Micron® Software-Defined Storage with Excelero NVMesh®

Reference Architecture

systems software storage memory

Contents

Executive Summary

To address the needs of high-performance enterprise and cloud-centric applications, Micron has collaborated with Excelero, Mellanox® and Supermicro® to develop a comprehensive solution for scalable, remote NVMe™ storage using:

- Excelero's NVMesh software-defined storage platform
- Micron® 9100 MAX SSDs and DRAM
- Supermicro SYS-2028U two-socket Intel® Xeon® server platform
- Mellanox Spectrum™-series Ethernet switches and ConnectX®-series high-bandwidth network interface card solutions

This reference architecture details the optimized configuration and performance of a three-node Excelero NVMesh solution that provides NVMe SSD-based logical volumes to Linux-based application servers running the NVMesh Intelligent Client Driver.

Performance of the Micron Software Defined Storage (SDS) with Excelero NVMesh Reference Architecture (RA) was recorded using real-world RAID-10 logical volumes that provide full data protection as part of an overall high-availability design. Focused on enterprise solutions such as online transaction processing (OLTP) database applications that depend on random, small-block I/O to centralized RAID protected volumes for bare-metal application servers, this solution illustrates the value of Excelero's NVMesh for OLTP databases.

Read/Write Ratio	IOPS	Throughput (GB/s)	Latency (99th Percentile, ms)
0% Read	1,740,000	6.96	36.8
70% Read	3,000,000	11.74	29.33
100% Read	7,550,000	30.21	27.53

Table 1: Summary Findings for the Micron SDS with Excelero NVMesh RA

Performance and configuration details are provided to enable reproduction of the results detailed in this RA.

Micron's Reference Architectures

Micron Reference Architectures are optimized, pre-engineered, enterprise-leading platforms that are developed by Micron with industry leading hardware and software companies.

Designed and tested at Micron's Storage Solutions Center by our software and platform partners, these best-in-class solutions enable end users, channel participants, independent software vendors (ISVs), and OEMs to have a broader choice in deploying next-generation solutions with reduced time investment and risk.

The Purpose of this Document

This document describes a reference architecture (RA) for deploying a performance-optimized software-defined storage (SDS) solution for Excelero NVMesh using Micron SSDs with NVMe.

Detailed within this document are the hardware and software building blocks used to characterize the RA's performance. This document covers the Micron SDS composition including the server and OS configuration for the NVMesh target nodes, the configuration of the application servers running the NVMesh Intelligent Client Driver (NVMesh Client) and the storage network infrastructure configurations.

The purpose of this document is to provide a pragmatic blueprint for administrators, solution architects and IT planners who need to build and tailor high-performance SDS infrastructures at scale for I/O-intensive and/or latency sensitive workloads. Micron assumes no liability for lost, stolen or corrupted data or performance differences arising from the use of any Micron product. Products are warranted only to meet Micron's production data sheet specifications. Products, programs and specifications are subject to change without notice.

Why Micron for this Solution

Storage (NVMe SSDs and DRAM) represents a major portion of the cost and performance potential in today's advanced server/storage solutions. Micron's storage expertise starts at memory technology research, innovation and design and extends through collaborating with customers and technology leaders on total data solutions. Micron develops and manufactures the storage and memory products that go into the enterprise solutions we architect.

Micron has worked extensively to apply our storage expertise in defining an optimal hardware and software solution that combines the SDS expertise of Excelero with leading networking provider Mellanox and high-performance server vendor Supermicro. With years of real-world application optimization experience, Micron is uniquely positioned to provide a complete, high-performance SDS solution to meet demanding workload requirements.

Solution Overview

Excelero NVMesh targets high-performance storage solutions that take advantage of the efficiency and performance of Micron's latest NVMe SSD and memory components within low cost, off-the-shelf industry standard servers with near server-local performance in a remote storage solution.

Based on our experience developing advanced storage solutions and ongoing discussions with our customers regarding what they were looking for in an advanced storage solution, Micron identified Excelero as an SDS solution that could provide the key functionality and features that are important for an advanced SDS solution, including:

- Deployment flexibility
- Scalability
- Data protection
- Manageability

Deployment Flexibility

Flexibility is the characteristic that enables potential users to deploy a solution in multiple ways and on a broad range of hardware. Micron's SDS with Excelero's NVMesh RA meets these requirements. By organizing the solution as a set of three independently deployable software components, users have the flexibility to build deployments consisting of storage-centric, converged compute and storage, or any combination of these.

The three primary SDS components work together to provide a highly-available and scalable NVMe block storage service for high-performance applications. Coordinated by a centralized, out-of-band management service, the NVMesh Intelligent Client Block Driver and the NVMesh Target Module provide access to virtualized block storage (Figure 1). This architecture allows customers to target specific requirements of one or more workloads by being able to decide if each server runs the Intelligent Client Driver, the Target Module, or both data components simultaneously.

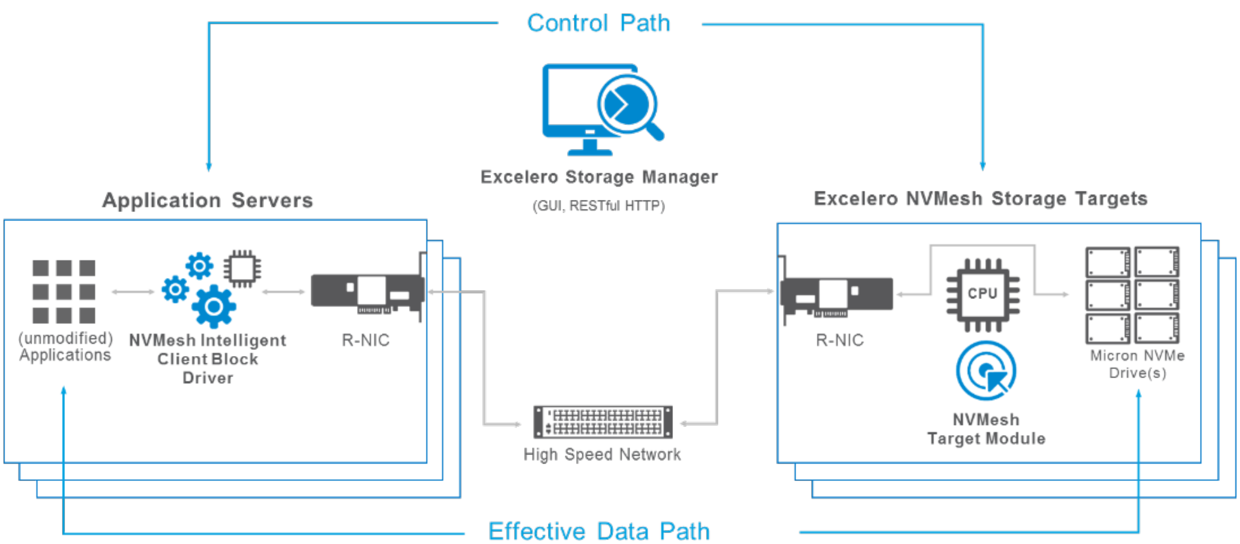

Figure 1: Micron SDS with Excelero NVMesh Reference Architecture Overview

In addition, Excelero NVMesh is supported on a wide variety of x86 server platforms from a broad range of vendors. The only requirements are that the server supports native NVMe SSDs and high-speed memory solutions such as those offered by Micron, in addition to high-bandwidth networking solutions based on InfiniBand® or Ethernet protocols that support Remote Direct Memory Access (RDMA). For this RA, Micron selected hardware components from Supermicro and Mellanox.

Scalability

The ability to scale seamlessly has been noted by many Micron customers as a critical component of any storage infrastructure. What scalability means to each customer can vary greatly; Micron focused on the following:

- **Capacity:** Any data center-focused storage solution should be able to increase capacity as needed—with no downtime—and grow capacities without reducing overall storage performance in unpredictable ways.

  Excelero NVMesh supports capacity growth in numerous ways. First, as a scale-out clustered SDS solution, users can add additional storage target nodes to the cluster without any additional downtime. Second, when using target node servers, each node can have additional SSDs added to the solution and incorporated into use for existing and new application servers. Third, Excelero NVMesh supports hundreds of target nodes supporting multiple 10s of petabytes of storage per cluster[1]

- **Transactional:** The solution should provide a well-defined method for providing more transactional performance—in terms of input output operations per second (IOPS) —with consistent, low latency.

  Increasing transactional performance can only be attained by adding additional storage devices to the solution. Users who do not have a high-capacity requirement, but who do have a high-transactional requirement, have the flexibility to use additional lower-capacity SSDs across more NVMesh target nodes to increase overall solution IOPS performance.

- **Hosts:** The solution needs to be able to support many application servers without additional hosts directly impacting the performance of other application servers also using the storage solution.

  Excelero NVMesh supports large numbers of application server hosts. It does so by moving all IO operational responsibility away from the storage target and into the application servers using the storage. This enables predictable application performance planning regardless of the number of hosts. Solutions are only limited by the total IOPS performance of the target nodes and the available network bandwidth available in the storage network.

Data Protection

An SDS solution must protect stored data from loss or unavailability. Historically, this has been accomplished by using proprietary hardware platforms that provide redundant storage controllers, power supplies and network interfaces. An SDS is designed to run on standard server platforms that may not provide all the traditional hardware redundancies of proprietary solutions. To continue to provide full availability and no data loss, SDS solutions use other means for success.

Depending on the type of application being deployed, data protection can be the responsibility of either the application or the storage solution. Many “cloud-ready” applications such as scale-out NoSQL or real-time analytics engines typically rely on their internal data protection schemes to support maximum efficiency and performance for that specific application. For these applications, Excelero NVMesh can provide simple, non-protected data volumes that support concatenated volumes or RAID-0 striped volumes.

For more traditional enterprise applications, such as SQL or unified communications applications that have an expectation of data protection being provided by the storage solution, Excelero NVMesh supports its own RAID-1 and RAID-1/0 data protection capabilities. Both data protection configurations ensure each mirror is located on different array nodes within the storage cluster, resulting in a solution that is completely fault-tolerant.

---

[1] 128 nodes using 24x 11TB Micron 9100 ECO SSDs results in 33PB of raw storage.

