Configuration Examples and Troubleshooting for VMDirectPath

VMware ESX 4.x

VMDirectPath allows guest operating systems to directly access an I/O device, bypassing the virtualization layer. This direct path, or passthrough can improve performance for VMware ESX™ systems that utilize high-speed I/O devices, such as 10 Gigabit Ethernet.

ESX Host Requirements

VMDirectPath supports a direct device connection for virtual machines running on Intel Xeon 5500 systems, which feature an implementation of the I/O memory management unit (IOMMU) called Virtual Technology for Directed I/O (VT-d). VMDirectPath can work on AMD platforms with I/O Virtualization Technology (AMD IOMMU), but this configuration is offered as experimental support.

Some machines might not have this technology enabled in the BIOS by default. Refer to your hardware documentation to learn how to enable this technology in the BIOS.

PCI Device Compatibility

Each virtual machine can connect to up to two passthrough devices, which include certain networking and storage PCI devices. Supported devices include the Intel 82598 10 Gigabit Ethernet controller and Broadcom 57710 and 57711 10 Gigabit Ethernet controllers.

To configure devices for VMDirectPath, see “Configure Passthrough Devices on a Host” and “Configure a PCI Device on a Virtual Machine” in the *ESX Configuration Guide*. Also see KB 1010789, “Configuring VMDirectPath I/O passthrough devices on an ESX host.”

For an example of a VMX file with two passthrough devices connected in the virtual machine, see “Editing the VMX File to Support Advanced Configuration” on page 2.

Enable or Disable VMDirectPath

Enable or disable VMDirectPath through the hardware advanced settings page of the vSphere Client. Reboot the ESX host after enabling or disabling VMDirectPath.

Disable VMDirectPath and reboot the ESX host before removing physical devices.

To find the VMDirectPath Configuration page in the vSphere Client

1 Select the ESX host from Inventory.

2 Select the Configuration tab.

3 Select **Advanced Settings** under Hardware.

To disable and disconnect the PCI Device

1 Use the vSphere Client to disable or remove the VMDirectPath configuration.

2 Reboot the ESX host.

3 Physically remove the device from the ESX host.

Verify IOMMU Is Enabled After ESX Is Booted

Use the `--list` option of the `esxcfg-module` command from the ESX service console or the `vicfg-module.pl` command from the vSphere CLI to verify if IOMMU (VT-d or AMD IOMMU) is enabled. For details about this command, see "VMkernel Module Manipulation with vicfg-module" in the *vSphere Command-Line Interface Installation and Reference Guide*.

If IOMMU is not enabled, refer to the hardware manual for your Intel or AMD machine to learn how to enable VT-d or AMD IOMMU.

Editing the VMX File to Support Advanced Configuration

While many devices work fine in the virtual machine with the default settings, advanced configuration is sometimes necessary in the following cases:

- If you are not able to boot the virtual machine configured with passthrough devices.
- If the device is not working properly inside the virtual machine.
- You know a specific configuration is required.

Typically, you should not edit the VMX file because changes to passthrough devices in the vSphere Client overwrite any changes you make to this file. In addition, the following complications might arise:

- If you manually upgrade the virtual machine's hardware version from 4 to 7 by editing its VMX file, the passthrough configuration might not be complete and you might not be able to boot the virtual machine.
- If you manually remove or add lines related to the pciBridge, new passthrough devices might not be added during later configuration.

Sample VMX File

The following example shows the contents of a VMX file after two passthrough devices are added to the corresponding virtual machine.

This file is for reference. Do not use it as a template to overwrite the VMX file for your virtual machines.

