs) can negotiate the speed down as needed: Enhanced Data Rate (EDR) 100Gbps; Fourteen Data Rate (FDR) 56Gbps; or Quad Data Rate (QDR) 40Gbps. A lower speed makes it easier to connect to legacy components.

3. NVMe and NVMe-oF: Protocols and Concepts

Command Set and Transport Protocol

This document uses the following definitions:

- **Command set**
  The set of commands used for moving data to and from the storage array and managing storage on that array.
- **Transport protocol**
  The underlying physical connection and the protocol used to carry the commands and data to and from the storage array.

Since its inception, ETERNUS AB/HB series storage software was designed to provide SCSI command set storage functionality. Over the years, this software has been extended, architecturally and functionally, to support additional SCSI transport protocols. However, in all cases, it has remained fundamentally a SCSI product. Therefore, it is unnecessary to distinguish between the concept of the supported command set and the concept of the underlying transport protocol. That is, because ETERNUS AB/HB series has traditionally been a SCSI product, the fact that it supports a FC host/fabric connect implies that it supports the FCP SCSI protocol carried by the FC transport protocol.

Table 1 lists the transport protocol and command set combinations supported by ETERNUS AB/HB series storage products. Historically, despite the variations in underlying transport mechanisms, SCSI has been the command set in use in every case.

Table 1 Historical ETERNUS AB/HB series transport protocol and command set combinations

Physical Connection	Transport Protocol	Command Set
Parallel SCSI bus	SCSI	SCSI
FC	FC SCSI Protocol (FCP)	SCSI
Serial-attached SCSI	Serial SCSI Protocol (SSP)	SCSI
Ethernet	Internet SCSI Protocol (iSCSI)	SCSI
IB	SCSI RDMA Protocol (SRP)	SCSI
IB	iSCSI extensions for RDMA (iSER)	SCSI
InfiniBand	NVMe-oF (NVMe/IB)	NVMe
RDMA over Converged Ethernet	NVMe-oF (NVMe/RoCE)	NVMe
FC	NVMe-oF (NVMe/FC)	NVMe

The ETERNUS AB/HB series storage software SANtricity OS supports a command set other than SCSI: NVMe-oF. SANtricity OS supports all three NVMe transport protocols: IB NVMe, RoCE v2 NVMe, and FC NVMe.

NVMe Concepts

NVMe

NVMe is a specification-defined, register-level interface for applications (through OS-supplied file systems and drivers) to communicate with nonvolatile memory data storage through a PCI Express (PCIe) connection. This interface is used when the storage devices reside in the same physical enclosure, and the host OS and application can be directly connected through PCIe, such as within servers or laptop computers.

NVMe-oF

NVMe-oF is a specification-defined extension to NVMe that enables NVMe-based communication over interconnects other than PCIe. This interface makes it possible to connect "external" storage enclosures to a server, either directly or through a switch, while still using NVMe as the fundamental communication mechanism.

The NVMe-oF protocol can be carried over multiple different physical connections, including transport protocols such as RoCE, IB, and FC. Regardless of the interconnect, NVMe-oF is transport agnostic. Server and storage communicate through NVMe-oF, independent of the underlying transport that is used to carry the NVMe-oF protocol. For the SANtricity OS software of the ETERNUS AB/HB series, the InfiniBand, RoCE v2, and FC transport protocols are all supported.

