January 2017

Demartek

Demartek RoCE Deployment Guide

Produced with the InfiniBand Trade Association in support of the RoCE Initiative.

INFINIBAND℠ TRADE ASSOCIATION

RoCE™

Executive Summary

Data center managers and administrators face increasing pressure to accelerate application performance because data continues to grow, and demand for quick access to this data continues to intensify.

The computing industry continues to develop new technologies such as flash storage and more powerful processors to keep pace with these demands. Product design engineers are also incorporating performance acceleration techniques into network adapters and switches.

One of these networking adapter technologies, Remote Direct Memory Access (RDMA) has been built into special network adapters that use RDMA over Converged Ethernet, or RoCE technology. This performance acceleration technology is built into specialized network adapters that work with a streamlined software stack in the operating system to improve performance and lower the host CPU consumption required to process network traffic.

This guide is designed for managers and technical professionals within IT departments who are exploring the benefits of deploying RoCE technology or who are looking for practical guidance and deployment examples of RoCE solutions.

As RoCE includes server, adapter and switching technologies, this guide provides information and guidance in each area. A basic understanding of each of these areas is needed to successfully deploy RoCE technology.

This guide is intended to be used as a reference and includes screen shots and information from actual deployments of specific RoCE-enabled products.

All of the work was performed in the Demartek lab in Golden, Colorado, USA.

Demartek RoCE Deployment Guide

RoCE Basic Definitions

RDMA – Remote Direct Memory Access - the remote memory management capability that allows server-to-server data movement directly between the application memory of each without any CPU involvement.

RoCE – (pronounced "Rocky") RDMA over Converged Ethernet – a network protocol that allows remote direct memory access over a converged Ethernet network.

Converged Ethernet – A term associated with a collection of standards-based extensions to traditional Ethernet that provide lossless characteristics that enable the convergence of LAN and SAN technology onto a single unified fabric.

RoCE Initiative

The RoCE Initiative is an education and resource program of the InfiniBand Trade Association (IBTA) that is committed to increasing awareness of RoCE by providing technical education and reference solutions for high-performance Ethernet topologies in traditional and cloud-based data centers.

Additional information is available on the RoCE Initiative website including white papers, solution briefs and contact information.

Use of Bookmarks

This document uses a feature known as "bookmarks" that can be used for navigation within this document. Enable the display of bookmarks in your PDF reader so that you can navigate to any section of this document by clicking on the bookmark entry.

About Demartek

Demartek is a computer industry analyst organization with its own ISO 17025 accredited computer test lab. We provide real-world, hands-on research and analysis by focusing on lab validation testing and power efficiency testing of data center computer equipment such as servers, networking and storage systems. We produce white papers and videos that provide the results of our tests along with our commentary. Additional information regarding our services is available on our lab testing page.

To be notified when new Deployment Guides and lab validation reports become available, you can subscribe to our free monthly newsletter, ***Demartek Lab Notes***, available on our website. We do not give out, rent or sell our email list.

RoCE and RDMA vs. Traditional TCP/IP

Remote Direct Memory Access (RDMA) enables more direct movement of data in and out of a server. RDMA bypasses the normal system software network stack components and the multiple buffer copy operations that they normally perform. This is displayed in the ***Standard TCP/IP vs. RDMA Data Movement*** diagram below that shows a standard network connection on the left and an RDMA connection on the right.

This elimination of buffer copy operations reduces overall CPU consumption and improves (lowers) the latency of the host software stack, since it uses fewer instructions to complete a data transfer.

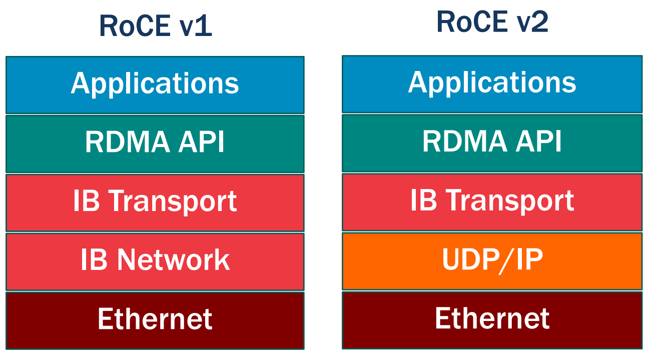
RDMA can be used in Ethernet applications by using specialized adapters that support RDMA. The adapters are sometimes known as RNICs or RDMA network interface cards (NICs).

RDMA can be implemented for networking and/or storage applications. Some RDMA adapters may have offload functions built into them. The initiator and the target adapters must use the same type of RDMA technology, such as RoCE v1, RoCE v2, etc.

RDMA can be used for storage traffic such as block protocols like iSCSI and file protocols such as NFS and SMB (formerly known as CIFS). RDMA can improve the latency of storage access and reduce CPU consumption of the host server performing I/O requests, allowing a higher rate of I/O requests to be sustained, or perhaps a smaller server to perform the same rate of I/O requests.

RoCE v1 vs. RoCE v2

The original implementation of RoCE, known as "v1", provided the semantics to allow devices to perform DMA transfers that significantly reduce the CPU activity by eliminating the copy functions as shown in the ***Standard TCP/IP vs. RDMA Data Movement*** diagram. RoCE v1 does not span across IP subnets. RoCE v2 enables communication across IP subnets.



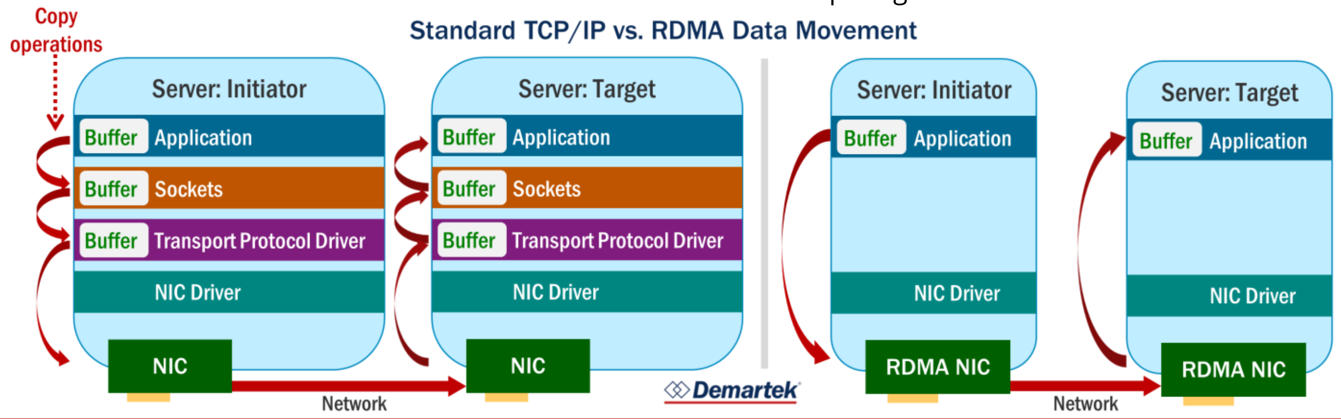
A more detailed technical explanation of the difference between RoCE v1 and v2 is available at https://community.mellanox.com/docs/DOC-1451

DCB Switch

To achieve low latency, RoCE adapters work best with Ethernet switches that support Data Center Bridging (DCB) that provide lossless characteristics and support priority flow control (PFC). DCB Ethernet switches have become more common over the last few years.

RoCE can also work without a DCB switch because the RoCE specification defines how to use standard IP explicit congestion notification (ECN) and a congestion notification packet response (CNP) which provides the mechanisms needed for congestion management without requiring PFC.

Standard TCP/IP vs. RDMA Data Movement



RoCE Applications

There are a variety of applications that can take advantage of RoCE technology, including the three described below. Typically, RoCE is available for 10Gbps and faster technologies. Increased throughput, reduced CPU consumption and low latency are among the benefits available.

Windows: SMB Direct & Storage Spaces Direct

Beginning with Windows Server 2012, Microsoft has included a feature known as SMB Direct that supports the use of RDMA-capable network adapters, such as RoCE network adapters. SMB Direct uses SMBv3 (SMB3) and these adapters to enable file transfer operations to operate at high speed and very low latency, improving the overall performance of a file server. SMB Direct is enabled automatically if RDMA-capable adapters are present.

Beginning with Windows Server 2016, Storage Spaces Direct takes advantage of RDMA-capable network adapters to provide highly scalable software-defined storage. Storage Spaces Direct can be deployed on physical machines or virtual machines with Hyper-V and can be deployed in converged or hyper-converged configurations.

Linux: iSER and NFS over RDMA

For Linux environments, iSCSI Extensions for RDMA (iSER) allows iSCSI (block storage) traffic to take advantage of RDMA-capable network adapters. Similarly, NFS over RDMA allows file traffic to take advantage of RDMA-capable network adapters.

NVMe over Fabrics

NVM Express over Fabrics (NVMe-oF™) defines a common architecture that supports a range of storage networking fabrics for NVMe block storage protocol over a storage networking fabric, including RDMA fabrics. RoCE is supported for NVMe-oF.

Products Tested in this Guide

We used several different products in various configurations to product this ***RoCE Deployment Guide***. These include the following:

> HPE 3PAR StoreServ File Controller v3 and HPE 3PAR 8450 all-flash storage array

> Mellanox SN2410 25GbE/100GbE switch

> Mellanox ConnectX-4 25GbE and 100GbE adapters

> QLogic QLE45212 25GbE adapters

HPE

The HPE 3PAR StoreServ File Controller v3 provides file services from any model of HPE 3PAR StoreServ storage. This was connected to an HPE 3PAR 8450 all-flash storage array.

The HPE target consists of an Apollo 2600 chassis with two HPE ProLiant XL190r Gen9 server nodes. It is designed to provide a RoCE-capable file server front end for our HPE 3PAR StoreServ 8450 block storage system. SMB 3.0 over TCP or SMB Direct over RoCE can both be used; the setup is similar for both. Each server node is configured with Windows Storage Server 2012 R2 and contains an Emulex 16GbFC HBA and an HPE Ethernet 10/25Gb 2-port 640FLR-SFP28 Adapter. This adapter uses the Mellanox ConnectX-4 Lx controller and provides 2 ports of 25GbE for each server node.

Mellanox

The Mellanox SN2410 is an Open Ethernet top-of-rack switch with 48-ports of 25GbE and 8-ports of 100GbE. The Mellanox ConnectX-4 Lx EN network controller supports 1/10/25/40/50GbE. The Mellanox ConnectX-4 EN network controller supports 100GbE.

QLogic

The QLogic® FastLinQ™ QL45212 dual-port Intelligent Ethernet Adapter supports 25GbE.

Deploying RoCE on a Network Switch

The user will need to appropriately configure the switch, configure the adapters according to vendor-specific instructions, and then follow the appropriate application setup.

General considerations for a RoCE deployment are as follows:

> Standard MTU: While some concurrently running TCP/IP applications may benefit from Jumbo frames, and Jumbo Frames may be used with RoCE, it does increase the latency and is not necessary. Keeping the MTU at 1500 is better in most cases.

> PFC (Priority Flow Control): Unlike TCP/IP that uses dropped packets as a form of implicit congestion notification; RoCE does not have its own congestion management and requires a lossless network to function properly when there is network congestion. While a lossless network may be achieved most simplistically by enabling basic flow control (global pause), standard practice is to implement Priority Flow Control and QoS with VLAN. These settings need to be implemented on the adapter, in the OS, and on the switch.

> Switch settings must match the server and driver settings. (We used VLAN 2 and priority 3 for RoCE Traffic in our RoCE examples because we are using a Mellanox switch. Any other switch would most likely have Priority 3 set to FCoE, and we would be unable to change it. In these cases, Priority 5 would be recommended.)

A typical way to achieve this setup on a Mellanox switch is as follows:
(Default username and password for a new Mellanox switch is admin,admin).

1. Gain access to the administrator command set:
   ```
   #enable
   #config t
   ```
2. Display the current configuration:
   ```
   #show running-config
   ```
3. Display an interface:
   ```
   #show interfaces ethernet 1/1
   ```
4. Make sure flow control is disabled (often it is best to revert to the initial configuration):
   ```
   #config switch-to initial
   ```
5. Enable PFC priority 3 on the switch:
   ```
   #dcb priority-flow-control enable force
   #dcb priority-flow-control priority 3 enable
   ```
6. Enable PFC on **each port**:
   ```
   #interface ethernet 1/1 dcb priority-flow-control
   mode on force
   #interface ethernet 1/2 dcb priority-flow-control
   mode on force
   …
   ```
7. Configure VLAN on switch:
   ```
   #vlan 2
   ```
8. Configure VLAN on **each port**:
   ```
   #interface ethernet 1/1 switchport trunk
   allowed-vlan 2
   #interface ethernet 1/2 switchport trunk
   allowed-vlan 2
   …
   ```
9. Configure lossless (RoCE) and lossy (all other traffic) buffers for **each port**:
   ```
   #interface 1/1 ingress-buffer iPort.pg0 map pool
   iPool0 type lossy reserved 20480 shared alpha 8
   #interface ethernet 1/1 ingress-buffer iPort.pg3
   map pool iPool0 type lossless reserved 90000
   xoff 17000 xon 17000 shared alpha 2
   #interface ethernet 1/1 ingress-buffer iPort.pg3
   bind switch-priority 3
   #interface ethernet 1/1 egress-buffer ePort.tc3
   map pool ePool0 reserved 4096 shared alpha inf
   #interface ethernet 1/2 ingress-buffer iPort.pg0
   map pool iPool0 type lossy reserved 20480
   shared alpha 8
   ```
   …Each port gets 4 commands total to configure the buffers.
10. Write the configuration to come back to later:
    ```
    #config write to RoCEPFCwVLAN
    ```

11. Switch back to it at any time if you make changes:
    #config switch-to RoCEPFCwVLAN
12. Double-check PFC with the command:
    #show dcb priority-flow-control

Most of the time, this configuration change will require a reboot. If PFC is not enabled but all configuration changes are complete, a manual reboot may be needed after the configuration is written.

Windows: Deploying SMB Direct over RoCE with Storage Spaces

The user will need to appropriately configure the switch, configure the adapters according to vendor-specific instructions, and then follow the appropriate application setup.

TCP/IP connection setup should be completed before proceeding to adapter and application setup steps. The steps to do so are not outlined here in this guide.

Adapter and application setup will need to be completed twice, once on the client and once on the server.

*Please note that the HPE File Controller uses Mellanox adapters.

