InfiniBand/RoCE Setup and Performance on vSphere 7.x

Using NVIDIA ConnectX adapter cards for HPC and machine learning workloads on vSphere in DirectPath I/O mode
Performance Study – September 13, 2022

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

Table of Contents

1. Introduction

Applications today benefit from the use of high-speed NICs for performance-critical workloads and hardware accelerators like GPUs for machine learning workloads or virtual desktops. To use accelerators in a virtual machine (VM), VMware DirectPath I/O (PCI passthrough) allows the direct assignment of physical PCI functions to VMs with minimal intervention from the ESXi host. DirectPath I/O improves the performance of the VMs, but it requires settings that use device-specific PCIe addresses. Dynamic DirectPath I/O, recently introduced as assignable hardware in VMware vSphere® 7.x, overcomes the above constraint to specify the device PCIe address and instead leverages attributes of the PCIe devices. Thus, Dynamic DirectPath I/O enables several important features of vSphere, such as high availability (HA) and DRS, to perform the initial placement of all available PCIe devices for a VM.

In this third installment of a series of technical guides, we walk through the steps to simultaneously enable DirectPath I/O with InfiniBand (IB) and RDMA over converged Ethernet (RoCE) on the separate ports of a dual-port Mellanox ConnectX-5 (CX-5) VPI adapter card in vSphere 7.x. Since the vSphere Client uses passthrough mode to expose the physical PCI functions directly to the VMs for both DirectPath I/O and Dynamic DirectPath I/O, we expect the performance to be similar between these two configurations. You can leverage the steps to enable IB or RoCE Dynamic DirectPath I/O on your adapters as needed. Even though we demonstrate this on different ports of a single card, we don't expect that there will be production environments running RoCE and IB on multiple ports of a single adapter.

We cover the steps from BIOS, ESXi, and the vSphere Client to the functionality test on the VM guest operating system. We also introduce how to use the vHPC toolkit, an open-source tool developed by VMware, to speed up the deployment of an HPC cluster in vSphere. Some of the steps are referenced from VMware documentation [1][2][7][8] about how to configure a VM to use DirectPath I/O devices and NVIDIA documentation [3][4][5][9] about how to set up and configure the firmware and driver of Mellanox ConnectX adapter cards in an ESXi environment.

We conclude with performance results from five HPC applications across multiple vertical domains and demonstrate that a virtual HPC cluster can achieve performance similar to a bare metal HPC cluster.

2. Configuration Workflow

For simplicity, Figure 1 illustrates the general idea of the DirectPath I/O InfiniBand and RoCE configuration on 2 VMs. Unlike the procedure we outlined in the previous two papers of this series, here we use an IB switch for the IB traffic and an Ethernet switch for the RoCE traffic simultaneously. When we do performance testing using benchmarks, we configure networking for 16 VMs on 16 servers.

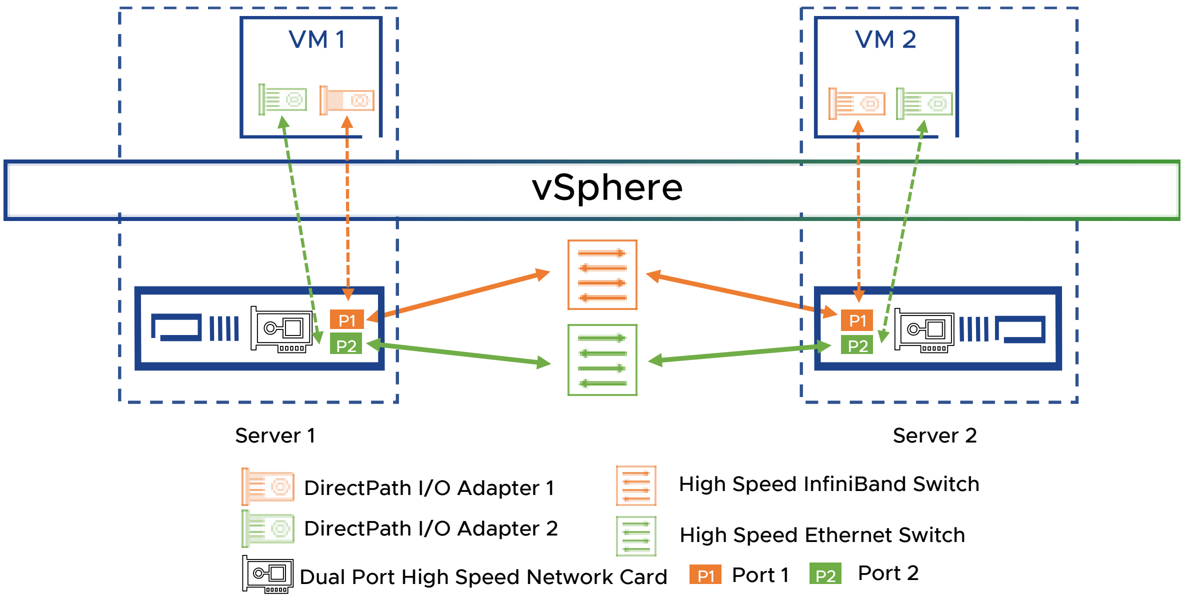


Figure 1. Illustration of IB and RoCE DirectPath I/O configuration

Figure 2 presents a flow chart demonstrating all the steps to enable DirectPath I/O IB and RoCE on the two ports of a ConnectX-5 adapter. First, we clean the VDS since it was set up for SR-IOV and reconfigure the firmware to clear the prior SR-IOV settings, and then configure the two ports of our CX-5 card to IB and RoCE separately. If there is no prior configuration on your adapter, you can skip the first two steps in the dashed box. Next, we enable the two ports as DirectPath I/O devices in the vSphere Client and attach both to a VM. The whole configuration workflow is generally divided into four stages, starting from the BIOS to ESXi to vSphere Client, and to the VM guest.

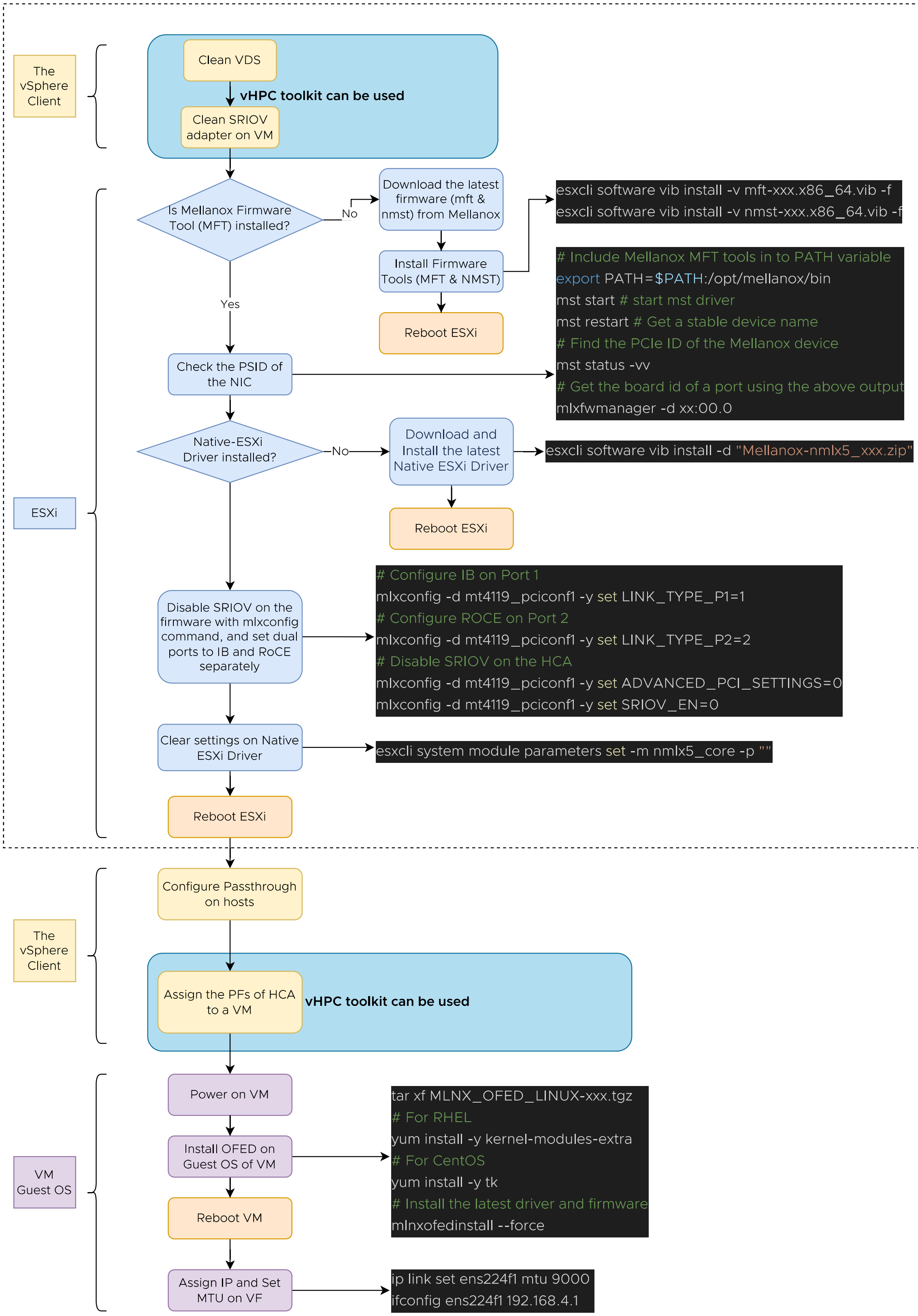


Figure 2. Flow chart to enable IB and RoCE DirectPath I/O on NVIDIA Mellanox ConnectX-5 in ESXi 7.x

2.1 BIOS configuration

DirectPath I/O has no requirement on the BIOS setting. If the processor settings **Virtualization Technology** and **SR-IOV Global** are enabled in the CPU settings of the BIOS, you can still use devices in DirectPath I/O mode.

> **Best Practice:** For HPC workloads, **Performance Per Watt (OS)** is the recommended system power profile setting (Figure 3). HPE servers have a similar profile setting. Then, when we get to the step where we can set the ESXi power management, we will choose **High Performance**.

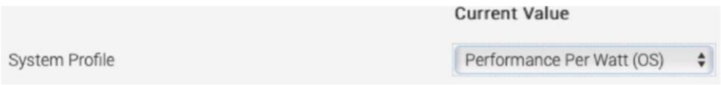


Figure 3. Power profile in BIOS

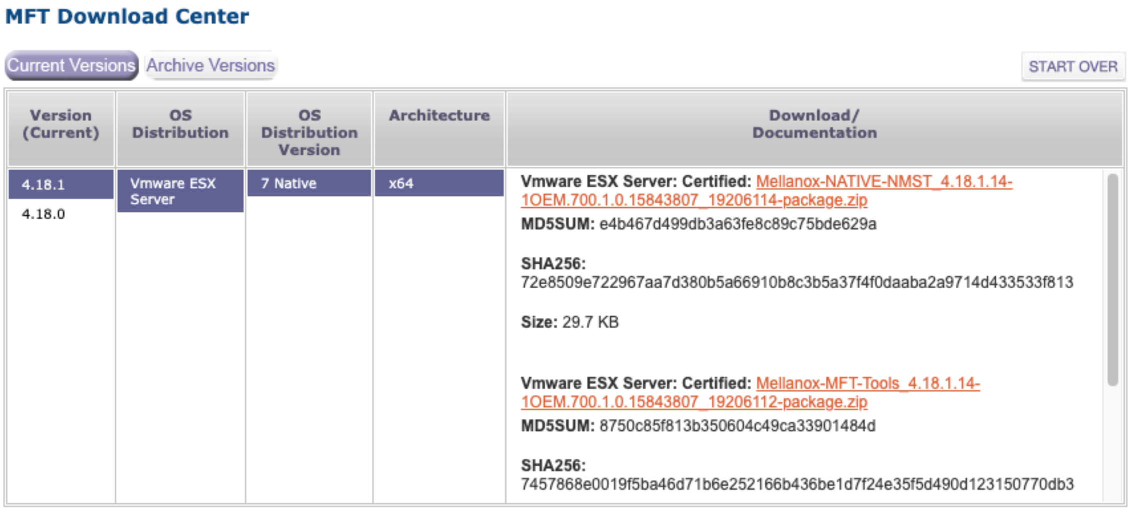
2.2. ESXi configuration

If SR-IOV settings are enabled in firmware and the ESXi native driver, we need to clear them. We need to download and install any missing software as described in section 2.2.1 and 2.2.2. If you have already installed them, please directly go to section 2.2.3. (Most of the steps in this section refer to NVIDIA's virtualization documentation [5], and you can check additional information there.)

