Configuring PVRDMA in VMware vSphere 6.5

Last Update: 17 November 2017

Introduces Paravirtual RDMA (PVRDMA) as a virtual NIC device offered to virtual machines

Explains how to enable PVRDMA in VMware vSphere 6.5

Shows how Linux guest VMs are configured to use PVRDMA

Provides the test results using PVRDMA in VMs

Yi Liao

LENOVO PRESS

Abstract

Paravirtual RDMA (PVRDMA) is a new PCIe virtual NIC which supports standard RDMA API and is offered to a VM on vSphere 6.5. This paper describes how to enable PVRDMA in VMware vSphere 6.5 via vCenter on Lenovo Purley servers.

This paper is suitable for people who are familiar with vSphere and vCenter and who know how to use ESXi commands in an ESXi environment. It is expected that the reader will have some knowledge of PVRDMA and its application.

At Lenovo Press, we bring together experts to produce technical publications around topics of importance to you, providing information and best practices for using Lenovo products and solutions to solve IT challenges.

See a list of our most recent publications at the Lenovo Press web site:

http://lenovopress.com

Do you have the latest version? We update our papers from time to time, so check whether you have the latest version of this document by clicking the **Check for Updates** button on the front page of the PDF. Pressing this button will take you to a web page that will tell you if you are reading the latest version of the document and give you a link to the latest if needed. While you're there, you can also sign up to get notified via email whenever we make an update.

Contents

Introduction

Remote Direct Memory Access (RDMA) allows direct access from the memory of one server to the memory of another server, without involving the operating system or CPU. The transfer of memory is off-loaded to the RDMA-capable Host Channel Adapters (HCA). Paravirtual RDMA (PVRDMA) is a new device that can emulate RDMA in the ESXi hypervisor.

In vSphere, a virtual machine can use a PVRDMA network adapter to communicate with other virtual machines that have PVRDMA devices. The virtual machines must be connected to the same vSphere Distributed Switch.

The PVRDMA device automatically selects the method of communication between the virtual machines:

- For virtual machines that run on the same ESXi host, with or without a physical RDMA device, the data transfer is a memory copy (memcpy) between the two virtual machines. The physical RDMA hardware is not used in this case.
- For virtual machines that reside on different ESXi hosts and that have a physical RDMA connection, the physical RDMA devices must be uplinks on the distributed switch. In this case, the communication between the virtual machines, by way of PVRDMA, use the underlying physical RDMA devices.
- For two virtual machines that run on different ESXi hosts, when at least one of the hosts does not have a physical RDMA device, the communication falls back to a TCP-based channel and the performance is reduced.

Hardware configurations

In our lab tests, we used three servers for RDMA verification:

- One server acts as a management node with vCenter, for both PVRDMA and vSphere Distributed Switch (vDS). This should be configured from vCenter and it only needs a management network NIC. In our lab tests, we used a ThinkServer RD340 system as the management node.
- Two servers are for the testing. We used ThinkSystem SR630 servers in our lab environment. Two network adapters in each server are required:
  - One adapter for network management. In our setup, we used the Intel I350 T4 Gigabit Ethernet Adapter.
  - One adapter for VM traffic. In our lab, we used the Mellanox ConnectX-3 Pro ML2 40 GbE adapter.

Details of the server configurations we used in our lab tests are shown in Table 1.

Table 1 Hardware configuration

Servers	Hardware	Software
vCenter node	► ThinkServer RD340	► Windows Server 2012 R2 ► VMware vCenter 6.5.0-4602587

Servers	Hardware	Software
ESXi node1	► Lenovo ThinkSystem SR630 ► NIC1 - ThinkSystem I350-T4 PCIe 1Gb 4-Port RJ45 Ethernet Adapter By Intel ► NIC2 - ThinkSystem Mellanox ConnectX-3 Pro ML2 FDR 2-Port QSFP VPI Adapter	► VMware ESXi 6.5 - 4564106
ESXi node2	► Lenovo ThinkSystem SR630 ► NIC1 - ThinkSystem I350-T4 PCIe 1Gb 4-Port RJ45 Ethernet Adapter By Intel ► NIC2 - ThinkSystem Mellanox ConnectX-3 Pro ML2 FDR 2-Port QSFP VPI Adapter	► VMware ESXi 6.5 - 4564106

UEFI settings

PVRDMA support is based on RDMA over Converged Ethernet (RoCE) in VMware vSphere 6.5. RoCE is a network protocol that uses RDMA to provide faster data transfers for network-intensive applications. RoCE allows direct memory transfer between hosts without involving the hosts' processors.

In this case, we need to set the "Network link type" to Ethernet to work with the ROCE protocol for all the physical ports on MT27520, as shown in Figure 1.