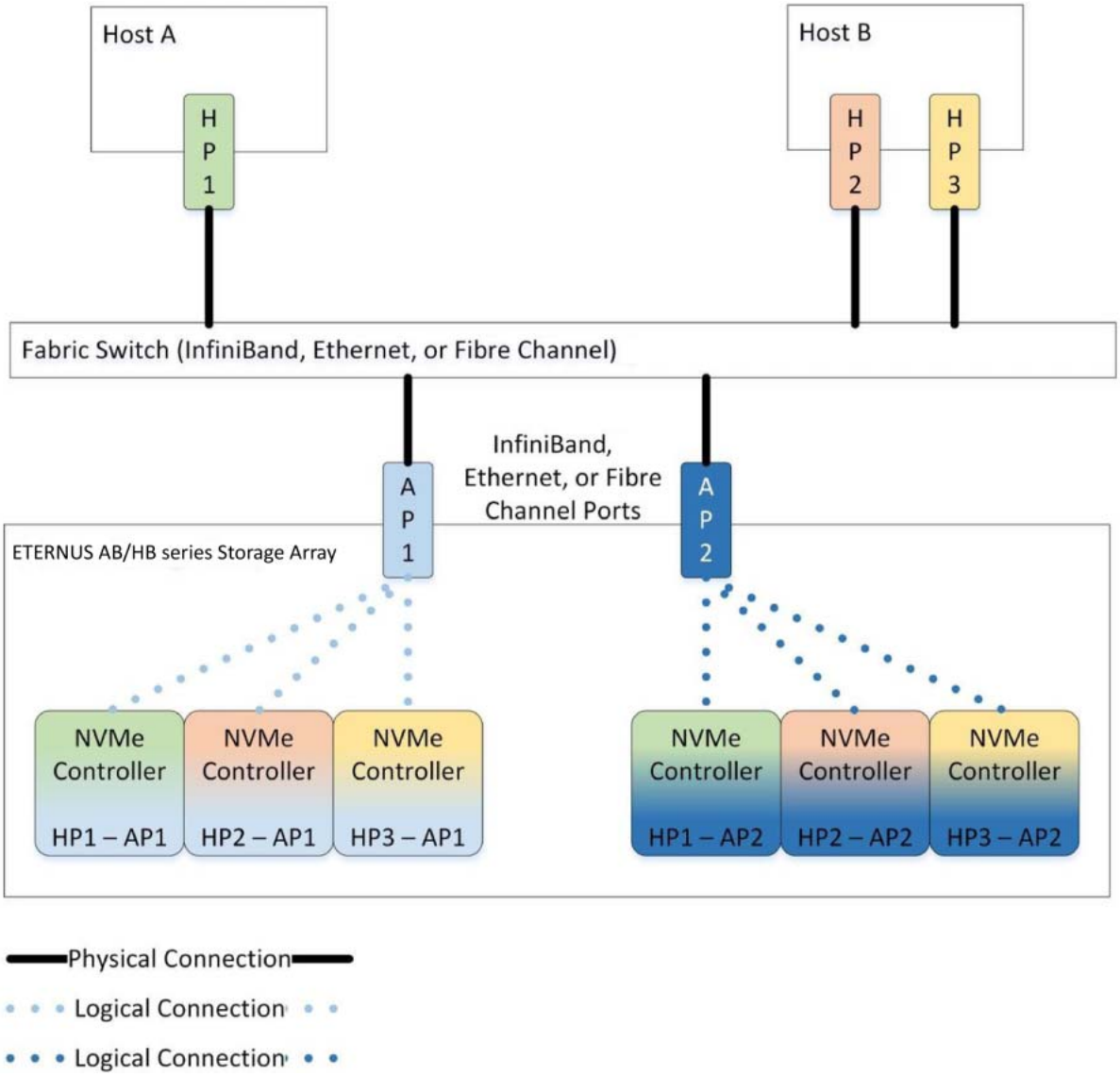
Controller

The term controller has a very specific meaning in the context of NVMe. An NVMe controller is a concrete entity defined by the specification as a PCI Express function that implements NVM Express. Essentially, a controller is the device that presents a set of hardware registers that can be accessed with PCIe for the purpose of storing nonvolatile data.

The NVMe-oF standard extends the definition of a controller, however, specifying that a controller is associated with one host at a time. Therefore, in the context of NVMe-oF, a controller becomes an abstract entity that represents the relationship between a single host port and a single storage array port. In SCSI terms, this is analogous to the concept of an initiator port-target port nexus. When an NVMe-oF connection is established between an initiator port and a target port, an NVMe controller is created to represent that connection.

Figure 3 illustrates the logical NVMe controller constructs for a typical NVMe-oF topology. Host A presents one port (HP1) connected to a switch, and Host B presents two ports (HP2 and HP3), also connected to the same switch. The ETERNUS AB/HB series controller presents two ports, AP1 and AP2, both connected to the same switch as the host ports. NVMe-oF connections are established from each host port to each array port. Therefore, ETERNUS AB/HB series software creates a logical NVMe controller for each connection. Figure 3 illustrates six NVMe controllers for the sample configuration, with the host-to-array port association described as HPx-APx.

Figure 3 NVMe-oF host/array topology with logical NVMe controllers



Namespace

An NVMe namespace is defined as a storage area that is logically divided. This definition is virtually identical to the SCSI concept of a logical unit. Therefore, volumes created on an ETERNUS AB/HB series storage array are presented to NVMe-oF interfaces as namespaces.