In this example, there are two passthrough devices: `pciPassthru0` and `pciPassthru1`. These indicate the first and second PCI passthrough devices, respectively.

```
#!/bin/vmx

# version for configuration
config.version = "8"
# version for virtual machine (Regular version is 4)
virtualHW.version = "7"

# enable vnc
RemoteDisplay.vnc.enabled = "TRUE"
RemoteDisplay.vnc.port = "5900"
```

```
# type of guest os
guestOS = "linux"

# display name for the VI Client/WebCenter
displayName = "RHEL3"

# scsi controller 0
scsi0.present = "true"
scsi0.virtualDev = "lsilogic"

# scsi hard drive
scsi0:0.present = "true"
scsi0:0.fileName = "/volumes/your-path/passthru.vmdk"
scsi0:0.deviceType = "scsi-hardDisk"
scsi0:0.redo = ""

# IDE CD drive
ide0:0.present ="true"
ide0:0.startConnected = "TRUE"
ide0:0.fileName = "/volumes/your-path/your-iso-image"
ide0:0.deviceType = "cdrom-image"

memsize = "512"

sched.mem.max = "512"
sched.mem.minsize = "512"
sched.swap.derivedName = "/volumes/your-path/passthru-12345.vswp"
svga.vramSize = "16777216"


# Please try not modify any of the following lines since they are auto
# generated by the VI Client when configuring passthru devices.
pciPassthru0.present = "TRUE"
pciPassthru0.deviceId = "3456"
pciPassthru0.vendorId = "12ab"
# The systemId is equivalent to output of "vsish -e cat /system/systemUuid"
pciPassthru0.systemId = "48c4619b-6d58-18db-2a0e-000423d1e6f6"
pciPassthru0.id = "03:00.0"

pciPassthru1.present = "TRUE"
pciPassthru1.deviceId = "6543"
pciPassthru1.vendorId = "78cd"
pciPassthru1.systemId = "48c4619b-6d58-18db-2a0e-000423d1e6f6"
pciPassthru1.id = "02:00.0"

# Please try not modify any of the following lines since they are auto
# generated by the VI Client when performing HW version upgrading.
pciBridge0.present = "TRUE"
pciBridge4.present = "TRUE"
pciBridge5.present = "TRUE"
pciBridge6.present = "TRUE"
pciBridge7.present = "TRUE"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.pciSlotNumber = "21"
pciBridge4.functions = "8"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.pciSlotNumber = "22"
pciBridge5.functions = "8"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.pciSlotNumber = "23"
pciBridge6.functions = "8"
pciBridge7.virtualDev = "pcieRootPort"
pciBridge7.pciSlotNumber = "24"
pciBridge7.functions = "8"

pciBridge0.pciSlotNumber = "19"
pciPassthru0.pciSlotNumber = "160"
pciPassthru1.pciSlotNumber = "192"
```

PCIPassthru Error Message

Sometimes the default passthrough setting for PCI devices might not work as expected. You might encounter the following error when you try to boot the virtual machine:

```
PCIPassthru: Device 016:00.0 barIndex 0 type Mem realaddr 0xfa000000 size 33554432
PCIPassthru: PCI device 016:00.0 is marked wrong PCIe
PCIPassthru: Failed to register PCI slot 016:00.0
```

Setting `pciPassthru0.virtualDev = "pci"` resolves this error and allows PCI and PCI-x devices to work properly as VMDirectPath devices. The default of this option is `pcie`.

Setting the Physical Mode for IOAPIC

Setting `pciPassthru0.msiEnabled = "FALSE"` sets the physical mode to IOAPIC, if the virtual mode is IOAPIC. The default setting for this option is `TRUE`, which means physical mode is MSI or MSI-X. This default works for most supported configurations. In some cases, however, you need to change the default.

One such case is when you have a Broadcom 57710 or 57711 device connected to a Windows 2003 or 2008 virtual machine. If either of these devices are not working as expected with passthrough (if you see a yellow icon displayed in the Device Manager panel for the VMDirectPath device), edit the `pciPassthru<N>.msiEnabled` setting in the virtual machine's VMX file to False, to enable IOAPIC.