Mellanox

To, enable PFC and QoS within the Windows Server 2012 OS, the Data Center Bridging Feature is installed using the Add Roles and Features Wizard. This will enable the following PowerShell commands which put RoCE on PFC Priority 3:

```
>Remove-NetQosTrafficClass
>Remove-NetQosPolicy -Confirm:$False
>Set-NetQosDcbxSetting -Willing 0
New-NetQosPolicy "SMB" -PolicyStore
ActiveStore -NetDirectPortMatchCondition 445 -
PriorityValue8021Action 3
>New-NetQosPolicy "DEFAULT" -PolicyStore
Activestore -Default -PriorityValue8021Action 3
New-NetQosPolicy "TCP" -PolicyStore
ActiveStore -IPProtocolMatchCondition TCP -
PriorityValue8021Action 0
>New-NetQosPolicy "UDP" -PolicyStore
ActiveStore -IPProtocolMatchCondition UDP -
PriorityValue8021Action 0
>Disable-NetQosFlowControl 0,1,2,4,5,6,7
>Enable-NetQosFlowControl -Priority 3
>Enable-NetAdapterQos -InterfaceAlias
"<ethernet interface>"
```

The adapter settings were as follows:

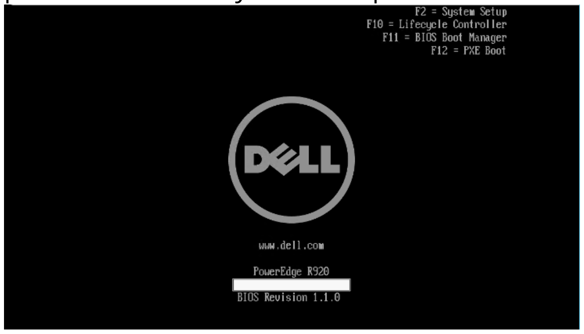
Name	Value
Encapsulated Task Offload	Enabled
Flow Control	Rx & Tx Enabled
Interrupt Moderation	Enabled
IPV4 Checksum Offload	Rx & Tx Enabled
Jumbo Packet	1500
Large Send Offload V2 (IPv4)	Enabled
Large Send Offload V2 (IPv6)	Enabled
Maximum number of RSS Proce…	20
NetworkDirect Functionality	Enabled
Preferred NUMA Node	Default Settings
Maximum Number of RSS Queues	20
Priority & Vlan Tag	Priority & VLAN Enabled
Quality Of Service	Enabled
Receive Buffers	4096
Recv Segment Coalescing (IPv4)	Enabled
Recv Segment Coalescing (IPv6)	Enabled
Receive Side Scaling	Enabled
RSS Base Processor Number	0
RSS Maximum Processor Number	40
RSS Load Balancing Profile	ClosestProcessor
SR-IOV	Disabled
TCP/UDP Checksum Offload (I…	Rx & Tx Enabled
TCP/UDP Checksum Offload (I…	Rx & Tx Enabled
Send Buffers	4096
Virtual Machine Queues	Disabled

VMQ VLAN Filtering	Disabled
Network Address	--
Receive Completion Method	Adaptive
R/RoCE Max Frame Size	Auto
Rx Interrupt Moderation Type	Adaptive
Rx Interrupt Moderation Pro...	Aggressive
Tx Interrupt Moderation Pro...	Aggressive
VLAN ID	2

QLogic

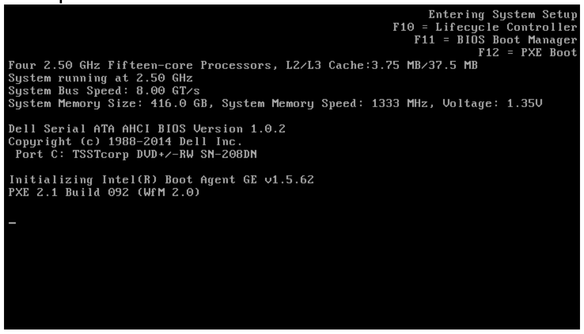
In order to use RoCE with QLogic cards, the DCBX and priority number are selected in the UEFI HII interface. Here is an example of this on a Dell server:

1. When the menu appears at boot in the top right, press F2 to enter System Setup.



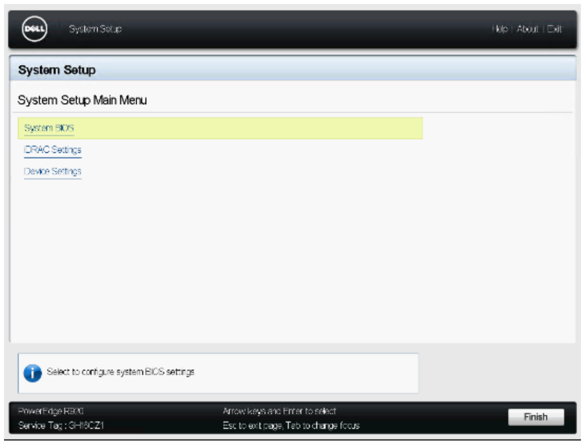
```
                                        F2 = System Setup
                                  F10 = Lifecycle Controller
                                     F11 = BIOS Boot Manager
                                              F12 = PXE Boot

                   www.dell.com
                  PowerEdge R920
                BIOS Revision 1.1.0
```

2. Wait while normal boot procedures are completed.



```
                                                   Entering System Setup
                                              F10 = Lifecycle Controller
                                                 F11 = BIOS Boot Manager
                                                          F12 = PXE Boot
Four 2.50 GHz Fifteen-core Processors, L2/L3 Cache:3.75 MB/37.5 MB
System running at 2.50 GHz
System Bus Speed: 8.00 GT/s
System Memory Size: 416.0 GB, System Memory Speed: 1333 MHz, Voltage: 1.35V

Dell Serial ATA AHCI BIOS Version 1.0.2
Copyright (c) 1988-2014 Dell Inc.
 Port C: TSSTcorp DVD+/-RW SN-208DN

Initializing Intel(R) Boot Agent GE v1.5.62
PXE 2.1 Build 092 (WfM 2.0)


_
```

3. Select Device Settings at the System Setup Main Menu.



System Setup

System Setup Main Menu

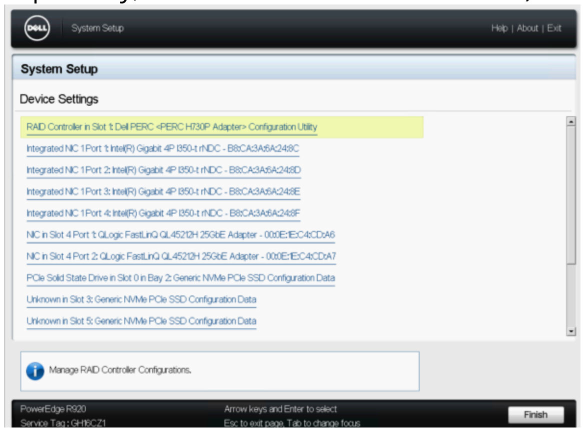
- System BIOS
- iDRAC Settings
- Device Settings

Select to configure system BIOS settings

PowerEdge R920
Service Tag : GH16CZ1

Arrow keys and Enter to select
Esc to exit page, Tab to change focus

4. Choose the QLogic Device Port to configure (Here there are two ports to be configured separately, both shown as “NIC in Slot 4…”).



System Setup

Device Settings

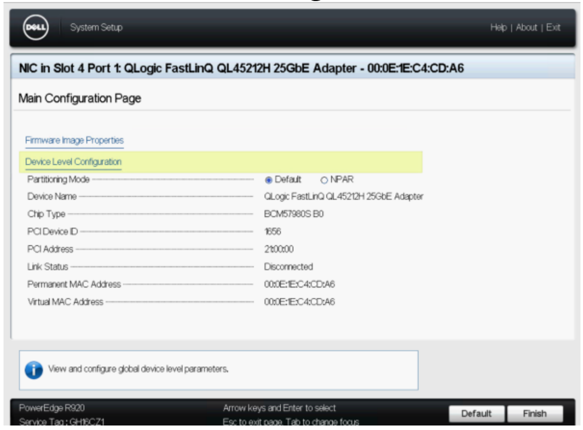
- RAID Controller in Slot 1: Dell PERC <PERC H730P Adapter> Configuration Utility
- Integrated NIC 1 Port 1: Intel(R) Gigabit 4P I350-t rNDC - B8:CA:3A:6A:24:8C
- Integrated NIC 1 Port 2: Intel(R) Gigabit 4P I350-t rNDC - B8:CA:3A:6A:24:8D
- Integrated NIC 1 Port 3: Intel(R) Gigabit 4P I350-t rNDC - B8:CA:3A:6A:24:8E
- Integrated NIC 1 Port 4: Intel(R) Gigabit 4P I350-t rNDC - B8:CA:3A:6A:24:8F
- NIC in Slot 4 Port 1: QLogic FastLinQ QL45212H 25GbE Adapter - 00:0E:1E:C4:CD:A6
- NIC in Slot 4 Port 2: QLogic FastLinQ QL45212H 25GbE Adapter - 00:0E:1E:C4:CD:A7
- PCIe Solid State Drive in Slot 0 in Bay 2: Generic NVMe PCIe SSD Configuration Data
- Unknown in Slot 3: Generic NVMe PCIe SSD Configuration Data
- Unknown in Slot 5: Generic NVMe PCIe SSD Configuration Data

Manage RAID Controller Configurations.

PowerEdge R920
Service Tag : GH16CZ1

Arrow keys and Enter to select
Esc to exit page, Tab to change focus

5. Select Device Level Configuration.



NIC in Slot 4 Port 1: QLogic FastLinQ QL45212H 25GbE Adapter - 00:0E:1E:C4:CD:A6

Main Configuration Page

- Firmware Image Properties
- Device Level Configuration

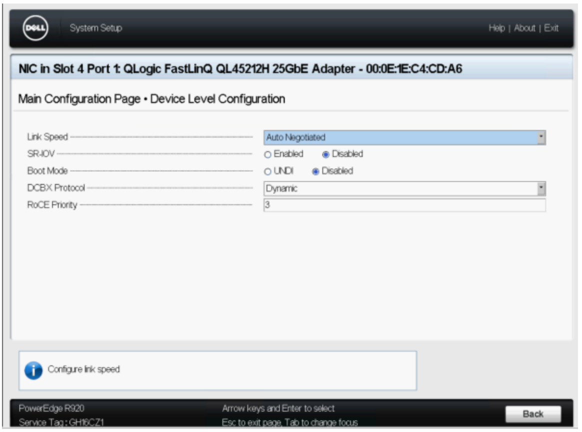
Partitioning Mode	[● Default] ○ NPAR
Device Name	QLogic FastLinQ QL45212H 25GbE Adapter
Chip Type	BCM57980S B0
PCI Device ID	1656
PCI Address	21:00:00
Link Status	Disconnected
Permanent MAC Address	00:0E:1E:C4:CD:A6
Virtual MAC Address	00:0E:1E:C4:CD:A6

View and configure global device level parameters.

PowerEdge R920
Service Tag : GH16CZ1

Arrow keys and Enter to select
Esc to exit page, Tab to change focus

6. Select Dynamic DCBX protocol and the appropriate RoCE Priority. Here Priority 3 is used, which is permitted with a Mellanox switch.



NIC in Slot 4 Port 1: QLogic FastLinQ QL45212H 25GbE Adapter - 00:0E:1E:C4:CD:A6

Main Configuration Page • Device Level Configuration

Link Speed	Auto Negotiated
SR-IOV	Enabled [✓ Disabled]
Boot Mode	UNDI [✓ Disabled]
DCBX Protocol	Dynamic
RoCE Priority	3

Configure link speed

PowerEdge R920
Service Tag: GH16CZ1

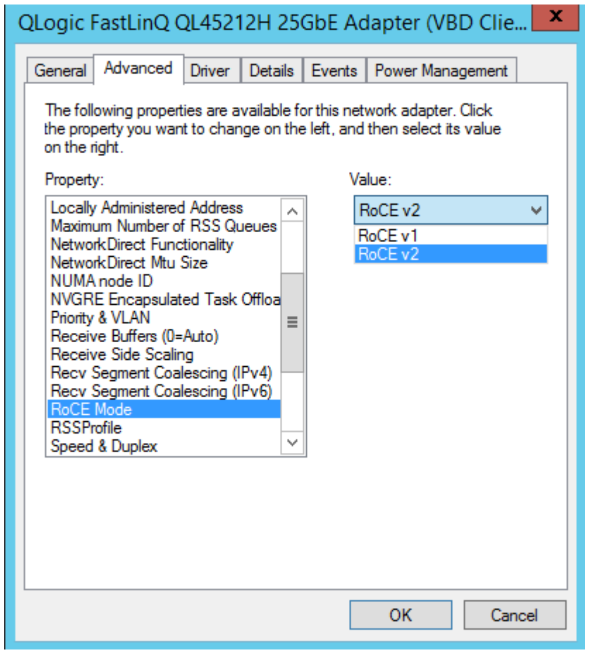
Arrow keys and Enter to select
Esc to exit page, Tab to change focus

Other manufacturers should have similar UEFI hook-ins for the QLogic cards allowing the DCBX option and traffic priority class to be set. If any difficulty is encountered using the HII interface on a server, contact QLogic technical support for an engineering solution.

The adapter settings were as follows:

Name	Value
Encapsulated Task Offload	Enabled
NVGRE Encapsulated Task Off…	Enabled
VXLAN Encapsulated Task Off…	Enabled
Flow Control	Rx & Tx Enabled
Interrupt Moderation	Enabled
Jumbo Packet	1514
Large Send Offload V2 (IPv4)	Enabled
Large Send Offload V2 (IPv6)	Enabled
NetworkDirect Functionality	Enabled
NUMA node ID	65535
Maximum Number of RSS Queues	16
Priority & VLAN	Priority & VLAN enabled
Receive Buffers (0=Auto)	3000
Recv Segment Coalescing (IPv4)	Enabled
Recv Segment Coalescing (IPv6)	Enabled
Receive Side Scaling	Enabled
RSSProfile	NUMAScalingStatic
Speed & Duplex	25 Gbps Full Duplex
SR-IOV	Disabled
TCP/UDP Checksum Offload (I…	Rx & Tx Enabled
TCP/UDP Checksum Offload (I…	Rx & Tx Enabled
Transmit Buffers (0=Auto)	5000
Virtual Machine Queues	Disabled
VXLAN UDP destination port …	4789
Locally Administered Address	--
NetworkDirect Mtu Size	1024
Link control	Preboot controlled
RoCE Mode	RoCE v2
VLAN ID	2

> Notice the option to specify the RoCE mode as RoCE v2 or RoCE v1. RoCE v2 was used for all our SMB Direct testing.


QLogic FastLinQ QL45212H 25GbE Adapter (VBD Clie...

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:

Locally Administered Address
Maximum Number of RSS Queues
NetworkDirect Functionality
NetworkDirect Mtu Size
NUMA node ID
NVGRE Encapsulated Task Offloa
Priority & VLAN
Receive Buffers (0=Auto)
Receive Side Scaling
Recv Segment Coalescing (IPv4)
Recv Segment Coalescing (IPv6)
RoCE Mode
RSSProfile
Speed & Duplex

Value:

RoCE v2
RoCE v1
RoCE v2

> Network Direct MTU size for QLogic may be increased to 4096 for possible performance gains, as long as the Ethernet MTU size is larger than 4096, and a 4096 MTU is configured on the switch, client, and server.

Application Setup

When deploying SMB Direct over RoCE, perform the following steps after switch configuration, OS install, cluster configuration (if needed, as in in the case of the HPE File Controller) and adapter configuration:

1. Make sure the global setting for SMB Direct is enabled using the PowerShell command:
   \>get-NetOffloadGlobalSetting
2. Make sure multichannel is enabled. (While this feature was enabled but not leveraged in our testing due to our simple setup, it should be used when there are, physically, multiple channels available)
   \> get-SMBClientConfiguration | Select Enable Multichannel
   OR
   \> get-SMBServerConfiguration | Select Enable Multichannel
3. Verify the network interface is listed as being RDMA capable:
   \>get-SMBClientNetworkInterface
   OR
   \>get-SMBServerNetworkInterface
   AND
   \>get-NetAdapterRDMA
4. Create the storage server.
   > HPE File Controller:
      > In the 3PAR management software, a small volume should be configured for Quorum and exported to both server nodes. For our testing, 1GB was sufficient. A larger volume should also be configured and exported to both nodes, so that the cluster configuration wizard will automatically configure a file server role on the cluster and make the larger volume available for shares.
      > Following the ICT will configure the failover cluster with file server role for the user. Once complete, the user needs to go to Server Manager, File and Storage Services, Shares, and configure a share on the large volume.
      > **If** an active/active configuration is desired, add a second file server role to run on the second node. First create a new large LUN exported to both nodes. Then rescan the storage in Server Manager, File and Storage Services, Volumes, Disks. Create a new volume on the new LUN. Then go to the Server Manager Tools Menu to find the Failover Clustering. Go to roles and create a new file server role, with a new IP address and the newly created volume for storage. After the role is created, assign the role to the other node, if it hasn't been already. After this, a share can be created on the volume as was done with the previously created file server role.
   > Generic Servers: Install the desired block device(s) locally in the target server and use Storage Spaces to create Virtual Drives containing File Shares.
5. If all is configured correctly, there should be two SMB Direct Listeners waiting for connections on both client and server:
   \>netstat –xan 1
   Once a connection to the remote file share is initiated and traffic has started, netstat will show the connections.
6. To view RDMA connection attempts, active connections, and bytes or frames sent, open the Perfmon program, right click to add counters, and select RDMA.