2.2.1. Install Mellanox firmware tools

First, we need to check whether the latest firmware tools [3] are installed on our ESXi host. Figure 4 shows that there are the two packages included: NMST and MFT.

> **Best Practice:** We recommend downloading the latest NVIDA firmware tools to the vSAN datastore or network file system (NFS) so that all ESXi hosts in the cluster can conveniently access these files for large-scale deployment.



MFT Download Center

Current Versions | Archive Versions | START OVER

Version (Current)	OS Distribution	OS Distribution Version	Architecture	Download/ Documentation
4.18.1 4.18.0	Vmware ESX Server	7 Native	x64	**Vmware ESX Server: Certified:** Mellanox-NATIVE-NMST_4.18.1.14-1OEM.700.1.0.15843807_19206114-package.zip **MD5SUM:** e4b467d499db3a63fe8c89c75bde629a **SHA256:** 72e8509e722967aa7d380b5a66910b8c3b5a37f4f0daaba2a9714d433533f813 **Size:** 29.7 KB **Vmware ESX Server: Certified:** Mellanox-MFT-Tools_4.18.1.14-1OEM.700.1.0.15843807_19206112-package.zip **MD5SUM:** 8750c85f813b350604c49ca33901484d **SHA256:** 7457868e0019f5ba46d71b6e252166b436be1d7f24e35f5d490d123150770db3

Figure 4. Download firmware tools from the NVIDIA Mellanox website

After extracting the two zip files, use the following commands to install them on the ESXi host as shown in Figure 5. We also add the installation directory to the `$PATH` variable for convenience in the remaining steps. When the installation completes, reboot the host.

```
# Install MFT and NMST
[esxi]$ esxcli software vib install -v mft-xxx.x86_64.vib -f
[esxi]$ esxcli software vib install -v nmst-xxx.x86_64.vib -f
# Best Practice: Add installation directory to PATH variable
[esxi]$ echo 'export PATH=$PATH:/opt/mellanox/bin' >> etc/profile.local
# Reboot the host for the first time
```

Figure 5. Commands to install Firmware Tools.

After the reboot, start the firmware tools service and check whether the tools function well—for example, by querying the status, firmware version and board id, and updating the firmware online.

```
# start mst driver
[esxi]$ mst start

# Restart mst to get a stable device name
[esxi]$ mst restart

# Find the PCIe ID of the Mellanox device
[esxi]$ mst status -vv
PCI devices:
------------
DEVICE_TYPE             MST                           PCI
ConnectX4LX(rev:0)      mt4117_pciconf0               1a:00.0
ConnectX4LX(rev:0)      mt4117_pciconf0.1             1a:00.1
ConnectX5(rev:0)        mt4119_pciconf1               3b:00.0
ConnectX5(rev:0)        mt4119_pciconf1.1             3b:00.1

# Get the board id of a port using the output of the above command
[esxi]$ mlxfwmanager -d 3b:00.0
Querying Mellanox devices firmware ...
Device #1:
----------
  Device Type:      ConnectX5
  Part Number:      MCX556A-ECA_Ax
  Description:      ConnectX-5 VPI adapter card; EDR IB (100Gb/s) and 100GbE;
dual-port QSFP28; PCIe3.0 x16; tall bracket; ROHS R6
  PSID:             MT_0000000008
  PCI Device Name:  3b:00.0
...
  Versions:         Current        Available
     FW             16.32.1010     N/A
     PXE            3.6.0502       N/A
     UEFI           14.25.0017     N/A
  Status:           No matching image found

# If the current firmware version is lower than the online version,
# an update can be done by this command.
[esxi]$ mlxfwmanager --online -u -d 3b:00.0 -f
```

Figure 6. Commands to query the HPC NIC with firmware tools.

Note: The `mst status` command in Figure 6 discovers four devices: we have two adapter cards on our ESXi host, and each card has two ports. The `ConnectX4LX` card is used as a service network interface card (NIC) for connection to the vSphere Client and vSAN, while the `ConnectX5`, on which we intend to enable IB and RoCE passthrough, is used for the HPC/ML workload. Setting up two NICs is typical for an HPC workload using vSphere [10].

Note: To query the firmware version and board ID (PSID) of our ConnectX-5, we use the `mlxfwmanager` command with the peripheral component interconnect express (PCIe) ID generated by `mst status`, which is `3b:00.0` in this case. We are currently using the 16.32.1010 firmware version. The PSID of the host channel adapter (HCA) is `MT_0000000008`, which we compare with the latest online firmware version on the NVIDIA website in Figure 7. If online updating the firmware is not available, choose to manually burn the firmware with the flint command [5].

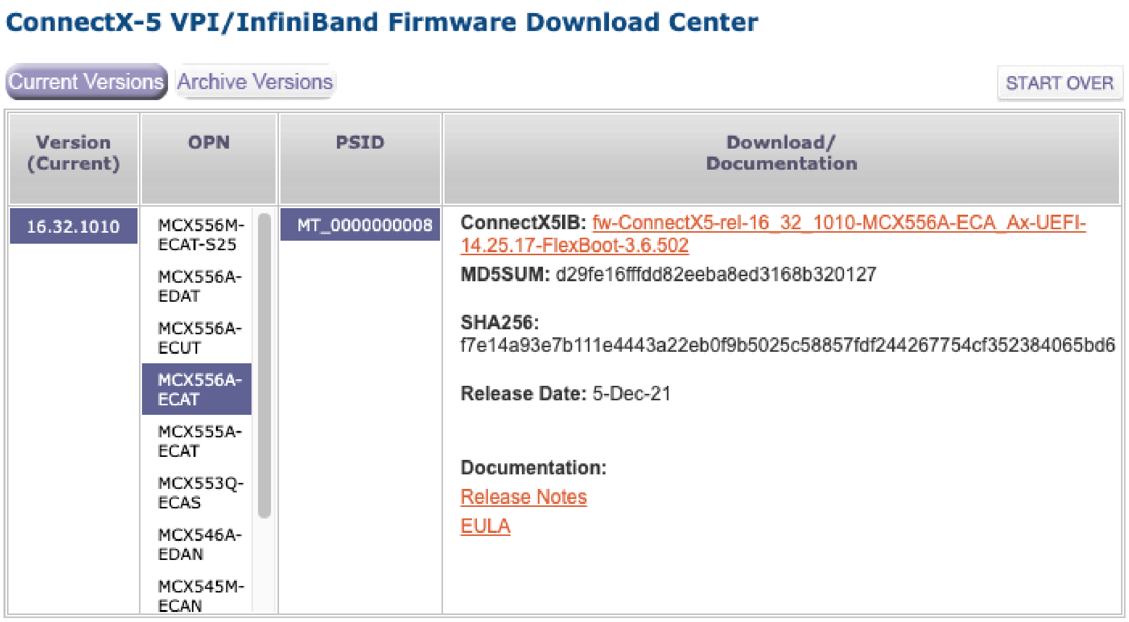


Figure 7. The latest firmware version of our HPC NIC

2.2.2. Install native Mellanox ESXi driver

After the firmware tools function well, configure the Native Mellanox ESXi (nmlx) driver. If it is not installed, please click this link [4]. At the time of this writing, the Mellanox website shows that the driver is defined for Ethernet only, not for InfiniBand. But we confirmed with the Mellanox support team that the 4.21.71.101 version can be used to configure IB and RoCE SR-IOV. This webpage then directs to a VMware site to download the nmlx_core driver.

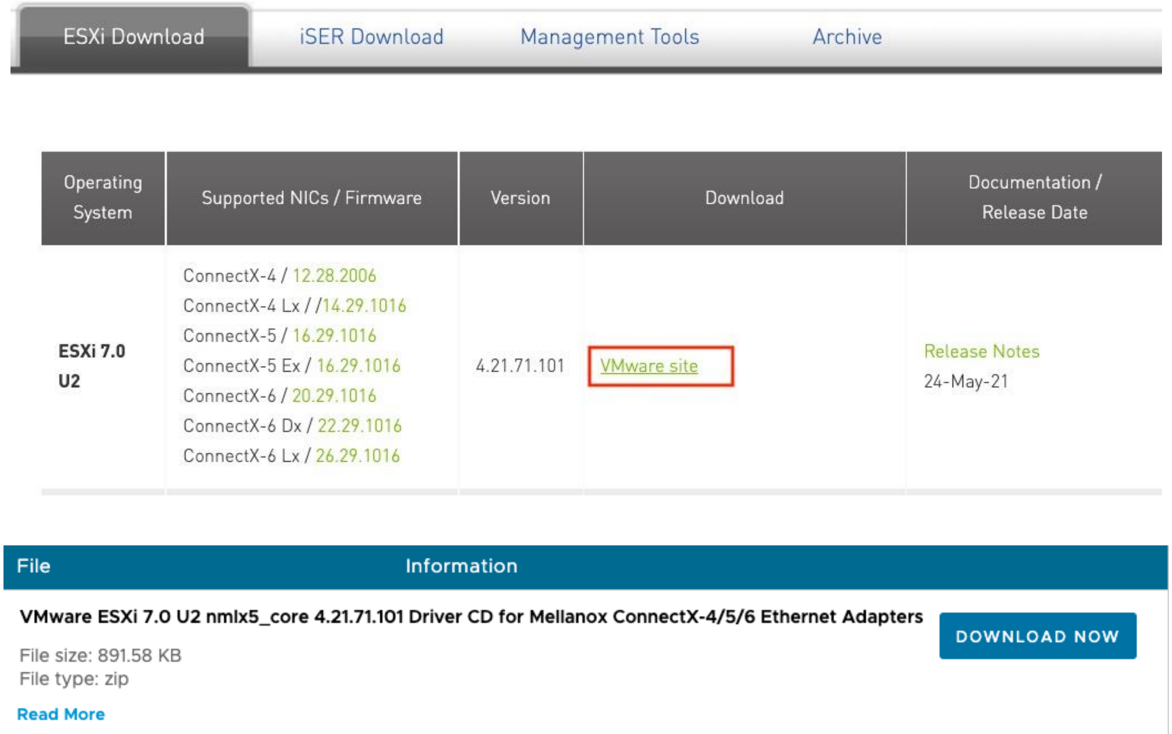


Figure 8. Download the native ESXi driver

> **Best Practice:** We also recommend downloading the nmlx driver to a location in the vSAN or NFS for the same reason as before.