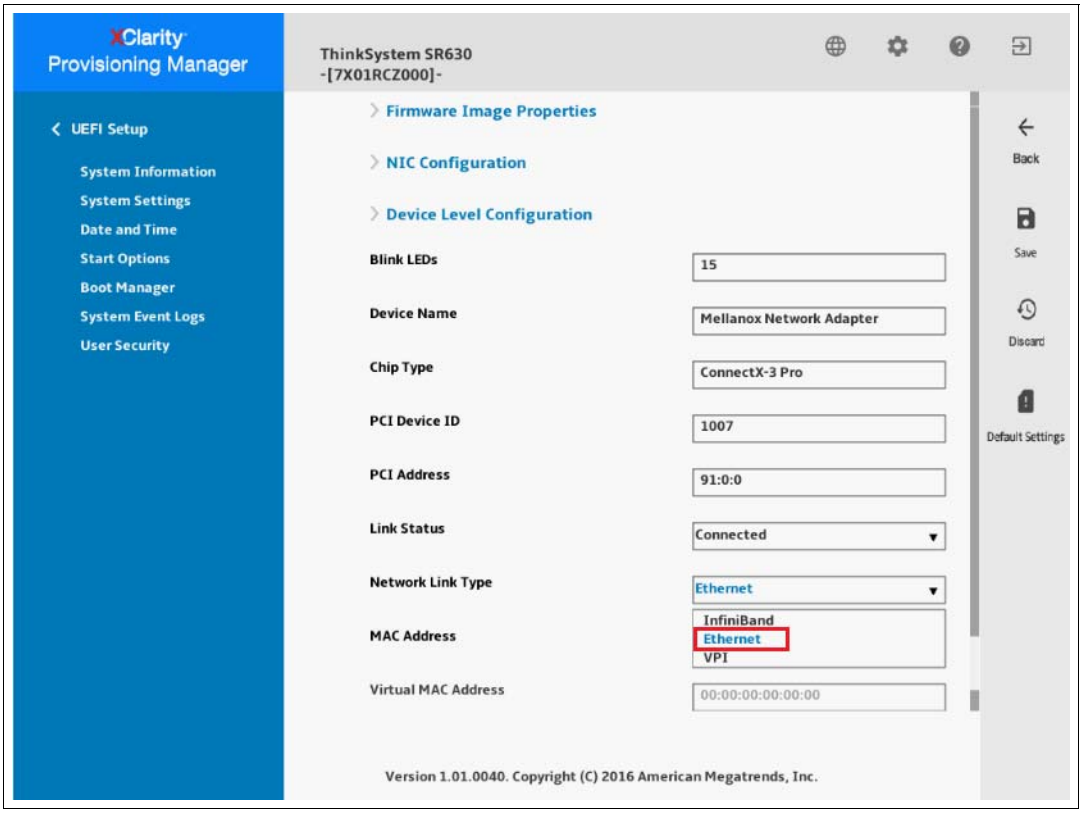

Figure 1 Lenovo ThinkSystem SR630 UEFI options for PVRDMA

Check host configurations

We used two ESXi hosts to validate and test the RDMA function. Each host's configuration on the OS side is shown in Figure 2 and Figure 3. Please keep in mind their IP addresses and vmnic names, because they will be used to describe nodes or parameters in what follows.

```
[root@WIN-1K93AFQG1DT:~]
[root@WIN-1K93AFQG1DT:~] esxcfg-nics -l
Name         PCI          Driver      Link Speed      Duplex MAC Address       MTU    Description
vmnic0       0000:08:00.0 igbn        Down 0Mbps      Half   a0:36:9f:34:8a:90 1500   Intel Corporation Ethernet Server Adapter I350
-T4
vmnic1       0000:08:00.1 igbn        Down 0Mbps      Half   a0:36:9f:34:8a:91 1500   Intel Corporation Ethernet Server Adapter I350
-T4
vmnic1000402 0000:5b:00.0 nmlx4_en    Up   40000Mbps  Full   24:8a:07:6d:3e:12 1500   Mellanox Technologies MT27520 Family
vmnic2       0000:08:00.2 igbn        Down 0Mbps      Half   a0:36:9f:34:8a:92 1500   Intel Corporation Ethernet Server Adapter I350
-T4
vmnic3       0000:08:00.3 igbn        Up   1000Mbps   Full   a0:36:9f:34:8a:93 1500   Intel Corporation Ethernet Server Adapter I350
-T4
vmnic4       0000:5b:00.0 nmlx4_en    Down 0Mbps      Half   24:8a:07:6d:3e:11 1500   Mellanox Technologies MT27520 Family
[root@WIN-1K93AFQG1DT:~]
[root@WIN-1K93AFQG1DT:~] esxcfg-vmknic -l
Interface  Port Group/DVPort/Opaque Network        IP Family IP Address                              Netmask         Broadcast
 MAC Address       MTU     TSO MSS   Enabled Type                NetStack
vmk0       Management Network                      IPv4      10.245.39.241                           255.255.255.0   10.245.39.255
 a0:36:9f:34:8a:93 1500    65535     true    DHCP                defaultTcpipStack
vmk0       Management Network                      IPv6      fe80::a236:9fff:fe34:8a93               64
 a0:36:9f:34:8a:93 1500    65535     true    STATIC, PREFERRED   defaultTcpipStack
vmk1       6                                       IPv4      192.168.1.241                           255.255.255.0   192.168.1.255
 00:50:56:6f:54:c0 1500    65535     true    STATIC              defaultTcpipStack
vmk1       6                                       IPv6      fe80::250:56ff:fe6f:54c0                64
 00:50:56:6f:54:c0 1500    65535     true    STATIC, PREFERRED   defaultTcpipStack
[root@WIN-1K93AFQG1DT:~]
[root@WIN-1K93AFQG1DT:~] esxcli rdma device list
Name     Driver      State    MTU  Speed    Paired Uplink  Description
-------  ----------  ------  ----  -------  -------------  ------------------------------
vmrdma0  nmlx4_rdma  Down    1024  0        vmnic4         MT27520 Family [ConnectX-3 Pro]
vmrdma1  nmlx4_rdma  Active  1024  40 Gbps  vmnic1000402   MT27520 Family [ConnectX-3 Pro]
```

Figure 2 Host configurations for 10.245.39.241

```
[root@localhost:~] esxcfg-nics -l
Name         PCI          Driver      Link Speed      Duplex MAC Address       MTU    Description
vmnic0       0000:06:00.0 nmlx4_en    Down 0Mbps      Half   7c:fe:90:9b:53:71 1500   Mellanox Technologies MT27520 Family
vmnic1       0000:2f:00.0 igbn        Down 0Mbps      Half   a0:36:9f:98:72:68 1500   Intel Corporation Ethernet Server Adapter I350
-T4
vmnic1000002 0000:06:00.0 nmlx4_en    Up   40000Mbps  Full   7c:fe:90:9b:53:72 1500   Mellanox Technologies MT27520 Family
vmnic2       0000:2f:00.1 igbn        Down 0Mbps      Half   a0:36:9f:98:72:69 1500   Intel Corporation Ethernet Server Adapter I350
-T4
vmnic3       0000:2f:00.2 igbn        Down 0Mbps      Half   a0:36:9f:98:72:6a 1500   Intel Corporation Ethernet Server Adapter I350
-T4
vmnic4       0000:2f:00.3 igbn        Up   1000Mbps   Full   a0:36:9f:98:72:6b 1500   Intel Corporation Ethernet Server Adapter I350
-T4
vusb0        Pseudo       cdce        Up   100Mbps    Full   7e:d3:0a:c6:9f:07 1500   IBM XClarity Controller
[root@localhost:~]
[root@localhost:~] esxcfg-vmknic -l
Interface  Port Group/DVPort/Opaque Network        IP Family IP Address                              Netmask         Broadcast
 MAC Address       MTU     TSO MSS   Enabled Type                NetStack
vmk0       Management Network                      IPv4      10.245.39.129                           255.255.255.0   10.245.39.255
 a0:36:9f:98:72:6b 1500    65535     true    DHCP                defaultTcpipStack
vmk0       Management Network                      IPv6      fe80::a236:9fff:fe98:726b               64
 a0:36:9f:98:72:6b 1500    65535     true    STATIC, PREFERRED   defaultTcpipStack
vmk1       7                                       IPv4      192.168.1.129                           255.255.255.0   192.168.1.255
 00:50:56:6a:c3:91 1500    65535     true    STATIC              defaultTcpipStack
vmk1       7                                       IPv6      fe80::250:56ff:fe6a:c391                64
 00:50:56:6a:c3:91 1500    65535     true    STATIC, PREFERRED   defaultTcpipStack
[root@localhost:~]
[root@localhost:~] esxcli rdma device list
Name     Driver      State    MTU  Speed    Paired Uplink  Description
-------  ----------  ------  ----  -------  -------------  ------------------------------
vmrdma0  nmlx4_rdma  Down    1024  0        vmnic0         MT27520 Family [ConnectX-3 Pro]
vmrdma1  nmlx4_rdma  Active  1024  40 Gbps  vmnic1000002   MT27520 Family [ConnectX-3 Pro]
```