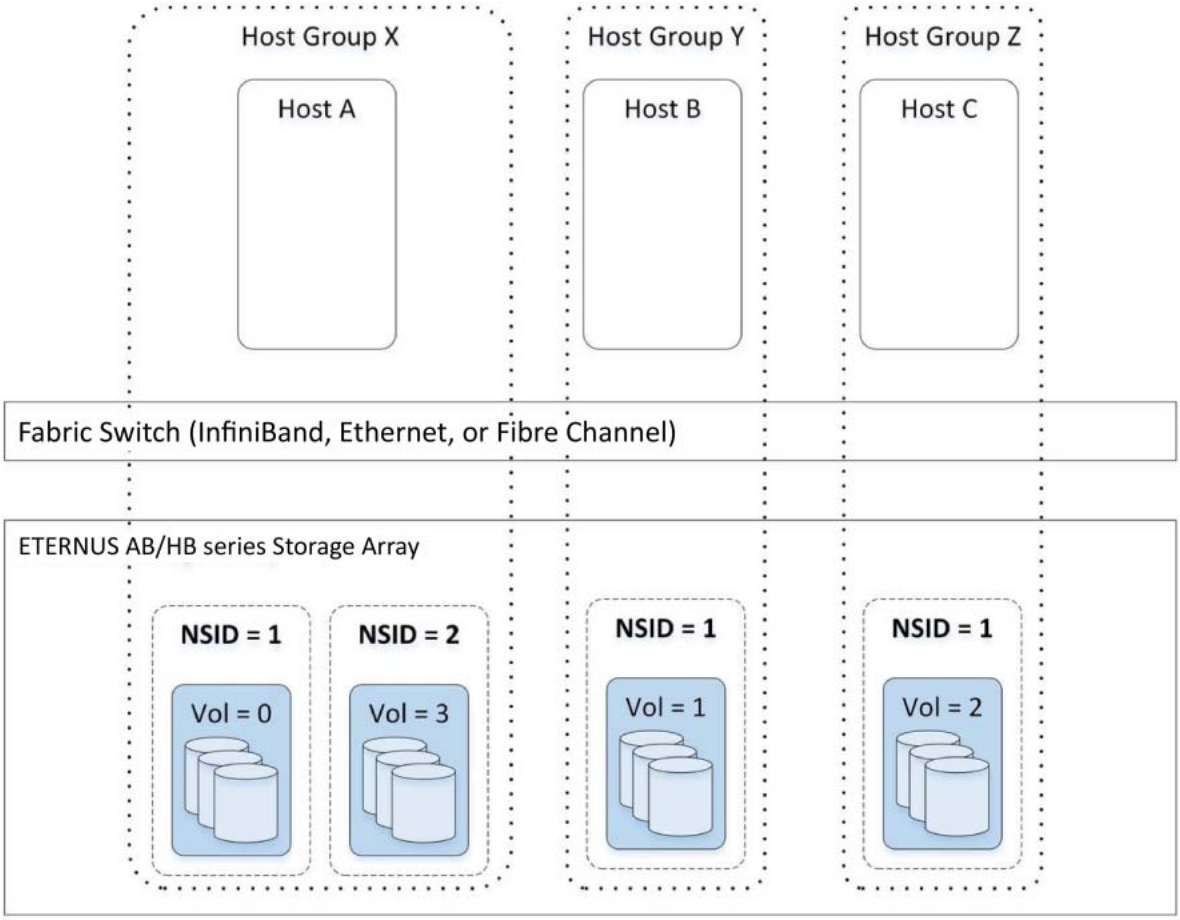
Namespace ID

A namespace ID (NSID) is an identifier used by an ETERNUS AB/HB series controller to provide access to a namespace. This is nearly equivalent to a logical unit number (LUN) in SCSI. There are two exceptions. First, an NSID cannot have a value of 0. Second, an NSID of FFFFFFFFh is a broadcast value that is used to specify all namespaces.

The accessibility of a volume by a host is configured from the management interfaces, along with setting the namespace ID for that host or host group. As with SCSI, a logical volume can be mapped to only a single host group at a time, and a given host group cannot have any duplicate NSIDs.

Figure 4 shows the logical partitioning of storage to multiple hosts. Logical Volumes 0 and 3 are assigned to Host Group X with an NSID of 1 and 2, respectively (considering that 0 is not a valid NSID). Logical Volume 1 is assigned to Host Group Y as NSID 1, and Logical Volume 2 is assigned to Host Group Z, also as NSID 1.

Figure 4 Namespace ID mapping to host groups



Queue Pairs

Like some other RDMA-based protocols, NVMe and NVMe-oF communication between devices relies on the concept of a queue pair (QP). A QP is the combination of a submission queue (SQ) and a completion queue (CQ). The host (initiator) places commands into an SQ that is read by the storage array (target).

The target places completion information relating to a received command into the CQ associated with the SQ on which the command was received. For NVMe-oF, there must be a 1:1 relationship between submission queues and completion queues; that is, every SQ must have a single, unique CQ associated with only that SQ.

When an NVMe controller is created, a minimum of two QPs are created. The first QP (using queue ID 0) is referred to as the admin queue; all remaining queues are referred to as I/O queues. The admin queue is used to process the admin command set (see "Admin Commands" (page 14), while the I/O queues are used to process the NVM command set (see "NVM Commands" (page 15).

The number of I/O queues created by the controller is negotiated by the initiator and target:

Procedure ▶▶▶

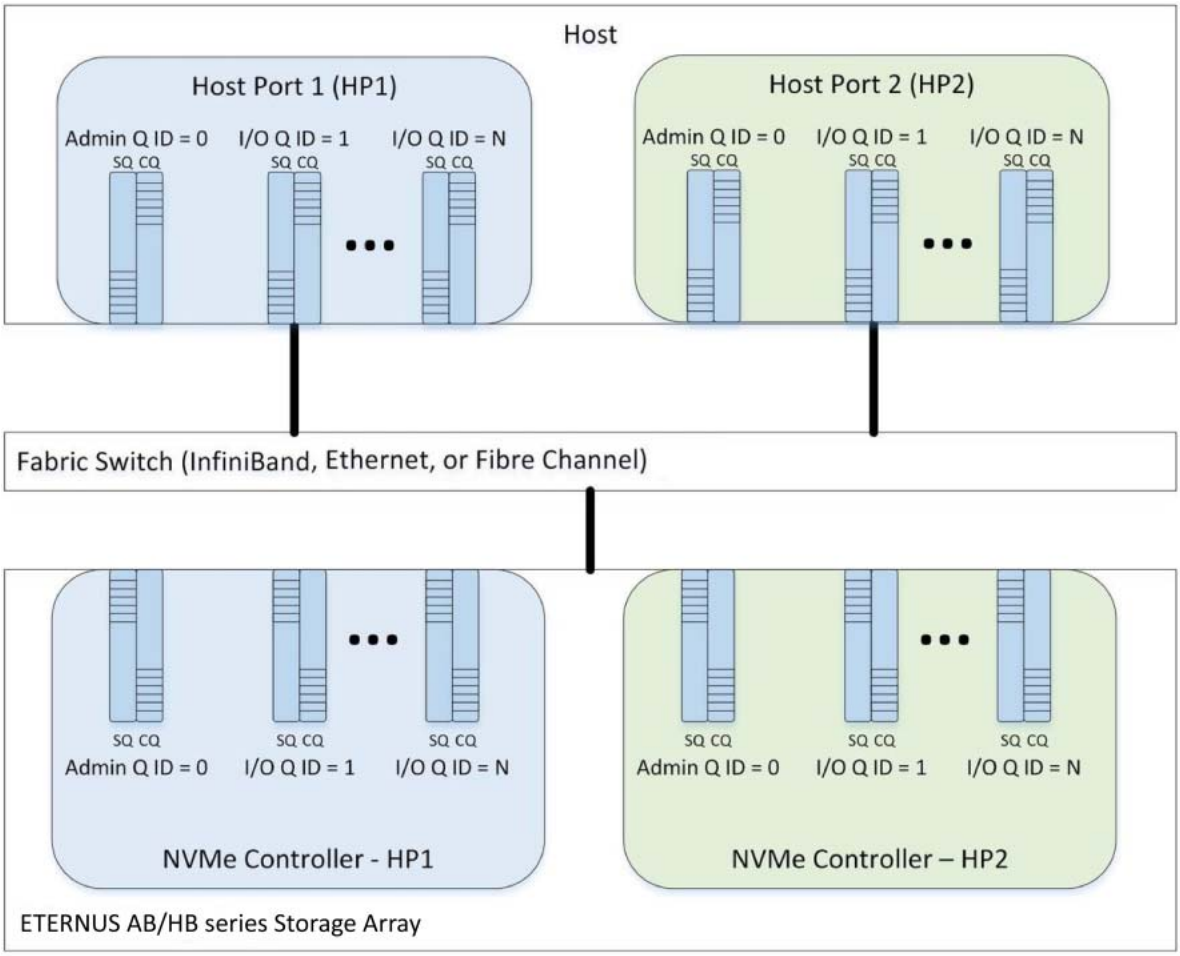
1. The initiator requests some number of I/O queues.
2. The target responds with the number of I/O queues that can be created on that controller.
3. The initiator issues I/O queue creation requests for the number of queues indicated by the target.

◀◀◀

The ETERNUS AB/HB series SANtricity OS software supports four I/O queues per controller.

Figure 5 illustrates the queue pair concept applied to NVMe controllers. The NVMe controller is a logical construct within the ETERNUS AB/HB series storage software. It represents the combination of the host port and the array port that forms the NVMe-oF connection between the host and the storage array. When an NVMe-oF connection is established, the host creates an admin queue (with a queue ID of 0) and then negotiates with the array to create N I/O queues (with queue IDs of 1 through N). Even though the host queue pairs and the storage array queue pairs are not part of the same physical memory system, NVMe-oF allows them to be a logically shared memory queue pair, with command and control structures being shared through RDMA between the host and storage array. Therefore, when the host puts a new command in a slot on its submission queue for I/O Queue 1, that command is placed, through RDMA, in the same slot in I/O Queue 1 for the associated NVMe controller on the storage array.

Figure 5 NVMe controller queue pairs



Command Sets

This section provides a brief overview of the NVMe and NVMe-oF commands supported by the ETERNUS AB/HB series storage array software. For details about specific commands, see the NVMe and NVMe-oF specifications, as well as the ETERNUS AB/HB series NVMe host interface software interface specification.

Admin Commands

Admin commands are received and completed through the controller admin QP. These commands are used, among other things, to configure and monitor the status of NVMe controllers connected to a host.

They are analogous to SCSI non-read/write commands such as `SCSI INQUIRY`, `MODE SELECT/SENSE`, and `LOG SELECT/SENSE`, among others.