Use the following commands to install the driver and reboot ESXi the second time.

```
# Install Native Mellanox ESXi Driver (nmlx)
[esxi]$ esxcli software vib install -d "Mellanox-nmlx5_xxx.zip"
# Reboot the host for the second time
```

Figure 9. Commands to install the nmlx ESXi driver and reboot the host

2.2.3. Configure InfiniBand and RoCE on firmware and ESXi driver

After the second reboot, we configure IB on Port 1 and RoCE on Port 2 in the firmware and the native ESXi driver using the commands in Figure 10.

```
1 # Configure IB on the firmware of Port 1 and RoCE on Port 2 of ConnectX-5
2 [esxi]$ mlxconfig -d mt4119_pciconf1 -y set LINK_TYPE_P1=1
3 [esxi]$ mlxconfig -d mt4119_pciconf1 -y set LINK_TYPE_P2=2
4
5 # Disable SRIOV on the firmware of ConnectX-5
6 # Clear Advanced PCI setting
7 [esxi]$ mlxconfig -d mt4119_pciconf1 -y set ADVANCED_PCI_SETTINGS=0
8 # Disable SRIOV
9 [esxi]$ mlxconfig -d mt4119_pciconf1 -y set SRIOV_EN=0
10
11 # Clear settings on Native ESXi Driver
12 [esxi]$ esxcli system module parameters set -m nmlx5_core -p ""
13
14 # Reboot the host for the third time
```

Figure 10. Commands to enable RoCE SR-IOV on the firmware and nmlx driver

In lines 2 and 3, we enable IB on port 1 by setting `LINK_TYPE_P1=1` and Ethernet on port 2 by setting `LINK_TYPE_P2=2`. In line 7, we set `ADVANCED_PCI_SETTINGS=0`, since DirectPath I/O doesn't need to enable it. In line 9, we set `SRIOV_EN=0` to disable SR-IOV. In line 12, we clear the parameters on nmlx5_core native driver with an empty string.

2.3. vSphere Client configuration

After configuring the port type on the firmware and driver on the ESXi, the PCIe devices can be enabled by logging to vSphere, going to the **Hosts and Clusters** view, and selecting the relevant ESXi server, followed by **Configure** → **Hardware** → **PCI Devices** → Check the two ports of **ConnectX-5 VPI adapter card** as shown in Figure 11.

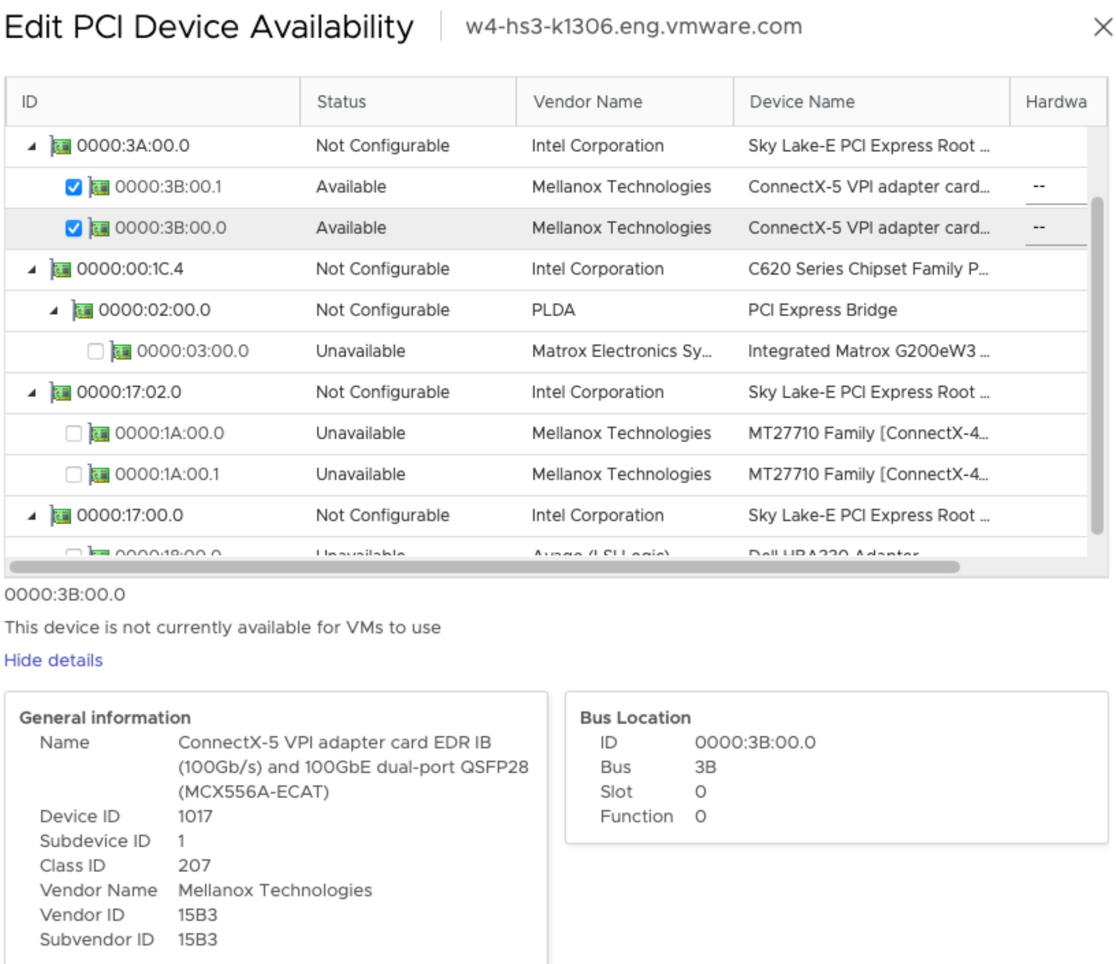


Figure 11. Check the mark to enable passthrough for PCI devices on a host in the vSphere Client

Best Practice: In Figure 12, we choose to use High Performance in the power policy by clicking the relevant ESXi server → Configure → Hardware → Overview, scrolling down to Power Management, clicking Edit Power Policy, and selecting High Performance.

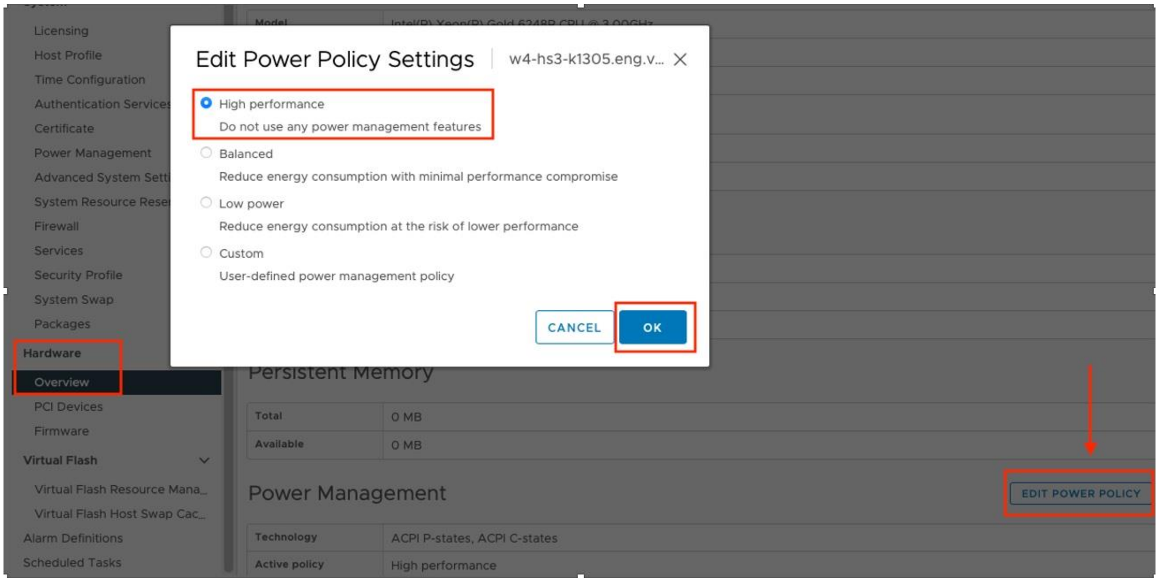


Figure 12. Choose **High Performance** as the power policy for the ESXi host

2.3.1. Attach a PCIe device on a virtual machine

In this step, we attach a PCIe passthrough device to a VM. Figure 13 shows this operation by following the steps in Configure a PCI Device on a Virtual Machine **Error! Reference source not found.** in the vSphere Client. Choose Dynamic DirectPath I/O if you want to use vSphere DRS or HA features for this VM.

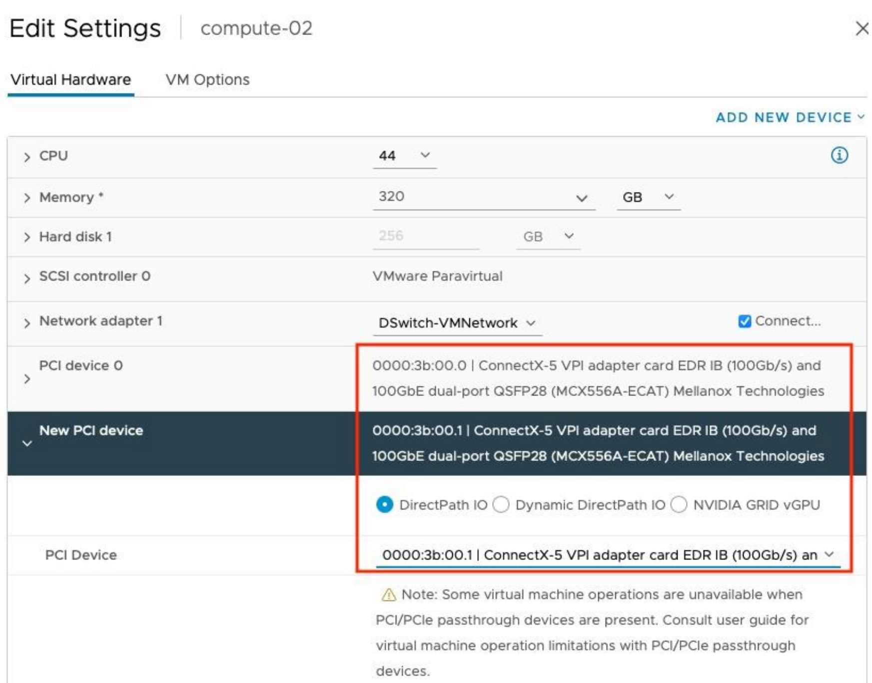


Figure 13. Use the vSphere Client to add a new PCIe device to a VM

Best Practice: You can use the vHPC toolkit to speed up the operation in Figure 14. The `--vm` flag can be replaced with `--file` flag to take an input file with a list of VMs' names.

```
[vhpc]$ IB_PCI_ID="0000:3B:00.0"
[vhpc]$ RoCE_PCI_ID="0000:3B:00.1"

# Assign a PF as a DirectPath I/O PCI device to a VM
[vhpc]$ ./vhpc_toolkit passthru --add --vm $vm_name --device $IB_PCI_ID
[vhpc]$ ./vhpc_toolkit passthru --add --vm $vm_name --device $RoCE_PCI_ID
```

Figure 14. Use the vHPC toolkit to assign a VF as a DirectPath I/O PCIe device to a VM

2.4. Guest OS configuration

Now, we can power on the VM. If Mellanox's version of OpenFabrics Enterprise Distribution (OFED) is not installed on the VM, download it using this link [9]. Figure 15 shows the command to install OFED. You must reboot the VM after installing OFED.

```
[guest OS]$ tar xf MLNX_OFED_LINUX-xxx.tgz
# For RHEL, install necessary dependent packages
[guest OS]$ dnf install -y kernel-modules-extra
# For CentOS, install necessary dependent packages
[guest OS]$ dnf install -y tk
# Install the latest driver and firmware
[guest OS]$ ./mlnxofedinstall --force
# Reboot VM
```

Figure 15. Install OFED on the guest

RoCE needs to enable PFC and set MTU=9000 on the Ethernet switch to achieve the performance requirement for HPC workloads. For the Ethernet switch commands to do so, please refer to the documentation of your Ethernet switch.