While not employed in our own testing, the following may be explored to potentially enhance performance of SMB Direct:

> Increase the number of SMB Direct RDMA connections from the default (2) to 8 or 16 using the PowerShell command
\>Set-ItemProperty –Path “HKLM:\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters” ConnectionCountPerRdmaNetworkInterface –Type DWORD –Value <number of connections> -Force

It should be noted that there will not be a key present in the registry if the default number of connections are being used; it will only be present once the value is changed. It should also be noted that the PowerShell command sets the value immediately, while using regedit necessitates a reboot. This option will have to be set on both the target server and on the client.

> If RoCE v2 is not necessary for routing purposes, RoCE v1 may give better performance.

Linux: Deploying iSER over RoCE

The user will need to appropriately configure the switch, configure the adapters according to vendor-specific instructions, and then follow the appropriate application setup.

TCP/IP connection setup is should be completed before proceeding to adapter and application setup steps. The steps to do so are not outlined here in this guide.

Adapter and application setup will need to be completed twice, once on the initiator and once on the target.

Mellanox

Using iSER with RoCE on Red Hat Enterprise Linux requires the use of InfiniBand software and tools, called OFED. While Red Hat Enterprise Linux does provide inbox InfiniBand, Mellanox also provides its own OFED. While Mellanox can use the inbox OFED, if Mellanox is the only brand of NIC using RDMA on a server, it is best to download and install the Mellanox OFED. Please note that the Mellanox OFED should not be installed on a server alongside the generic OFED and cannot be used with non-Mellanox NICs.

1. Download the Mellanox OFED. 3.3 is available from http://www.mellanox.com/page/products_dyn?product_family=26&mtag=linux_sw_drivers
2. Install prerequisite packages.
   #yum install cpp-4.8.5-4.el7.x86_64.rpm gcc-4.8.5-4.el7.x86_64.rpm gcc-gfortran-4.8.5-4.el7.x86_64.rpm glibc-2.17-106.el7_2.8.x86_64.rpm glibc-common-2.17-106.el7_2.8.x86_64.rpm glibc-devel-2.17-106.el7_2.8.x86_64.rpm glibc-headers-2.17-106.el7_2.8.x86_64.rpm kernel-headers-3.10.0-327.36.1.el7.x86_64.rpm libgfortran-4.8.5-4.el7.x86_64.rpm libmpc-1.0.1-3.el7.x86_64.rpm libquadmath-4.8.5-4.el7.x86_64.rpm libquadmath-devel-4.8.5-4.el7.x86_64.rpm mpfr-3.1.1-4.el7.x86_64.rpm
3. Install Mellanox OFED.
   #tar -xvf MLNX_OFED_LINUX-3.3-1.0.4.0-rhel7.2-x86_64.tgz
   #cd MLNX_OFED_LINUX-3.3-1.0.4.0-rhel7.2-x86_64
   #./mlnxofedinstall
   It may warn of additional missing packages. These should be installed with yum and then the Mellanox OFED install should be attempted again.
4. Check if the drivers are loaded correctly.
   #ibdev2netdev
   If the drivers are loaded correctly, this should list the name of the Mellanox interface.
   If nothing was present:
   ```
   #/etc/init.d/openibd restart
   (It may be necessary to unload several
   other modules in order to run this
   command:
       #rmmod ib_isert
       #rmmod xprtrdma
       #rmmod ib_srpt
       #/etc/init.d/openibd restart
       #ibdev2netdev)
   ```
5. Configure PFC.

   > Display the current priorities. This should show that there is no priority set.
   #mlnx_qos -i <interface name>

   > Set the desired priority. Each number corresponds to a priority, 0 through 7. Setting the 4th number to 1 enables the 4th priority in the sequence, which is 3 (0,1,2,3,4,5,6,7).
   #mlnx_qos -i <interface name> 0,0,0,1,0,0,0,0

   > Set the egress priority mappings.

There is supposed to be a way to set the egress mapping by adding a line to the network script. However, in our case it didn't work and we used alternate means. Unfortunately, this set of commands will have to be re-entered each time the server or network service is restarted. It may be useful to put them in a script.

#ip link set <VLAN interface> type vlan egress 0:<RoCE PFC priority>

#ip link set <VLAN interface> type vlan egress 1:<RoCE PFC priority>

… repeat for all priorities up to 15. Only priorities 0 and 1 are shown above.

```
#ip link set p9p1.2 type vlan egress 0:3
#ip link set p9p1.2 type vlan egress 1:3
#ip link set p9p1.2 type vlan egress 2:3
#ip link set p9p1.2 type vlan egress 3:3
#ip link set p9p1.2 type vlan egress 4:3
#ip link set p9p1.2 type vlan egress 5:3
#ip link set p9p1.2 type vlan egress 6:3
#ip link set p9p1.2 type vlan egress 7:3
#ip link set p9p1.2 type vlan egress 8:3
#ip link set p9p1.2 type vlan egress 9:3
#ip link set p9p1.2 type vlan egress 10:3
#ip link set p9p1.2 type vlan egress 11:3
#ip link set p9p1.2 type vlan egress 12:3
#ip link set p9p1.2 type vlan egress 13:3
#ip link set p9p1.2 type vlan egress 14:3
#ip link set p9p1.2 type vlan egress 15:3
```

> Check the egress commands worked.

```
#cat /proc/net/vlan/p9p1.2

p9p1.2  VID: 2
   total frames received            168
    total bytes received           9229
Broadcast/Multicast Rcvd              0

total frames transmitted             15
 total bytes transmitted            942

Device: p9p1
INGRESS priority mappings: 0:0  1:0  2:0
3:0  4:0  5:0  6:0 7:0
 EGRESS priority mappings: 0:3 1:3 2:3
3:3 4:3 5:3 6:3 7:3 8:3 9:3 10:3 11:3
12:3 13:3 14:3 15:3
```

> Optional: Check the GIDs with the Mellanox provided script show-gids (below). This script can be obtained from the Mellanox website: https://community.mellanox.com/docs/DOC-2421

Index 7 is running RoCE v2 on VLAN 2 interface p9p1.2, which is what is needed.

```
#./show-gids
DEV     PORT    INDEX   GID                                     IPv4        VER   DEV
---     ----    -----   ---                                     ----------  ---   ---
mlx5_0    1     0       fe80:0000:0000:0000:7efe:90ff:fe22:a00a             v1    p9p1
mlx5_0    1     1       fe80:0000:0000:0000:7efe:90ff:fe22:a00a             v2    p9p1
mlx5_0    1     2       fe80:0000:0000:0000:7efe:90ff:fe22:a00a             v1.5  p9p1
mlx5_0    1     3       0000:0000:0000:0000:0000:ffff:0a00:01fc 10.0.1.252  v1    p9p1
mlx5_0    1     4       0000:0000:0000:0000:0000:ffff:0a00:01fc 10.0.1.252  v2    p9p1
mlx5_0    1     5       0000:0000:0000:0000:0000:ffff:0a00:01fc 10.0.1.252  v1.5  p9p1
mlx5_0    1     6       0000:0000:0000:0000:0000:ffff:0a00:01fc 10.0.1.252  v1    p9p1.2
mlx5_0    1     7       0000:0000:0000:0000:0000:ffff:0a00:01fc 10.0.1.252  v2    p9p1.2
mlx5_0    1     8       0000:0000:0000:0000:0000:ffff:0a00:01fc 10.0.1.252  v1.5  p9p1.2
n_gids_found=9
```

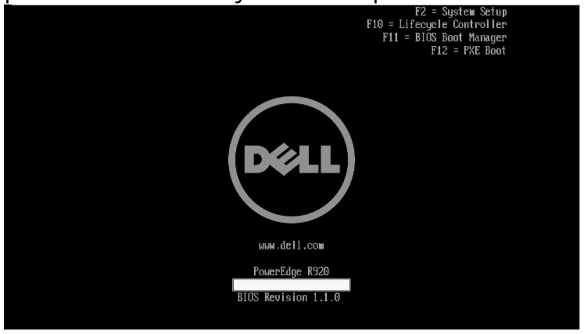
> Check everything with tc_wrap.py. Make sure everything for VLAN 2 is priority 3.

```
#tc_wrap.py -i p9p1
priority  0
        skprio: 0
        skprio: 8
        skprio: 9
        skprio: 10
        skprio: 11
        skprio: 12
        skprio: 13
        skprio: 14
        skprio: 15
priority  1
        skprio: 1
priority  2
        skprio: 2 (tos: 8)
priority  3
        skprio: 3
        skprio: 0 (vlan 2)
        skprio: 1 (vlan 2)
        skprio: 2 (vlan 2 tos: 8)
        skprio: 3 (vlan 2)
        skprio: 4 (vlan 2 tos: 24)
        skprio: 5 (vlan 2)
        skprio: 6 (vlan 2 tos: 16)
        skprio: 7 (vlan 2)
        skprio: 8 (vlan 2)
        skprio: 9 (vlan 2)
        skprio: 10 (vlan 2)
        skprio: 11 (vlan 2)
        skprio: 12 (vlan 2)
        skprio: 13 (vlan 2)
        skprio: 14 (vlan 2)
        skprio: 15 (vlan 2)
priority  4
        skprio: 4 (tos: 24)
priority  5
        skprio: 5
priority  6
        skprio: 6 (tos: 16)
priority  7
```

QLogic

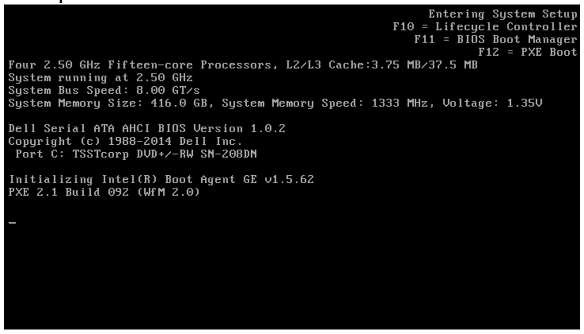
In order to use RoCE with QLogic cards, the DCBX and priority number are selected in the UEFI HII interface. Here is an example of this on a Dell server:

1. When the menu appears at boot in the top right, press F2 to enter System Setup.



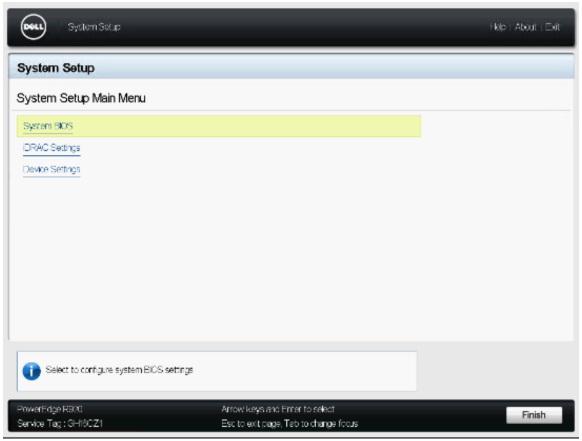
```
                                            F2 = System Setup
                                   F10 = Lifecycle Controller
                                      F11 = BIOS Boot Manager
                                              F12 = PXE Boot

                         www.dell.com
                       PowerEdge R920
                     BIOS Revision 1.1.0
```

2. Wait while normal boot procedures are completed.



```
                                                       Entering System Setup
                                                   F10 = Lifecycle Controller
                                                      F11 = BIOS Boot Manager
                                                              F12 = PXE Boot
Four 2.50 GHz Fifteen-core Processors, L2/L3 Cache:3.75 MB/37.5 MB
System running at 2.50 GHz
System Bus Speed: 8.00 GT/s
System Memory Size: 416.0 GB, System Memory Speed: 1333 MHz, Voltage: 1.35V

Dell Serial ATA AHCI BIOS Version 1.0.2
Copyright (c) 1988-2014 Dell Inc.
 Port C: TSSTcorp DVD+/-RW SN-208DN

Initializing Intel(R) Boot Agent GE v1.5.62
PXE 2.1 Build 092 (WfM 2.0)

_
```

3. Select Device Settings at the System Setup Main Menu.

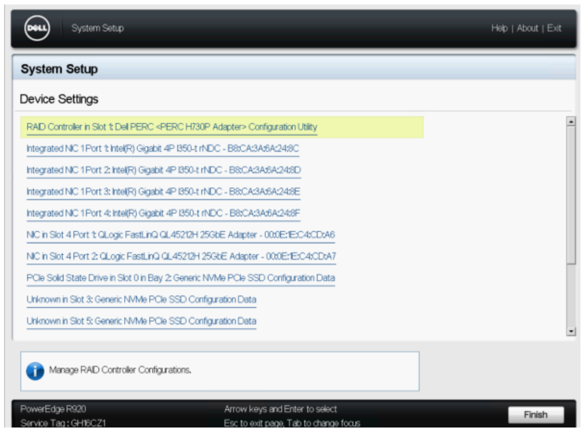


Select to configure system BIOS settings

PowerEdge R920
Service Tag : GH16CZ1

Arrow keys and Enter to select
Esc to exit page, Tab to change focus

4. Choose the QLogic Device Port to configure (Here there are two ports to be configured separately, both shown as "NIC in Slot 4…").



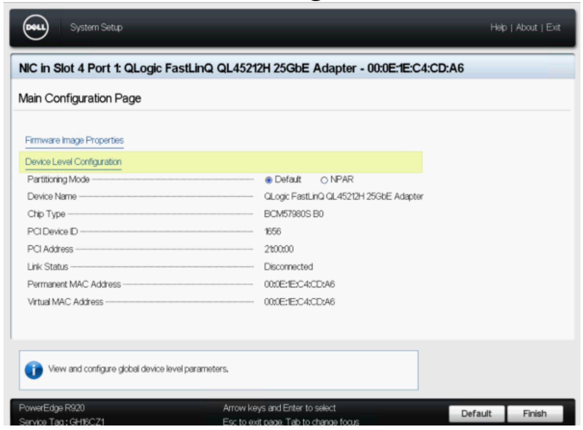
RAID Controller in Slot 1: Dell PERC <PERC H730P Adapter> Configuration Utility
Integrated NIC 1 Port 1: Intel(R) Gigabit 4P I350-t rNDC - B8:CA:3A:6A:24:8C
Integrated NIC 1 Port 2: Intel(R) Gigabit 4P I350-t rNDC - B8:CA:3A:6A:24:8D
Integrated NIC 1 Port 3: Intel(R) Gigabit 4P I350-t rNDC - B8:CA:3A:6A:24:8E
Integrated NIC 1 Port 4: Intel(R) Gigabit 4P I350-t rNDC - B8:CA:3A:6A:24:8F
NIC in Slot 4 Port 1: QLogic FastLinQ QL45212H 25GbE Adapter - 00:0E:1E:C4:CD:A6
NIC in Slot 4 Port 2: QLogic FastLinQ QL45212H 25GbE Adapter - 00:0E:1E:C4:CD:A7
PCIe Solid State Drive in Slot 0 in Bay 2: Generic NVMe PCIe SSD Configuration Data
Unknown in Slot 3: Generic NVMe PCIe SSD Configuration Data
Unknown in Slot 5: Generic NVMe PCIe SSD Configuration Data

Manage RAID Controller Configurations.