Manageability

The last major requirement for a good storage solution is that it must be easily managed. It must have the ability to easily create, manage, and destroy volumes, add and remove target nodes as well as application servers, and it should be able to do this in either an interactive or automated manner. Excelero NVMesh has a robust, out-of-band storage management capability that has many valuable features and capabilities, including:

- Inventory maintenance for all client and target nodes as well as all NVMe SSDs in each target node
- Management support for hundreds of NVMesh-target nodes with thousands of NVMe SSDs
- Database maintenance for all logical volumes that have been defined and a list of the NVMe SSDs on which those logical volumes reside, as well as what application servers have access to each logical volume
- Statistics generation for all logical volume performance
- RESTful application programming interface (API) to support remote, scripted interaction via standards-based operational tools
- Open-source database deployed in a three-way replica-set to provide high-availability and data protection
- Provisioning support for NVMe storage resources
- Standard HTTP-based graphical user interface for ease of use and visualization

Together, these key storage functional areas can determine the value of a storage solution. Based on these criteria, Micron has selected Excelero's NVMesh solution for this RA to highlight the value of Micron NVMe SSDs as a foundation for a highly scalable, low-latency storage solution.

Reference Architecture Design Overview

The Micron SDS with Excelero NVMesh RA uses enterprise SSDs with NVMe interfaces and high-speed memory from Micron together with servers from Supermicro and a dedicated storage network infrastructure with switches and host adapters from Mellanox (Figure 2).

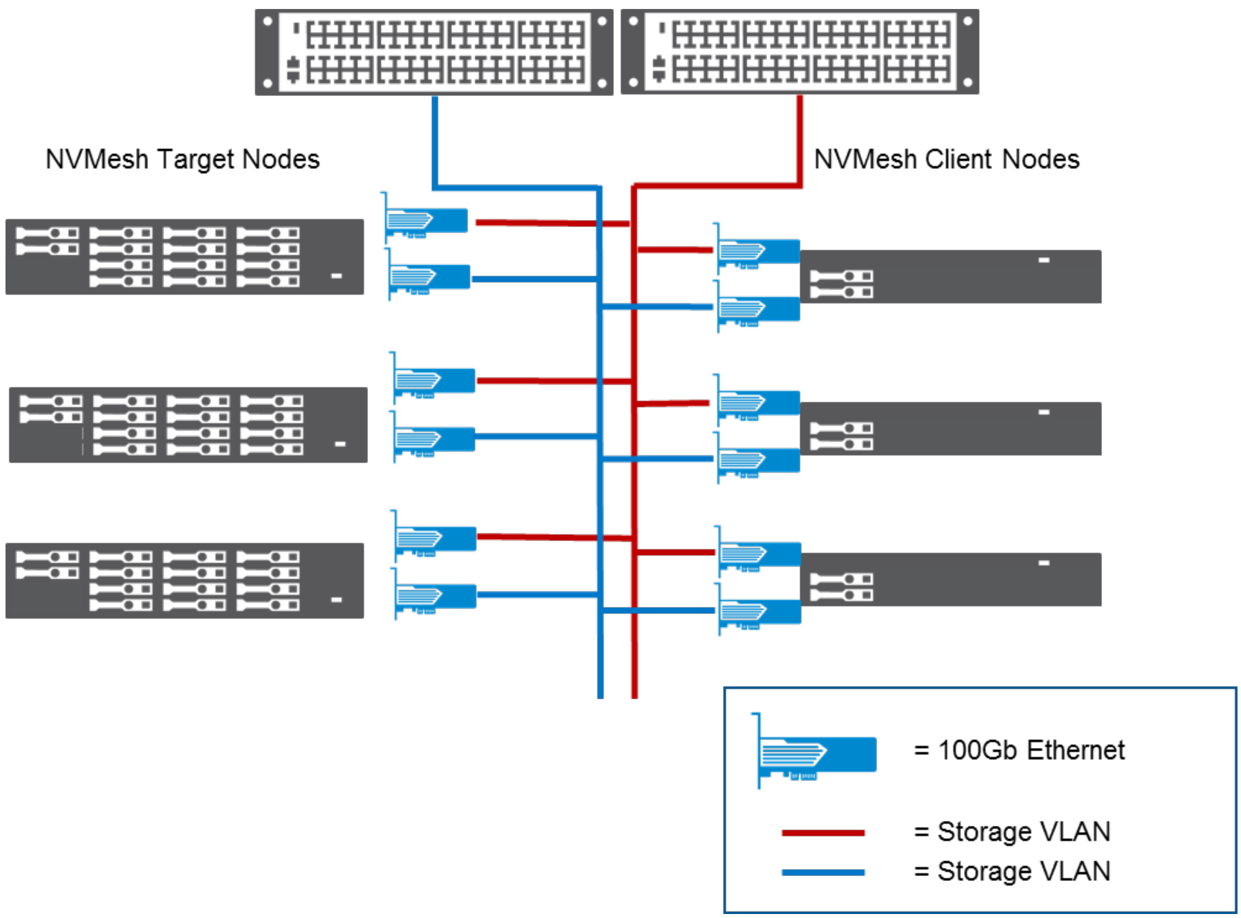

Figure 2: Reference Architecture Elements

To support the Micron SDS with Excelero NVMesh RA, we need a high-performance network that can more easily support the movement of NVMe traffic between application hosts and the storage target nodes. NVMe-based SDS solutions need to have additional networking capabilities over traditional Ethernet networks; in particular, they need to support RDMA over Converged Ethernet (RoCE).

For this RA, Micron and Mellanox have been working closely together on optimizing access to the NVMe SSDs and on maximizing the performance of the solution when connected by Mellanox end-to-end networking solutions, which support industry networking standards such as TCP/IP, RoCE and InfiniBand from 10 Gb/s to 100 Gb/s.

The following sections describe a basic building block architecture that can be easily expanded in terms of capacity and performance as your storage needs grow.

Reference Architecture Elements

The Micron SDS with Excelero NVMesh RA is composed of three primary components:

- NVMesh target nodes
- NVMesh client nodes
- 100Gb Ethernet storage area network

NVMesh Target Nodes

The RA consists of three target nodes configured as an NVMesh storage cluster. Each target node runs the Excelero NVMesh Target Module and provides all NVMe-hosted storage services for all application servers in the RA. NVMesh is scalable to 128 target nodes in the tested release.

NVMesh Client Nodes

For testing purposes, the RA is configured with three application servers used for load-generation purposes during testing. All nodes are running the Excelero NVMesh Intelligent Client Driver. NVMesh is scalable to thousands of client nodes.

Mellanox High-Bandwidth RDMA over Converged Ethernet Storage Network

The RA is built upon a set of core Ethernet functionalities called RDMA over Converged Ethernet (RoCE), which extends server-local DMA protocols that are industry standard interfaces for high-performance applications across advanced, high-bandwidth Ethernet networks. Built upon industry standard IEEE 802.1 Data Center Bridging protocols and InfiniBand® technology, RoCE ensures that NVMesh has high bandwidth, low latency connection between application servers and their assigned remotely deployed NVMe devices.

Data Center Bridging

802.1Qbb – Priority Flow Control (PFC)	An Ethernet flow control mechanism that can independently pause different classes of traffic based on their 802.1p Class of Service value.
802.1Qaz – Enhanced Transmission Selection (ETS)	Provides a common method for managing bandwidth access for different classes of traffic, ensuring fair access to an Ethernet path based on application priorities.
Data Center Bridging Exchange Protocol (DCBX)	A part of the 802.1Qaz standard, provides a standard interaction framework between two network bridging devices to ensure each device understands the capabilities of the peer devices on the network.

Micron Components

Micron 9100 Max SSDs with NVMe

The 9100 family of NVMe SSDs provides workload-focused endurance and capacities for both read-centric and mixed-use applications and environments.

Offered in half-height, half-length (HHHL) and U.2 industry standard form factors in capacities up to 3.2TB, the 9100 is available in either read-centric (9100 PRO) or mixed-use (9100 MAX) classes, providing an exceptional balance of performance, endurance and price.

Micron's SDS with Excelero's NVMesh RA is designed from the ground up to take advantage of NVMe primary storage to provide remotely accessible 9100 MAX SSDs with near server-local performance. For this RA, 2.4TB 9100 MAX U.2 SSDs were used. Features of the Micron 9100 family of SSDs include:

- **Low Total Cost of Ownership:** NVMe SSDs can accelerate data with throughput compared with SATA and SAS SSDs. You can calculate TCO of your solution using the Micron TCO Tool.
- **Enhanced Performance:** Improve workload performance with transfer speeds up to 3 GB/s[2]
- **Reliability and Quality:** Protect mission-critical data with power-loss protection and data path protection features
- **Optimized Endurance:** Choose from endurance options matched to read-centric or mixed-use workloads
- **XPERT Firmware Features:** eXtended Performance and Enhanced Reliability Technology (XPERT) features such as power-loss protection, Redundant Array of Independent NAND (RAIN), data path protection, reduced command access latency, adaptive read and thermal protection

The table below shows the general specifications of the 2.4TB U.2 9100 MAX SSD used in this RA. Additional information about this SSD and the entire 9100 series is available in the family datasheet.

Model	9100 MAX	Interface	PCIe Gen 3 x4 lanes
Form Factor	Pluggable U.2 (PCIe x4)	Capacity	2.4TB
Sequential Read (128KB)	Up to 3GB/s	Sequential Write (128KB)	Up to 2 GB/s
Random Read (4KB)	Up to 750K IOPS	Random Write (4KB)	Up to 300K IOPS
Endurance	9.6PB total bytes written		

Table 2: Micron 9100 Specifications[2]

Micron 32GB DDR4 ECC RDIMM Memory

Micron DRAM modules provide the performance and reliability required for a wide range of mission-critical applications. For this RA, each Excelero target node was configured with 512GB of RAM, and each application server was configured with 256GB of RAM using 32GB RDIMMs.

[2] Source: Micron 9100 Technical Specification Data Sheet

Solution Design—Hardware

Excelero Target Nodes

The Supermicro SYS-2028U-TN24R4T+ server was used for each NVMesh target node. This server has a design that provides advantages for a storage-centric solution platform. A two socket, Intel Broadwell chipset provides 24x NVMe U.2 front-facing drive bays, each supporting four PCIe lanes as well as two x16 width PCIe Gen 3 option slots in the rear. (See Appendix A for detailed server configuration information.)

Each CPU complex (NUMA Node) hosts a single x16 PCIe adapter slot as well as x16 PCIe switch connected to 12x 4-lane U.2 drive bays on the midplane. In this RA, this server provides a balanced architecture ensuring storage traffic can be organized and directed with minimal use of the inter-CPU complex inter-connect (Figure 3).