Table 2 ETERNUS AB/HB series supported admin commands

Command	Specification	Mandatory/Optional
`Get Log Page`	NVMe	M
`Identify`	NVMe	M

Command	Specification	Mandatory/Optional
Abort	NVMe	M
Set Features	NVMe	M
Get Features	NVMe	M
Asynchronous Event Request	NVMe	M
Keep Alive	NVMe	O

NVM Commands

NVM commands are received and completed through the controller I/O QP. These commands are used for operations such as data movement and namespace access and are analogous to SCSI commands such as READ, WRITE, SYNC CACHE, PERSISTENT RESERVE IN, and so on.

Table 3 ETERNUS AB/HB series supported NVM commands

Command	Specification	Mandatory/Optional
Flush	NVMe	M
Write	NVMe	M
Read	NVMe	M
Reservation Register	NVMe	O
Reservation Report	NVMe	O
Reservation Acquire	NVMe	O
Reservation Release	NVMe	O

As noted in Table 3, in addition to the mandatory NVM commands, ETERNUS AB/HB series arrays using the NVMe-oF host interface support the optional reservation feature. This feature allows a host to reserve a namespace, which prevents other hosts from writing to the namespace (reads might also be restricted). This feature is commonly used by clustering packages to make sure that nodes within the cluster do not accidentally access the same namespace at the same time.

Those familiar with SCSI persistent reservations might find this feature to be very similar, although there are several differences. In SCSI, the registrant is tied to one and only one initiator-target-LUN (I_T_L) nexus, whereas in NVMe-oF, the registrant is tied to a namespace and host ID. Multiple controllers can have the same host ID, and all controllers using the same host ID are treated the same.

Another difference is that the Persist Through Power Loss setting can be set outside of a register command, and a register has the option of specifying No Change in addition to On and Off.

NVMe-oF does not have a specify initiator or all target port concept. Using the same host ID for all connections from a given host essentially solves the same problem. It is worth noting that NVMe-oF standard 1.2 has an Ignore Existing Key (IEKEY) parameter for every reservation command, but we have chosen to take guidance from 1.3, which dropped this parameter for all but the register command. The actual format of the commands and parameter data is noticeably different from SCSI.

Fabrics Commands

Table 4 lists the fabrics commands that are used to create queue pairs and initialize NVMe controllers. They are somewhat analogous to protocol-specific initialization commands such as the FC port login.

Table 4 ETERNUS AB/HB series supported fabrics commands

Command	Specification	Mandatory/Optional
`Property Set`	NVMe-oF	Mandatory
`Connect`	NVMe-oF	Mandatory
`Property Get`	NVMe-oF	Mandatory