PowerEdge R920
Service Tag : GH16CZ1

Arrow keys and Enter to select
Esc to exit page, Tab to change focus

5. Select Device Level Configuration.



NIC in Slot 4 Port 1: QLogic FastLinQ QL45212H 25GbE Adapter - 00:0E:1E:C4:CD:A6

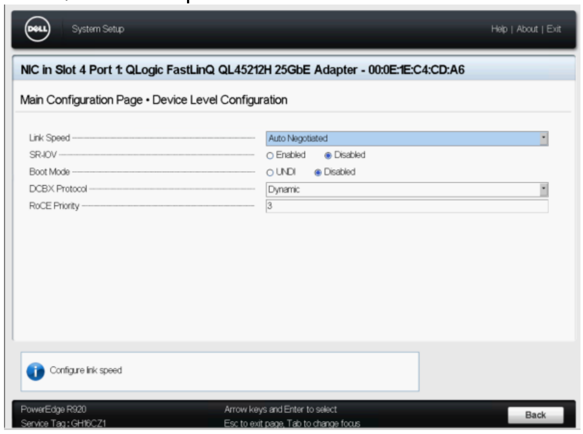
Partitioning Mode	[✓ Default] NPAR
Device Name	QLogic FastLinQ QL45212H 25GbE Adapter
Chip Type	BCM57980S B0
PCI Device ID	1656
PCI Address	21:00:00
Link Status	Disconnected
Permanent MAC Address	00:0E:1E:C4:CD:A6
Virtual MAC Address	00:0E:1E:C4:CD:A6

View and configure global device level parameters.

PowerEdge R920
Service Tag : GH16CZ1

Arrow keys and Enter to select
Esc to exit page, Tab to change focus

6. Select Dynamic DCBX protocol and the appropriate RoCE Priority. Here Priority 3 is used, which is permitted with a Mellanox switch.



NIC in Slot 4 Port 1: QLogic FastLinQ QL45212H 25GbE Adapter - 00:0E:1E:C4:CD:A6

Main Configuration Page • Device Level Configuration

Link Speed	Auto Negotiated
SR-IOV	Enabled [✓ Disabled]
Boot Mode	UNDI [✓ Disabled]
DCBX Protocol	Dynamic
RoCE Priority	3

Configure link speed

PowerEdge R920
Service Tag : GH16CZ1

Arrow keys and Enter to select
Esc to exit page, Tab to change focus

Other manufacturers should have similar UEFI hook-ins for the QLogic cards allowing the DCBX option and traffic priority class to be set. If any difficulty is encountered using the HII interface on a server, contact QLogic technical support for an engineering solution.

While in system setup, configure the following in the System BIOS for optimal performance later:

1. Disable hyperthreading.
2. Set the power management to performance, and disable any Energy Performance Tuning or Energy Efficient Turbo settings.
3. Set Package C State Limit C0/C1 state.
4. Disable CPU C3 report and CPU C6 report and Enhanced Halt State (C1E).
5. Disable ACPI T-States.

Once the system has booted:

1. Install an InfiniBand OFED. There are two choices: either use the Inbox OFED provided by Red Hat Enterprise Linux 7.2 or to compile and install the OFED provided at the OpenFabrics website. QLogic recommends using the inbox OFED:
   # yum –y groupinstall “InfiniBand Support”
2. Install extra packages for configuration and testing:
   #yum –y install perftest infiniband-diags
3. Install the RDMA drivers. The NIC drivers should already have been installed when TCP/IP was set-up. Go back to the same driver install directory.
   #cd fastlinq-minor-8.10.4.0/libqedr-8.10.3.0
4. The readme contains the appropriate configure line to use for Red Hat systems. Compile the RDMA driver.
   #./configure –prefix=/usr –libdir=${exec_prefix}/lib64 –sysconfdir=/etc
   #make install
5. Check RDMA drivers. There should already be qed and qede. The RDMA driver is qedr, and may need to be loaded manually.
   #lsmod | grep qed
   If necessary
   #modprobe qedr
6. Check to make sure everything is working. ibstat should list all QLogic ports available for RDMA.
   #ibstat
   #service rdma status
7. Make sure the CPUs are set to performance mode. Check each processor.
   #cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
   #cat /sys/devices/system/cpu/cpu1/cpufreq/scaling_governor
   …
   It should read performance for all. If any processors are in powersave, manually change each of them to performance.
   # echo –n performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
   # echo –n performance > /sys/devices/system/cpu/cpu1/cpufreq/scaling_governor
   …
8. Check memory limits.
   #ulimit –l
   If there is a limit, this must be rectified:
   > Add the following lines to /etc/security/limits.conf:

   ```
   * soft memlock unlimited
   * hard memlock unlimited
   ```

   > Then log off and log back in again for changes to take effect.

   > Check the status again.
   #ulimit –l
   Should give “unlimited”
9. Disable throttling.
   > in /etc/default/grub, find the end of the line “GRUB_CMDLINE_LINUX=” and add “intel_idle.max_cstate=0 processor.max_cstate=1 mce=ignore_ce idle=poll”

   > #grub2-mkconfig –o /boot/grub2/grub.cfg
10. Reboot. Do checks starting at step 5 again.
11. Disable firewall.

#systemctl disable firewalld

12. Check InfiniBand.
    On the target:
    #ib_write_bw –p 12500 –x 0 –R -a
    On the initiator:
    #ib_write_bw <target address> -p 12500 –x 0 –R -a

Application Setup

When deploying iSER over RoCE, perform the following steps after switch configuration, OS install, and adapter configuration:

1. On the initiator system, install the iscsi-initiator-utils if it was not installed with the OS.
   #yum install iscsi-initiator-utils

2. Find out the initiator IQN for future reference.
   #cat /etc/iscsi/initiatorname.iscsi

```
#cat /etc/iscsi/initiatorname.iscsi
iqn.1994-05.com.Red Hat:2ffa3543d275
```

3. Set up the target system.
   > install targetcli if it was not installed with the OS.
   #yum install targetcli

   > List the block devices and take note for future reference.
   #lsblk

   > Run targetcli. At the present time, the ls command should give us an empty tree as pictured below. As the target is configured, the tree will fill out. (If the tree is not empty, "\>clearconfig confirm=true") will clear it.
   #targetcli
   \>ls

```
targetcli shell version 2.1.fb41
Copyright 2011-2013 by Datera, Inc and
others.
For help on commands, type 'help'.
/> ls
o- /.................................. [...]
  o- backstores......................... [...]
  | o- block ........... [Storage Objects: 0]
  | o- fileio........... [Storage Objects: 0]
  | o- pscsi  .......... [Storage Objects: 0]
  | o- ramdisk  ........ [Storage Objects: 0]
  o- iscsi  .................... [Targets: 0]
  o- loopback  ................. [Targets: 0]
  o- srpt  ..................... [Targets: 0]
```

   > Create backstores out of the block devices. (In our case we used NVMe drives.)
   \>cd backstores/block
   \>create name=<backstorename1> dev=/dev/<device name>
   \>create name=<backstorename2> dev=/dev/<device name>
   … Repeat for additional backstores.

```
\>cd backstores/block
/backstores/block> create name=nvme0
dev=/dev/nvme0n1
 Created block storage object nvme0 using
/dev/nvme0n1.
/backstores/block> create name=nvme1
dev=/dev/nvme1n1
 Created block storage object nvme1 using
/dev/nvme1n1.
/backstores/block> create name=nvme2
dev=/dev/nvme2n1
 Created block storage object nvme2 using
/dev/nvme2n1.
/backstores/block> create name=nvme3
dev=/dev/nvme3n1
 Created block storage object nvme3 using
/dev/nvme3n1.
/backstores/block> create name=nvme4
dev=/dev/nvme4n1
 Created block storage object nvme4 using
/dev/nvme4n1.
```

   > Create a target and give it LUNs created from the previously created backstores.
   \>cd /iscsi
   \>create
   \>cd <new target created >/tpg1/luns
   \>create /backstores/block/<backstorename1>
   \>create /backstores/block/<backstorename2>
   …

```
/backstores/block> cd /iscsi
/iscsi> create
/iscsi> cd iqn.2003-01.org.linux-
iscsi.dmrtk-srvr-
d.x8664:sn.cdf664e56d8c/tpg1/luns
/iscsi/iqn.20...d8c/tpg1/luns> create
/backstores/block/nvme0
Created LUN 0.
/iscsi/iqn.20...d8c/tpg1/luns> create
/backstores/block/nvme1
Created LUN 1.
/iscsi/iqn.20...d8c/tpg1/luns> create
/backstores/block/nvme2
Created LUN 2.
/iscsi/iqn.20...d8c/tpg1/luns> create
/backstores/block/nvme3
Created LUN 3.
/iscsi/iqn.20...d8c/tpg1/luns> create
/backstores/block/nvme4
Created LUN 4.
```

   > Designate an IP address from the target server as a portal for the initiators to reach the

target. Always designate port 3260 as this is the default port for iSCSI.
\>cd ../portals
\>delete 0.0.0.0 3260
\>create <target portal IP> 3260

```
/iscsi/iqn.20...d8c/tpg1/luns> cd
../portals
/iscsi/iqn.20.../tpg1/portals> delete
0.0.0.0 3260
  Deleted network portal 0.0.0.0:3260
/iscsi/iqn.20.../tpg1/portals> create
10.0.8.252 3260
  Using default IP port 3260
  Created network portal 10.0.8.252:3260.
```

> Enable iSER for each portal.
\>cd <target portal IP>:3260
\>enable_iser true

```
/iscsi/iqn.20.../tpg1/portals> cd
10.0.8.252:3260
/iscsi/iqn.20....0.8.252:3260 >
enable_iser true
  iSER enable now: True
```

> Designate which initiators can connect to this target, using the initiator name From Step 2.
\>cd ../../acls
\>create <initiator name>

```
/iscsi/iqn.20....0.8.254:3260> cd
../../acls
/iscsi/iqn.20...d8c/tpg1/acls> create
iqn.1994-05.com.Red Hat:2ffa3543d275
  Created Node ACL for iqn.1994-
05.com.Red Hat:2ffa3543d275
  Created mapped LUN 4.
  Created mapped LUN 3.
  Created mapped LUN 2.
  Created mapped LUN 1.
  Created mapped LUN 0.
```

> Check the configuration using ls to view the whole tree, then save and exit (as shown below).
\>ls
\>saveconfig
\>exit

```
/> ls
o- / .............................................................................[...]
  o- backstores ..................................................................[...]
  | o- block ......................................................[Storage Objects: 5]
  | | o- nvme0 ............................[/dev/nvme0n1 (1.5TiB) write-thru activated]
  | | o- nvme1 ............................[/dev/nvme1n1 (1.5TiB) write-thru activated]
  | | o- nvme2 ............................[/dev/nvme2n1 (1.5TiB) write-thru activated]
  | | o- nvme3.............................[/dev/nvme3n1 (1.5TiB) write-thru activated]
  | | o- nvme4.............................[/dev/nvme4n1 (1.5TiB) write-thru activated]
  | o- fileio .....................................................[Storage Objects: 0]
  | o- pscsi ......................................................[Storage Objects: 0]
  | o- ramdisk ....................................................[Storage Objects: 0]
  o- iscsi ..............................................................[Targets: 1]
  | o- iqn.2003-01.org.linux-iscsi.dmrtk-srvr-d.x8664:sn.cdf664e56d8c .........[TPGs: 1]
  |   o- tpg1 ...................................................[no-gen-acls, no-auth]
  |     o- acls ...............................................................[ACLs: 1]
  |     | o- iqn.1994-05.com.Red Hat:2ffa3543d275 ......................[Mapped LUNs: 5]
  |     |   o- mapped_lun0 .....................................[lun0 block/nvme0 (rw)]
  |     |   o- mapped_lun1 .....................................[lun1 block/nvme1 (rw)]
  |     |   o- mapped_lun2 .....................................[lun2 block/nvme2 (rw)]
  |     |   o- mapped_lun3 .....................................[lun3 block/nvme3 (rw)]
  |     |   o- mapped_lun4 .....................................[lun4 block/nvme4 (rw)]
  |     o- luns ...............................................................[LUNs: 5]
  |     | o- lun0 ........................................[block/nvme0 (/dev/nvme0n1)]
  |     | o- lun1 ........................................[block/nvme1 (/dev/nvme1n1)]
  |     | o- lun2 ........................................[block/nvme2 (/dev/nvme2n1)]
  |     | o- lun3 ........................................[block/nvme3 (/dev/nvme3n1)]
  |     | o- lun4 ........................................[block/nvme4 (/dev/nvme4n1)]
  |     o- portals .........................................................[Portals: 1]
  |       o- 10.0.8.252:3260 .......................................................[iser]
  o- loopback ...........................................................[Targets: 0]
  o- srpt ...............................................................[Targets: 0]
/> saveconfig
  Last 10 configs saved in /etc/target/backup.
  Configuration saved to /etc/target/saveconfig.json
/> exit
```

4. Start a connection from the initiator and login to the target. Once this is done, new devices should show up with lsblk.
   #iscsiadm –m discovery --op=new --op=delete
      --type sendtargets --portal <target portal IP>
      -I iser
   #iscsiadm –m node –l
   #lsblk

The new LUNs will be visible to the initiator.