Figure 3 Host configurations for 10.245.39.129

vSphere Distributed Switch

A vSphere Distributed Switch provides centralized management and monitoring of the networking configuration of all hosts that are associated with the switch. You must set up a distributed switch on a vCenter Server system, and its settings will be propagated to all hosts that are associated with the switch.

Creating a vSphere Distributed Switch

Perform the following steps to create a new vSphere Distributed Switch:

1. Launch the vSphere Web Client and connect to a vCenter Server instance.
2. On the vSphere Web Client home screen, select the **vCenter** object from the list on the left. Then, select **Distributed Switches** from the Inventory Lists area.
3. On the right side of the vSphere Web Client, click the **Create a New Distributed Switch** icon (it looks like a switch with a green plus mark in the corner). This launches the New Distributed Switch wizard.
4. Supply a name for the new distributed switch and select the location within the vCenter inventory (a datacenter object or a folder) where you would like to store the new distributed switch. Click **Next**.
5. Select the version of the vDS you would like to create. Figure 4 shows the options for distributed switch versions.

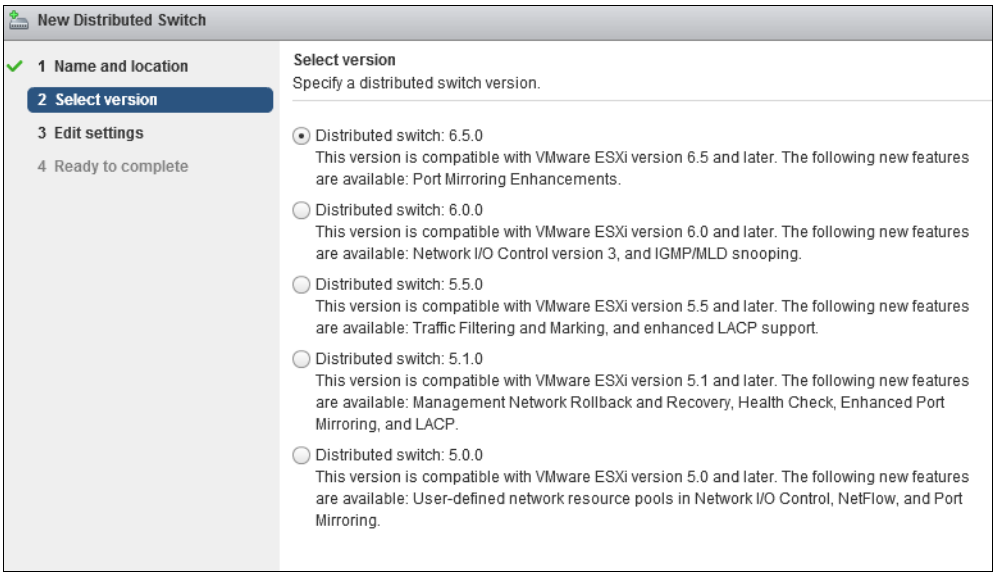

Figure 4 Options for distributed switch versions

6. Specify the number of uplink ports as **2** and create a default port group with the name **vg-RDMA** as shown in Figure 5 on page 7. Click **Next** and then **Finish**.

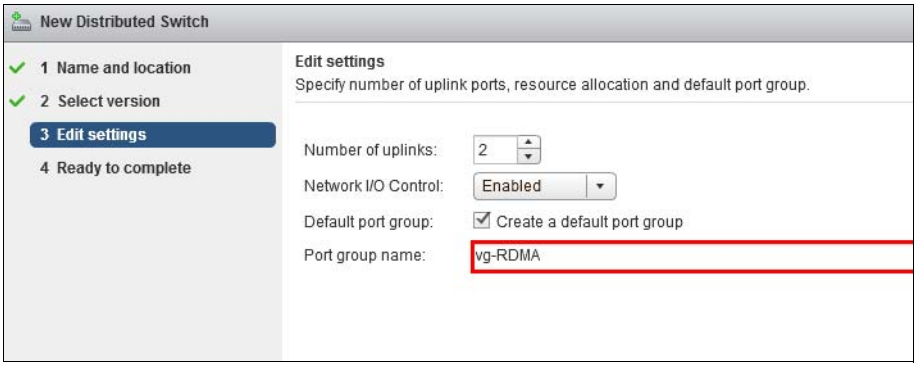

Figure 5 Number of uplinks and port group name

Adding and managing hosts

Perform the following steps to add an ESXi host to an existing distributed switch:

1. Launch the vSphere Web Client, and connect to a vCenter Server instance.
2. Navigate to the list of distributed switches. One way of getting there is to start at the vCenter home screen and click **Distributed Switches** in the Inventory Lists area.
3. Select an existing distributed switch in the list of objects on the right, and select **Add and Manage Hosts** from the Actions menu shown in Figure 6.

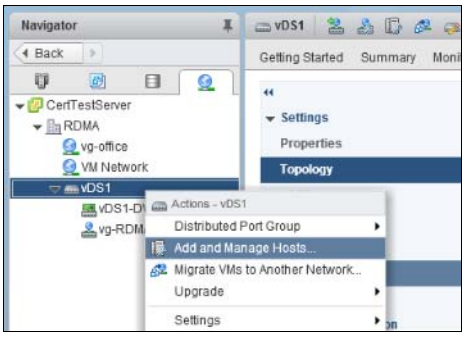

Figure 6 Launching the Add and Manage Hosts wizard

4. Select the **Add Hosts** radio button and click **Next**.
5. Click the green plus icon to add an ESXi host. This opens the Select New Host dialog box.
6. From the list of new hosts to add, place a check mark next to the name of each ESXi host you would like to add to the distributed switch. Click OK when you are done, and then click Next to continue, as shown in Figure 7 on page 8.