Host Driver and Tools

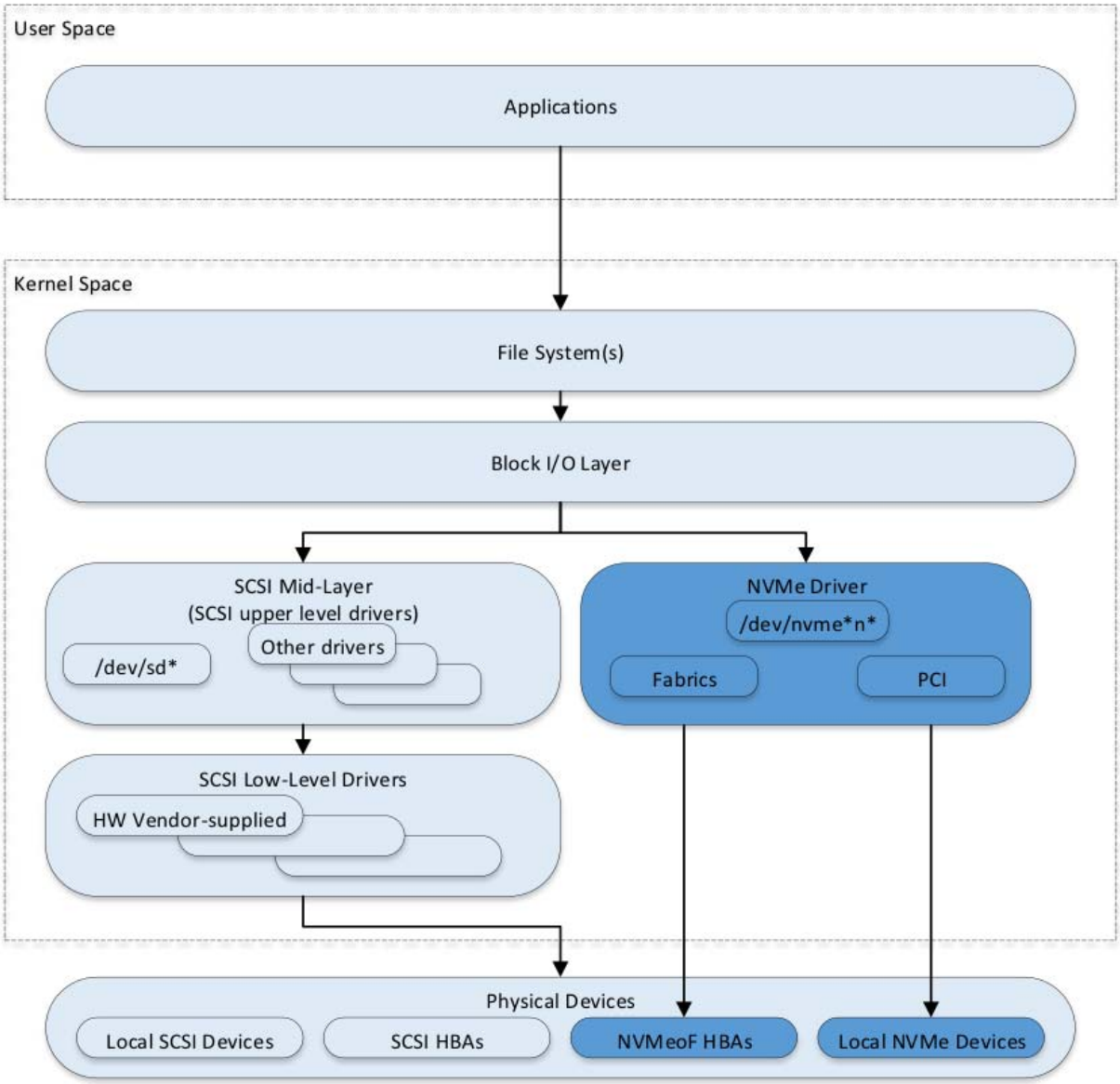
Driver Stack

One of the advertised advantages of NVMe (and NVMe-oF) relative to SCSI is that it can support lower-latency I/O. This is not only because the devices are faster, but also because of some advantages in the host OS driver stack. Therefore I/O spends less time getting from the application to the storage, which reduces response times.

Figure 6 shows a simplified view of the Linux OS driver stack. Like the SCSI driver, the NVMe driver sits below the block I/O layer. The NVMe driver, however, is not split into upper and lower levels. The driver presents the NVMe devices to the block I/O system and contains drivers to support both PCIe-attached devices and fabrics-attached devices. The fabrics portion of the driver contains the transport-specific code to handle operations such as IB-based RDMA.

Another advantage of using an NVMe driver is that they are designed with fast, nonvolatile memory storage devices in mind; that is, they are optimized to work well with low-latency storage. The SCSI driver, on the other hand, was originally created and used with rotating media-based storage such as spinning disk drives. For those devices, the I/O response time is mainly due to the operation of the device (for example, rotational seek times). Therefore, the driver itself did not have to be highly optimized for execution-time performance. Additionally, the SCSI driver stack was designed to support multiple different types of SCSI devices such as disks, tapes, printers, and so on. Therefore, there is some built-in overhead in the driver for handling I/O to various types of devices that is not present in the NVMe driver.

Figure 6 Linux OS driver structure



Multipathing and Failover

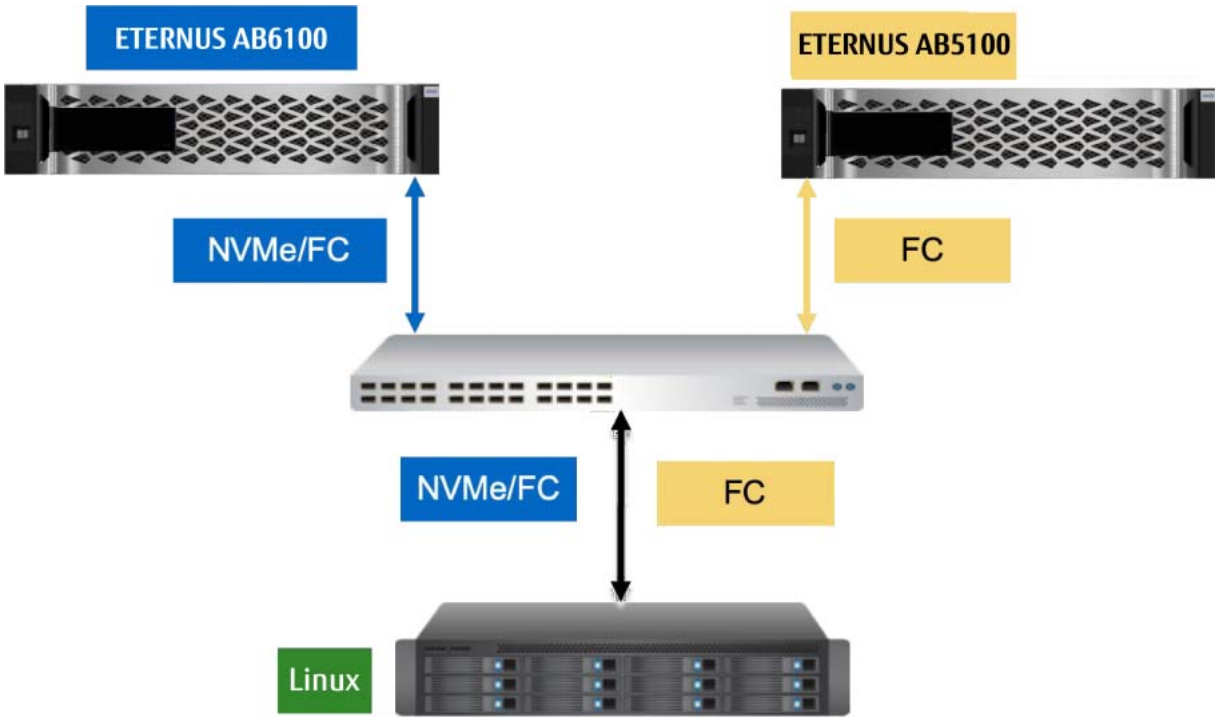
Currently, multipathing and failover functionality is provided by the Linux Device Mapper Multipath (DM-MP) module. Multiple paths to the namespaces on an ETERNUS AB/HB series array are automatically configured into a single logical device. The Linux multipath tools use the NVMe Asynchronous Namespace Access (ANA) feature to obtain information about the paths and works with DM-MP to ensure that I/O is sent to the ETERNUS AB/HB series namespaces through the optimal path, and to assist with failover and failback operations.