For our testing, we ran vdbench against our LUNs raw, but in most cases, the user would want to employ parted and mkfs to implement file systems on the new devices.

```
# iscsiadm -m discovery --op=new --op=delete --type sendtargets --portal 10.0.8.253 -I iser
10.0.8.252:3260,1 iqn.2003-01.org.linux-iscsi.dmrtk-srvr-d.x8664:sn. cdf664e56d8c
#iscsiadm -m node -l
Logging in to [iface: iser, target: iqn.2003-01.org.linux-iscsi.dmrtk-srvr-d.x8664:sn.cdf664e56d8c, portal:
10.0.8.252,3260]
#lsblk
NAME           MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda              8:0    0   477G  0 disk
├─sda1           8:1    0   500M  0 part /boot
└─sda2           8:2    0 476.5G  0 part
  ├─rhel-root 253:0    0    50G  0 lvm  /
  ├─rhel-swap 253:1    0     4G  0 lvm  [SWAP]
  └─rhel-home 253:2    0 422.4G  0 lvm  /home
sdb              8:16   0   1.5T  0 disk
sdc              8:32   0   1.5T  0 disk
sdd              8:48   0   1.5T  0 disk
sde              8:64   0   1.5T  0 disk
sdf              8:80   0   1.5T  0 disk
sr0             11:0    1  1024M  0 rom
```

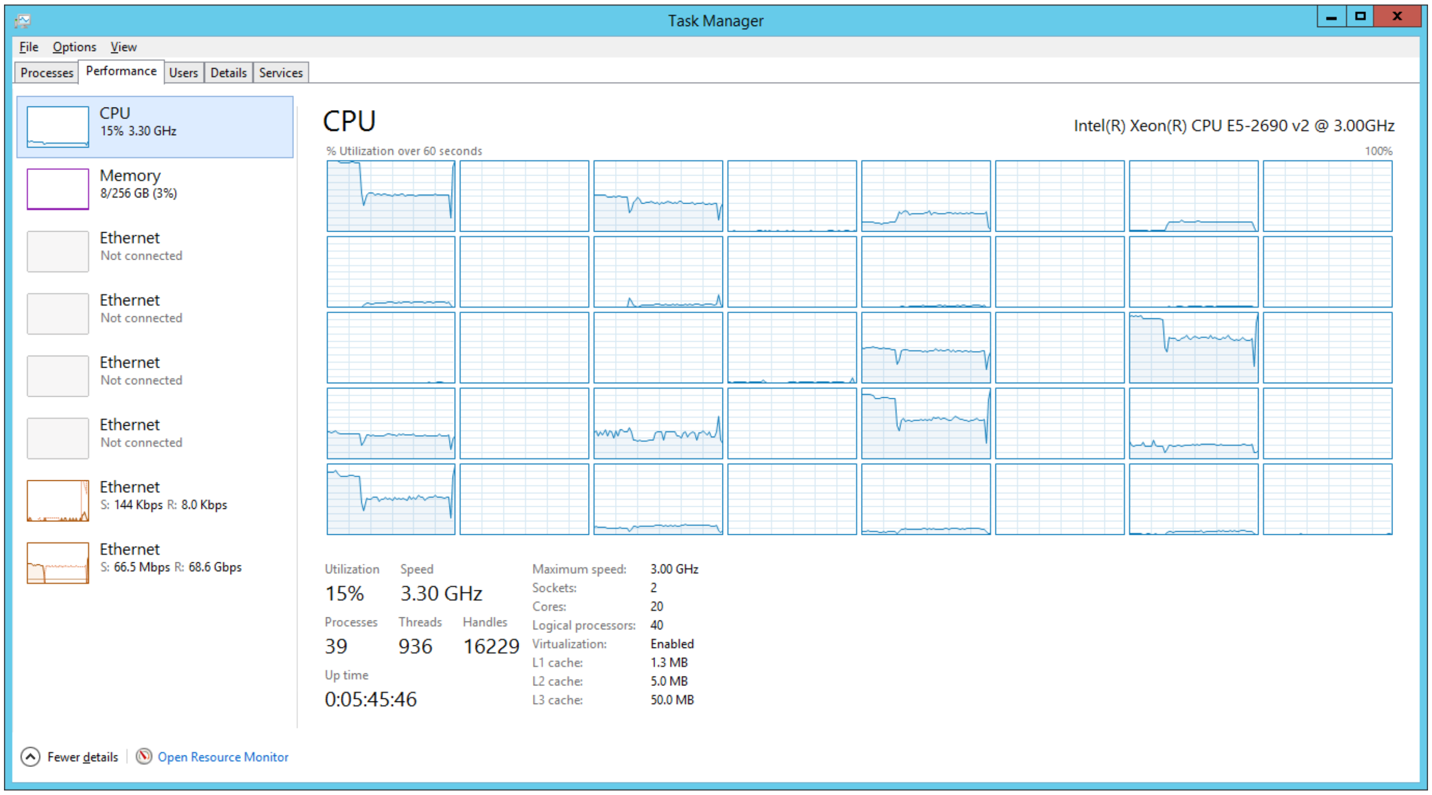
Test Results

Expected Performance from RoCE

When we replace TCP/IP with RoCE, we expect CPU consumption to decrease due to the Remote Direct Memory Access capabilities eliminating CPU cycles use for data movement and transport layer processing. In addition when using TCP/IP, despite configuring Receive Side Scaling (RSS) on NICs, often certain CPU cores are heavily utilized for transport processing while the rest of the cores are left idle. On a smaller server with less processing power, this can result in a throughput performance cap and large latencies when TCP/IP is used. If RoCE is instead deployed on the same small server, we would expect cores to utilized more evenly and less CPU to be required for the same or greater amounts of traffic. If RoCE is deployed on a large server, we may notice the same throughput performance increase due to better traffic distribution between the cores. In most cases this leads to a greater amount of throughput, and in some cases, the throughput is so much greater that the CPU consumption will increase.

Below is a typical example of what a server running SMB over TCP/IP might look like. Several cores are heavily utilized, while many are left idle. The server in our example is a larger, more powerful server.

For additional detail on CPU and core consumption levels, see the ***Processor Consumption Comments*** section on page **30**.



CPU
15% 3.30 GHz

Memory
8/256 GB (3%)

Ethernet
Not connected

Ethernet
Not connected

Ethernet
Not connected

Ethernet
Not connected

Ethernet
S: 144 Kbps R: 8.0 Kbps

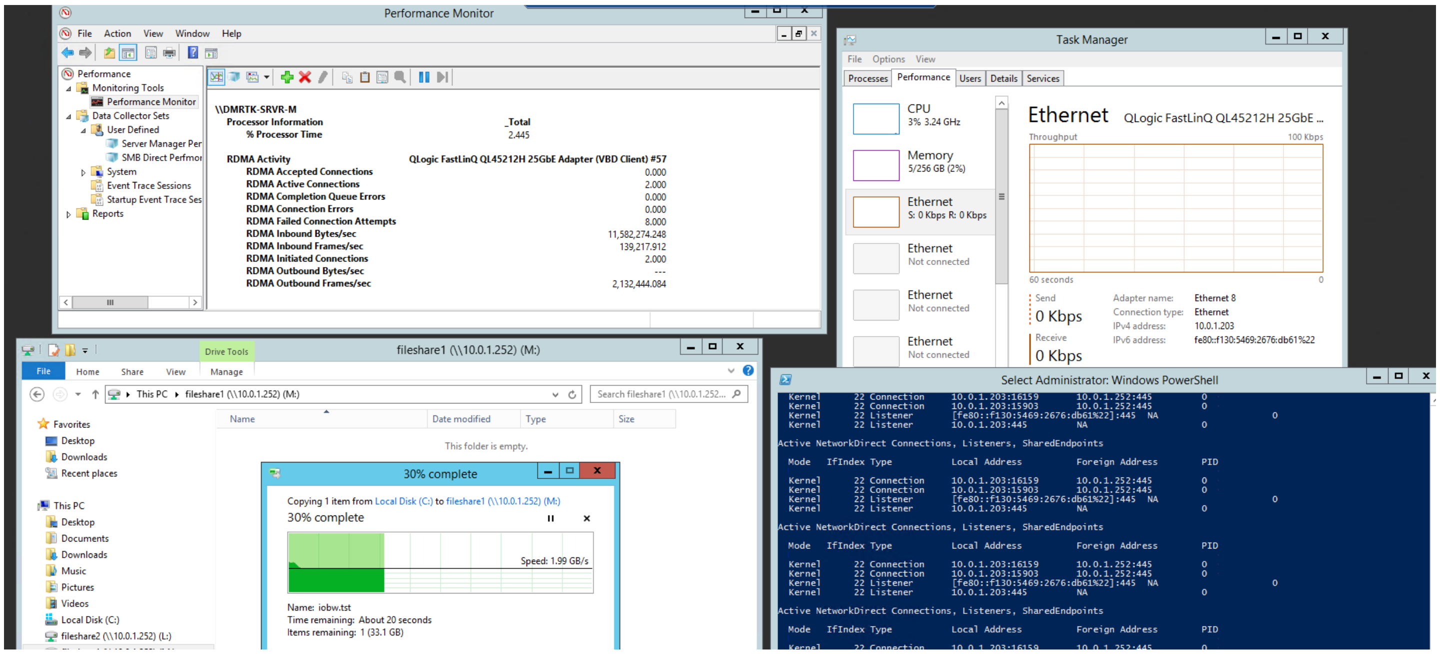
Ethernet
S: 66.5 Mbps R: 68.6 Gbps

CPU Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz

Utilization	Speed	
15%	3.30 GHz	
Processes	**Threads**	**Handles**
39	936	16229
Up time		
0:05:45:46		

Maximum speed:	3.00 GHz
Sockets:	2
Cores:	20
Logical processors:	40
Virtualization:	Enabled
L1 cache:	1.3 MB
L2 cache:	5.0 MB
L3 cache:	50.0 MB

Below is a typical example of what a server running SMB Direct over RoCE might look like. Notice that the Ethernet is not registering any traffic, however a file is being copied across the network and RDMA traffic is showing up in Perfmon. To the right is netstat, showing an active SMB Direct Connection.



```
\\DMRTK-SRVR-M
  Processor Information                                    _Total
     % Processor Time                                       2.445

  RDMA Activity          QLogic FastLinQ QL45212H 25GbE Adapter (VBD Client) #57
     RDMA Accepted Connections                                        0.000
     RDMA Active Connections                                          2.000
     RDMA Completion Queue Errors                                     0.000
     RDMA Connection Errors                                           0.000
     RDMA Failed Connection Attempts                                  8.000
     RDMA Inbound Bytes/sec                                  11,582,274.248
     RDMA Inbound Frames/sec                                    139,217.912
     RDMA Initiated Connections                                       2.000
     RDMA Outbound Bytes/sec                                            ---
     RDMA Outbound Frames/sec                                 2,132,444.084
```

Send
0 Kbps
Receive
0 Kbps

Adapter name:	Ethernet 8
Connection type:	Ethernet
IPv4 address:	10.0.1.203
IPv6 address:	fe80::f130:5469:2676:db61%22

Drive Tools
fileshare1 (\\10.0.1.252) (M:)
File Home Share View Manage
This PC fileshare1 (\\10.0.1.252) (M:)
Search fileshare1 (\\10.0.1.252...
Favorites
Desktop
Downloads
Recent places
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
fileshare2 (\\10.0.1.252) (L:)
Name Date modified Type Size
This folder is empty.
30% complete

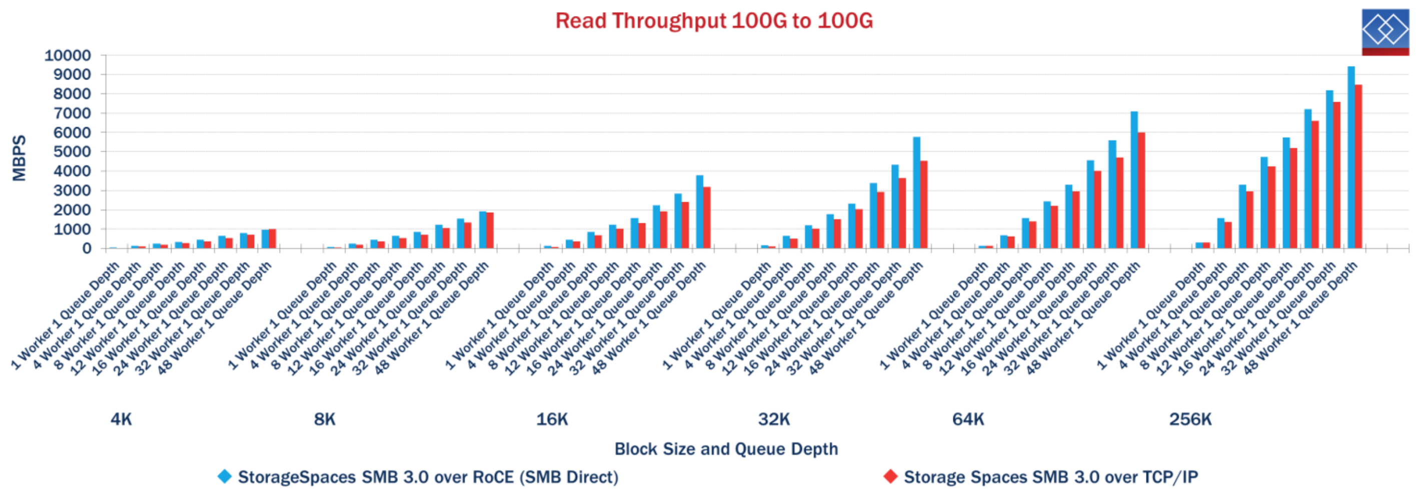
Copying 1 item from Local Disk (C:) to fileshare1 (\\10.0.1.252) (M:)
30% complete
Speed: 1.99 GB/s
Name: iobw.tst
Time remaining: About 20 seconds
Items remaining: 1 (33.1 GB)

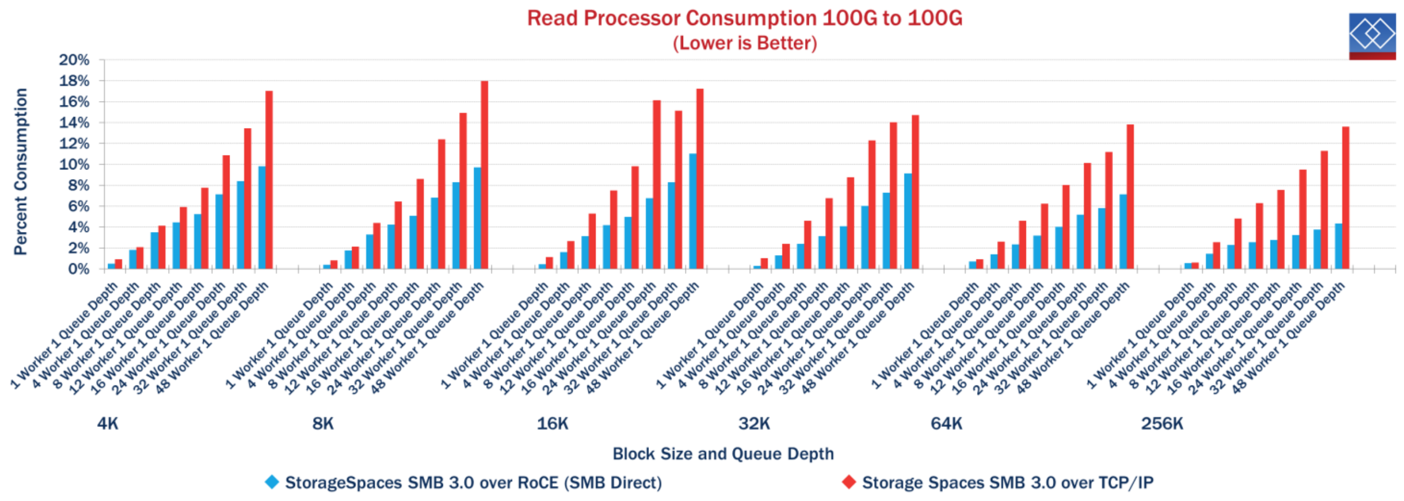
```
  Kernel      22 Connection     10.0.1.203:16159      10.0.1.252:445        0
  Kernel      22 Connection     10.0.1.203:15903      10.0.1.252:445        0
  Kernel      22 Listener       [fe80::f130:5469:2676:db61%22]:445  NA                 0
  Kernel      22 Listener       10.0.1.203:445        NA                    0

Active NetworkDirect Connections, Listeners, SharedEndpoints

  Mode   IfIndex Type           Local Address         Foreign Address       PID

  Kernel      22 Connection     10.0.1.203:16159      10.0.1.252:445        0
  Kernel      22 Connection     10.0.1.203:15903      10.0.1.252:445        0
  Kernel      22 Listener       [fe80::f130:5469:2676:db61%22]:445  NA                 0
  Kernel      22 Listener       10.0.1.203:445        NA                    0

Active NetworkDirect Connections, Listeners, SharedEndpoints

  Mode   IfIndex Type           Local Address         Foreign Address       PID

  Kernel      22 Connection     10.0.1.203:16159      10.0.1.252:445        0
  Kernel      22 Connection     10.0.1.203:15903      10.0.1.252:445        0
  Kernel      22 Listener       [fe80::f130:5469:2676:db61%22]:445  NA                 0
  Kernel      22 Listener       10.0.1.203:445        NA                    0

Active NetworkDirect Connections, Listeners, SharedEndpoints

  Mode   IfIndex Type           Local Address         Foreign Address       PID

  Kernel      22 Connection     10.0.1.203:16159      10.0.1.252:445        0
```

100GbE Windows Storage Spaces – Reads

Throughput limitations do not hit line rate for writes. It is suspected that we are nearing the performance limit for the storage utilized (Windows Storage Spaces with NVMe drives.)