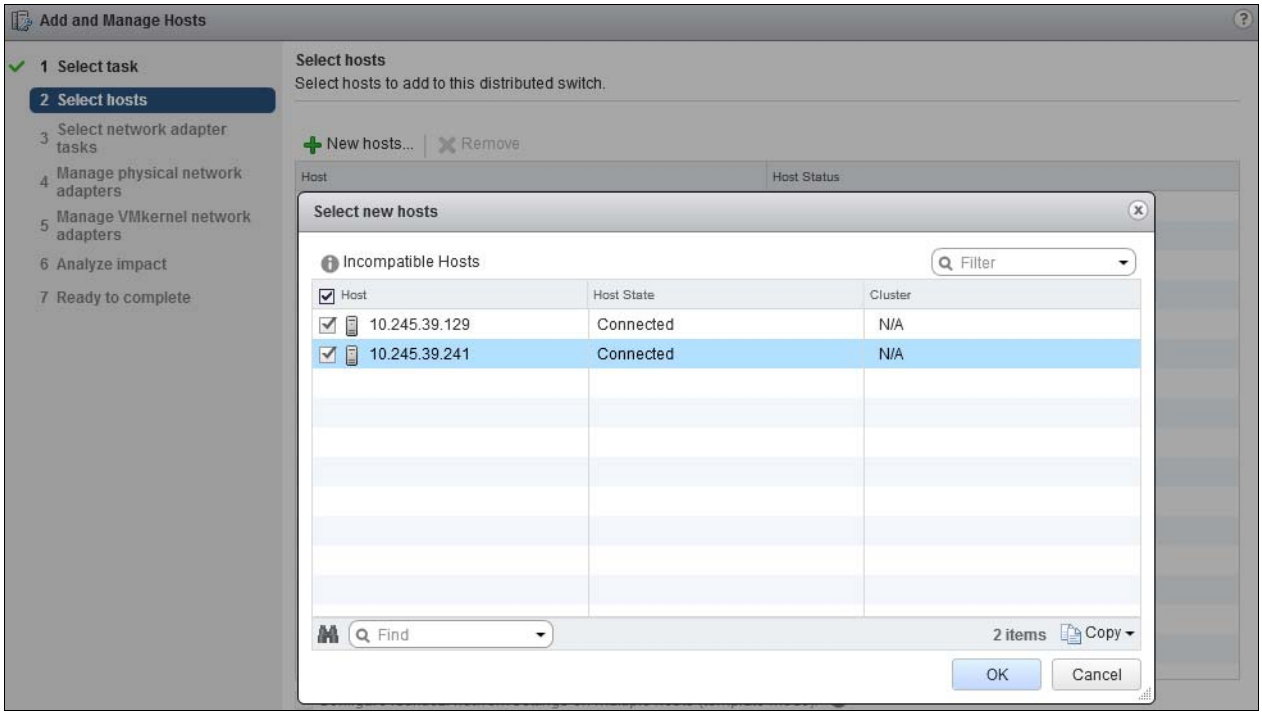

Figure 7 Adding hosts to vDS

7. The next screen offers three different adapter-related tasks to perform, as shown in Figure 8. In this case, make sure both the **Manage physical adapters** and **Manage VMkernel adapters** options are selected. Click **Next** to continue.

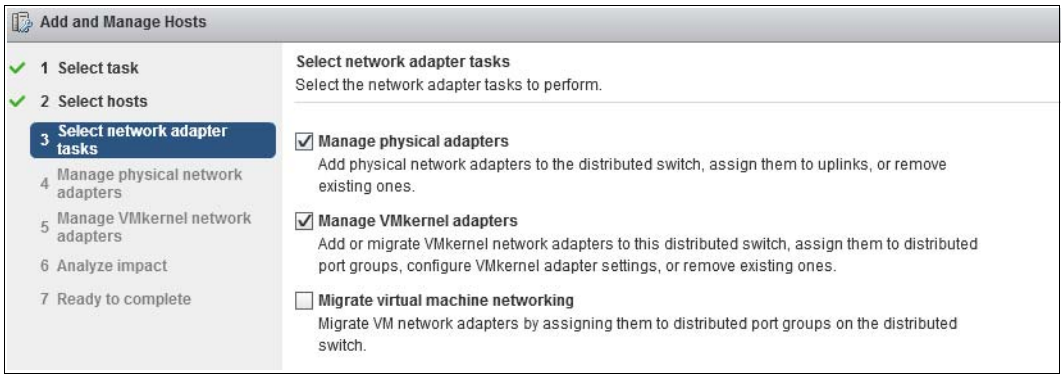

Figure 8 Select network adapter tasks

8. Configure vmnic1000002 in ESXi host (10.245.39.129) and vmnic1000402 in ESXi host (10.245.39.241) as uplinks for vDS, as shown in Figure 9 on page 9.