After OFED is installed on the guest, we first need to force restart OFED driver. Then we can check its version with `ofed_info -s`. Using `ip a` to list the network interface, we see the IB and RoCE interface—ib0 and ens256—show up. Next, we set MTU=9000 and assign an IP to the RoCE interface. Note the IP should be different from existing subnets on the VM. Otherwise, IP conflict will appear [12]. Then we can use `ibv_devinfo` or `ibstatus` to check the status of the RoCE port. Figure 16 shows the port `mlx5_1` is in the **active** state with `active_MTU=4096` and is using **Ethernet** as the link layer.

```
# Load the updated OFED driver
[guest OS]$ /etc/init.d/openibd force-restart

# Check OFED version
[guest OS]$ ofed_info -s
MLNX_OFED_LINUX-5.4-3.0.3.0:

# Check interface, ens256 is the interface of RoCE
[guest OS]$ # ip a
. . .
3: ens224f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
1000
    link/ether 0c:42:a1:d3:9b:37 brd ff:ff:ff:ff:ff:ff
4: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 4092 qdisc mq state UP group default qlen 256
    link/infiniband 00:00:01:c3:fe:80:00:00:00:00:00:00:0c:42:a1:03:00:d3:9b:36 brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff

# Set MTU on the RoCE interface
[guest OS] ip link set ens224f1 mtu 9000
# Assign an IP on the RoCE interface. Note: this IP should be different from existing subnets
on the host. Otherwise, IP conflict will appear.
[guest OS] ip addr add 192.168.xx.xx/24 dev ens224f1
# If you want to ping the IB interface, IP address is required to add to ib0 with command ip
addr add XXX.XXX.XXX.XX/XX dev ib0

# Check device information
[guest OS]$ ibstatus
Infiniband device 'mlx5_0' port 1 status:
    default gid:     fe80:0000:0000:0000:0c42:a103:00d3:9846
    base lid:    0x14
    sm lid:      0x1
    state:       4: ACTIVE
    phys state:  5: LinkUp
    rate:        100 Gb/sec (4X EDR)
    link_layer:  InfiniBand

Infiniband device 'mlx5_1' port 1 status:
    default gid:     fe80:0000:0000:0000:0e42:a1ff:fed3:9847
    base lid:    0x0
    sm lid:      0x0
    state:       4: ACTIVE
    phys state:  5: LinkUp
    rate:        100 Gb/sec (4X EDR)
    link_layer:  Ethernet
```

Figure 16. Load OFED and check the OFED version and device information on the guest

3. Functionality Evaluation

In this section, we evaluate the functionality of the DirectPath I/O IB and RoCE that we configured using two tests: ibverbs utility test and the OSU microbenchmark suite.

Table 1 describes our testbed: hardware, BIOS settings, firmware, and driver versions used in the above DirectPath I/O configuration. These versions were the latest available when we conducted these experiments. We recommend that you consult your product vendor and use the appropriate versions.

Environment		Bare Metal	Virtual Machine
Hardware	Server	PowerEdge R740 vSAN ReadyNode	
	Processor	2 x Intel Xeon Gold 6248R @ 3.00GHz	
	HPC InfiniBand Network NIC	100 GbE NVIDIA Mellanox ConnectX-5 VPI Dual Ports	
	Service Network NIC	10/25 GbE NVIDIA Mellanox ConnectX-4 Dual Ports	
	HPC InfiniBand Switch	Mellanox SB7800 100 Gb IB switch	
	HPC Ethernet Network Switch	Dell PowerSwitch S5232F 100GbE	
	Service Network Switch	Dell PowerSwitch S5296F-ON	
	ConnectX-5 firmware	16.32.1010	
BIOS	Power Profile	Performance Per Watt (OS controlled)	
	Hyperthreading	Enabled	
	Virtualization	Intel VT-d Enabled	
Cores		All 48 cores used	44 vCPU reserved, High Latency sensitivity
Memory		24 * 16GB RDIMM, All 384 GB used	144 GB reserved for the VM
Operating system	Host	RHEL 8.1	VMware vSphere 7.0U2, Guest OS: RHEL 8.1
	Power Policy	Default	High Performance
	Mellanox Firmware tools	MFT & NMST 4.18.1	

	Native Mellanox (NMLX) Driver	N/A	4.21.71.101
	OFED	5.4-3.0.3.0	
Build Libraries	Compiler	GCC 9.3.0	
	MPI	OpenMPI 4.1.2	
	UCX	1.12.0	
	Intel One API / Cluster Checker	2022.2 / 2021 Update 6 (build 20220318)	
	Spack	0.17.1	
	OSU MicroBenchmark	5.7.1	

Table 1. Testbed details of the virtual clusters

3.1. ibverbs utility test

We first evaluate the performance of IB DirectPath I/O between two VMs by using the ibverbs bandwidth test in Figure 17 and the latency test in Figure 18.

```
[root@compute-02 ~]# ib_send_bw -a --report_gbits -d mlx5_0 compute-03
---------------------------------------------------------------------------------------
                    Send BW Test
 Dual-port       : OFF      Device          : mlx5_0
 Number of qps   : 1        Transport type : IB
 Connection type : RC       Using SRQ      : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 TX depth        : 128
 CQ Moderation   : 100
 Mtu             : 4096[B]
 Link type       : IB
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0x14 QPN 0x004b PSN 0x971bc7
 remote address: LID 0x15 QPN 0x004c PSN 0xcf9184
---------------------------------------------------------------------------------------
 #bytes    #iterations  BW peak[Gb/sec]  BW average[Gb/sec] MsgRate[Mpps]
 2         1000         0.11             0.11               6.769307
 4         1000         0.22             0.22               6.952953
 8         1000         0.45             0.45               6.995139
 16        1000         0.90             0.90               7.004699
 32        1000         1.85             1.81               7.076277
 64        1000         3.72             3.70               7.230020
 128       1000         7.39             5.68               5.549830
 256       1000         14.77            12.37              6.039957
 512       1000         28.31            27.44              6.698045
 1024      1000         53.65            53.44              6.523100
 2048      1000         79.86            79.70              4.864574
 4096      1000         93.05            93.00              2.838018
 8192      1000         95.08            95.04              1.450152
 16384     1000         95.68            95.66              0.729866
 32768     1000         95.81            95.81              0.365475
 65536     1000         96.07            96.06              0.183225
 131072    1000         96.17            96.17              0.091716
 262144    1000         96.14            96.14              0.045841
 524288    1000         96.19            96.19              0.022933
 1048576   1000         96.17            96.16              0.011463
 2097152   1000         96.26            96.26              0.005737
 4194304   1000         96.24            96.24              0.002868
 8388608   1000         96.21            96.21              0.001434
---------------------------------------------------------------------------------------
```

Figure 17. ibverbs IB bandwidth test

Figure 17 shows that the `ib_send_bw` bandwidth of 95 Gbps on larger packet sizes is close to the line rate of 100 Gbps of the ConnectX-5 adapter card, which indicates that DirectPath I/O IB is configured correctly.

```
[root@compute-02 ~]# ib_send_lat -a --report_gbits -d mlx5_0 compute-03
---------------------------------------------------------------------------------------
                    Send Latency Test
 Dual-port       : OFF          Device         : mlx5_0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 TX depth        : 1
 Mtu             : 4096[B]
 Link type       : IB
 Max inline data : 236[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0x14 QPN 0x004c PSN 0x7520b
 remote address: LID 0x15 QPN 0x004d PSN 0x6041dd
---------------------------------------------------------------------------------------
 #bytes  #iterations t_min[usec]  t_max[usec]  t_typical[usec] t_avg[usec] t_stdev[usec] 99%[usec]  99.9%[usec]
 2       1000        1.04         4.91         1.09            1.10        0.11          1.27       4.91
 4       1000        1.03         3.72         1.08            1.09        0.12          1.23       3.72
 8       1000        1.03         3.35         1.07            1.08        0.10          1.18       3.35
 16      1000        1.04         3.53         1.07            1.08        0.09          1.20       3.53
 32      1000        1.07         3.77         1.11            1.12        0.14          1.26       3.77
 64      1000        1.18         3.52         1.22            1.23        0.10          1.37       3.52
 128     1000        1.22         3.41         1.26            1.27        0.10          1.41       3.41
 256     1000        1.63         4.01         1.68            1.69        0.12          2.00       4.01
 512     1000        1.69         4.05         1.74            1.75        0.11          1.92       4.05
 1024    1000        1.81         4.29         1.94            1.95        0.14          2.17       4.29
 2048    1000        2.07         4.44         2.11            2.13        0.09          2.36       4.44
 4096    1000        2.58         4.50         2.65            2.67        0.09          2.87       4.50
 8192    1000        3.23         5.92         3.35            3.36        0.10          3.55       5.92
 16384   1000        4.64         7.92         4.87            4.89        0.14          5.16       7.92
 32768   1000        6.82         8.90         7.05            7.06        0.14          7.52       8.90
 65536   1000        9.56         11.90        9.77            9.79        0.20          11.01      11.90
 131072  1000        15.06        18.04        15.53           15.52       0.18          15.88      18.04
 262144  1000        26.64        29.43        27.42           27.44       0.27          28.17      29.43
 524288  1000        49.88        52.94        51.16           51.19       0.46          52.37      52.94
 1048576 1000        96.48        100.26       98.23           98.24       0.61          99.81      100.26
 2097152 1000        187.19       192.06       189.47          189.49      0.83          191.57     192.06
 4194304 1000        361.09       367.53       364.16          364.26      0.85          366.47     367.53
 8388608 1000        710.80       717.97       714.27          714.25      1.00          716.42     717.97
---------------------------------------------------------------------------------------
```

Figure 18. ibverbs IB latency test

Figure 18 shows that the latency of `ib_send_lat` averages 1.1 microseconds for small messages. We look up the manual for the IB Mellanox SB7800 switch, and we find that it has 90 nanoseconds of latency. We consider IB has been correctly configured.

Next, we use the same utility test to evaluate the RoCE DirectPath I/O performance between two VMs in Figure 19 and Figure 20.

```
[root@compute-02 ~]# ib_send_bw -a --report_gbits -d mlx5_1 compute-03
---------------------------------------------------------------------------------------
                    Send BW Test
 Dual-port       : OFF        Device          : mlx5_1
 Number of qps   : 1          Transport type : IB
 Connection type : RC         Using SRQ       : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 TX depth        : 128
 CQ Moderation   : 100
 Mtu             : 4096[B]
 Link type       : Ethernet
 GID index       : 3
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x0149 PSN 0x891617
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:192:168:05:02
 remote address: LID 0000 QPN 0x0149 PSN 0x5d0ec0
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:192:168:05:03
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 2          1000           0.11               0.11                 6.669537
 4          1000           0.22               0.22                 6.868166
 8          1000           0.44               0.33                 5.091346
 16         1000           0.88               0.84                 6.534750
 32         1000           1.78               1.51                 5.885992
 64         1000           3.50               3.36                 6.562548
 128        1000           7.01               6.64                 6.482748
 256        1000           14.03              14.00                6.834633
 512        1000           27.24              23.12                5.645554
 1024       1000           49.53              49.42                6.032969
 2048       1000           74.64              74.49                4.546635
 4096       1000           91.32              91.29                2.785981
 8192       1000           94.89              94.89                1.447954
 16384      1000           96.08              96.07                0.732986
 32768      1000           96.49              96.49                0.368073
 65536      1000           96.80              96.80                0.184628
 131072     1000           97.01              97.00                0.092511
 262144     1000           96.66              96.66                0.046090
 524288     1000           90.02              89.98                0.021454
 1048576    1000           93.10              93.07                0.011094
 2097152    1000           95.13              95.12                0.005670
 4194304    1000           96.01              95.96                0.002860
 8388608    1000           96.27              96.21                0.001434
---------------------------------------------------------------------------------------
```