For the Windows Storage Spaces tests, the throughput was higher and the CPU consumption was lower for the RoCE technology compared to traditional TCP/IP.

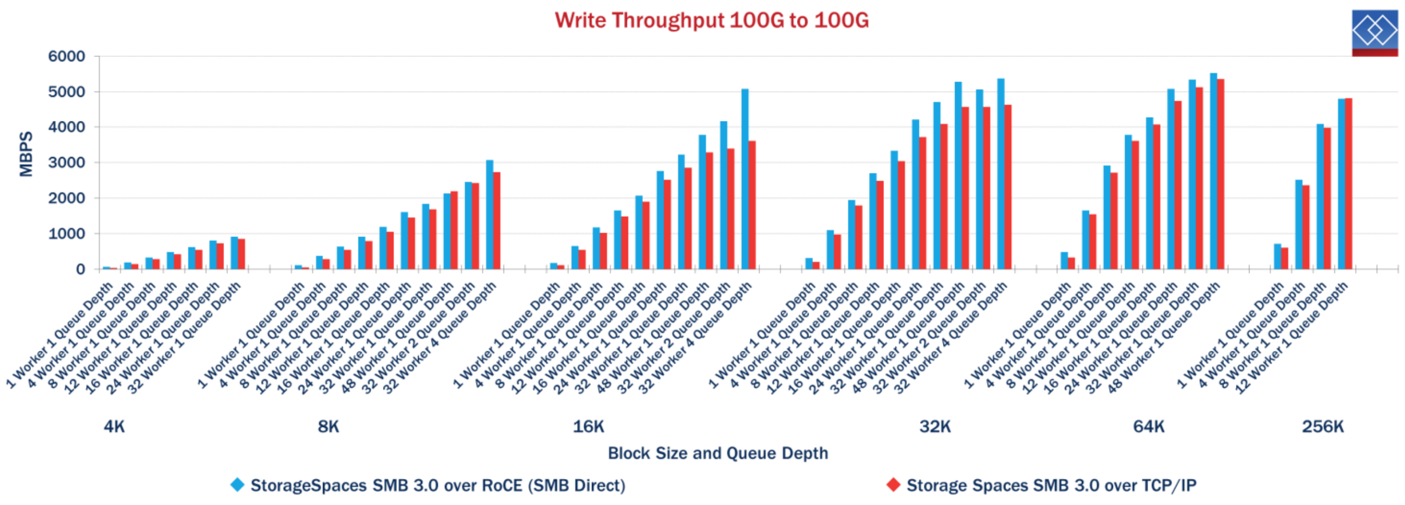


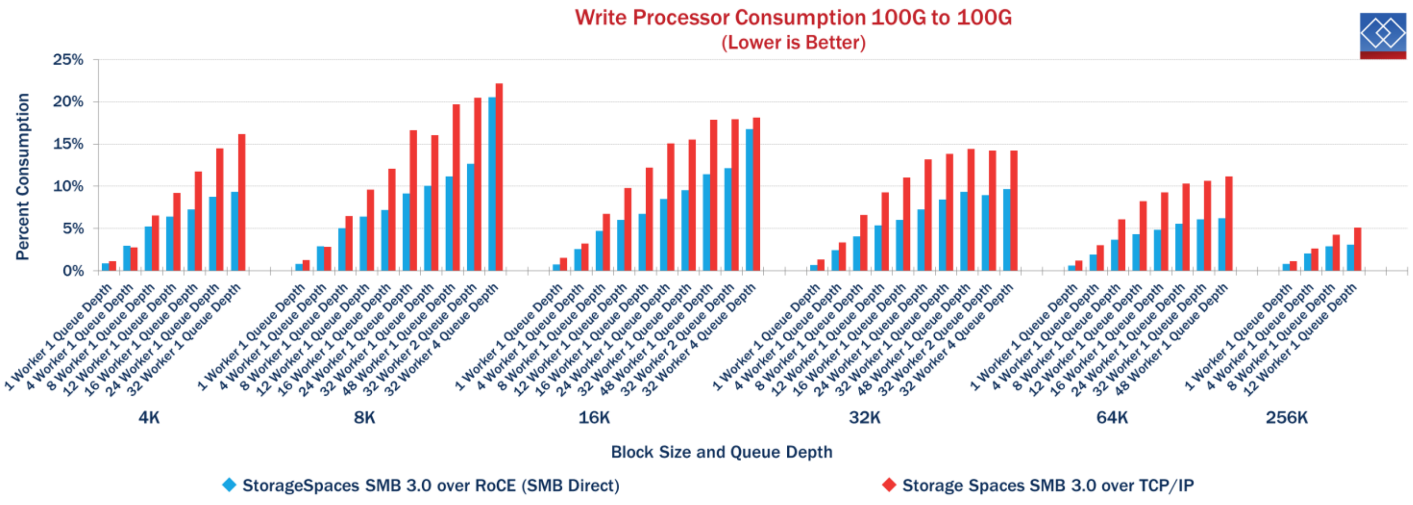


100GbE Windows Storage Spaces – Writes

Throughput limitations do not hit line rate for writes. It is suspected that we are nearing the performance limit for the storage utilized (Windows Storage Spaces with NVMe drives.)

For the Windows Storage Spaces tests, the throughput was higher and the CPU consumption was lower for the RoCE technology compared to traditional TCP/IP.

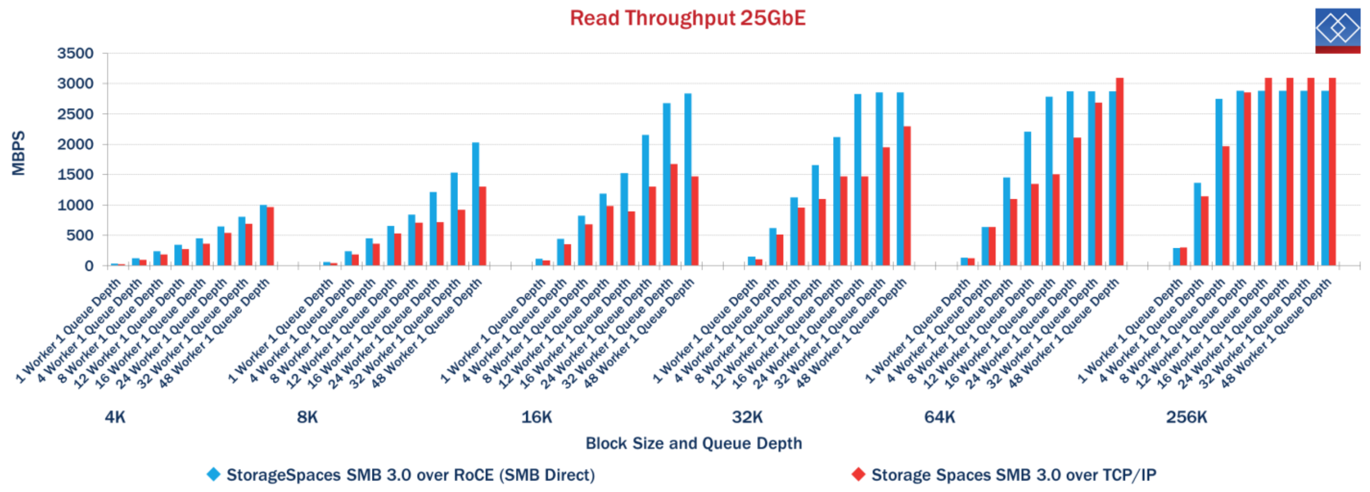


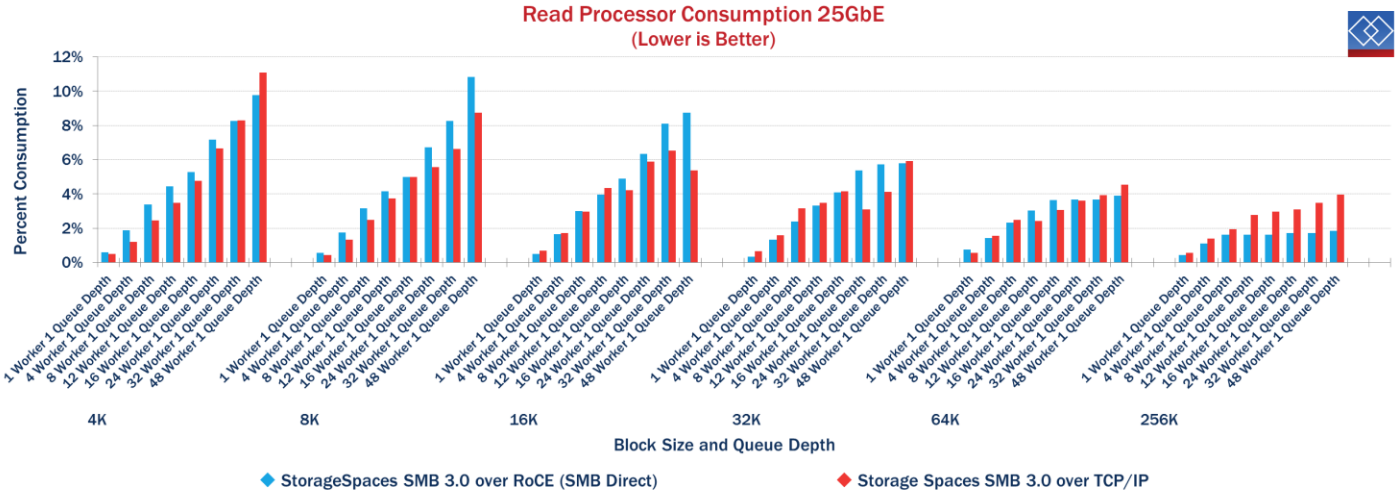


25GbE Windows Storage Spaces – Reads

In this example, the client is capable of processing 4x's as much IO as our NIC allows. This results in the NIC causing much of the bottleneck, in the case of both TCP/IP and RoCE.

For the Windows Storage Spaces tests, the throughput was higher and the CPU consumption was lower for the RoCE technology compared to traditional TCP/IP.

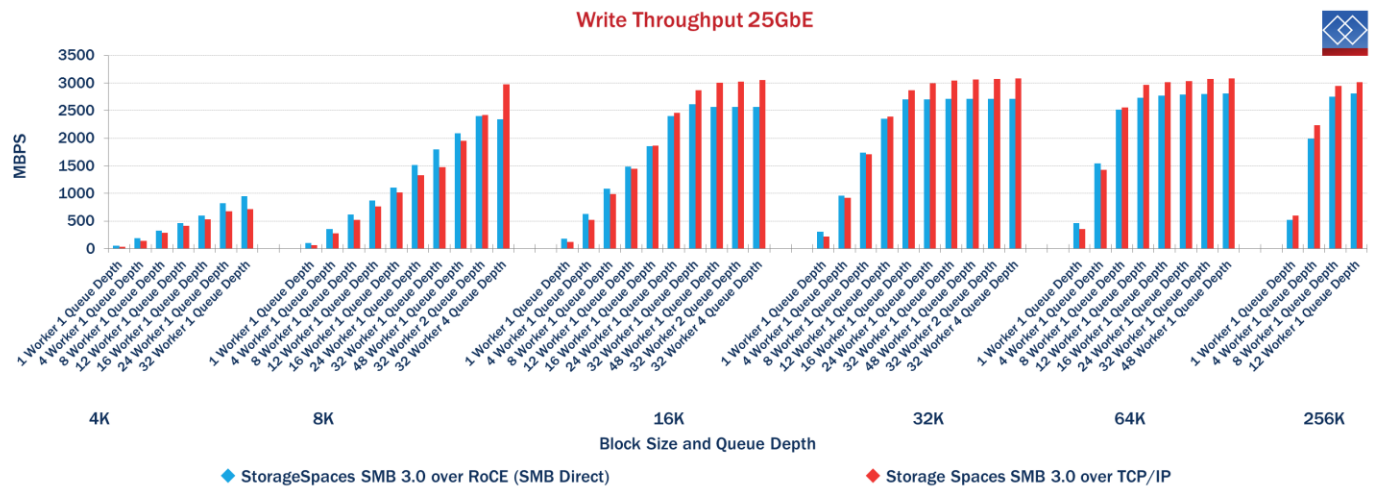


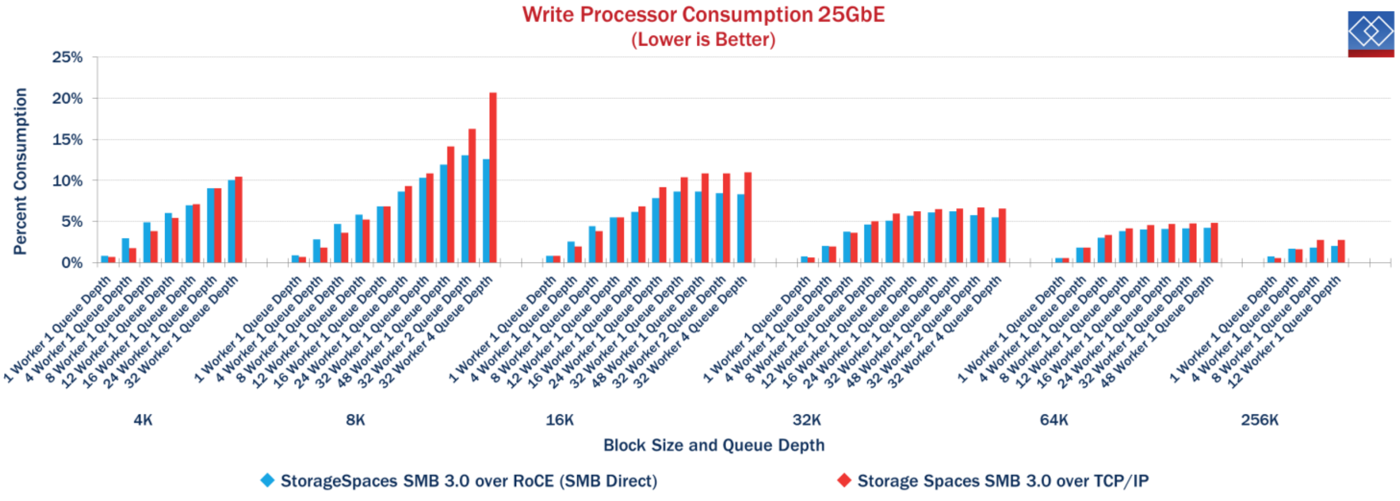


25GbE Windows Storage Spaces – Writes

In this example, the client is capable of processing 4x's as much IO as our NIC allows. This results in the NIC causing much of the bottleneck, in the case of both TCP/IP and RoCE.