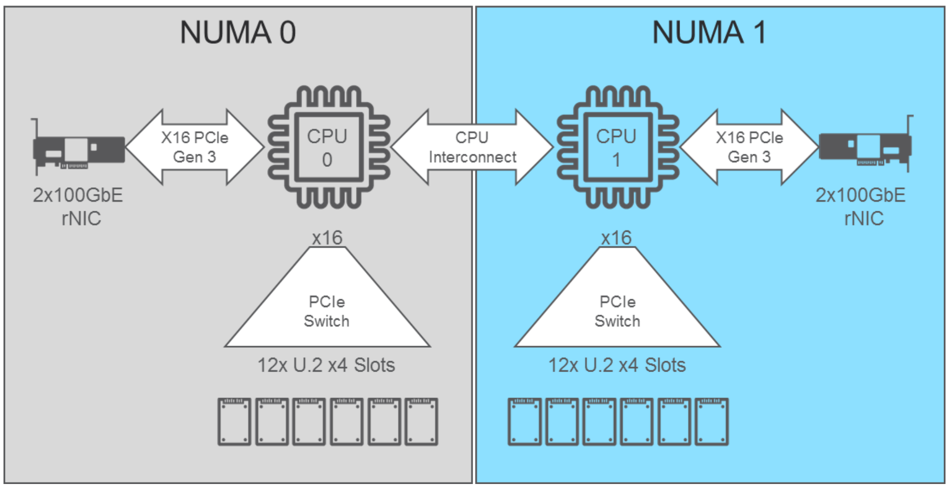

Figure 3: Excelero NVMesh Target Node Block Diagram

For this solution, each target node was configured with:

- Two Mellanox ConnectX-5 dual-port 100Gb Ethernet NICs installed in the two x16 PCIe slots to provide up to 22 GB/s bandwidth into and out of each target node
- 12x Micron 9100 NVMe U.2 SSDs installed such that six drives are attached to each CPU complex (Figure 3)

Application Servers (Excelero Intelligent Client Nodes)

The Supermicro SYS-1028U-TR4T+ server was used for each client node. A two socket, Intel Broadwell chipset provides a powerful platform for hosting most application solutions.

Each application server was configured with two Mellanox ConnectX®-5 100 GbE Ethernet adapters for connection to the dedicated storage network.

See Appendix A for detailed server configuration information.

Storage Network Configuration

A dedicated storage network is recommended for this solution. The network is designed to provide a fully redundant interconnect fabric, ensuring no single component will interrupt data availability in production.

The network consists of two Mellanox Spectrum SN2100 Open Ethernet Switches running MLNX-OS® and providing 16x 100 GbE ports each.

Physical connections are illustrated in Figure 4. Each application server has one 100Gb network interface connected to each switch to ensure maximum load was placed on the NVMesh target nodes. Each NVMesh target node has two ports connected to each switch, with one port from each adapter attached to each switch. Finally, each switch is interconnected using 4x 100Gb ports to provide alternate data paths between application servers and storage targets.

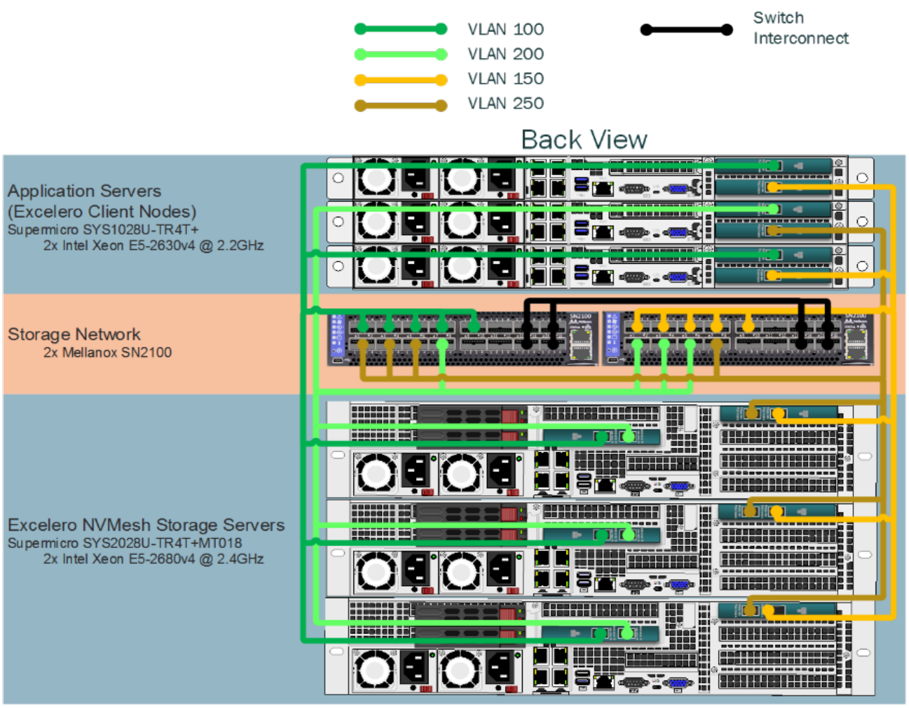

Figure 4: Excelero NVMesh Network Configuration

Each switch is configured to support two different VLANs (four VLANs in the solution), with each VLAN associated with a single 100 GbE port on each target node. VLANs 100 and 150 are configured on the left switch; VLANs 200 and 250 are configured on the right switch.

The switch-to-switch interconnect is configured to forward traffic for all VLANs.

In the case of a network component (switch, cable, or port on either the target nodes or application server) failure, the NVMesh Intelligent Client Driver will continue sending traffic through the remaining paths.

Solution Design—Software

Excelero NVMesh

Excelero NVMesh target nodes and application servers were configured with NVMesh 1.2.0 build 855. Target nodes were installed with the NVMesh Target Module. Application servers were installed with the NVMesh Intelligent Client Driver.

NVMesh-Management 1.2.0 build 142 was installed on each NVMesh target node and was deployed in a HA fashion using MongoDB configured in a replica-set.

Operating System

All Excelero target nodes and application servers used for this solution used CentOS Linux release 7.4. The CentOS Project is an open-source, community supported project that allows users access to a robust, Linux operating system environment suitable for almost any workload. Built upon the well-known Red Hat Enterprise Linux® distributions, CentOS is a well-accepted solution in the industry. To find out more about CentOS, visit www.centos.org.

For this solution, Linux CentOS 7.4 with kernel version 3.10.0-693.2.2.el7.x86_64 was used for all testing.

MongoDB

Excelero NVMesh Storage Manager utilizes MongoDB as its data storage solution. MongoDB is an open-source NoSQL database solution that uses a document paradigm for all data storage where each document is composed of key-value pairs. Values can be any structured or unstructured data such as a document, array, or binary object. MongoDB uses a distributed, scale-out architecture that leverages a cluster paradigm of individual nodes with each node hosting a part (shard) of the overall database as a primary manager and replicas of database shards that are hosted on other MongoDB nodes within the cluster for redundancy.

For this RA, MongoDB version 3.2.17 was installed on each NVMesh target node along with the Excelero NVMesh Storage Manager.

Planning Considerations

The following planning considerations should be reviewed when creating an SDS using Excelero NVMesh:

- **Balanced target server architectures:** It is important to consider the design of the server platform used to run the NVMesh software as a *storage-centric* solution. Many multi-socket server designs targeted at general-purpose computing workloads offer unbalanced architectures. An unbalanced architecture is one where some critical components (PCIe slots, drive slots, etc.) may be tied to one CPU complex or another, but not necessarily both complexes.

  An example would be a server offering a design where a large proportion of the physical PCIe option slots are connected to one CPU complex, and all storage I/O interfaces are assigned to a different CPU complex. This design would require that all disk I/O traverse the CPU complex interconnect as data moves between network interfaces and storage devices.

  Look for server offerings that provide more balanced designs like the server platform chosen for this RA.

- **Minimum number of target nodes:** Excelero NVMesh storage software is architected as a shared-nothing, scale-out cluster. To ensure all data is redundant and cluster health can be assured, at least three NVMesh Target Nodes must be deployed. This ensures that high-availability may be maintained in the event of a node loss or communications issue on the network. It also enables the use of mirrored volumes with an arbiter on a third node for cluster quorum majority.

- **Data redundancy:** Depending on the applications using the NVMesh storage services, it should always be ensured that data is protected. For traditional data center applications, such as SQL databases or mail servers, this may require the creation of NVMesh logical volumes that are configured with RAID-1 or RAID-1/0 for data protection. For applications based on newer, open source, scale-out cloud-ready architectures—such as NoSQL database applications—ensure the application is using a replication factor of 2 or higher or erasure coding.

- **Application server network bandwidth:** While the RA described in this document utilizes 100 Gb/s Ethernet interfaces, not all applications require this much bandwidth. It is acceptable to use interfaces with higher or lower bandwidths to more efficiently meet the requirements of the workload running on the application server.

- **NVMesh target node network bandwidth:** While the RA described in this document utilizes 100 Gb/s network interfaces, it is not a requirement. It is recommended that the target nodes be given as much bandwidth as possible to provide all application servers using the storage solution adequate bandwidth to meet application service-level objectives.

- **NVMesh target node CPU sizing:** While the target nodes in this RA are dual-socket servers using Intel Xeon 2680v4 CPUs, the Excelero NVMesh Target Module does not consume CPU in its normal operational mode; therefore, it is recommended to use a lower-performing CPU option for dedicated target nodes.

- **Storage network:** NVMesh relies on RDMA over Converged Ethernet (RoCE v2) for all data movement within the storage network. Switches must support IEEE Data Center Bridging functionality.

- **Application server file system choice:** The file system type used for application servers that are assigned NVMesh volumes directly impact performance. NVMesh supports the use of almost any file system supported by Linux. All tests presented in this document utilized the XFS file system.

Measuring Performance

An NVMesh storage solution such as the one described in this document can provide storage services for a wide variety of applications. Accurately running and documenting every possible workload or application is not possible.

To test the performance of this solution, a synthetic workload generator was used to simulate a small-block, random 4 KiB I/O workload typically used as a benchmark for OLTP database-type applications.

To determine the optimum configuration, a series of tests were run to illustrate a small subset of workload profiles that should provide general performance guidance in support of planning a production solution based on this RA.

Test Methodology

The following test methodology was used for this RA:

- A baseline test was performed to ensure that each node was properly configured and to determine the maximum performance for a single node. The baseline test consisted of having the three NVMesh client nodes simultaneously accessing logical volumes from a single NVMesh target node.
- A primary test configuration focused on using a set of RAID-1/0 logical volumes. All three NVMesh target nodes were utilized and I/O was generated using all three application servers running the NVMesh Intelligent Client Driver.