Figure 19. ibverbs RoCE bandwidth test

Figure 19 shows that the `ib_send_bw` bandwidth of 95 Gbps on larger packet sizes is close to the line rate 100 Gbps of the ConnectX-5 adapter card, which indicates that IB passthrough is configured correctly.

```
[root@compute-02 ~]# ib_send_lat -a --report_gbits -d mlx5_1 compute-03
---------------------------------------------------------------------------------------
                    Send Latency Test
 Dual-port       : OFF      Device         : mlx5_1
 Number of qps   : 1        Transport type : IB
 Connection type : RC       Using SRQ      : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 TX depth        : 1
 Mtu             : 4096[B]
 Link type       : Ethernet
 GID index       : 3
 Max inline data : 236[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x014b PSN 0xcf614
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:192:168:05:02
 remote address: LID 0000 QPN 0x014b PSN 0x3ae0e0
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:192:168:05:03
---------------------------------------------------------------------------------------
 #bytes    #iterations  t_min[usec]  t_max[usec]  t_typical[usec] t_avg[usec] t_stdev[usec] 99%[usec]  99.9%[usec]
 2         1000         1.91         8.81         1.96            1.98        0.14          2.32       8.81
 4         1000         1.91         4.67         1.95            1.96        0.11          2.28       4.67
 8         1000         1.90         4.09         1.95            1.96        0.08          2.28       4.09
 16        1000         1.92         5.65         1.96            1.97        0.13          2.30       5.65
 32        1000         1.95         5.50         1.99            2.01        0.17          2.32       5.50
 64        1000         2.01         4.01         2.06            2.07        0.11          2.41       4.01
 128       1000         2.05         4.38         2.10            2.11        0.12          2.42       4.38
 256       1000         2.46         4.43         2.51            2.54        0.11          2.89       4.43
 512       1000         2.52         4.20         2.57            2.61        0.11          3.02       4.20
 1024      1000         2.64         5.40         2.71            2.74        0.16          3.01       5.40
 2048      1000         2.96         6.02         3.02            3.06        0.12          3.44       6.02
 4096      1000         3.45         6.50         3.55            3.57        0.14          3.80       6.50
 8192      1000         4.16         6.43         4.26            4.28        0.12          4.72       6.43
 16384     1000         5.53         11.41        5.75            5.77        0.16          6.08       11.41
 32768     1000         7.71         10.80        7.92            7.93        0.11          8.26       10.80
 65536     1000         10.37        13.59        10.71           10.72       0.18          11.15      13.59
 131072    1000         15.69        19.24        16.15           16.15       0.14          16.46      19.24
 262144    1000         27.57        33.23        28.42           28.43       0.28          29.12      33.23
 524288    1000         51.43        54.63        52.84           52.86       0.47          54.01      54.63
 1048576   1000         99.32        103.52       101.12          101.10      0.67          102.73     103.52
 2097152   1000         191.43       197.97       194.48          194.52      0.98          197.02     197.97
 4194304   1000         365.22       371.38       368.09          368.12      0.99          370.61     371.38
 8388608   1000         712.16       720.54       715.99          716.02      1.22          719.22     720.54
---------------------------------------------------------------------------------------
```

Figure 20. ibverbs RoCE latency test

Figure 20 shows that the latency of `ib_send_lat` averages 2 microseconds for small messages, and we look up the Dell PowerSwitch S5232F 100 GbE to see that it has 877 nanoseconds latency in one hop. Thus, two hops between two VMs in around 2 microseconds is an acceptable value.

3.2. OSU benchmark test

Since our server has been configured as a dual-boot system (that is, bare metal and ESXi), we use the OSU benchmark to compare the communication performance first on the 16 bare metal nodes, then on the 16 VMs. We run OSU multiple bandwidth/message rate benchmark (mbw_mr) Figure 21with 2, 4, 8, 12, and 16 VMs. Each data point uses an average of five runs. Because the VMs are using 44 vCPUs, for a fair comparison, we run 48 and 44 processes per node on `BareMetal (BM)` nodes. The legend `VM.44.144.LatSens.IB.Passthru` means the VM uses 44 vCPUs, 144 GB memory, sets latency sensitivity to high, and uses IB DirectPath I/O. The legend format is also used in the later HPC application tests.

Figure 21 and Figure 22 show that IB and RoCE DirectPath I/O can achieve near bare metal performance of IB and RoCE on all message sizes for the aggregate bandwidth/message rate test, respectively.

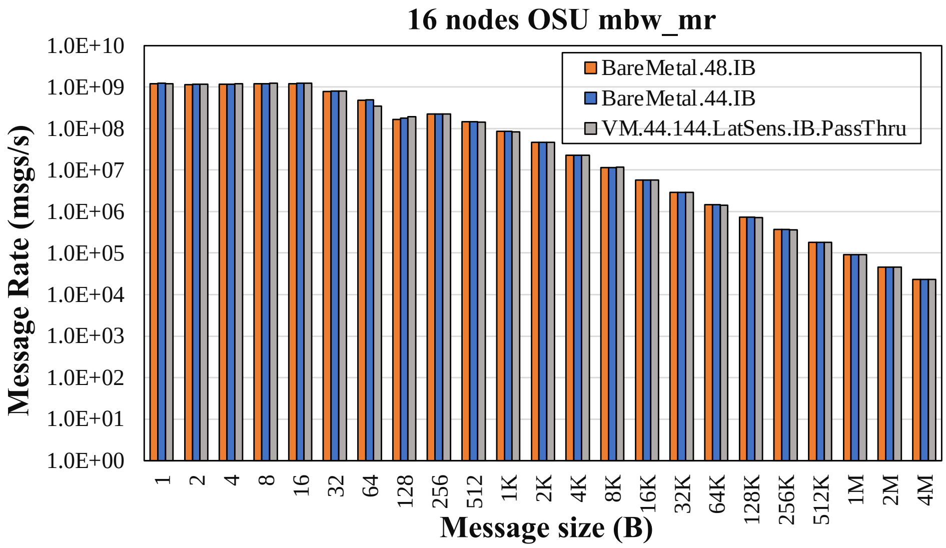


Figure 21. IB OSU MBW_MR test on 16 nodes

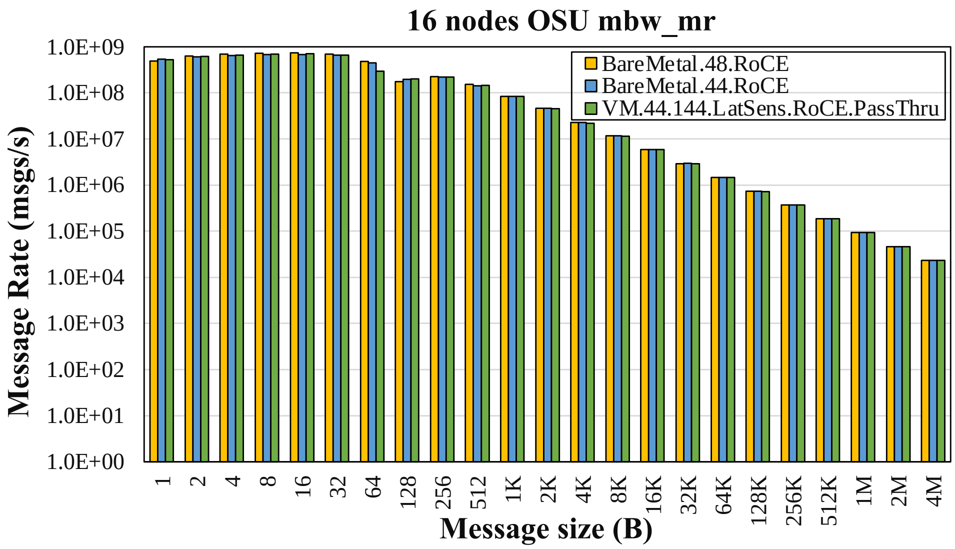


Figure 22. RoCE OSU MBW_MR test on 16 nodes

4. Performance Study of HPC Applications

In this section, we compare the performance and strong scalability between bare metal and virtual systems by using a range of different HPC applications across multiple vertical domains, along with the benchmark datasets used in Table 2. We use the tuning best practices in Performance Study of HPC Scale-Out Workloads on VMware vSphere 7 [10] to achieve MPI application performance running in a virtualized infrastructure that is close to the performance observed for the bare metal infrastructure. Since 48 PPN in bare metal systems uses 8.3% more cores than 44 PPN in virtual, we use this number as a gauge. Thus, if the performance delta falls within 8.3%, we consider this acceptable since vSphere offers other features like vSAN, vMotion, high availability, security, isolation, and more.

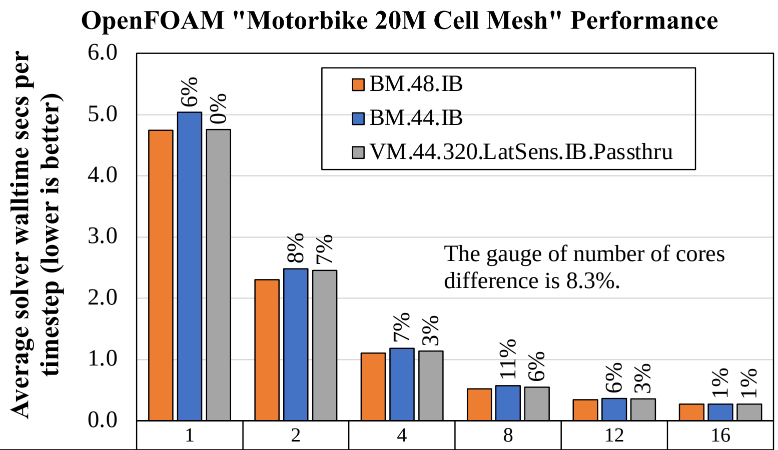
Application	Vertical Domain	Benchmark Dataset	Version
OpenFOAM	Manufacturing – Computational Fluid Dynamics (CFD)	Motorbike 20M cell mesh	9
Weather Research and Forecasting (WRF)	Weather and Environment	Conus 2.5KM	3.9.1.1
Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)	Molecular Dynamics	EAM Metallic Solid Benchmark	20210310
GROMACS	Life Sciences – Molecular Dynamics	HECBioSim BenchPEP 12M Atoms	2020.5
Nanoscale Molecular Dynamics (NAMD)	Life Sciences – Molecular Dynamics	STMV – 8M Atoms	2.14

Table 2. Application and benchmark details

4.1. OpenFOAM

We begin with the OpenFOAM software for computational fluid dynamics. Since the 20M Motorbike benchmark needs a larger memory than 144 GB to run, we expand the VM's memory to 320 GB only in this HPC application.

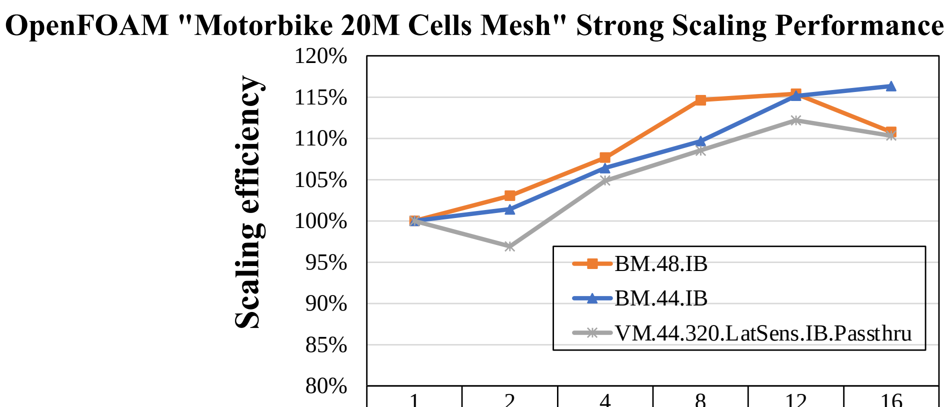
To compare the IB performance, we use the `BM.48.IB` as the baseline in Figure 23, so the percentage number on the top of the columns `BM.44.IB` and `VM.44.320.LatSens.IB.Passthru` shows the performance delta compared to the baseline. We observe that `VM.44.IB.Passthru` has at most a 7% delta compared to `BM.48.IB` on 2 nodes and performs better than `BM.44.IB` on all node counts.



