NVIDIA

# MULTI-USER VIRTUAL REALITY SYSTEM

DG-08567-001_v01 | May 2017

## Reference Design Guide

**To access the latest version of this document, visit: http://info.nvidia.com/vrsystem.html.**

## DOCUMENT CHANGE HISTORY

DG-08567-001_v01

| Version | Date | Authors | Description of Change |
|---|---|---|---|
| 01 | May, 2017 | FD | Initial release |
| | | | |
| | | | |

# TABLE OF CONTENTS

# LIST OF FIGURES



# LIST OF TABLES

# BUILDING A MULTI-USER VR SYSTEM

When considering uses of virtual reality (VR) in the enterprise, many enterprises require collaboration, or simultaneous access, or both, to VR by multiple participants at the same physical location. Most companies already have places set aside for collaborative work such as conference rooms, operations or design centers, classrooms, or auditoriums. Some businesses are built specifically around a group experience. Examples of such businesses include educational institutions, theaters, theme parks, and convention centers. Multi-user VR solutions are applicable to these use cases.

The conventional approach to applying a multi-user VR solution would be to deploy multiple individual systems with one system for each user. This approach is adopted because consumer VR systems have been designed with the assumption that each user's head-mounted display (HMD) and hand controllers are attached to a dedicated host machine.

This approach is impracticable for enterprise use in multipurpose venues because the bulkiness of multiple systems, the jumble of wiring, and the time consumed by setup and teardown would tie up the space for too long. Managing separate systems and keeping them all up to date is also more challenging.

A multi-user VR system with capabilities that meet enterprise standards for availability, maintenance, form-factor, scalability and portability can overcome these drawbacks.

This document describes how NVIDIA designed and built an **experimental** multi-user VR system capable of addressing a variety of enterprise needs.

**Note:** This document is **not** intended to provide complete instructions for designing and building a production system. The system described in this document is suggested as a reference configuration that may be used as a basis for further experimentation, evolution, and refinement of the multi-user VR concept. NVIDIA **cannot** provide technical support or services to the DIY community based on the information contained in this document.

# DESIGN REQUIREMENTS

The goal of this experiment was to build a single machine capable of delivering a rich user experience to multiple HMD devices. VR participants could then use the same physical hardware, thereby minimizing space, power, cooling, maintenance, and complexity of setup.

The system should appear logically identical to a conventional arrangement of multiple networked PCs to ensure compatibility with existing VR ecosystem software and peripherals. It must be easily packaged and ruggedized to enable rapid relocation and shipping while each user experience must meet or exceed that of an individual PC.

To deliver a rich user experience to multiple HMD devices from a single machine, the design had to meet these requirements:

- The ability to connect two or more HMDs to a single machine
- DisplayPort 1.2 or HDMI 2.0/2.1 video output with graphics at 90 FPS and 4K resolution to be delivered to each HMD
- USB 3.0 root hub data interconnect to each HMD
- No dependency on a specific make of HMD, such as HTC or Oculus
- 3200 W redundant power limit (two 110 or 220 kVA circuits to maintain redundancy)

# TYPES OF VR PAYLOADS

## Real-Time 3D Rendered Models

Each frame shown on the HMD display must be rendered instantly by using the latest virtual camera position, which is determined by the position and attitude of the user's head and hand controllers in physical space. Apart from a little warping to account for lens distortion and interpolation of the latest available tracking data, the only pixels that are rendered are generally the pixels that will ultimately be visible in the HMD display at the current time. Low-latency tracking, rendering, and delivery of pixels is critical for good VR. The consensus is that good VR requires less than 20 ms motion-to-photon latency.

## Rendered 360 Video Content

360 Video content, whether captured from reality or rendered offline, generally refers to 360-degree panoramic images and video in either mono or stereo formats.

For each timestep, a wrap-around image covering all possible viewing angles surrounding the camera is delivered to the user's host system. Based on head orientation, a sub-image is extracted from the wrap-around image, which completely fills the user's HMD.

This technique is convenient for streaming 360 video over a network to a PC or wireless smartphone because the final pixel selection is made at the receiving end, thereby avoiding a round trip to the server and preserving the required low motion-to-photon latency.

# VIRTUAL REALITY ON VIRTUAL MACHINES

Conventional VDI configurations are not best suited for either real-time 3D rendered video content or rendered 360 video content. VDI involves use of a broker for remoting over network, which is not typically required in a VR setup. More importantly, latency requirements in typical VDI setup are not as stringent as VR. Good VR requires motion-to-photon latency of less than 20 ms, whereas most VDI applications require 120ms for input, upload, rendering, encoding, download, and decoding.