Prior to running benchmarks for each test phase, all volumes were pre-conditioned as a best-practice when I/O testing with SSD media.

Each test was run using the Flexible I/O workload generator (FIO) version 3.2, an open-source I/O workload generator commonly used for general benchmarking in the Linux community. For more information about FIO, see https://github.com/axboe/fio.

Workload profiles for each test run consisted of those shown in Table 3.

- All tests were executed five times
- Results shown are based on the statistical averages of all five test runs
- Each test run was executed for one hour, resulting in a total of 15 test hours

Test	FIO Settings
100% 4KiB Random Reads	rw=randrw numjobs=28 blocksize=4k iodepth=64 rwmixread=100 ioengine=libaio direct=1 refill_buffers norandommap randrepeat=0 group_reporting
100% 4KiB Random Writes	rw=randrw numjobs=28 blocksize=4k iodepth=64 rwmixread=0 ioengine=libaio direct=1 refill_buffers norandommap randrepeat=0 group_reporting
4KiB 70% Read/ 30% Write	rw=randrw numjobs=28 blocksize=4k iodepth=64 rwmixread=70 ioengine=libaio direct=1 refill_buffers norandommap randrepeat=0 group_reporting

Table 3: Workload Profiles

Drive Conditioning

For each configuration tested, all volumes were pre-conditioned to ensure that data was completely and evenly distributed across each volume. Table 4 shows the FIO configuration settings used to condition the volumes:

Setting	Value
Global Settings	rw=write numjobs=1 ioengine=libaio direct=1 refill_buffers norandommap randrepeat=0
Volume Settings	filesize=3200G blocksize=2M iodepth=64

Table 4: FIO Configuration Settings

Test Results and Analysis

Baseline Test

A baseline test was executed to determine the maximum limits of the solution. This test does not necessarily represent a real-world solution, but helps provide context for the actual testing described later in this document.

For this test, the goal is to determine the maximum performance of a single NVMesh target node. To accomplish this, each of the three NVMesh client servers was assigned a set of four 2 TiB RAID-0 volumes. Each volume was configured to use an equal amount of storage space allocated from each of the 12 drives in the single NVMesh target node (Figure 5).

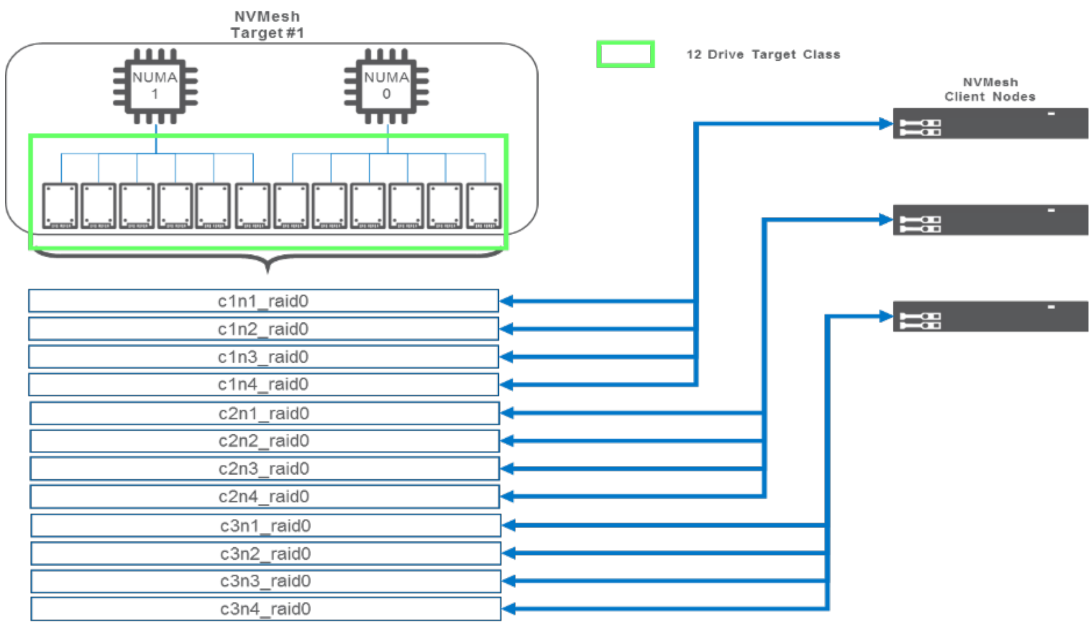

Figure 5: Baseline Test Configuration

A FIO thread value of 28 was used to ensure each CPU core (each Intel Xeon 2680v4 provided 14 hardware cores) executed a single thread. Our testing showed that executing more threads—thus leveraging the CPU's hyper-threading functionality—did not result in better performance.

A queue depth of 64 was used as it showed the highest performance value at 28 threads. The goal of this baseline test was to show the maximum performance capable for a single node.

This baseline test utilized all drives in a single target node, ensuring that the clients generated enough storage I/O to fully utilize the network bandwidth of the target node.

The test results showed that this configuration resulted in a total of 4.3 million IOPS. This equates to approximately 16 GiB/s when using 4KiB block size. This result is in line with our expectations based on the following factors:

- While our target nodes have 4x 100 GbE ports (2x dual-port PCIe x16 network interface cards), the effective throughput of these four ports is approximately 200 Gb/s since each adapter resides in a x16 PCIe port capable of supporting throughput of approximately 100 Gb/s each.
- There will be some overhead imparted by the operating system I/O stack.
- The FIO tool and the `libaio` I/O engine used for the testing imparts some performance impact due to its CPU utilization and other factors.

Mainstream Test

One criteria for the mainstream performance test is to use a configuration that accurately represents a highly available production OLTP-focused solution using RAID-10 volume definitions.

In this RA, RAID (both RAID-1 and RAID-1/0) is “node aware” and ensures that each side of a volume mirror is located on a different NVMesh target node. Leveraging the drive class functionality of NVMesh-Management, three drive classes were created. Each drive class contained 6x SSDs from two different nodes, as illustrated in Figure 6. This ensures that logical volumes created on a given storage class will always use those SSDs. A byproduct of this configuration is that each application server utilizes storage resources of only two of the three storage target nodes.

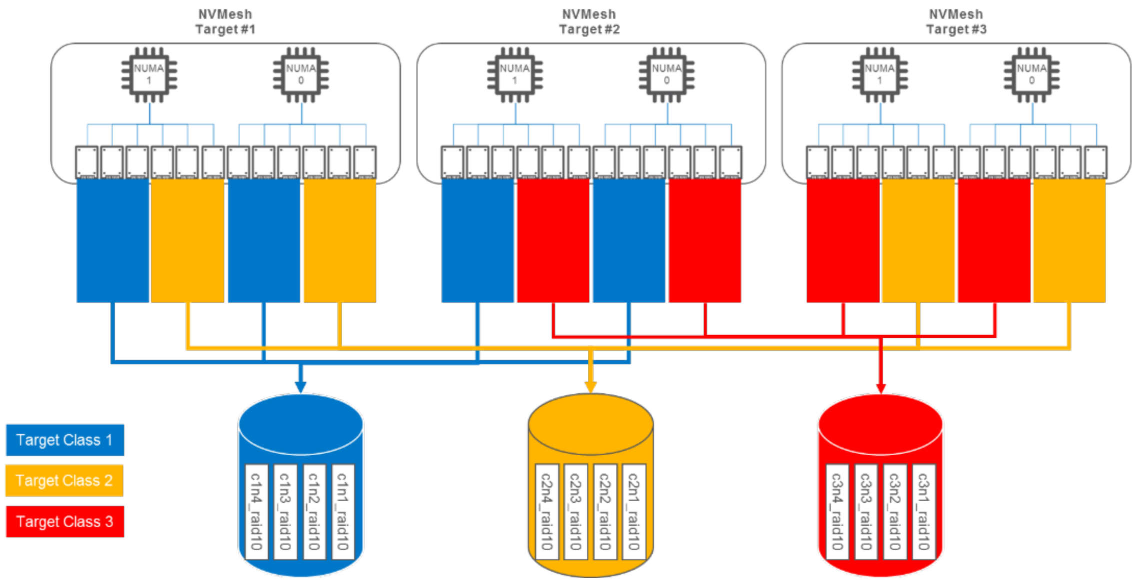

Figure 6: Logical Volume Configuration for Mainstream Test

In each storage class, four logical volumes were created with each logical volume capacity being set at 3.2 TiB mirrored (raw storage consumed is 6.4 TiB per volume). This resulted in 12x RAID-1/0 logical volumes consuming a total of 76.8 TiB. This is approximately 89% of the total capacity of the three-node Excelero cluster.

Each NVMesh client node was attached to the four logical volumes from a single target class. The result is that each client node is performing I/O with two different target nodes, and each target node is performing I/O with two different client nodes.

Analyzing the test results (Figure 7, Figure 8), several observations can be made:

- I/O resources provided by the NVMesh target nodes are well balanced, as each client server received an even distribution of the performance from the 3-node storage cluster (See Table 5).
- Using a 4 KiB I/O size, a 70%/30% read-to-write ratio workload resulted in over 11 GiB/s of throughput.
- 100% read performance is 4.3X the 100% write performance, which is in line with the specifications of the 9100 SSD when mirroring is considered and each client write I/O results in two SSD write operations (See Table 2: Micron 9100 Specifications).
- 99th percentile latencies are less than 30ms for 70%/30% read/write workload and a RAID-1/0 volume configuration (Figure 8). This is in line with our expectations based on real-world, measured local SSD latency for the 9100 Max SSD and the use of the XFS file system and RAID level at higher queue depths for this workload profile.