	1	2	4	8	12	16
BM.48.IB	4.74	2.30	1.10	0.52	0.34	0.27
BM.44.IB	5.03	2.48	1.18	0.57	0.36	0.27
VM.44.320.LatSens.IB.Passthru	4.76	2.45	1.13	0.55	0.35	0.27

Figure 23. OpenFOAM performance comparison between virtual and bare metal systems

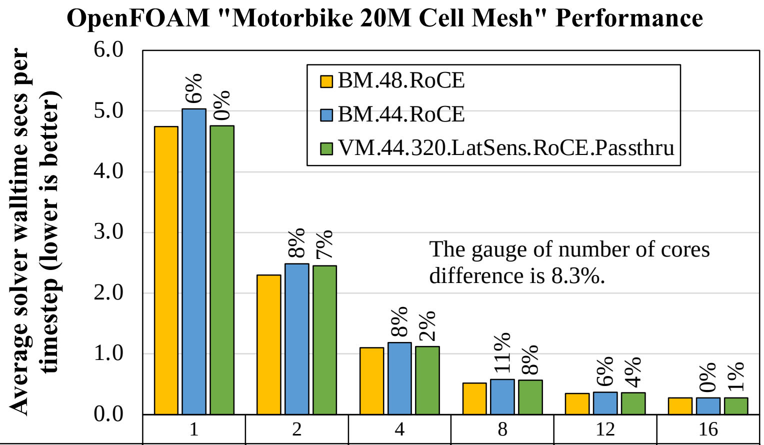
Figure 24 presents the strong scaling bare metal and virtual efficiency of OpenFOAM using IB on different node counts. `VM.44.IB.Passthru` has the same trend as the other two bare metal configurations and all of them show an efficiency of nearly or above 100%.



	1	2	4	8	12	16
BM.48.IB	100%	103%	108%	115%	115%	111%
BM.44.IB	100%	101%	106%	110%	115%	116%
VM.44.320.LatSens.IB.Passthru	100%	97%	105%	109%	112%	110%

Figure 24. OpenFOAM strong scaling comparison between virtual and bare metal systems

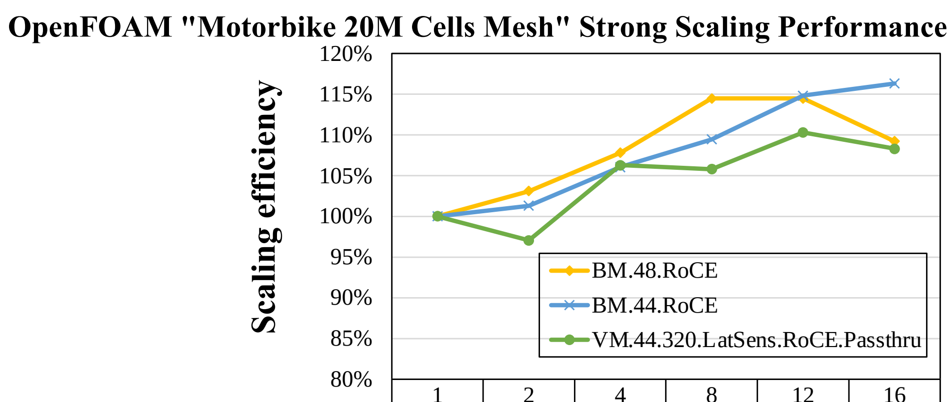
Similarly for the RoCE performance comparison in Figure 25, we use the `BM.48.RoCE` as the baseline, so the percentage number on the top of the columns `BM.44.RoCE` and `VM.44.320.LatSens.RoCE.Passthru` shows the performance delta compared to the baseline. We observe that `VM.44.RoCE.Passthru` has at most an 8% delta compared to `BM.48.RoCE` on 8 nodes and also has better performance than `BM.44.RoCE` on all node counts.



	1	2	4	8	12	16
BM.48.RoCE	4.74	2.30	1.10	0.52	0.35	0.27
BM.44.RoCE	5.03	2.48	1.19	0.57	0.37	0.27
VM.44.320.LatSens.RoCE.Passthru	4.76	2.45	1.12	0.56	0.36	0.27

Figure 25. OpenFOAM performance comparison using RoCE between virtual and bare metal systems

Figure 26 shows the strong scaling bare metal and virtual efficiency of OpenFOAM using RoCE on different node counts. We again find the RoCE DirectPath I/O follows a similar trend over 100% scaling efficiency as other bare metal configurations.



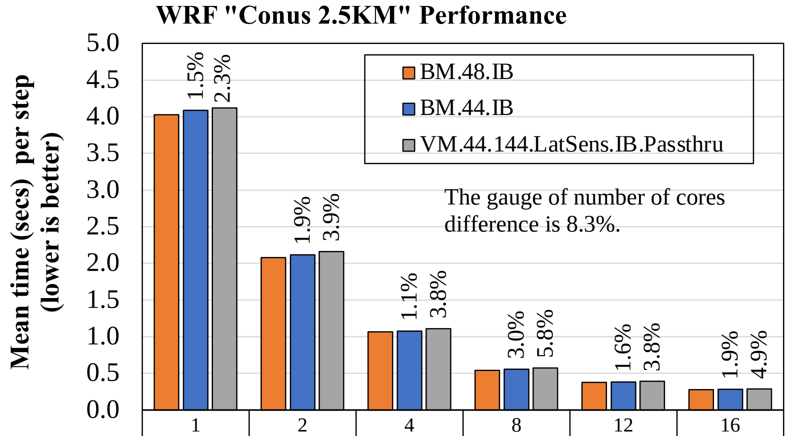
	1	2	4	8	12	16
BM.48.RoCE	100%	103%	108%	114%	114%	109%
BM.44.RoCE	100%	101%	106%	109%	115%	116%
VM.44.320.LatSens.RoCE.Passthru	100%	97%	106%	106%	110%	108%

Figure 26. OpenFOAM strong scaling comparison using RoCE between virtual and bare metal systems

4.2. WRF

For our following example, we try the Weather Research and Forecasting (WRF) model, a numerical weather prediction system used in atmospheric research and other applications. For IB performance comparison in Figure 27, we observe that `VM.44.IB.Passthru` has at most a 5.6%

performance delta on 16 nodes compared to `BM.48.IB`. Other node counts still present the performance delta within the 8.3% gauge.

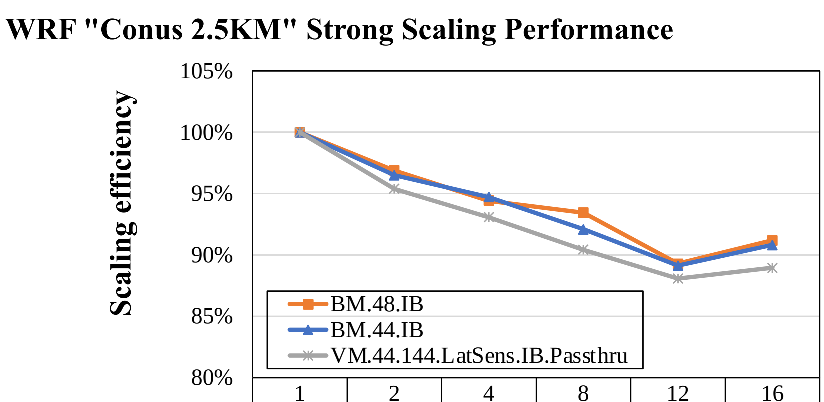


	1	2	4	8	12	16
BM.48.IB	4.03	2.08	1.07	0.54	0.38	0.28
BM.44.IB	4.09	2.12	1.08	0.55	0.38	0.28
VM.44.144.LatSens.IB.Passthru	4.12	2.16	1.11	0.57	0.39	0.29

Number of nodes

Figure 27. WRF performance comparison using IB between virtual and bare metal systems

Figure 28 presents the strong scaling efficiency of WRF using IB on different node counts. We notice the IB DirectPath I/O has the same trend as the other two bare metal configurations.

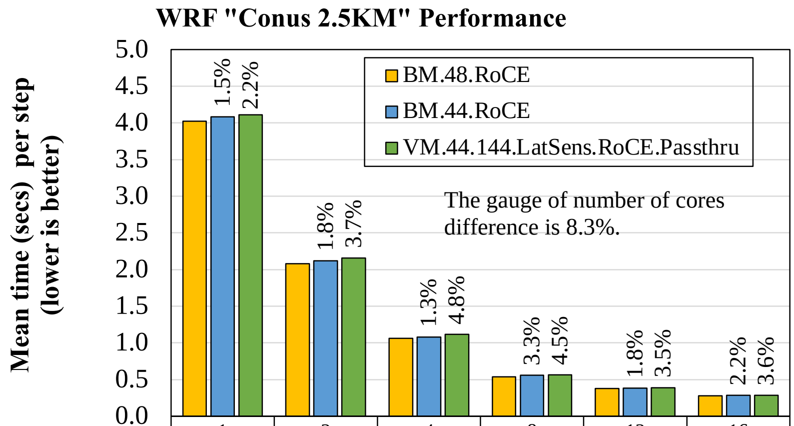


	1	2	4	8	12	16
BM.48.IB	100%	97%	94%	93%	89%	91%
BM.44.IB	100%	96%	95%	92%	89%	91%
VM.44.144.LatSens.IB.Passthru	100%	95%	93%	90%	88%	89%

Number of Nodes

Figure 28. WRF strong scaling comparison using IB between virtual and bare metal systems.

For the RoCE performance comparison, in Figure 29 and Figure 30, we observe that `VM.44.RoCE.Passthru` has at most a 4.8% delta on 4 nodes compared to `BM.48.IB`, and also has a similar performance to `BM.44.RoCE`.

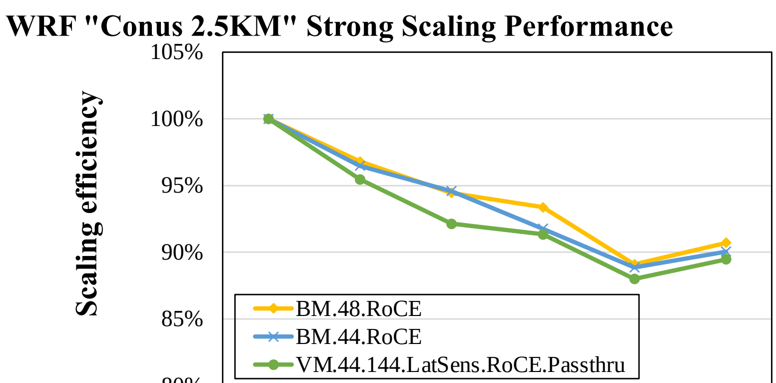


Number of nodes	1	2	4	8	12	16
BM.48.RoCE	4.02	2.08	1.07	0.54	0.38	0.28
BM.44.RoCE	4.08	2.12	1.08	0.56	0.38	0.28
VM.44.144.LatSens.RoCE.Passthru	4.11	2.15	1.12	0.56	0.39	0.29

Number of nodes

Figure 29. WRF performance comparison using RoCE between virtual and bare metal systems

Figure 30 presents the strong scaling efficiency of WRF using RoCE on different node counts. We find the RoCE DirectPath I/O efficiency has the same trend as bare metal.