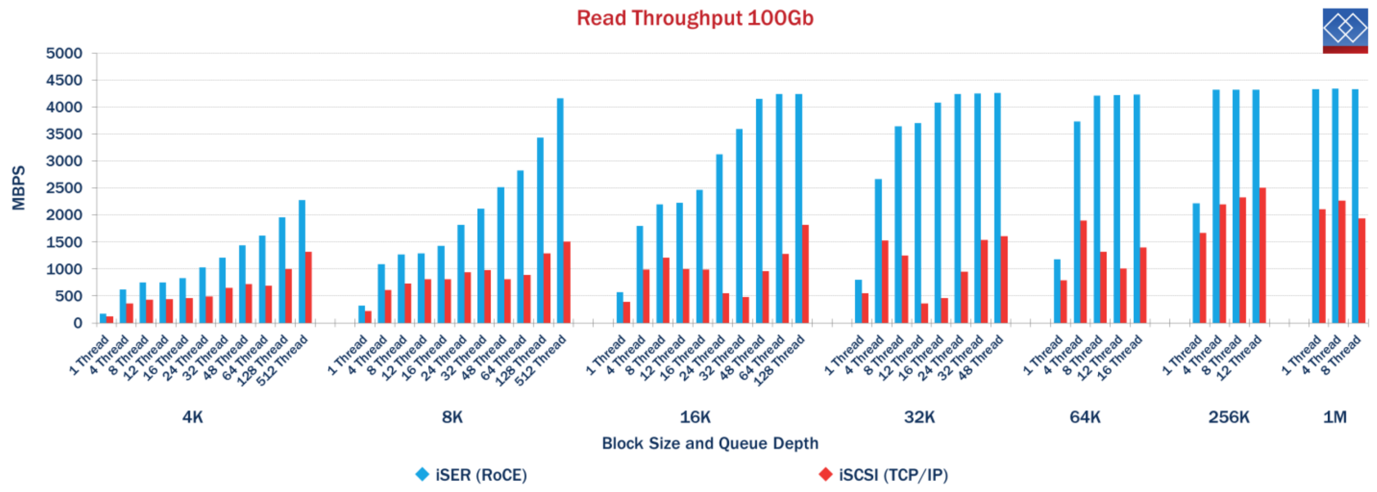
For the Windows Storage Spaces tests, the throughput was generally higher and the CPU consumption was generally lower for the RoCE technology compared to traditional TCP/IP.

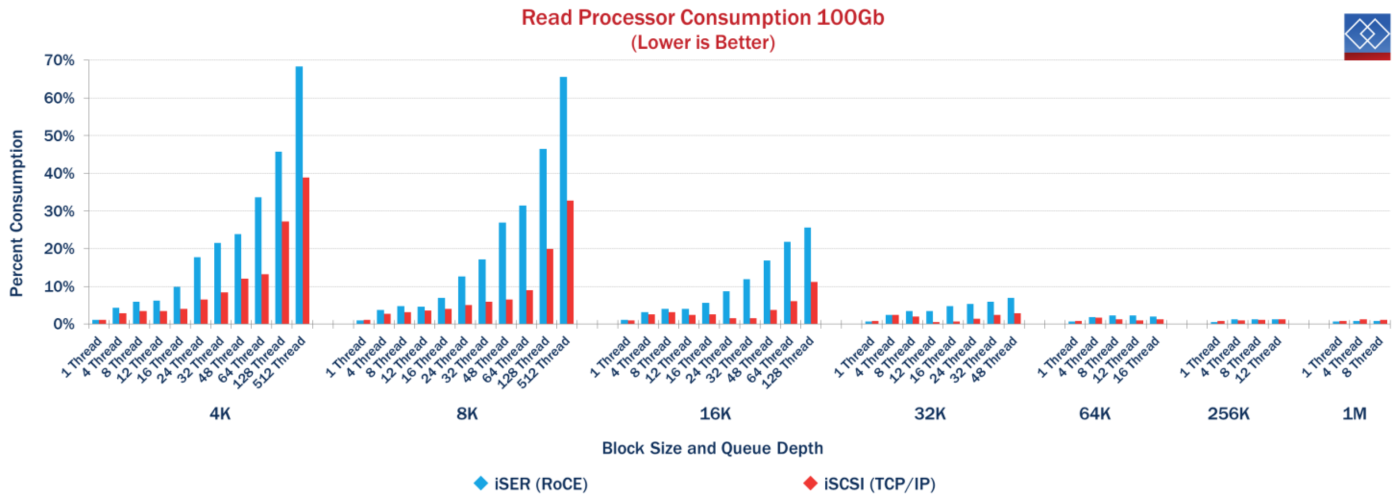




100GbE Linux iSER/iSCSI – Reads

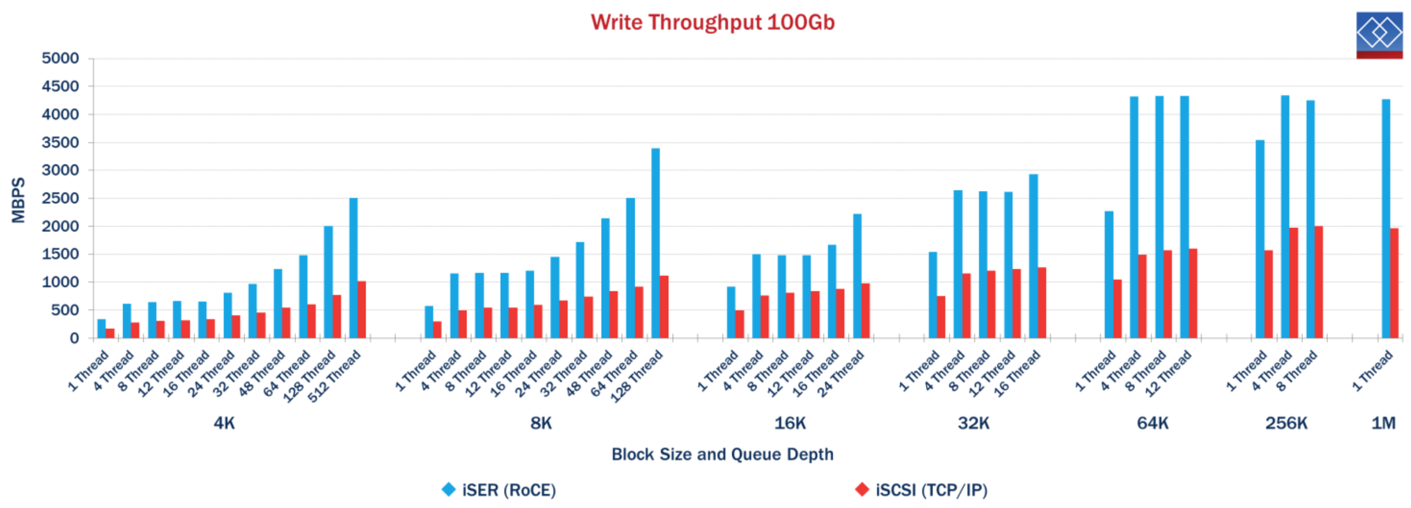
iSCSI over TCP/IP has a low performance cap, enabling RoCE to vastly outperform it in throughput in both the case of 100Gb and 25Gb connections.

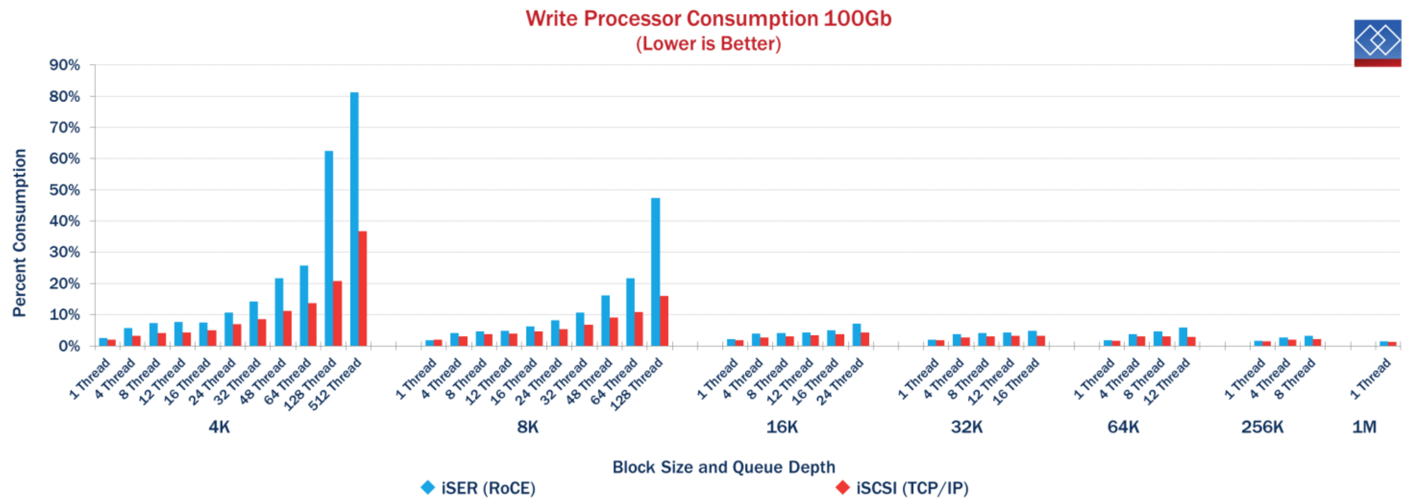




100GbE Linux iSER/iSCSI – Writes

iSCSI over TCP/IP has a low performance cap, enabling RoCE to vastly outperform it in throughput in both the case of 100Gb and 25Gb connections.

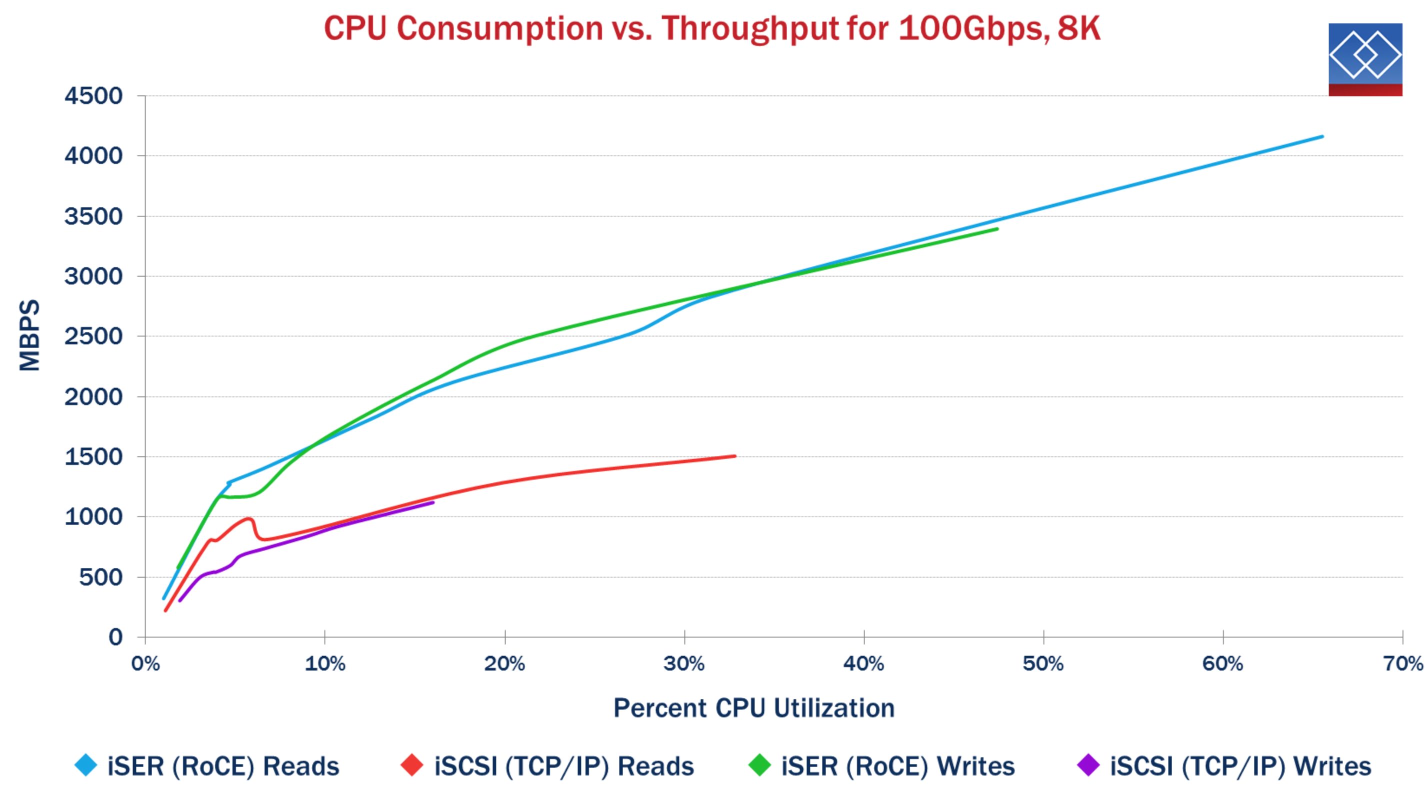




100GbE Linux iSER/iSCSI CPU Consumption vs. Throughput

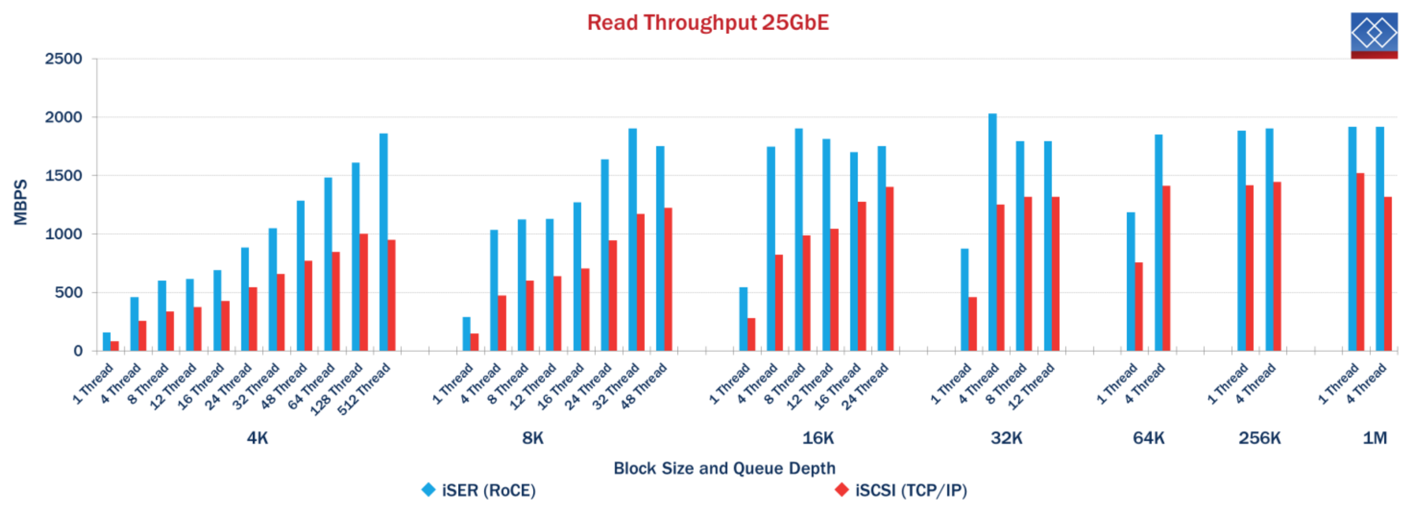
Taking a deeper look into CPU consumption and throughput, we took the 8K block size data and plotted the results on the chart below.