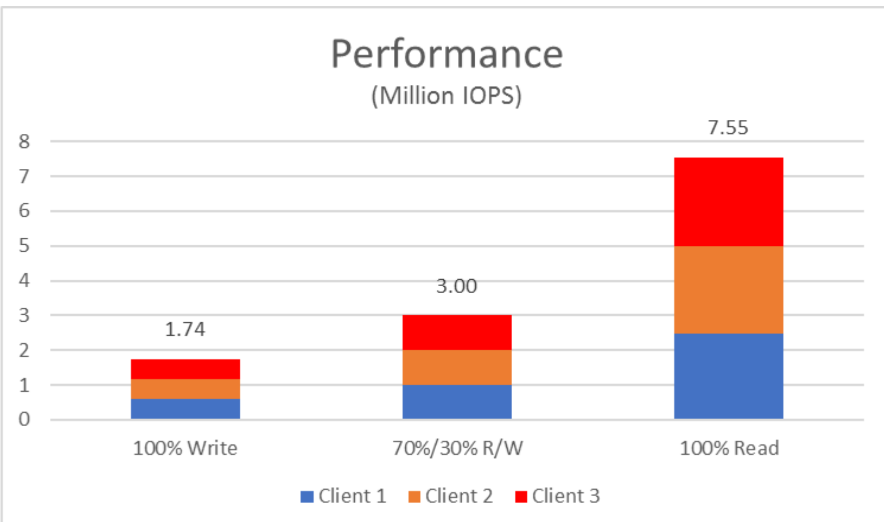

Figure 7: Performance Results for 3-Node Excelero NVMesh Cluster

Read/Write Ratio	Client 1	Client 2	Client 3	Total
0% Read	0.58	0.58	0.58	1.74
70% Read	1.00	1.00	1.00	3.00
100% Read	2.46	2.54	2.56	7.55

Table 5: Performance: 4K Random I/O

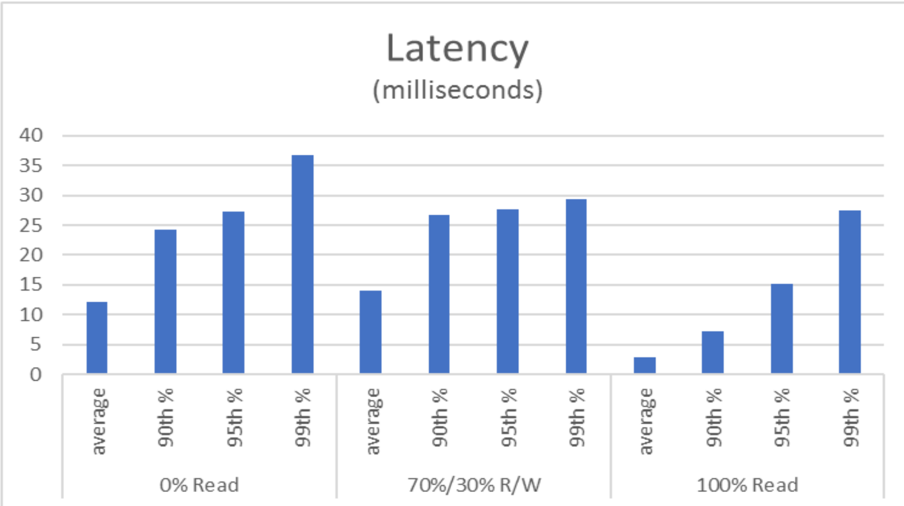

Figure 8: Latency Results for 3-Node Excelero NVMesh Cluster

Read/Write Ratio	Average	90th Percentile	95th Percentile	99th Percentile
0% Read	12.15	24.28	27.32	36.80
70% Read	14.10	26.76	27.69	29.33
100% Read	2.82	7.31	15.10	27.53

Table 6: Latency: 4K Random I/O

An interesting metric that is highlighted by Excelero about the NVMesh SDS is that there is almost no CPU utilization on the NVMesh Target Nodes. To verify this fact, during our testing, we measured CPU utilization. The results are shown in Figure 9.

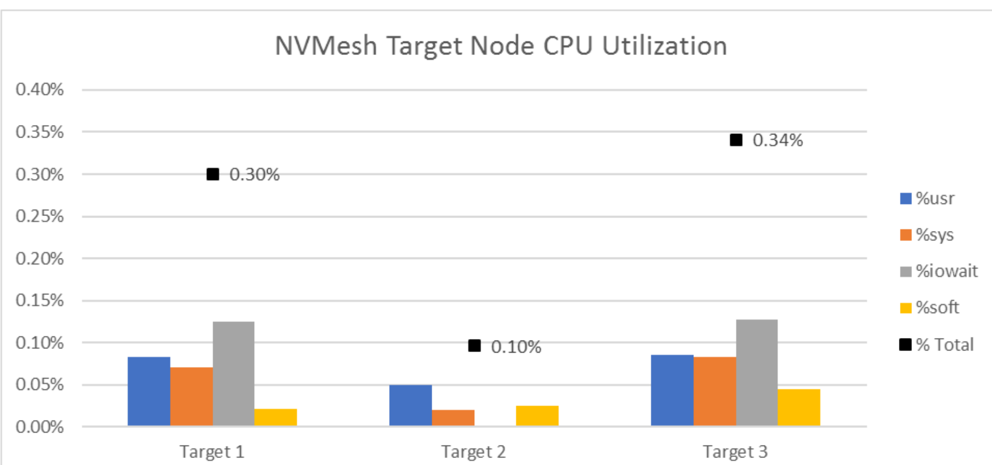

Figure 9: NVMesh Target Node Average CPU Utilization

Based on this CPU data, each NVMesh target node utilizes less than ½ of one percent of CPU capacity to manage 12 NVMe SSDs in that node. This has some interesting implications:

- The actual number of SSDs that can be supported by NVMesh Target Module is only limited by the physical PCIe architecture of the CPU and motherboard.
- NVMesh target nodes can be configured with single CPU sockets and with the lower CPU performance parts, resulting in a dramatic reduction in the cost of the servers used for NVMesh storage.
- Since the NVMesh architecture is based on separate client and target functional software modules, it is possible—and in some specific application use cases recommended—to install the NVMesh client driver onto the same servers running the NVMesh target module and run application workloads on the target nodes as well. While this is beyond the scope of this RA, it is a deployment model that Excelero promotes.

Alternative Hardware Architecture

CPU Selection

Storage-Centric SDS Deployments

Based on the CPU utilization data, for a storage-centric SDS deployment alternative, lower cost servers and CPU “bins” can be used in the NVMesh Target Nodes. By reducing the number of CPU sockets and the number of CPU cores purchased, overall solution costs can be reduced.

As an example, the Supermicro server configured with the Intel Xeon E5-2680v4 used in this RA has a suggested retail price of $41,256[3], while the same server with Intel Xeon E5-2603v4 processors has a suggested retail price of $37,750—a 9% reduction in per node costs.

Depending on the application solution requirements, it is also possible that a completely different CPU architecture—such as AMD’s EPYC™ architecture—could be used to create even more cost reduction benefits as well as for gaining access to additional PCIe lanes.

Compute-Storage Converged Deployments

As mentioned earlier, it is possible to deploy the NVMesh Target Module and the NVMesh Intelligent Client Driver on the same server, thus creating a converged compute and storage solution.

As we have shown, the Target Module does not take any additional CPU cycles to provide NVMe SSD storage services to any application server. All CPU selections should be made based on the requirements for the application(s) being run on the converged servers. Testing should be performed to identify the total CPU load of the application and the NVMesh Client Driver before selecting any specific CPU.

Network Interface Cards

Application servers, in most cases, will not require 200 Gb/s of total bandwidth per server. Alternately, it is acceptable to install lower bandwidth adapters such as:

- Mellanox MCX512A-ACAT 25GbE dual-port
- Mellanox MCX515A-GCAT 50GbE single-port
- Mellanox MCX516A-GCAT 50GbE dual-port

For lower bandwidth applications, the NVMesh Target Nodes can also be configured with the dual-port adapters listed above as alternatives to the two dual-port 100 GbE adapters used in this RA.

Network Switches

While this RA utilized Mellanox Spectrum SN2100 16-port 100 GbE switches, other switches are suitable, such as the other members of the Mellanox SN2000 Series. While Micron has not directly tested another vendor’s offerings, Excelero does provide support for other RoCE-capable Ethernet switch solutions.

[3] Based on pricing at Supermicro reseller. Micron does not endorse any one vendor, but uses this for demonstration purposes only.

Appendix A: Hardware List

Excelero NVMesh Target Nodes (x3)

Component	Description
Server Platform	Supermicro SYS-2028U-TN24R4T+
CPU	2x Intel® Xeon® E5-2680v4 @ 2.4 GHz
Firmware/BIOS/CPLD/BMC	Firmware: 03.47 BIOS: 2.0b Redfish 1.0.1 CPLD: 03.a1.30 BMC: 3.47
Memory	16x 32GB Micron DRAM
Operating System	Centos Linux 7.4 Kernel: 3.10.0-693.2.2.el7.x86_64
Network Interfaces	2x Mellanox MCX516A-CCAT dual-port, 100Gb Ethernet adapters Firmware: 16.20.1010 Driver Package: 4.1-1.0.2
Boot Drive	2x Micron 240GB M510DC SSDs (RAID 1)
Data Drives	12x Micron 2.4TB 9100 MAX SSDs

Application Server Nodes (x3)

Component	Description
Server Platform	Supermicro SYS-1028U-TR4T+ MT018
CPU	2x Intel Xeon E5-2690v4 @ 2.6 GHz
Firmware/BIOS/CPLD/BMC	Firmware: 03.47 BIOS: 2.0b Redfish 1.0.1 CPLD: 03.a1.32 BMC: 3.47
Memory	16x 16GB Micron DRAM
Operating System	Centos Linux 7.4 Kernel: 3.10.0-693.2.2.el7.x86_64
Network Interfaces	2x Mellanox CX515A-CCAT single-port, 100Gb Ethernet adapters Firmware: 14.20.1010 Driver Package: 4.1-1.0.2
Boot Drive	2x Micron 960GB M510DC SSDs (RAID 1)

Storage Network Switches

Component	Description
Switch Model	2x Mellanox Spectrum SN2100 16 port 100Gb Ethernet
Firmware	3.6.4112

Excelero NVMesh Software

Component	Description
NVMesh Intelligent Client Block Driver	1.2.0 Build 855
NVMesh Target Driver	1.2.0 Build 855
NVMesh Management	1.2.0 Build 142

Appendix B: Server BIOS Configuration Settings

The following sections describe the non-default BIOS configuration settings used for this solution. Consult the Supermicro documentation for specifics on entering and saving BIOS configuration information.