Devices that are known to work in the default configuration of physical MSI for Windows 2003:

- QLogic Fibre Channel 2500
- LSI SAS 1068E
- Intel 82598

Problems with Device Assignment Dependencies

Because of the limits of the PCI bus or individual devices, it is sometimes necessary for multiple devices to be assigned to passthrough mode on the host together. These dependencies exist because passthrough requires a way to reset devices and it is sometimes not possible to reset a device without also resetting other devices. If you know that a device will reset itself properly through a D3 to D0 power transition, you can edit `/etc/vmware/passthru.map` to add an entry for your device (on ESX hosts through the service console).

In addition to reset method constraints, it is also possible that although a device has multiple PCI functions, they cannot be used simultaneously by different virtual machines or by a virtual machine and the VMkernel, due to hardware and vendor-supplied software limitations.

A sample `pcipassthru.map` follows. The top part of the file describes the format for device entries.

```
# passthrough attributes for devices
# file format: vendor_id device_id resetMethod fptShareable
# vendor/device id: xxxx (in hex) (ffff can be used for wildchar match)
# reset methods: flr, d3d0, link, bridge, default
# fptShareable: true/default, false

# Intel 82598 (Oplin) 10Gig cards can be reset with d3d0
8086 10b6 d3d0 default
8086 10c6 d3d0 default
8086 10c7 d3d0 default
8086 10c8 d3d0 default
8086 10dd d3d0 default
# Broadcom 57710/57711 10Gig cards are not shareable
14e4 164e default false
14e4 164f default false
14e4 1650 default false
# Qlogic 8Gb FC card can not be shared
1077 2532 default false
# LSILogic 1068 based SAS controllers
1000 0056 d3d0 default
1000 0058 d3d0 default
```

Vendor ID and Device ID

You can obtain vendor and device IDs using the `lspci -n` command on the ESX service console. The vendor ID and the device ID appear at the end of the output string in brackets, separated by a colon. Refer to the *lspci(8)* man page for more information.

Reset Method

Possible values for the reset method include flr, d3d0, link, bridge, or default.

The default setting is described as follows. If a device supports function level reset (FLR), ESX always uses FLR. If the device does not support FLR, ESX next defaults to link reset and bus reset in that order. Link reset and bus reset might prevent some devices from being assigned to different virtual machines, or from being assigned between the VMkernel and virtual machines. In the absence of FLR, it is possible to use PCI Power Management capability (D3 to D0 transitions) to trigger a reset. Most of the Intel NICs and various other HBAs support this mode.

Full Passthrough Shareable

The values for fptShareable can be `true` or `false`. The default is true, which means the PCI device can be shared. Sharing refers to using multiple functions of a multi-function device in different contexts. That is, sharing between two virtual machines or between a virtual machine and VMkernel.

NOTE Dual-function QLogic 2532 adapters cannot be shared between two virtual machines because the QLogic driver uses function numbers to retrieve FC credentials. This process does not work well when the functions are split between two virtual machines or between VMkernel and virtual machines.

Problems with Virtual IRQ Sharing

In some cases when a full passthrough device in a guest shares virtual IRQ (IOAPIC) with another virtual device (for example, vmxnet), performance problems might occur. These performance problems are due to the chaining of interrupt service routines. Depending on the order in which drivers are loaded inside the guest and the order of the chaining of interrupt service routines, you might see different performance behavior. Recent versions of Linux or Windows typically do not exhibit this behavior.

As a workaround, disable and re-enable the virtual device that shares IRQ to change the sharing order. Another alternative is to disable the virtual device that shares virtual IRQ.

Problems with Physical IRQ Sharing

Due to an IRQ sharing issue, the PCI device and its driver might stop communicating when the physical device mode is MSI and the virtual device is in IOAPIC mode.

To resolve this problem, change the VMX file setting `pciPassthru0.msiEnabled` from true (MSI) to false (IOAPIC). See "Setting the Physical Mode for IOAPIC" on page 4.

If you have comments about this documentation, submit your feedback to: docfeedback@vmware.com

VMware, Inc. 3401 Hillview Ave., Palo Alto, CA 94304 www.vmware.com

Item: EN-000217-01