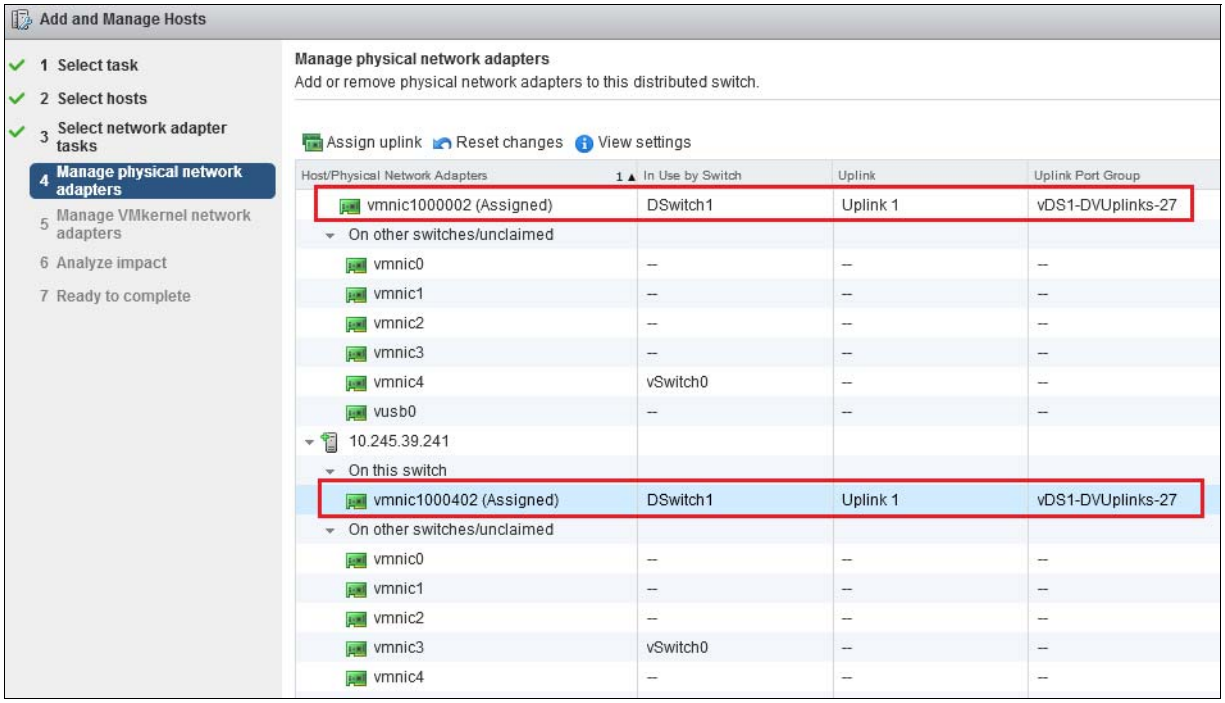

Figure 9 Manage physical network adapters

9. Attach the vmkernel adapter vmk1 to vDS port group vg-RDMA, as shown in Figure 10.

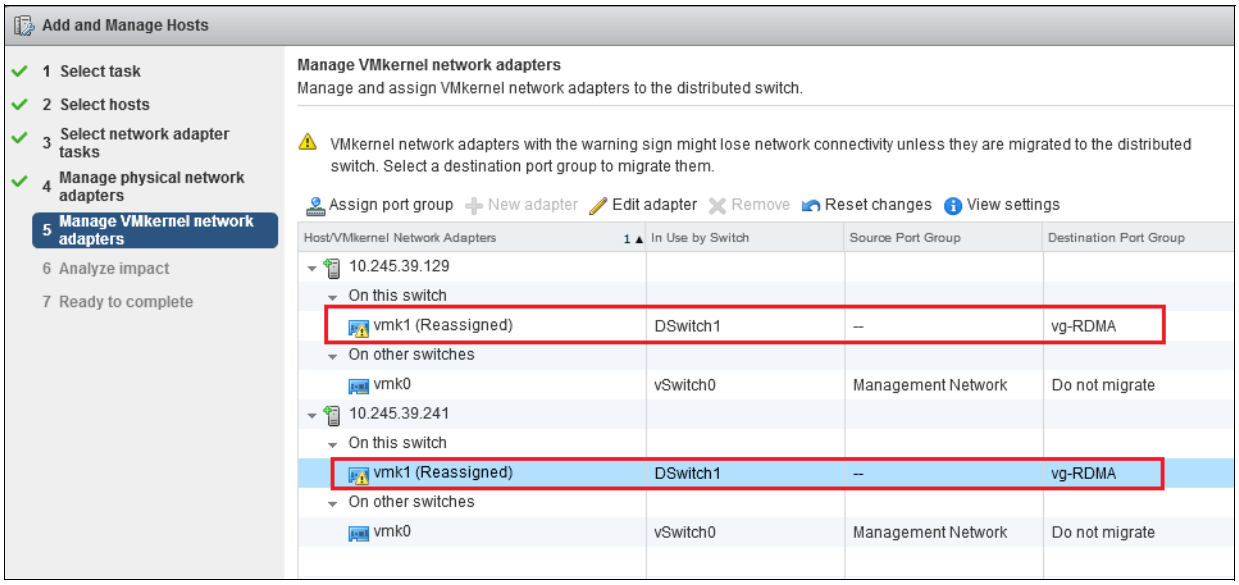

Figure 10 Manage VMkernel network adapters

Configure an ESXi Host for PVRDMA

To use PVRDMA in vSphere 6.5, your environment must meet several configuration requirements, as shown in Table 2 on page 10.

Table 2 Environment configuration requirements

Component	Requirement
vSphere	► ESXi 6.5 or later ► vCenter Server or vCenter Server Appliance 6.5 or later ► vSphere Distributed Switch
Physical host	Must be compatible with the ESXi release
Host Channel Adapter (HCA)	Must be compatible with the ESXi release **Notes:** Virtual machines that reside on different ESXi hosts require HCA to use RDMA. You must assign the HCA as an uplink for the vSphere Distributed Switch. PVRDMA does not support NIC teaming. The HCA must be the only uplink on the vSphere Distributed Switch. For virtual machines on the same ESXi hosts or virtual machines using the TCP-based fallback, the HCA is not required.
Virtual machine	Virtual hardware version 13 or later
Guest OS	Linux (64-bit)

To configure an ESXi host for PVRDMA, perform the following steps:

1. Tag a VMkernel Adapter for PVRDMA
2. Enable the Firewall Rule for PVRDMA
3. Assign a PVRDMA Adapter to a Virtual Machine

The following subsections describe each of these steps.

Tag a VMkernel Adapter for PVRDMA

Select a VMkernel adapter and enable it for PVRDMA communication using the following steps:

1. In the vSphere Web Client, navigate to the host.
2. On the Configure tab, expand **System**.
3. Click **Advanced System Settings**.
4. Locate Net.PVRDMA vmknic and click **Edit**.
5. Enter the value of the VMkernel adapter that you want to use and click OK. In our lab environment, we entered **`vmk1`** as shown in Figure 11 on page 11.