Excelero NVMesh Target Nodes

Category	Section	Setting	Value
Performance	Advanced => Boot Features	Restore on AC power loss	**Stay Off**
	Advanced => CPU Configuration => Advanced Power Management Configuration	Power Technology	**Disable**
		Energy Performance Bias	**Performance**
		Energy Efficient Turbo	**Disable**
	Advanced => PCIe/PCI/PnP	Maximum Payload	**256 Bytes**
		Maximum Read Request	**4096 Bytes**
		OPROM	**EFI**
IPMI	BMC Network Configuration	IPMI LAN Configuration	Yes
		Station IP Address	Enter IP address compatible with management network. Our example uses **172.016.132.102**
		Subnet Mask	Enter subnet mask suitable with target management network. Our example uses **255.255.252.000**
		Gateway IP Address	Enter suitable gateway address for target management network. Our example uses **172.016.132.001**
RAID (Boot)	Advanced => sSATA	sSATA RAID as	**RAID**
		sSATA RAID Option ROM/UEFI	**EFI**
		sSATA Device Type	**Solid State Drive**
	Advanced => Intel RSTe sSATA Controller => Create RAID Volume	Name	**OSBoot**
		RAID Level	**RAID1 (MIRROR)**
	Boot	Boot Mode Select	**UEFI**
		Legacy to EFI support	**Disabled**
		Boot Order	UEFI Boot Order #1 = UEFI Hard Disk UEFI Boot Order #2 = UEFI USB Key UEFI Boot Order #3 = UEFI USB Hard Disk All others should be disabled.

Application Server Nodes (SYS-1028U-TR4T)

Category	Section	Setting	Value
Performance	Advanced => Boot Features	Restore on AC power loss	**Stay Off**
	Advanced => CPU Configuration => Advanced Power Management Configuration	Power Technology	**Disable**
		Energy Performance Bias	**Performance**
		Energy Efficient Turbo	**Disable**
	Advanced => PCIe/PCI/PnP	Maximum Payload	**256 Bytes**
		Maximum Read Request	**4096 Bytes**
		OPROM	**EFI**
IPMI	BMC Network Configuration	IPMI LAN Configuration	**Yes**
		Station IP Address	Enter IP Address compatible with management network. Our example uses **172.016.132.102**
		Subnet Mask	Enter subnet mask suitable with target management network. Our example uses **255.255.252.000**
		Gateway IP Address	Enter suitable gateway address for target management network. Our example uses **172.016.132.001**
RAID (Boot)	Advanced => sSATA	sSATA RAID as	**RAID**
		sSATA RAID Option ROM/UEFI	**EFI**
		sSATA Device Type	**Solid State Drive**
	Advanced => Intel RSTe sSATA Controller => Create RAID Volume	Name	**OSBoot**
		RAID Level	**RAID1 (MIRROR)**
	Boot	Boot Mode Select	**UEFI**
		Legacy to EFI support	**Disabled**
		Boot Order	UEFI Boot Order #1 = UEFI Hard Disk UEFI Boot Order #2 = UEFI USB Key UEFI Boot Order #3 = UEFI USB Hard Disk All others should be disabled.

Appendix C: Network Infrastructure Configuration Settings

VLAN Configuration Information

Category	Switch 1	Switch 2
VLAN	VLAN 100: • ID: 100 • Network: 172.31.100.1/24 • Ports: 1/1, 1/3, 1/5, 1/7, 1/9 VLAN 200: • ID: 200 • Network: 172.31.200.1/24 • Ports: 1/2, 1/4, 1/6, 1/8	VLAN 150: • ID: 150 • Network: 172.31.150.1/24 • Ports: 1/1, 1/3, 1/5, 1/7, 1/9 VLAN 250: • ID: 250 • Network: 172.31.250.1/24 • Ports: 1/2, 1/4, 1/6, 1/8

Switch Configuration Settings

Category	Switch 1 (Left)	Switch 2 (Right)
General	no cli default prefix-modes enable	no cli default prefix-modes enable
Interface	interface port-channel 1 interface ethernet 1/1–1/9 mtu 9216 force interface ethernet 1/13–1/16 mtu 9216 force interface port-channel 1 mtu 9216 force interface ethernet 1/1–1/9 flowcontrol receive on force interface ethernet 1/1–1/9 flowcontrol send on force interface ethernet 1/13–1/16 flowcontrol receive on force interface ethernet 1/13–1/16 flowcontrol send on force interface port-channel 1 flowcontrol receive on force interface port-channel 1 flowcontrol send on force interface ethernet 1/1–1/9 switchport mode trunk interface ethernet 1/13–1/16 switchport mode trunk interface port-channel 1 switchport mode trunk interface ethernet 1/13-1/16 channel-group 1 mode active interface port-channel 1 switchport mode trunk	interface port-channel 1 interface ethernet 1/1–1/9 mtu 9216 force interface ethernet 1/13–1/16 mtu 9216 force interface port-channel 1 mtu 9216 force interface ethernet 1/1–1/9 flowcontrol receive on force interface ethernet 1/1–1/9 flowcontrol send on force interface ethernet 1/13–1/16 flowcontrol receive on force interface ethernet 1/13–1/16 flowcontrol send on force interface port-channel 1 flowcontrol receive on force interface port-channel 1 flowcontrol send on force interface ethernet 1/1–1/9 switchport mode trunk interface ethernet 1/13–1/16 switchport mode trunk interface port-channel 1 switchport mode trunk interface ethernet 1/13-1/16 channel-group 1 mode active interface port-channel 1 switchport mode trunk
LACP	lacp port-channel load-balance ethernet source-destination-ip source-destination-mac	lacp port-channel load-balance ethernet source-destination-ip source-destination-mac

VLAN	vlan 100 vlan 200 vlan 100 name "VLAN 100" vlan 200 name "VLAN 200" interface ethernet 1/1–1/9 switchport trunk allowed-vlan none interface ethernet 1/1,1/3,1/5,1/7,1/9 switchport trunk allowed-vlan 100 interface ethernet 1/2,1/4,1/6,1/8 switchport trunk allowed-vlan 200 mac-address-table aging-time 1800	vlan 150 vlan 250 vlan 150 name "VLAN 150" vlan 250 name "VLAN 250" interface ethernet 1/1–1/9 switchport trunk allowed-vlan none interface ethernet 1/1,1/3,1/5,1/7,1/9 switchport trunk allowed-vlan 150 interface ethernet 1/2,1/4,1/6,1/8 switchport trunk allowed-vlan 250 mac-address-table aging-time 1800
Spanning Tree	no spanning-tree	no spanning-tree
L3 Routing	ip routing vrf default interface vlan 100 interface vlan 150 interface vlan 200 interface vlan 150 interface vlan 100 ip address 172.31.100.1 255.255.255.0 interface vlan 100 mtu 9216 interface vlan 200 ip address 172.31.200.1 255.255.255.0 interface vlan 200 mtu 9216 ip route vrf default 172.31.150.0 /24 10.0.0.2 ip route vrf default 172.31.250.0 /24 10.0.0.2	ip routing vrf default interface vlan 100 interface vlan 150 interface vlan 200 interface vlan 150 interface vlan 150 ip address 172.31.150.1 255.255.255.0 interface vlan 150 mtu 9216 interface vlan 250 ip address 172.31.250.1 255.255.255.0 interface vlan 250 mtu 9216 ip route vrf default 172.31.100.0 /24 10.0.0.2 ip route vrf default 172.31.200.0 /24 10.0.0.2
LLDP	lldp	lldp
Management Network	no interface mgmt0 dhcp no interface mgmt0 dhcp hostname interface mgmt0 ip address 172.16.135.94/22 no interface mgmt0 ipv6 address autoconfig default no interface mgmt0 ipv6 enable	no interface mgmt0 dhcp no interface mgmt0 dhcp hostname interface mgmt0 ip address 172.16.135.93/22 no interface mgmt0 ipv6 address autoconfig default no interface mgmt0 ipv6 enable
Miscellaneous	ip route 0.0.0.0 0.0.0.0 172.16.132.1 no ipv6 enable clock timezone America North United_States Central web http redirect cli default prefix-modes enable	ip route 0.0.0.0 0.0.0.0 172.16.132.1 no ipv6 enable clock timezone America North United_States Central web http redirect cli default prefix-modes enable

Appendix D: Excelero Target Node Network Configuration Settings

Management Interfaces Configuration

Location: /etc/sysconfig/network-scripts/

Config File	Target Node 1	Target Node 2	Target Node 3
Ifcfg-enp1s0f0	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp1s0f0 DEVICE=enp1s0f0 IPADDR=172.16.135.115 PREFIX=22	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp1s0f0 DEVICE=enp1s0f0 IPADDR=172.16.135.116 PREFIX=22	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp1s0f0 DEVICE=enp1s0f0 IPADDR=172.16.135.117 PREFIX=22
rule-enp1s0f0	from 172.16.135.115/30 table lom1 to 172.16.135.115 table lom1	from 172.16.135.116/30 table lom1 to 172.16.135.116 table lom1	from 172.16.135.117/30 table lom1 to 172.16.135.117 table lom1
route-enp1s0f0	172.16.132.0/22 via 172.16.135.115 table lom1 default via 172.16.132.1	172.16.132.0/22 via 172.16.135.116 table lom1 default via 172.16.132.1	172.16.132.0/22 via 172.16.135.117 table lom1 default via 172.16.132.1

Storage Network Interfaces Configuration

Location: /etc/sysconfig/network-scripts/

Config File	Target Node 1	Target Node 2	Target Node 3
ifcfg-ens1f1	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f1 DEVICE=ens1f1 MTU=4200		
ifcfg-ens1f1.150	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f1.150 DEVICE=ens1f1.150 IPADDR=172.31.150.11 PREFIX=24 MTU=4200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f1.150 DEVICE=ens1f1.150 IPADDR=172.31.150.12 PREFIX=24 MTU=4200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f1.150 DEVICE=ens1f1.150 IPADDR=172.31.150.13 PREFIX=24 MTU=4200

Config File	Target Node 1	Target Node 2	Target Node 3
route-ens1f1.150	172.31.100.0/24 dev ens1f1.150 table roce3 172.31.250.0/24 via 172.31.150.1 table roce3 172.31.150.0/24 via 172.31.150.1 table roce3 172.31.200.0/24 via 172.31.150.1 table roce3		
rule-ens1f1.150	from 172.31.150.11/32 table roce3 prio 303 to 172.31.100.0/24 table roce3 prio 403 to 172.31.250.0/24 table roce3 prio 503 to 172.31.150.0/24 table roce3 prio 603 to 172.31.200.0/24 table roce3 prio 703	from 172.31.150.12/32 table roce3 prio 303 to 172.31.100.0/24 table roce3 prio 403 to 172.31.250.0/24 table roce3 prio 503 to 172.31.150.0/24 table roce3 prio 603 to 172.31.200.0/24 table roce3 prio 703	from 172.31.150.13/32 table roce3 prio 303 to 172.31.100.0/24 table roce3 prio 403 to 172.31.250.0/24 table roce3 prio 503 to 172.31.150.0/24 table roce3 prio 603 to 172.31.200.0/24 table roce3 prio 703
ifcfg-ens1f0	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0 DEVICE=ens1f0 MTU=4200		
ifcfg-ens1f0.200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f1.200 DEVICE=ens1f0.200 IPADDR=172.31.200.11 PREFIX=24 MTU=4200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0.200 DEVICE=ens1f0.200 IPADDR=172.31.200.12 PREFIX=24 MTU=4200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0.200 DEVICE=ens1f0.200 IPADDR=172.31.200.13 PREFIX=24 MTU=4200
route-ens1f0.200	172.31.200.0/24 dev ens1f0.200 table roce1 172.31.200.0/24 via 172.31.200.1 table roce1 172.31.100.0/24 via 172.31.200.1 table roce1 172.31.250.0/24 via 172.31.200.1 table roce1		