Newer Linux releases have the option of using multipathing/failover functionality that is built into the Linux NVMe driver. This native-NVMe-multipathing eliminates the need for the DM-MP layer when sending I/O to ETERNUS AB/HB series namespaces. The NVMe kernel driver uses NVMe ANA events to keep the path information updated and route I/O to the optimal paths. This information is also used for failover and failback operations.

Coexistence Between NVMe/FC and FC

On the host side, both FC and NVMe/FC can run on the same host bus adapter at the same time but to different ETERNUS AB/HB series targets, as shown in Figure 7.

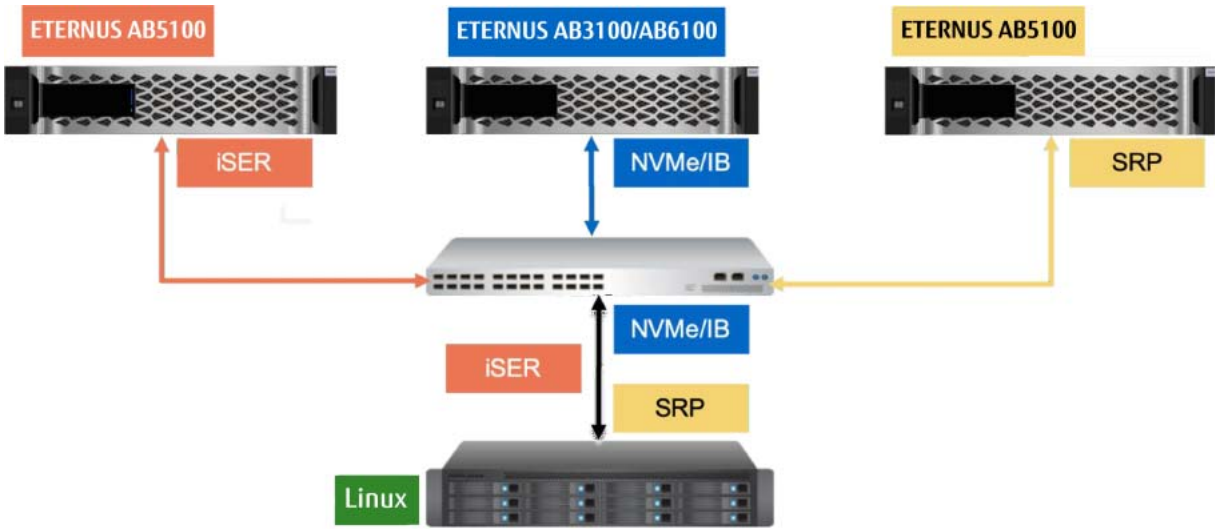
Figure 7 Coexistence example of NVMe/FC and FC



Coexistence Between NVMe/IB, iSER, and SRP

On the host side, all the protocols that ETERNUS AB/HB series devices support on IB (SRP, iSER, and NVMe/IB) can run on the same host channel adapter at the same time but to different ETERNUS AB/HB series targets, as shown in Figure 8.