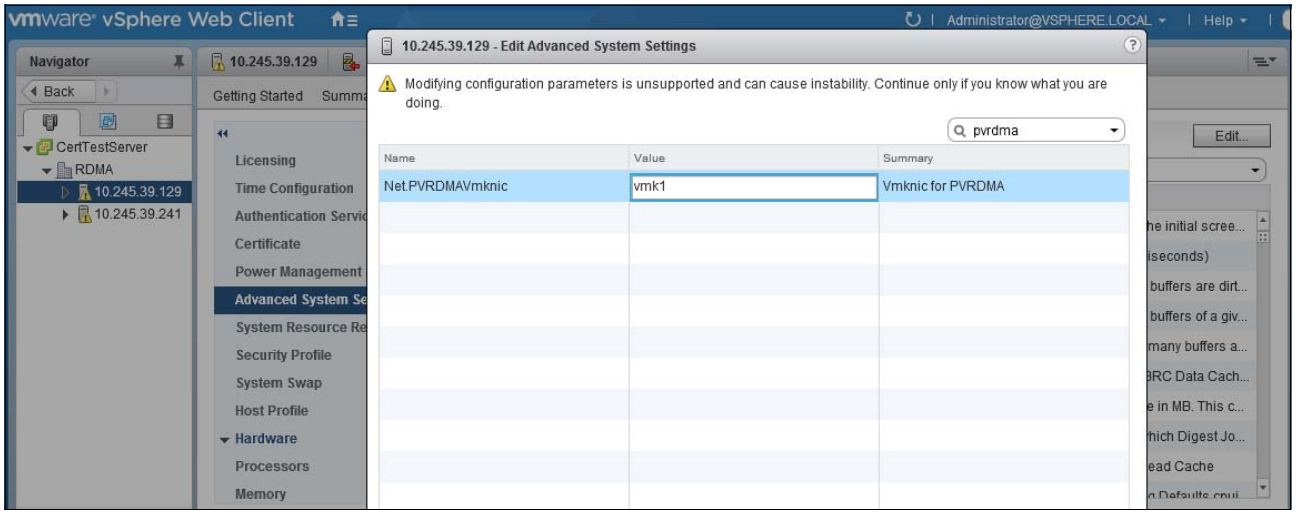

Figure 11 Tag a VMKernel Adapter for PVRDMA

Enable the Firewall Rule for PVRDMA

Enable the firewall rule for PVRDMA in the security profile of the ESXi host using the following procedure:

1. In the vSphere Web Client, navigate to the host.
2. On the Configure tab, expand **System**.
3. Click **Security Profile**.
4. In the Firewall section, click **Edit**.
5. Scroll to the pvrdma rule and select the check box next to it. See Figure 12.
6. Click **OK**.

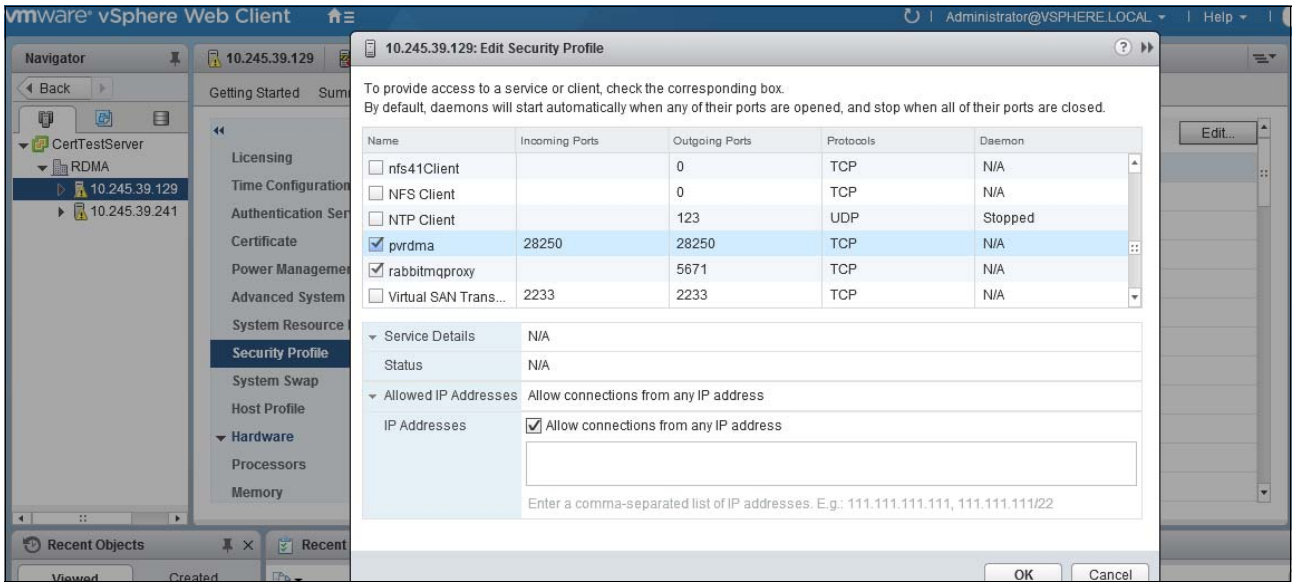

Figure 12 Enable the Firewall Rule for PVRDMA

Assign a PVRDMA Adapter to a Virtual Machine

To enable a virtual machine to exchange data using RDMA, you must associate the virtual machine with a PVRDMA network adapter. The steps are as follows:

1. Locate the virtual machine in the vSphere Web Client.
   a. Select a data center, folder, cluster, resource pool, or host and click the **VMs** tab.
   b. Click **Virtual Machines** and double-click the virtual machine from the list.
2. Power off the virtual machine.
3. In the **Configure** tab of the virtual machine, expand **Settings** and select **VM Hardware**.
4. Click **Edit** and select the **Virtual Hardware** tab in the dialog box displaying the settings, Figure 13.
5. At the bottom of the window next to New device, select **Network** and click **Add**.
6. Expand the **New Network** section and connect the virtual machine to a distributed port group.
7. For Adapter Type, select **PVRDMA**.