Config File	Target Node 1	Target Node 2	Target Node 3
rule-ens1f0.200	from 172.31.200.11/32 table roce1 prio 301 to 172.31.150.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.100.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701	from 172.31.150.12/32 table roce1 prio 301 to 172.31.150.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.100.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701	from 172.31.150.13/32 table roce1 prio 301 to 172.31.150.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.100.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701
ifcfg-enp143s0f1	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp143s0f1 DEVICE=enp143s0f1 MTU=4200		
ifcfg-enp143s0f1.100	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp143s0f1.100 DEVICE=enp143s0f1.100 IPADDR=172.31.100.11 PREFIX=24 MTU=4200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp143s0f1.100 DEVICE=enp143s0f1.100 IPADDR=172.31.100.12 PREFIX=24 MTU=4200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp143s0f1.100 DEVICE=enp143s0f1.100 IPADDR=172.31.100.13 PREFIX=24 MTU=4200
route-enp143s0f1.100	from 172.31.100.11/32 table roce4 prio 304 to 172.31.200.0/24 table roce4 prio 404 to 172.31.150.0/24 table roce4 prio 504 to 172.31.250.0/24 table roce4 prio 604 to 172.31.100.0/24 table roce4 prio 704		
rule-enp143s0f1.100	from 172.31.200.11/32 table roce1 prio 301 to 172.31.150.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.100.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701	from 172.31.150.12/32 table roce1 prio 301 to 172.31.150.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.100.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701	from 172.31.150.13/32 table roce1 prio 301 to 172.31.150.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.100.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701

Config File	Target Node 1	Target Node 2	Target Node 3
ifcfg-enp143s0f0	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp143s0f0 DEVICE=enp143s0f0 MTU=4200		
ifcfg-enp143s0f0.250	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp143s0f1.250 DEVICE=enp143s0f1.250 IPADDR=172.31.250.11 PREFIX=24 MTU=4200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp143s0f1.250 DEVICE=enp143s0f1.250 IPADDR=172.31.250.12 PREFIX=24 MTU=4200	TYPE=Ethernet VLAN=yes BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp143s0f1.250 DEVICE=enp143s0f1.250 IPADDR=172.31.250.13 PREFIX=24 MTU=4200
route-enp143s0f0.250	172.31.250.0/24 dev enp143s0f0.250 table roce2 172.31.100.0/24 via 172.31.250.1 table roce2 172.31.200.0/24 via 172.31.250.1 table roce2 172.31.150.0/24 via 172.31.250.1 table roce2		
rule-enp143s0f0.250	from 172.31.250.11/32 table roce2 prio 302 to 172.31.250.0/24 table roce2 prio 402 to 172.31.100.0/24 table roce2 prio 502 to 172.31.200.0/24 table roce2 prio 602 to 172.31.150.0/24 table roce2 prio 702	from 172.31.250.12/32 table roce2 prio 302 to 172.31.250.0/24 table roce2 prio 402 to 172.31.100.0/24 table roce2 prio 502 to 172.31.200.0/24 table roce2 prio 602 to 172.31.150.0/24 table roce2 prio 702	from 172.31.250.13/32 table roce2 prio 302 to 172.31.250.0/24 table roce2 prio 402 to 172.31.100.0/24 table roce2 prio 502 to 172.31.200.0/24 table roce2 prio 602 to 172.31.150.0/24 table roce2 prio 702

Location: /etc/sysctl.d

Config File	Target Node 1	Target Node 2	Target Node 3
50-roce-multipath.conf	net.ipv4.conf.all.arp_ignore=2 net.ipv4.conf.all.arp_notify=1 net.ipv4.conf.all.arp_announce=2 net.ipv6.conf.all.disable_ipv6=1 net.ipv6.conf.default.disable_ipv6=1 net.ipv4.conf.ens1f0/200.rp_filter=2 net.ipv4.conf.enp143s0f0/250.rp_filter=2 net.ipv4.conf.ens1f1/150.rp_filter=2 net.ipv4.conf.enp143s0f1/100.rp_filter=2		

Appendix E: Application Server (Excelero Storage Client) Configuration Settings

Management Interface Configuration

Location: /etc/sysconfig/network-scripts/

Config File	Client Node 1	Client Node 2	Client Node 3
Ifcfg-enp1s0f0	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp1s0f0 DEVICE=enp1s0f0 IPADDR=172.16.135.111 PREFIX=22	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp1s0f0 DEVICE=enp1s0f0 IPADDR=172.16.132.112 PREFIX=22	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp1s0f0 DEVICE=enp1s0f0 IPADDR=172.16.135.113 PREFIX=22
rule-enp1s0f0	from 172.16.135.111/30 table lom1 to 172.16.135.111 table lom1	from 172.16.135.112/30 table lom1 to 172.16.135.112 table lom1	from 172.16.135.113/30 table lom1 to 172.16.135.113 table lom1
route-enp1s0f0	172.16.132.0/22 via 172.16.135.111 table lom1 default via 172.16.132.1	172.16.132.0/22 via 172.16.135.112 table lom1 default via 172.16.132.1	172.16.132.0/22 via 172.16.135.113 table lom1 default via 172.16.132.1

Storage Network Configuration

Because there are four separate VLANs, clients should be distributed such that some use VLANs 100 and 150 and other clients use VLANs 200 and 250. Below are two sections that describe the two client configurations.

Client Config 1 (VLANs 100 & 150)

Location: /etc/sysconfig/network-scripts/

Config File	Client Node 1	Client Node 3
ifcfg-ens1f0	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0 DEVICE=ens1f0 MTU=4200	

Config File	Client Node 1	Client Node 3
ifcfg-ens1f0.100	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0.100 DEVICE=ens1f0.100 IPADDR=172.16.100.21 PREFIX=22	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp1s0f0.100 DEVICE=enp1s0f0.100 IPADDR=172.16.100.23 PREFIX=22
route-ens1f0.100	172.31.100.0/24 dev ens1f0.100 table roce1 172.31.200.0/24 via 172.31.100.1 table roce1 172.31.150.0/24 via 172.31.100.1 table roce1 172.31.250.0/24 via 172.31.100.1 table roce1	
rule-ens1f0.100	from 172.31.100.21/32 table roce1 prio 301 to 172.31.100.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.150.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701	from 172.31.100.23/32 table roce1 prio 301 to 172.31.100.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.150.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701
ifcfg-ens2f0	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0 DEVICE=ens1f0 MTU=4200	
ifcfg-ens2f0.150	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens2f0 DEVICE=ens2f0 IPADDR=172.16.150.21 PREFIX=22	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=enp2s0f0 DEVICE=enp2s0f0 IPADDR=172.16.150.23 PREFIX=22
route-ens2f0.150	172.31.150.0/24 dev ens2f0.150 table roce4 172.31.250.0/24 via 172.31.150.1 table roce4 172.31.200.0/24 via 172.31.150.1 table roce4 172.31.100.0/24 via 172.31.150.1 table roce4	

Config File	Client Node 1	Client Node 3
rule-ens2f0.150	from 172.31.150.21/32 table roce4 prio 304 to 172.31.250.0/24 table roce4 prio 404 to 172.31.150.0/24 table roce4 prio 504 to 172.31.200.0/24 table roce4 prio 604 to 172.31.100.0/24 table roce4 prio 704	from 172.31.150.23/32 table roce4 prio 304 to 172.31.250.0/24 table roce4 prio 404 to 172.31.150.0/24 table roce4 prio 504 to 172.31.200.0/24 table roce4 prio 604 to 172.31.100.0/24 table roce4 prio 704

Location: /etc/sysctl.d

Config File	Client Node 1	Client Node 3
50-roce-multipath.conf	net.ipv4.conf.all.arp_ignore=2 net.ipv4.conf.all.arp_notify=1 net.ipv4.conf.all.arp_announce=2 net.ipv6.conf.all.disable_ipv6=1 net.ipv6.conf.default.disable_ipv6=1 net.ipv4.conf.ens1f0/100.rp_filter=2 net.ipv4.conf.ens1f1/150.rp_filter=2	

Client Config 2 (VLANs 200 & 250)

Location: /etc/sysconfig/network-scripts/

Config File	Client Node 2
ifcfg-ens1f0	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0 DEVICE=ens1f0 MTU=4200
ifcfg-ens1f0.200	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0.200 DEVICE=ens1f0.200 IPADDR=172.16.200.22 PREFIX=22

Config File	Client Node 2
route-ens1f0.200	172.31.200.0/24 dev ens1f0.200 table roce1 172.31.100.0/24 via 172.31.200.1 table roce1 172.31.150.0/24 via 172.31.200.1 table roce1 172.31.250.0/24 via 172.31.200.1 table roce1
rule-ens1f0.200	from 172.31.200.22/32 table roce1 prio 301 to 172.31.100.0/24 table roce1 prio 401 to 172.31.200.0/24 table roce1 prio 501 to 172.31.150.0/24 table roce1 prio 601 to 172.31.250.0/24 table roce1 prio 701
ifcfg-ens2f0	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens1f0 DEVICE=ens1f0 MTU=4200
ifcfg-ens2f0.250	TYPE=Ethernet BOOTPROTO=none IPV6INIT=no ONBOOT=yes NAME=ens2f0 DEVICE=ens2f0 IPADDR=172.16.250.22 PREFIX=22
route-ens2f0.250	172.31.250.0/24 dev ens2f0.250 table roce4 172.31.150.0/24 via 172.31.250.1 table roce4 172.31.200.0/24 via 172.31.250.1 table roce4 172.31.100.0/24 via 172.31.250.1 table roce4
rule-ens2f0.250	from 172.31.250.22/32 table roce4 prio 304 to 172.31.250.0/24 table roce4 prio 404 to 172.31.150.0/24 table roce4 prio 504 to 172.31.200.0/24 table roce4 prio 604 to 172.31.100.0/24 table roce4 prio 704

Location: /etc/sysctl.d

Config File	Client Node 2
50-roce-multipath.conf	net.ipv4.conf.all.arp_ignore=2 net.ipv4.conf.all.arp_notify=1 net.ipv4.conf.all.arp_announce=2 net.ipv6.conf.all.disable_ipv6=1 net.ipv6.conf.default.disable_ipv6=1 net.ipv4.conf.ens1f0/200.rp_filter=2 net.ipv4.conf.ens1f1/250.rp_filter=2

Appendix F: MongoDB Configuration

The following tables describe the configuration settings to be modified on each Excelero Target node using the `mongod.conf` file located at `/etc/`.