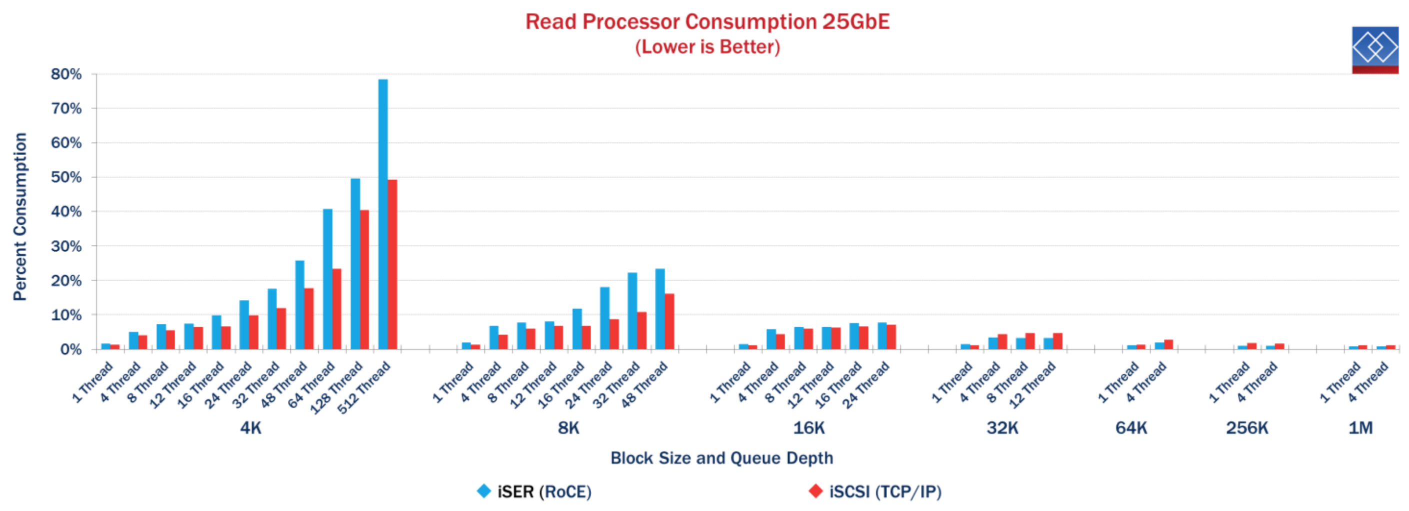
Using iSER (RoCE) provided higher throughput at various levels of CPU consumption compared to traditional TCP/IP. In addition, using the traditional TCP/IP technology the throughput levels did not reach the levels achieved using the RoCE technology.



25GbE Linux iSER/iSCSI – Reads

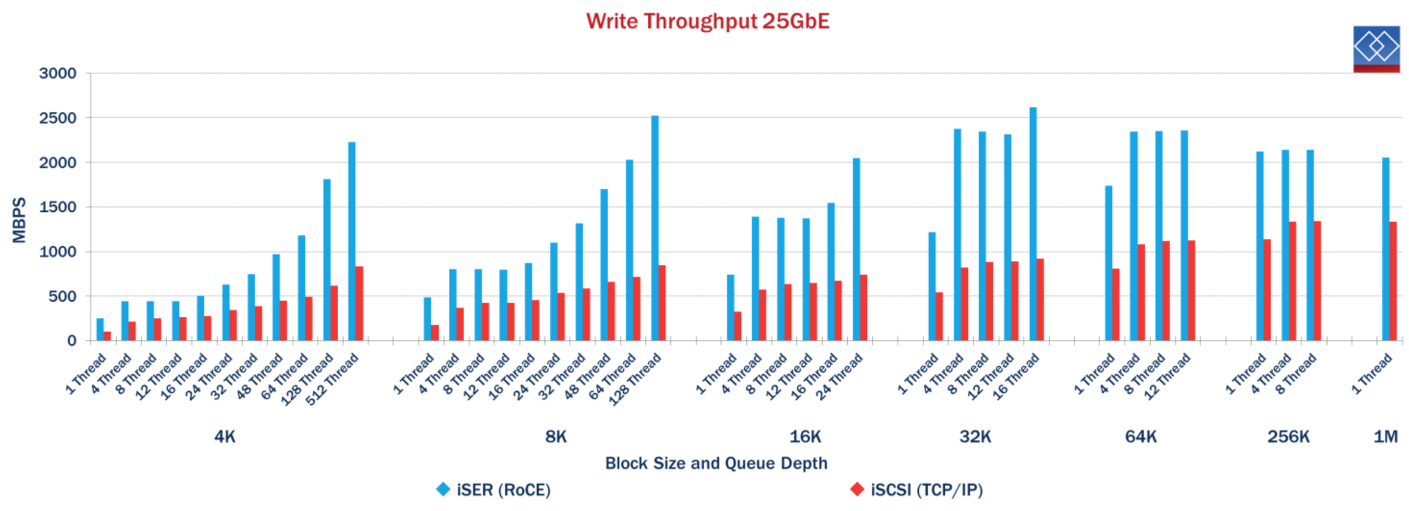
iSCSI over TCP/IP has a low performance cap, enabling RoCE to vastly outperform it in throughput in both the case of 100Gb and 25Gb connections.

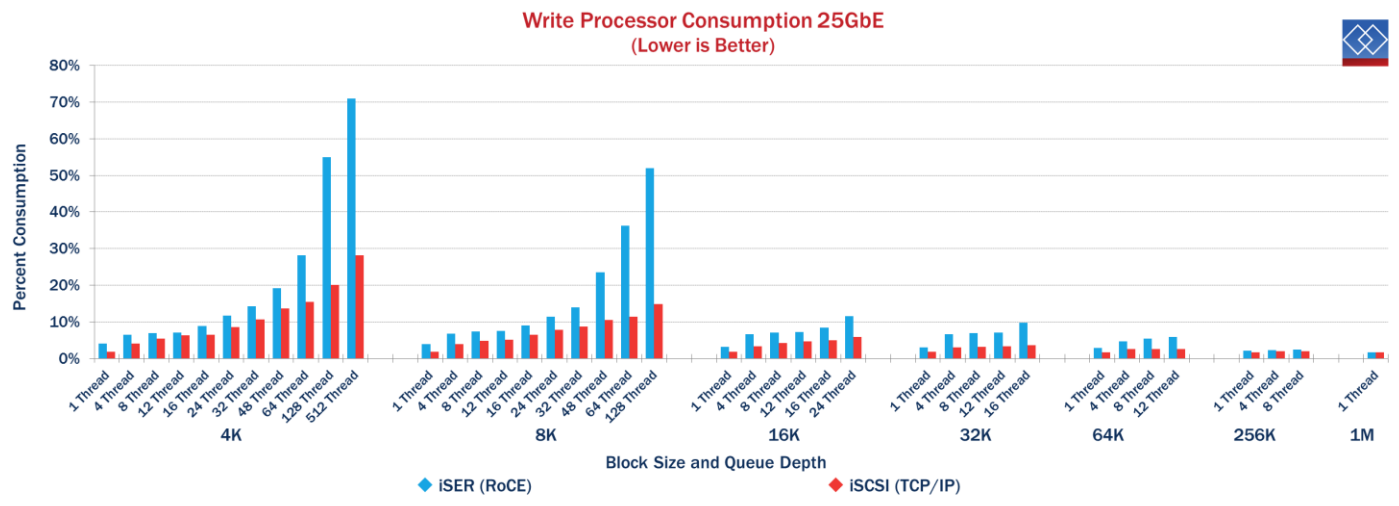




25GbE Linux iSER/iSCSI – Writes

iSCSI over TCP/IP has a low performance cap, enabling RoCE to vastly outperform it in throughput in both the case of 100Gb and 25Gb connections.





Processor Consumption Comments

It should be noted that the client in these examples has plenty of processor to spare. Had the client been equipped with less processor, we would expect a cap on the throughput performance shown earlier and processor be near 100%. Below we show what the utilization was of the individual processor cores on our client. We can see that certain cores are heavily utilized.

Processor Consumption for 4K Reads with TCP/IP

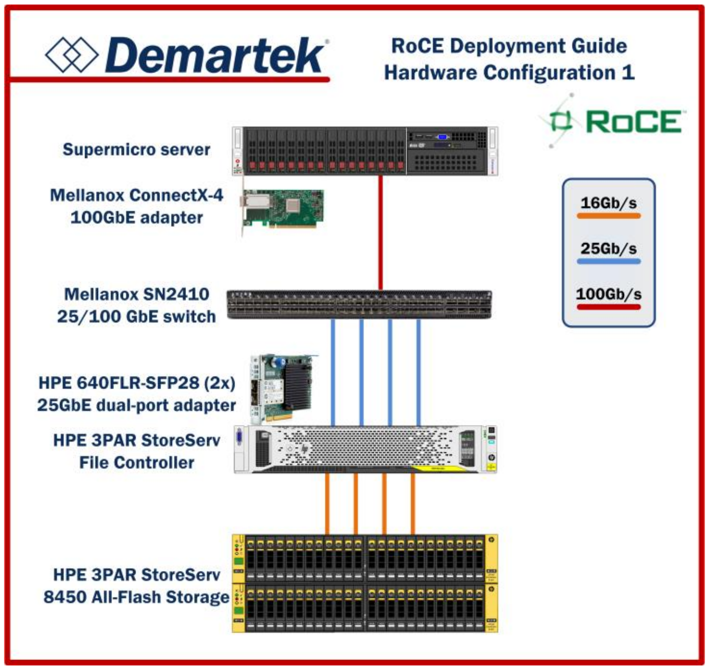
Processor	16 Worker 1 Queue Depth	24 Worker 1 Queue Depth	32 Worker 1 Queue Depth	48 Worker 1 Queue Depth
0	37%	54%	67%	83%
1	0%	0%	0%	0%
2	22%	30%	36%	47%
3	1%	1%	1%	0%
4	13%	19%	25%	35%
5	0%	0%	0%	0%
6	6%	9%	14%	19%
7	0%	0%	0%	0%
8	0%	1%	1%	2%
9	0%	0%	0%	0%
10	0%	1%	0%	0%
11	0%	0%	0%	0%
12	0%	0%	0%	0%
13	0%	0%	0%	0%
14	0%	0%	0%	0%
15	0%	0%	0%	0%
16	0%	0%	0%	0%
17	0%	0%	0%	0%
18	0%	0%	0%	0%
19	1%	1%	2%	2%
20	29%	37%	44%	52%
21	0%	0%	0%	0%
22	50%	65%	78%	90%
23	0%	0%	0%	0%
24	21%	28%	35%	46%
25	0%	0%	0%	0%
26	19%	29%	35%	45%
27	0%	0%	0%	0%
28	39%	59%	72%	87%
29	0%	0%	0%	0%
30	11%	17%	22%	35%
31	0%	0%	0%	0%
32	35%	52%	65%	83%
33	0%	0%	0%	0%
34	5%	11%	15%	28%
35	0%	0%	0%	0%
36	2%	6%	10%	23%
37	0%	0%	0%	0%
38	1%	2%	6%	17%
39	0%	0%	0%	1%

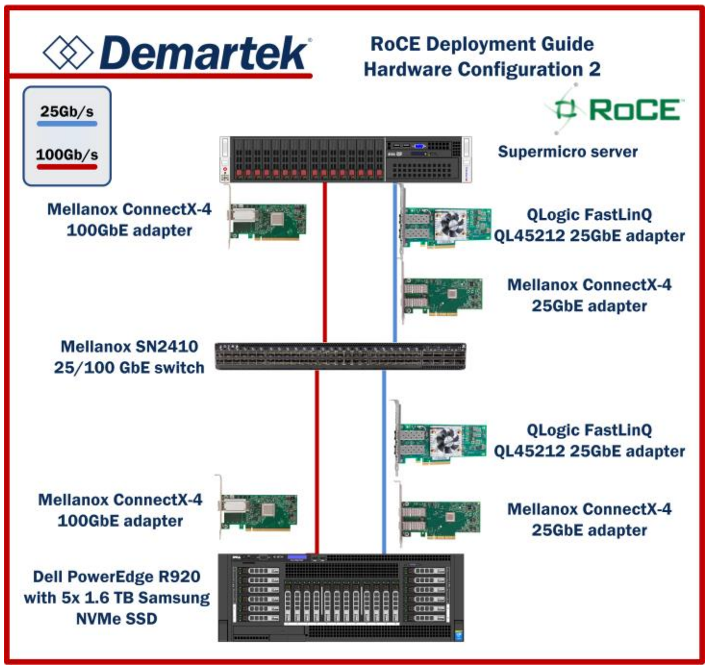
Processor Consumption for 256 K Reads with TCP/IP

Processor	16 Worker 1 Queue Depth	24 Worker 1 Queue Depth	32 Worker 1 Queue Depth	48 Worker 1 Queue Depth
0	57%	70%	82%	94%
1	0%	0%	0%	0%
2	20%	24%	29%	33%
3	1%	2%	1%	1%
4	3%	6%	4%	6%
5	0%	0%	0%	0%
6	0%	1%	0%	1%
7	0%	0%	0%	0%
8	0%	0%	0%	0%

9	0%	0%	0%	0%
10	0%	0%	0%	1%
11	0%	0%	0%	0%
12	0%	0%	0%	0%
13	0%	0%	0%	0%
14	0%	0%	0%	0%
15	0%	0%	0%	0%
16	0%	0%	0%	0%
17	0%	0%	0%	0%
18	0%	0%	0%	0%
19	1%	1%	2%	2%
20	24%	32%	35%	36%
21	0%	0%	0%	0%
22	54%	66%	80%	94%
23	0%	0%	0%	0%
24	15%	18%	25%	31%
25	0%	0%	0%	0%
26	13%	17%	23%	31%
27	0%	0%	0%	0%
28	50%	62%	75%	92%
29	1%	0%	0%	0%
30	2%	4%	6%	13%
31	0%	0%	0%	0%
32	47%	60%	70%	90%
33	0%	0%	0%	0%
34	1%	1%	2%	7%
35	0%	0%	0%	0%
36	0%	0%	0%	2%
37	0%	0%	0%	0%
38	0%	0%	0%	2%
39	1%	1%	0%	1%

RoCE Configuration Diagrams


Demartek
RoCE Deployment Guide
Hardware Configuration 1
RoCE
Supermicro server
Mellanox ConnectX-4 100GbE adapter
16Gb/s
25Gb/s
100Gb/s
Mellanox SN2410 25/100 GbE switch
HPE 640FLR-SFP28 (2x) 25GbE dual-port adapter
HPE 3PAR StoreServ File Controller
HPE 3PAR StoreServ 8450 All-Flash Storage


Demartek
RoCE Deployment Guide
Hardware Configuration 2
25Gb/s
100Gb/s
RoCE
Supermicro server
Mellanox ConnectX-4 100GbE adapter
QLogic FastLinQ QL45212 25GbE adapter
Mellanox ConnectX-4 25GbE adapter
Mellanox SN2410 25/100 GbE switch
QLogic FastLinQ QL45212 25GbE adapter
Mellanox ConnectX-4 100GbE adapter
Mellanox ConnectX-4 25GbE adapter
Dell PowerEdge R920 with 5x 1.6 TB Samsung NVMe SSD

Servers

Supermicro Server

> 2x Intel Xeon E5-2690 v2, 3.0GHz, 20 total cores, 40 total threads

> 256 GB RAM

Dell PowerEdge R920 Server

> 4x Intel Xeon E7-4880 v2, 2.5GHz, 60 total cores, 120 total threads

> 416 GB RAM

> 5x Samsung SM-1715 1.6 GB NVMe SSD

Switches

Mellanox SN2410 25/100GbE Switch

> 48 ports 25GbE

> 8 ports 100GbE

Adapters

Mellanox

> ConnectX-4 Lx EN single-port 25GbE

> ConnectX-4 EN single-port 100GbE

QLogic

> FastLinQ QL45212 dual-port 25GbE

Storage Systems

HPE Storage

> HPE 3PAR StoreServ File Controller

> HPE 3PAR StoreServ 8450 All-Flash Storage

Conclusion

Generally, we found that using RoCE technology provided higher throughput and lower latency for 25GbE and 100GbE applications.

Having knowledge of Ethernet switch commands is beneficial to deploying RoCE solutions, and these commands may vary by switch vendor.

Deploying RoCE technology can boost the performance or lower the CPU consumption of applications that use Ethernet networks.