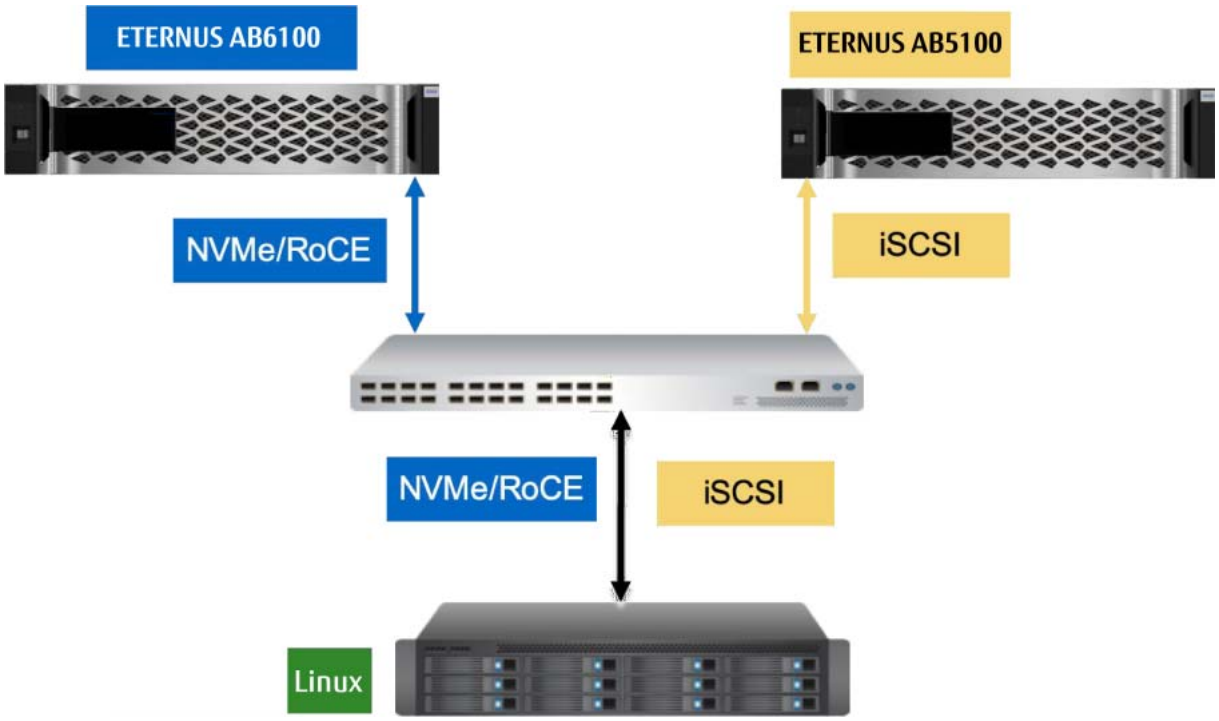
Figure 8 Coexistence example of NVMe/IB, iSER, and SRP on the host side



Coexistence Between NVMe/RoCE and iSCSI

On the host side, iSCSI and NVMe/RoCE can run on the same network adapter, provided that it supports RDMA at the same time, but to different ETERNUS AB/HB series targets, as shown in Figure 9.

Figure 9 Coexistence example of NVMe/RoCE and iSCSI



NVMe CLI

The NVMe CLI is a command line utility that provides management tools for NVMe devices on a Linux host. The utility can list NVMe namespaces that are configured as block devices on the host and can provide details about those namespaces. For NVMe-oF, it has subcommands for discovering and connecting to NVMe controllers on the fabric. Table 5 lists some useful commands. For more information, see the Linux man pages by running the `man nvme` command from the Linux shell.

Table 5 Some useful NVMe CLI commands

Command	Description
`nvme list`	Lists information about each NVMe namespace configured as a block device on the system.
`nvme discover`	Discovers NVMe controllers on the fabric.
`nvme connect`	Connects to an NVMe controller on the fabric.
`nvme connect-all`	Discovers and connects to multiple NVMe controllers on the fabric by using settings contained in `/etc/nvme/discovery.conf`.

A. Frequently Asked Questions

- Are regular network interface cards (NICs) supported for NVMe/RoCE?

  Answer:
  No. The cards must support RDMA. They are called RDMA NICs (rNICs) and are widely available.

- Is direct connect supported, or is a switch required?

  Answer:
  Both direct connect and fabric connection through a switch are supported for NVMe/IB, NVMe/RoCE and NVMe/FC.

- Do ETERNUS AB/HB series controllers support direct connect to multiple hosts using a 4x 25GbE by 100GbE cable for NVMe/RoCE?

  Answer:
  No. Direct connect is not supported.

- What are the different form factors for different speeds?

  Answer:
  - NVMe/IB: QSFP28 for 100Gbps, 56Gbps, and 40Gbps
  - NVMe/RoCE: QSFP28 for 100Gbps and 50Gbps; QSFP+ for 40Gbps; SFP28 for 25Gbps; and SFP+ for 10Gbps.

- What protocols do the AB5100 and HB5000 support with the 100Gbps InfiniBand host interface card?

  Answer:
  The SRP, iSER, NVMe/IB, and NVMe/RoCE protocols are supported, but not at the same time. Migrating between protocols is software-based, and no hardware change is required.

- Do ETERNUS AB/HB series systems support RoCE, which runs SCSI?

  Answer:
  No. Only the NVMe protocol is supported to run over RoCE, not SCSI protocol.

- Which version of NVMe/RoCE does ETERNUS AB/HB series support?

  Answer:
  Supports NVMe/RoCE version 2 and is also compatible with NVMe/RoCE version 1.

FUJITSU Storage
ETERNUS AB series All-Flash Arrays,
ETERNUS HB series Hybrid Arrays
NVMe over Fabrics Support

P3AG-6392-01ENZ0

Date of issuance: December 2021
Issuance responsibility: FUJITSU LIMITED


FUJITSU