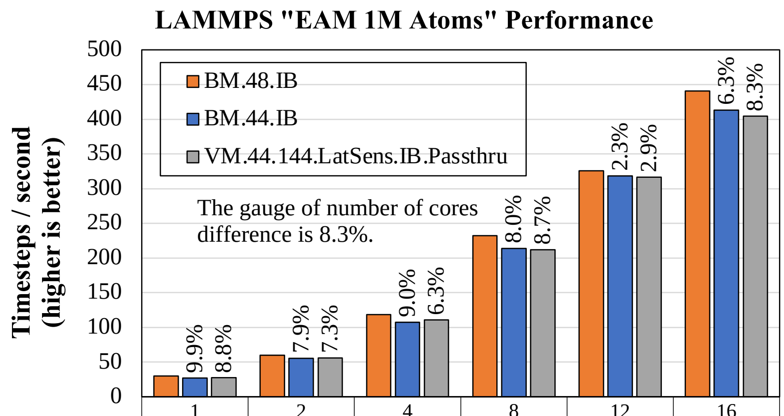
Number of Nodes	1	2	4	8	12	16
BM.48.RoCE	100%	97%	94%	93%	89%	91%
BM.44.RoCE	100%	96%	95%	92%	89%	90%
VM.44.144.LatSens.RoCE.Passthru	100%	95%	92%	91%	88%	89%

Number of Nodes

Figure 30. WRF strong scaling comparison using RoCE between virtual and bare metal systems

4.3. LAMMPS

Next, we use the molecular dynamics simulator LAMMPS. For the IB performance comparison, Figure 31 shows that `VM.44.IB.Passthru` has the largest delta at 8.8% compared to `BM.48.IB` on the single node. But `BM.44.IB` also has a delta of 9.9%, which we attribute to the domain decomposition. Other node counts still present the virtual performance delta within the 8.3% gauge.

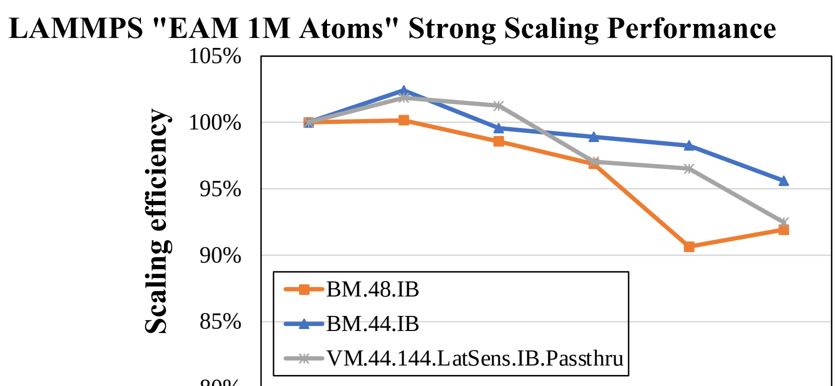


	1	2	4	8	12	16
BM.48.IB	30	60	118	232	326	441
BM.44.IB	27	55	108	214	318	413
VM.44.144.LatSens.IB.Passthru	27	56	111	212	316	404

Number of Nodes

Figure 31. LAMMPS performance comparison using IB between virtual and bare metal systems

Figure 32 presents the strong scaling efficiency of LAMMPS using IB on different node counts. We notice that `BM.44.IB` and `VM.44.IB.Passthru` has better scaling efficiency than `BM.48.IB`.

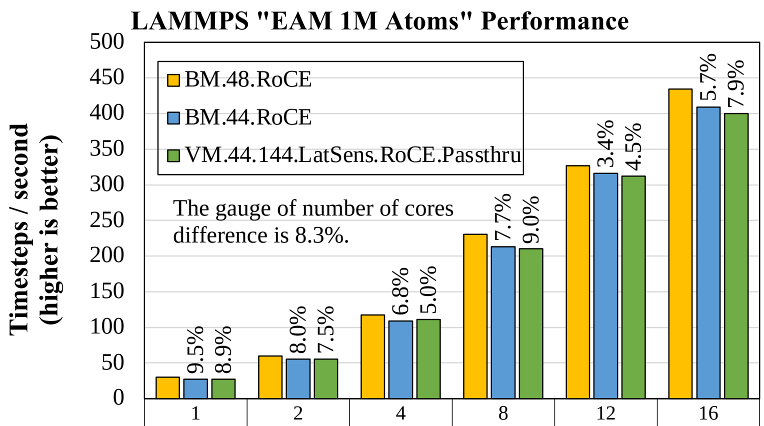


	1	2	4	8	12	16
BM.48.IB	100%	100%	99%	97%	91%	92%
BM.44.IB	100%	102%	100%	99%	98%	96%
VM.44.144.LatSens.IB.Passthru	100%	102%	101%	97%	97%	92%

Number of nodes

Figure 32. LAMMPS strong scaling comparison using IB between virtual and bare metal systems

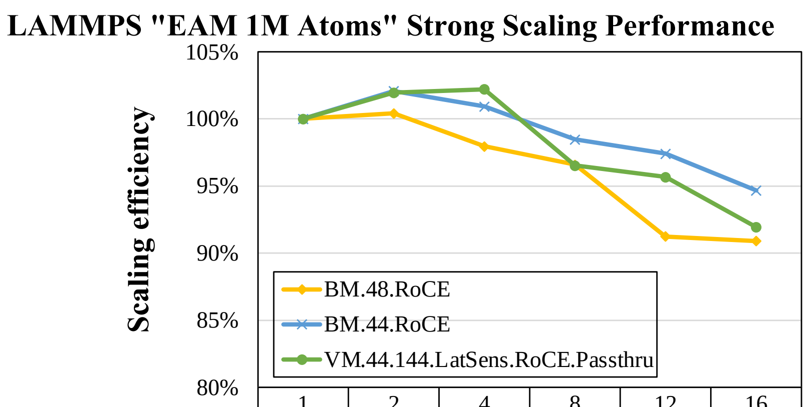
For the RoCE performance comparison in Figure 33, we observe that `VM.44.RoCE.Passthru` has at most a 4.8% delta on 4 nodes compared to `BM.48.IB`, and also has a similar performance to `BM.44.RoCE`.



	1	2	4	8	12	16
BM.48.RoCE	30	60	117	231	327	434
BM.44.RoCE	27	55	109	213	316	409
VM.44.144.LatSens.RoCE.Passthru	27	55	111	210	312	400

Figure 33. LAMMPS performance comparison using RoCE between virtual and bare metal systems

Figure 34 presents the strong scaling bare metal and virtual efficiency of LAMMPS using RoCE on different node counts. We again notice that `BM.44.RoCE` and `VM.44.RoCE.Passthru` has better scaling efficiency than `BM.48.RoCE`.

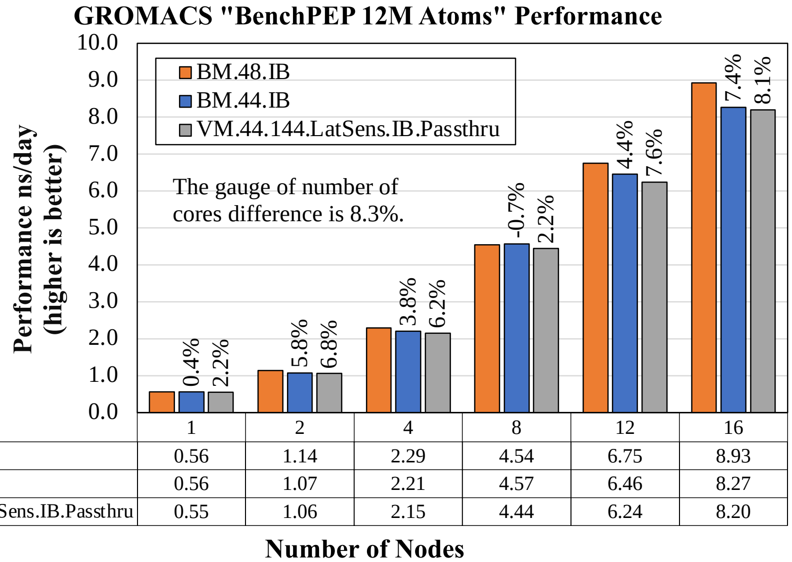


	1	2	4	8	12	16
BM.48.RoCE	100%	100%	98%	97%	91%	91%
BM.44.RoCE	100%	102%	101%	98%	97%	95%
VM.44.144.LatSens.RoCE.Passthru	100%	102%	102%	97%	96%	92%

Figure 34. LAMMPS strong scaling comparison using RoCE between virtual and bare metal systems

4.4. GROMACS

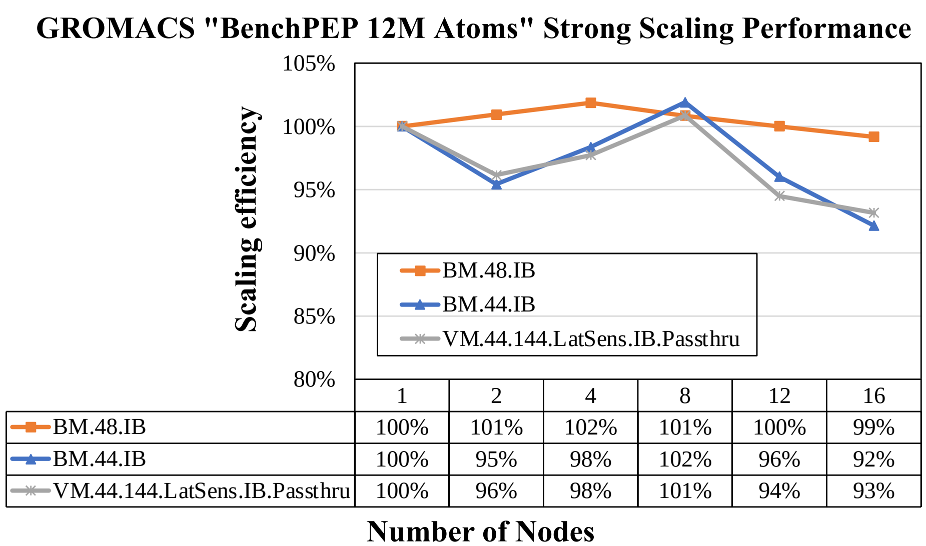
Next, we use GROMACS, a simulator often used for the study of biomolecules. For the IB performance comparison in Figure 35, we see that the largest delta between `VM.44.IB.Passthru` and `BM.48.IB` is 8.1% on 16 nodes. Note that `BM.44.IB` also has a 7.4% delta compared to `BM.48.IB`, which we consider acceptable because it is related to the difference in domain decomposition.



	1	2	4	8	12	16
BM.48.IB	0.56	1.14	2.29	4.54	6.75	8.93
BM.44.IB	0.56	1.07	2.21	4.57	6.46	8.27
VM.44.144.LatSens.IB.Passthru	0.55	1.06	2.15	4.44	6.24	8.20

Figure 35. GROMACS performance comparison using IB between virtual and bare metal systems

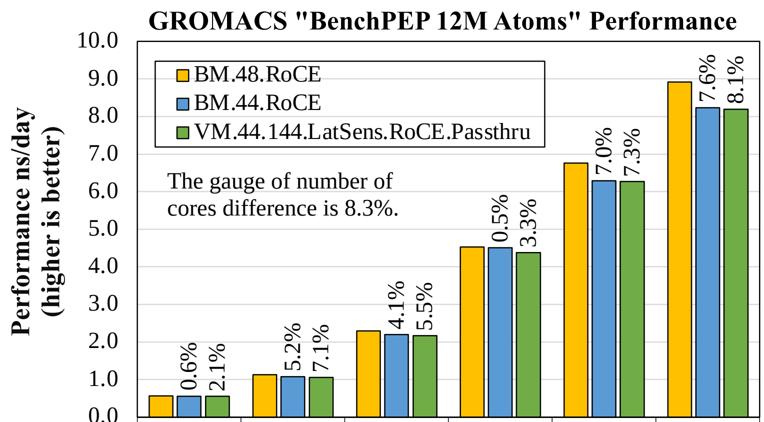
Figure 36 presents the strong scaling efficiency of GROMACS using IB on different node counts. We notice that `VM.44.IB.Passthru` and `BM.44.IB` have nearly identical scaling efficiency.