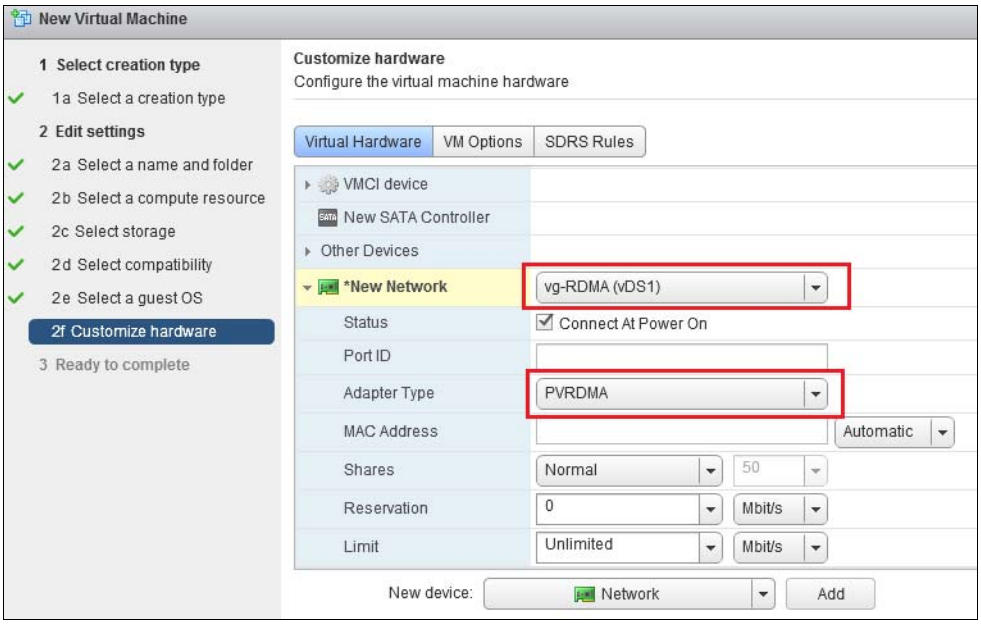

Figure 13 Guest OS network configuration for PVRDMA

8. Expand the **Memory** section, select **Reserve all guest memory (All locked)** as shown in Figure 14 on page 13.

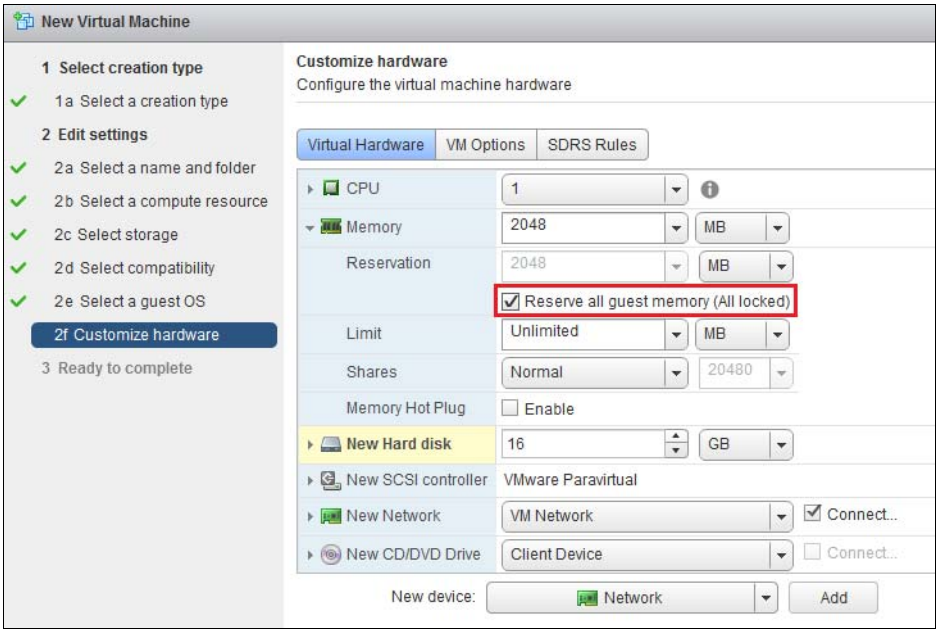

Figure 14 Guest OS network configuration for PVRDMA

9. Click **OK** to close the dialog window.
10. Power on the virtual machine.

Configure Guest OS for PVRDMA

For our lab test, we need two guest Linux OSes, each one located on an ESXi host. Our test topology is as shown in Figure 15.

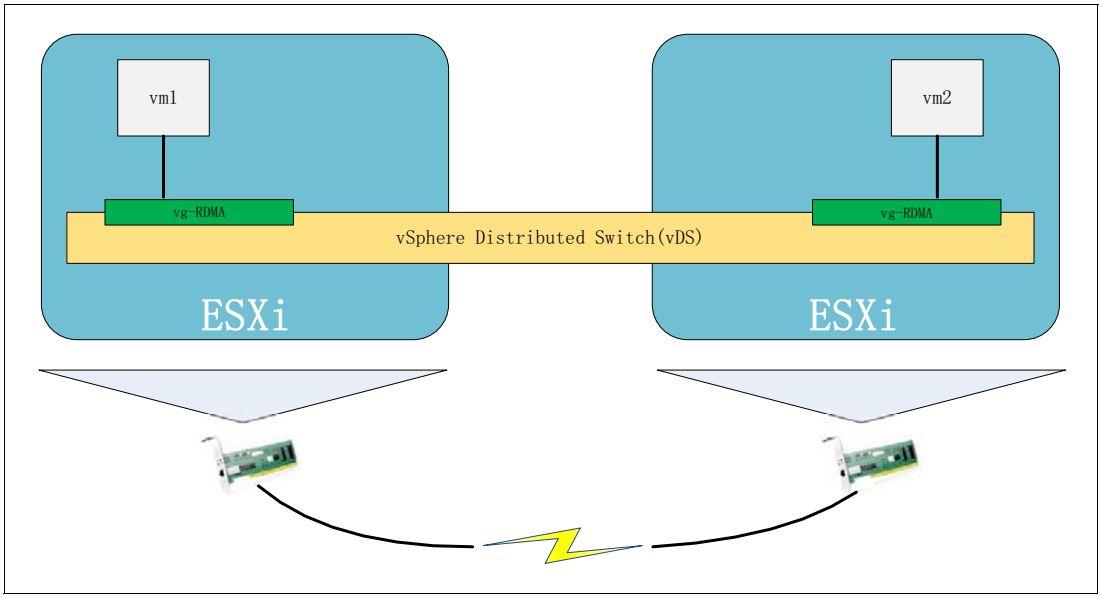

Figure 15 RDMA hardware topology

Perform the following three steps on each guest Linux OS:

1. Install the OFED Software Stack
2. Install the PVRDMA Driver and Lib
3. Check InfiniBand status

The following subsections describe each of these steps.

Install the OFED software stack

Download and install the software stack as follows:

1. Download OFED-3.18-3.tgz from the following page:

   http://www.openfabrics.org/downloads/OFED/OFED-3.18-3.tgz
2. Unzip it and execute install.pl
3. Select **2) Install OFED Software**
4. Select **3) All packages (all of Basic, HPC)**
5. Press **Q** to exit the installation

Install the PVRDMA Driver and Lib

Prerequisites: Kernel modules and libraries to support RDMA are required. You can download them from http://www.openfabrics.org or install them using the Linux-distributed package management tool.