Target Node 1

```
# mongod.conf
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1,172.16.135.115  # Listen to local interface only, comment to listen on all interfaces.
replication:
  replSetName: rs0
```

Target Node 2

```
# mongod.conf
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log
```

Target Node 2

```
# Where and how to store data.
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1,172.16.135.116  # Listen to local interface only, comment to listen on all interfaces.
replication:
  replSetName: rs0
```

Target Node 3

```
# mongod.conf
# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1,172.16.135.117  # Listen to local interface only, comment to listen on all interfaces.
replication:
  replSetName: rs0
```

Appendix G: NVMesh Configuration

Client Nodes

Location: /etc/opt/NVMesh/nvmesh.conf

All Client Nodes

```
# NVMesh configuration file
# This configuration file is utilized by Excelero NVMesh(tm) applications for various options.
# Define the management protocol
# MANAGEMENT_PROTOCOL="<https/http>"
# Example
# MANAGEMENT_PROTOCOL="https"
MANAGEMENT_PROTOCOL="https"
# Define the location of the NVMesh Management Websocket servers
# MANAGEMENT_SERVERS="<server name or IP>:<port>,<server name or IP>:<port>,..."
# Example:
# MANAGEMENT_SERVERS="nvmesh-management1:4001,nvmesh-management2:4001"
MANAGEMENT_SERVERS="172.16.135.115:4001,172.16.135.116:4001,172.16.135.117:4001"
# Define the nics that will be available for NVMesh Client/Target to work with
# CONFIGURED_NICS="<interface name;interface name;...>"
# To allow all nics to be available leave empty. Example: CONFIGURED_NICS=""
# Example:
# CONFIGURED_NICS="ib0;eno1;eth0"
CONFIGURED_NICS="ens1f0;ens1f1"
# Used to set the maximum burst size of queries to the IB Session Manager. This parameter is not relevant for
RoCE.
# A smaller number here will decrease the load on the SM, but will increase the initial bring-up time.
# Example:
# MAX_SM_QUERY_BURST="32"
MAX_SM_QUERY_BURST="32"
# Used to enable RDDA for ConnectX-4 or ConnectX-5 adapters
# MLX5_RDDA_ENABLED="<Yes/No>"
# Example:
# MLX5_RDDA_ENABLED="No"
MLX5_RDDA_ENABLED="Yes"
# Used to dump fast log (ftrace) buffer on kernel panic
# DUMP_FTRACE_ON_OOPS="<Yes/No>"
# Example:
# DUMP_FTRACE_ON_OOPS="Yes"
DUMP_FTRACE_ON_OOPS="Yes"
# Used to determine whether previously attached volumes should be attached on restart
# AUTO_ATTACH_VOLUMES="<Yes/No>"
# Example:
# AUTO_ATTACH_VOLUMES="No"

AUTO_ATTACH_VOLUMES="Yes"
```

Target Nodes

Location: /etc/opt/NVMesh/nvmesh.conf

All Target Nodes

```
MANAGEMENT_PROTOCOL="https"
MANAGEMENT_SERVERS="172.16.135.115:4001,172.16.135.116:4001,172.16.135.117:4001"
CONFIGURED_NICS="ens1f0;ens1f1"
MAX_SM_QUERY_BURST="32"
MLX5_RDDA_ENABLED="Yes"
DUMP_FTRACE_ON_OOPS="Yes"
AUTO_ATTACH_VOLUMES="Yes"
```

Location: /etc/opt/NVMesh/management.js.conf

All Target Nodes

```
var config = {};
config.loggingLevel = "INFO"; /*Possible values are: DEBUG, INFO, WARNING, ERROR*/
config.productionEnvironment = false;
//SSL Configuration
config.useSSL = true;
config.cert = 'cert/server.crt';
config.key = 'cert/server.key';
config.port = 4000;
config.webSocketServerPort = 4001;
//SMTP configuration.
config.SMTP = {
    host: 'localhost',
    port: 25,
    secure: false,
    authRequired: false,
    username: '<username@gmail.com>',
    password: '<password>',
    //Determine whether to use gmail SMTP.
    useDefault: true
}
config.mongoConnection = {
    hosts: '172.16.135.115:27017,172.16.135.116:27017,172.16.135.117:27017', /* Example for replica set
connection: nvme21:27017,nvme31:27017,nvme23:27017 */
    options: {
        replicaSetName: 'rs0'
    }
}
config.mongoConnectOptions = {
```

```
        server: {
                socketOptions: {
                        keepAlive: 1,
                        connectTimeoutMS: 30000,  // 30 secs
                        socketTimeoutMS: 30000,   // 30 secs
                }
        }
};
config.websocket = {
        serverConfig: {
                keepalive: true,
                keepaliveInterval: 2000,
                dropConnectionOnKeepaliveTimeout: true,
                keepaliveGracePeriod: 3000,
                closeTimeout: 3000,
                outOfSyncInterval: 5000,
                outOfsyncThreshold: 5000
        },
        clientConfig: {
                closeTimeout: 3000
        },
        keepAliveTimeout: 5000,
        keepAliveInterval: 2000,
        maxReconnectAttempts: 1000,
        reconnectAttemptInterval: 3000
};
//Phone home email
config.exceleroEmail = 'support+customerName@excelero.com'
config.sendStatsInterval = 1000*3600*24*7 //millis*seconds*hours*days (1week)
config.RESERVED_BLOCKS = 0.5;
config.MAX_JSON_SIZE = '2mb';
//Determine whether to run with ldlinux. RPM should contain the `libraries` dir.
config.compatibilityMode = false;
//Backup configuration
config.Backup = {
        backupPath : "/var/opt/NVMesh/backups",
        minutelyBackupInterval: 5,
        dailyBackupTime: "00:00",
        minutelyRotationThreshold: 36,
        hourlyRotationThreshold: 36,
        dailylRotationThreshold: 30
}
module.exports = config;
```

Glossary

Term	Definition
Management Server	The server or OS image instance running the NVMesh management module software
Target Node/Target	A physical server containing one or more NVMe SSDs running the storage target module
Client Node/Client	An OS image instance running the block storage client software
Converged Node	A target node that is also running the block storage client software
Logical Volume/Volume	A logical block device defined with the NVMesh management module
RDDA	Remote Direct Drive Access. Excelero's patent-pending low-latency and CPU bypass transport technology.
TOMA	[To]pology [Ma]nager. The storage target module component that handles error detection and volume rebuild activities.

Appendix H: About the Test(s) Used

Flexible IO (FIO)

FIO is an open source I/O workload generator originally developed by Jens Axboe in 2005. FIO can generate almost any desired workload profile based on criteria such as block size, read/write mix, queue depth, etc., as well as scale out by spawning multiple instances, and is manageable across multiple nodes using standard Dev/Ops scripting and tools. In addition, it provides a flexible set of output options that ensures resulting performance data can be easily processed and communicated for maximum results.

FIO Test Script Settings
[global] rw=randrw numjobs=28 blocksize=4k iodepth=64 rwmixread=0 ioengine=libaio direct=1 refill_buffers norandommap randrepeat=0 group_reporting [RAND_VOL1] filesize=3200G filename=/mnt/vol1/file runtime=600 time_based [RAND_VOL2] filesize=3200G filename=/mnt/vol2/file runtime=600 time_based [RAND_VOL3] filesize=3200G filename=/mnt/vol3/file runtime=600 time_based [RAND_VOL4] filesize=3200G filename=/mnt/vol4/file runtime=600 time_based

Appendix I: About

Micron

Micron Technology (Nasdaq: MU) is a world leader in innovative memory solutions. Through our global brands—Micron, Crucial® and Ballistix®—our broad portfolio of high-performance memory technologies, including DRAM, NAND, NOR Flash and 3D XPoint™ memory, is transforming how the world uses information. Backed by more than 35 years of technology leadership, Micron's memory solutions enable the world's most innovative computing, consumer, enterprise storage, data center, mobile, embedded, and automotive applications. Micron's common stock is traded on the Nasdaq under the MU symbol. To learn more about Micron Technology, Inc., visit micron.com.

Excelero

Founded in 2014 by a team of storage veterans and inspired by the Tech Giants' shared-nothing architectures for web-scale applications, the company has created a Software-Defined Block Storage solution that meets performance and scalability requirements of the largest web-scale and enterprise applications.

With Excelero's NVMesh, customers can build distributed, high-performance Server SAN for mixed application workloads. Customers benefit from the performance of local flash, with the convenience of centralized storage while avoiding proprietary hardware lock-in and reducing the overall storage TCO. The solution has been deployed for hyper-scale Industrial IoT services, machine learning applications and massive-scale simulation visualization. For more information visit www.excelero.com.

Mellanox

Mellanox Technologies (NASDAQ: MLNX) is a leading supplier of end-to-end Ethernet and InfiniBand Intelligent interconnect solutions and services for servers, storage, and hyper-converged infrastructure. Mellanox interconnect solutions increase data center efficiency by providing the highest throughput and lowest latency, delivering data faster to applications and unlocking system performance. Mellanox offers a choice of high performance solutions: network and multicore processors, network adapters, switches, cables, software and silicon, that accelerate application runtime and maximize business results for a wide range of markets including high performance computing, enterprise data centers, Web 2.0, cloud, storage, network security, telecom and financial services. More information is available at: www.mellanox.com.

Supermicro

Supermicro (NASDAQ: SMCI), a leading innovator in high-performance, high-efficiency server technology is a premier provider of advanced server Building Block Solutions® for Data Center, Cloud Computing, Enterprise IT, Hadoop/Big Data, HPC and Embedded Systems worldwide. Supermicro is committed to protecting the environment through its "We Keep IT Green®" initiative and provides customers with the most energy-efficient, environmentally-friendly solutions available on the market.

micron.com

Rev. A 3/18 CCM004-676576390-11003