	1	2	4	8	12	16
BM.48.IB	100%	101%	102%	101%	100%	99%
BM.44.IB	100%	95%	98%	102%	96%	92%
VM.44.144.LatSens.IB.Passthru	100%	96%	98%	101%	94%	93%

Figure 36. GROMACS strong scaling comparison using IB between virtual and bare metal systems

For the next RoCE performance comparison in Figure 37, we observe that `VM.44.RoCE.Passthru` has at most an 8.1% delta on 16 nodes compared to `BM.48.RoCE`, and also has a similar performance to `BM.44.RoCE`.

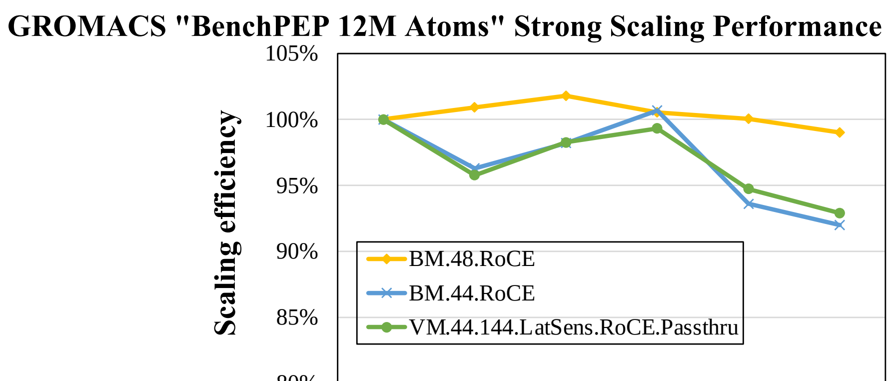


	1	2	4	8	12	16
BM.48.RoCE	0.56	1.14	2.29	4.53	6.76	8.92
BM.44.RoCE	0.56	1.08	2.20	4.51	6.29	8.24
VM.44.144.LatSens.RoCE.Passthru	0.55	1.06	2.17	4.38	6.27	8.20

Number of Nodes

Figure 37. GROMACS performance comparison using RoCE between virtual and bare metal systems

Figure 38 presents the strong scaling efficiency of GROMACS using RoCE on different node counts. We notice that `VM.44.RoCE.Passthru` and `BM.44.RoCE` have nearly identical scaling efficiency.



	1	2	4	8	12	16
BM.48.RoCE	100%	101%	102%	101%	100%	99%
BM.44.RoCE	100%	96%	98%	101%	94%	92%
VM.44.144.LatSens.RoCE.Passthru	100%	96%	98%	99%	95%	93%

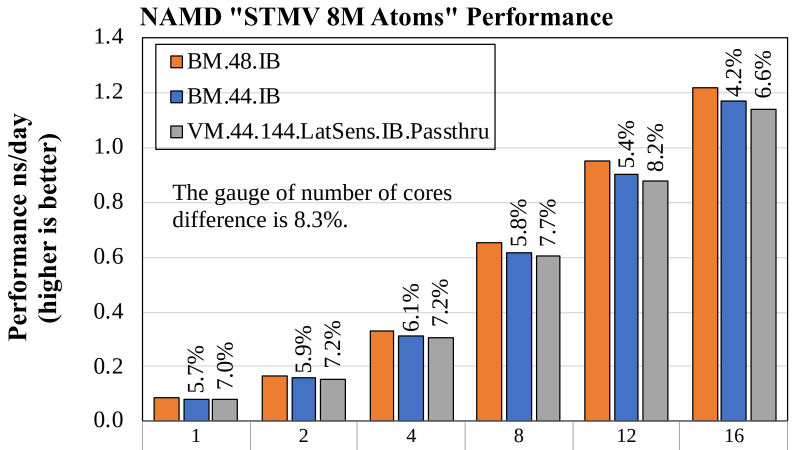
Number of Nodes

Figure 38. GROMACS strong scaling comparison using RoCE between virtual and bare metal systems

4.5. NAMD

Last, we use NAMD, a simulator of large biomolecular systems. We run NAMD in a hybrid mode, such as a 44 PPN, 1 MPI process with 43 computing threads, and 1 communication thread launched on a node. For the IB performance comparison in Figure 39, we see an 8.2% performance delta on the 12 nodes comparing `VM.44.IB.Passthru` and `BM.48.IB`. Since our servers don't enable sub-

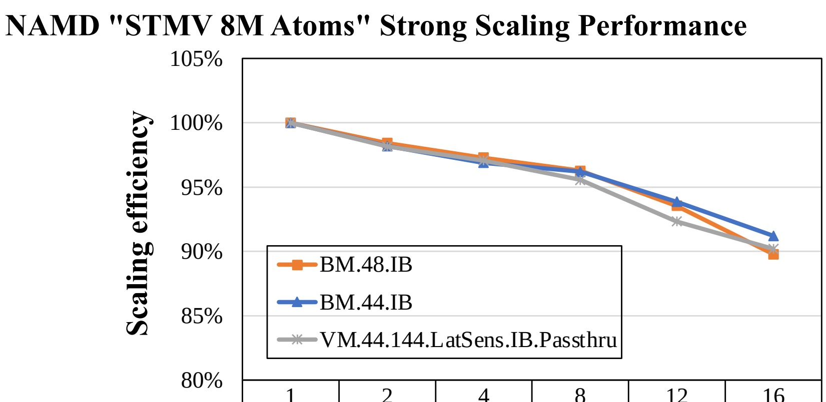
NUMA clustering (SNC), we plan to further investigate the performance improvement with SNC enabled in future.



Number of nodes	1	2	4	8	12	16
BM.48.IB	0.08	0.17	0.33	0.65	0.95	1.22
BM.44.IB	0.08	0.16	0.31	0.62	0.90	1.17
VM.44.144.LatSens.IB.Passthru	0.08	0.16	0.31	0.60	0.88	1.14

Figure 39. NAMD performance comparison using IB between virtual and bare metal systems

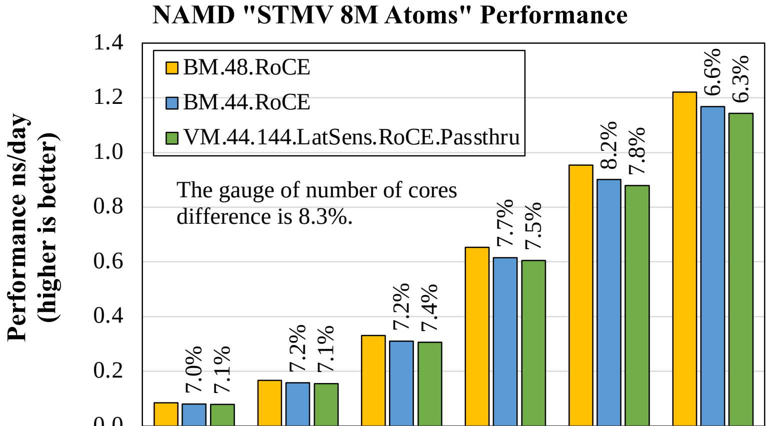
Figure 40 presents the strong scaling bare metal and virtual efficiency of NAMD using IB on different node counts. We notice that `VM.44.IB.Passthru` has nearly the same scaling efficiency trend as the other two bare metal configurations.



Number of Nodes	1	2	4	8	12	16
BM.48.IB	100%	98%	97%	96%	94%	90%
BM.44.IB	100%	98%	97%	96%	94%	91%
VM.44.144.LatSens.IB.Passthru	100%	98%	97%	96%	92%	90%

Figure 40. NAMD strong scaling comparison between virtual and bare metal systems

For the RoCE performance comparison in Figure 41, we observe that `VM.44.RoCE.Passthru` has at most an 8.1% delta on 16 nodes compared to `BM.48.RoCE`, and also has a similar performance to `BM.44.RoCE`.

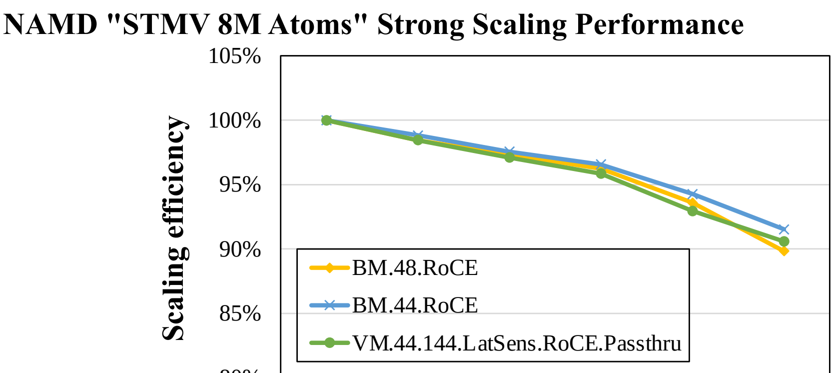


	1	2	4	8	12	16
BM.48.RoCE	0.08	0.17	0.33	0.65	0.95	1.22
BM.44.RoCE	0.08	0.16	0.31	0.62	0.90	1.17
VM.44.144.LatSens.RoCE.Passthru	0.08	0.16	0.31	0.60	0.88	1.14

Number of nodes

Figure 41. NAMD performance comparison between virtual and bare metal systems

Figure 42 presents the strong scaling bare metal and virtual efficiency of NAMD using RoCE on different node counts. We again notice that `VM.44.IB.Passthru` has nearly the same scaling efficiency trend as the other two bare metal configurations.



	1	2	4	8	12	16
BM.48.RoCE	100%	98%	97%	96%	94%	90%
BM.44.RoCE	100%	99%	98%	97%	94%	92%
VM.44.144.LatSens.RoCE.Passthru	100%	98%	97%	96%	93%	91%

Number of Nodes

Figure 42. NAMD strong scaling comparison between virtual and bare metal systems

5. Summary

In this document, we walked through the steps to configure IB and RoCE DirectPath I/O on NVIDIA Mellanox ConnectX-5 adapter cards in vSphere 7.x. We evaluated this setup's functionality with two benchmarks and studied its performance using five typical HPC applications. In all cases, our virtual HPC cluster approached the performance of a bare metal cluster.

We aimed to construct this technical guide to remain useful even when software versions and products evolve in the future. This is the last of three papers in this series for vSphere 7.

6. References

[1] DirectPath I/O

[2] vSphere VMDirectPath I/O and Dynamic DirectPath I/O: Requirements for Platforms and Devices (2142307)

[3] NVIDIA Firmware Tools (MFT) Documentation v4.18.1

[4] Native ConnectX Driver for VMware® ESXi Server

[5] NVIDIA ConnectX-4 onwards NICs NATIVE ESXi Driver for VMware vSphere User Manual v4.19.71.1

[6] vHPC-toolkit github page

[7] Enable Passthrough for a Network Device on a Host

[8] Configure a PCI Device on a Virtual Machine

[9] Guest OS OFED Download link

[10] Performance Study of HPC Scale-Out Workloads on VMware vSphere 7

[11] NVIDIA CONNECTX-5INFINIBAND ADAPTER CARDS

[12] Avoid Assigning Multiple NICs in the Same Computer to the Same Subnet

About the Author

Yuankun Fu is a member of technical staff in the HPC/ML group of VMware OCTO and had 10 years of HPC experience before joining VMware in July 2021. He focuses on HPC/ML application performance on the VMware platform and works on a wide variety of HPC projects, from creating technical guides and performance best practices to root-causing performance challenges when running highly technical workloads on customer platforms. Previously, he graduated from Purdue University with a PhD degree in Computer Science. He was also a research assistant at Purdue University and interned at the Los Alamos National Lab.

Acknowledgments

The author thanks Ramesh Radhakrishnan and Chris Gully from VMware, Rizwan Ali and Martin Hilgeman from Dell Technologies, and Martin Feyereisen for their contribution to the study. The author also thanks Pamela Gorder and Julie Brodeur for editing support and improving the paper.