Currently, the PVRDMA library and driver are not distributed in Linux packages. You can download them from these Git repositories:

- Userspace library: git://git.openfabrics.org/~aditr/libpvrdma.git
- Kernel driver: git://git.openfabrics.org/~aditr/pvrdma_driver.git

Run these commands, as root to install the kernel driver:

```
$ cd /path/to/kerneldriver.git
$ make
# insmod ./pvrdma.ko
```

Run these commands as root to install the userspace library:

```
$ cd /path/to/library.git
$ ./autogen.sh
$ ./configure --sysconfdir=/etc/ --libdir=/usr/lib
$ make
# make install
```

InfiniBand information checking

Now we can verify the InfiniBand information from the guest OS shell, as shown in Figure 16.

```
[root@localhost OFED-3.18-3]# ibv_devinfo
hca_id: vmw_pvrdma0
        transport:                      InfiniBand (0)
        fw_ver:                         1.0.000
        node_guid:                      0050:5600:00ab:3976
        sys_image_guid:                 0000:0000:0000:0000
        vendor_id:                      0x15ad
        vendor_part_id:                 2080
        hw_ver:                         0x1
        board_id:                       1
        phys_port_cnt:                  1
                port:   1
                        state:                  PORT_ACTIVE (4)
                        max_mtu:                4096 (5)
                        active_mtu:             1024 (3)
                        sm_lid:                 0
                        port_lid:               0
                        port_lmc:               0x00
                        link_layer:             Ethernet

[root@localhost OFED-3.18-3]#
```

Figure 16 InfiniBand information

We can see a port named `vmw_pvrdma0`, which is paired to the PVRDMA network adapter in the guest hardware configuration. In this case, the IP address is 172.16.1.200 and another one in the other guest OS will be configured as 172.16.1.100. Both of these IP addresses will be used in the RDMA traffic test.

```
[root@localhost ~]#
[root@localhost ~]# dmesg | grep -i paired
[  342.281343] vmw_pvrdma 0000:13:00.1: paired device to eno33559296
[root@localhost ~]#
[root@localhost ~]# ifconfig eno33559296
eno33559296: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.1.200  netmask 255.255.255.0  broadcast 172.16.1.255
        inet6 fe80::250:56ff:fe90:3b2f  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:90:3b:2f  txqueuelen 1000  (Ethernet)
        RX packets 336  bytes 25886 (25.2 KiB)
        RX errors 0  dropped 232  overruns 0  frame 0
        TX packets 94  bytes 8417 (8.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]#
```

Figure 17 InfiniBand info paired device

Running the RDMA traffic test

We will specify the guest OS with IP address 172.16.1.100 to act as the RDMA server and it will listen for RDMA traffic. The other one will act as an RDMA client. In our test, we will use the ib_write_bw command. In this case, the test lasts for 30 seconds and both the server and client results are shown in Figure 18 and Figure 19 on page 16 respectively.

```
[root@localhost ~]# ifconfig eno33559296 | grep -i inet
        inet 172.16.1.100  netmask 255.255.255.0  broadcast 172.16.1.255
        inet6 fe80::250:56ff:fe90:8cc4  prefixlen 64  scopeid 0x20<link>
[root@localhost ~]#
[root@localhost ~]# ib_write_bw -d vmw_pvrdma0 -F -D 30 --cpu_util

************************************
* Waiting for client to connect... *
************************************
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : vmw_pvrdma0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 CQ Moderation   : 100
 Mtu             : 1024[B]
 Link type       : Ethernet
 Gid index       : 0
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x0004 PSN 0x92bde RKey 0x000005 VAddr 0x007fc802d8f000
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:172:16:01:100
 remote address: LID 0000 QPN 0x0004 PSN 0xad27de RKey 0x000005 VAddr 0x007f4148afd000
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:172:16:01:200
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]    CPU_Util[%]
 65536      1117000          0.00               4363.28            0.069812         0.00
---------------------------------------------------------------------------------------
```

Figure 18 RDMA server result

```
[root@localhost ~]# ib_write_bw -d vmw_pvrdma0 -F -D 30 --cpu_util 172.16.1.100
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : vmw_pvrdma0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 TX depth        : 128
 CQ Moderation   : 100
 Mtu             : 1024[B]
 Link type       : Ethernet
 Gid index       : 0
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x0003 PSN 0x729a78 RKey 0x000004 VAddr 0x007fa14b286000
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:172:16:01:200
 remote address: LID 0000 QPN 0x0003 PSN 0x81333e RKey 0x000004 VAddr 0x007f712f7f0000
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:172:16:01:100
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]    CPU_Util[%]
 65536      1117000          0.00               4363.28            0.069812         100.00
---------------------------------------------------------------------------------------
```

Figure 19 RDMA client result

Conclusion

From the result, we can see that the RDMA bandwidth of the guest OS is 4,363.28 MB/sec, which is close to the physical port max bandwidth of 40 Gbps. This means that PVRDMA, *a* new feature in ESXi 6.5, has been successfully enabled in our test environment.

This function is compatible with most vSphere advanced features, such as vMotion, HA, Snapshots, and DRS. If you want higher performance and to use these features together in vSphere, PVRDMA is a good choice.

Further information

For more information, see the following web pages.

- Configuring Mellanox RDMA I/O Drivers for ESXi 5.x

  https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2058261

- Programming the Verbs API for a PVRDMA Device on ESXi 6.5 Hosts (2147694)

  https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2147694

- Toward a Paravirtual vRDMA Device for VMware ESXi Guests

  https://labs.vmware.com/vmtj/toward-a-paravirtual-vrdma-device-for-vmware-esxi-guests

Change history

- November 17, 2017:
  - Grammar and readability corrections

Author

Yi Liao is a VMware Engineer in the Lenovo Data Center Group in Beijing, China. He has a Master's Degree from China University of Mining and Technology. He joined Lenovo in 2015.

Thanks to the following people for their contributions to this project:

- Yonggang Wang, Lenovo OS enablement tester
- Junwu Liang, Lenovo OS enablement tester
- David Watts, Lenovo Press
- Mark T. Chapman, Lenovo Editor

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
1009 Think Place - Building One
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document was created or updated on November 17, 2017.

Send us your comments via the **Rate & Provide Feedback** form found at
http://lenovopress.com/lp0759

Trademarks

Lenovo, the Lenovo logo, and For Those Who Do are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. These and other Lenovo trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by Lenovo at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of Lenovo trademarks is available on the Web at http://www.lenovo.com/legal/copytrade.html.

The following terms are trademarks of Lenovo in the United States, other countries, or both:

Flex System™	ServeRAID™	System x®
Lenovo®	ServerGuide™	vNIC™
Lenovo(logo)®	ServerProven®	

The following terms are trademarks of other companies:

Intel, Intel Xeon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.