However, VR engines can be run on virtual machines for multi-user VR if conventional VDI configurations are modified as follows:

- Each HMD is directly connected to a dedicated physical GPU and USB card in a local server only a few feet away from the user.
- A dedicated physical GPU and USB card are directly passed through to each virtual machine to provide full acceleration, driver compatibility, and low-latency benefits.
- Enough CPU cores are dedicated to each virtual machine.

These design decisions enable a high-quality user experience that matches the quality on physically separate PCs to be maintained on virtualized hardware. Though still limited by the HMD tether length, this solution still benefits from the hardware density, shared storage, and IT manageability that virtualized systems provide. It does so by avoiding the latency associated with pushing pixels over a network as VDI does.

## HOW PHYSICS AFFECTS THE REQUIREMENTS

Quality HMDs support high resolutions and frame rates with low-latency head tracking to deliver wide field-of-view immersion with minimal motion-to-photon latency. Consequently, the best visual experiences today are being delivered over tethered cables between the host system and HMD. With future expectations of even greater resolutions and frame rates, tethered solutions are likely to be around for some time. Even with emerging wireless solutions, some cables between the system and wireless transmitters will still be required.

The maximum distance between the content rendering and the HMD is somewhat limited. HDMI 2.1 cables have a maximum specified length of 15 meters (49 feet) without the need for a range extender to rebroadcast the video content. USB 3.0, unlike previous versions of the USB standard, does not directly specify a maximum cable length. It requires only that all cables meet an electrical specification: For copper cabling with AWG 26 wires, the maximum practical length of a USB 3.0 cable is 3 meters (9.8 feet). These figures are only guidelines and NVIDIA has successfully tested the use of 25-feet USB 3.0 active cables at several trade shows.

However, NVIDIA has not succeeded in enabling an HMD to tolerate the latency that an IP range extender adds to a USB 3.0 connection. Fortunately, HTC offers the VIVE Business Edition HMD, which increases the total tether distance, for video, power, and USB cables, from 5 meters to 10 meters. It achieves this increase through the addition of a second link box and tether extension cable. This increased tether distance allows more flexibility in routing HMD tethers from a centralized multi-user VR system to the user position for a wide variety of use-case physical layouts.

# HARDWARE

The choice of hardware significantly affects the stability, ruggedness and performance of the system. Where possible, enterprise systems should use enterprise-grade components and not rely on consumer-grade components to reduce hardware costs.

## CHASSIS AND MOTHERBOARD

Multi-user VR systems require a machine that can integrate multiple GPUs and USB controllers with the motherboard. Each HMD requires a dedicated GPU controller paired with a dedicated USB controller. The requirement for external access to these controllers is an additional limiting factor.

For optimal performance, both the GPU controller and the USB controller must have access to the same I/O hub with a very fast PCIe switch. This requirement also ensures that the PCIe traffic between the USB controller, GPU, and the VM does not cross the host bridge between the CPUs, but instead stays on the same contiguous PCIe bus. Keeping this traffic on the same PCIe bus enables the CPU traffic of the VM to be isolated to a single socket, which improves performance.

Server motherboards based on the Intel® C612 chipset have a 5 GT/s DMI2 bus speed and can support up to eight PCIe generation 3×16 cards at either full speed or with minimal drop-off. Performance depends on the communication speed between the GPU cards and the USB controllers, as well as overall traffic patterns.

**Note:** The sample build uses the Supermicro 4028GR-TR motherboard with integrated X10DRG-O+-CPU and the X9DRG-O-PCIE daughterboard that supports two Intel® Xeon® E5-2600 v4 processors, up to 160 W TDP.

The X9DRG-O-PCIE daughterboard has a dual-root complex that balances PCIe slots across CPU1 and CPU2, whereas the X10DRG-O-PCIE daughter board has a single-root complex with nearly all PCIE slots connected to CPU1 and is therefore **not** recommended.

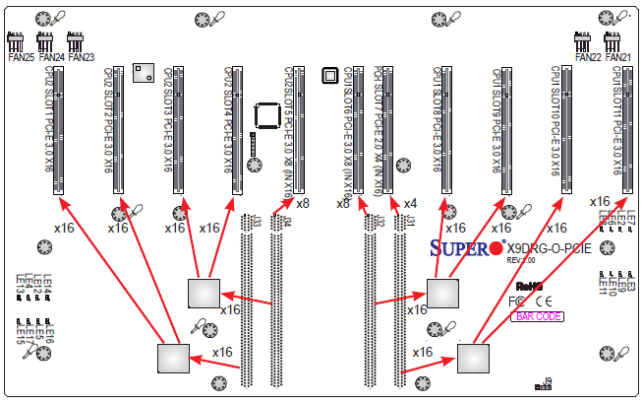
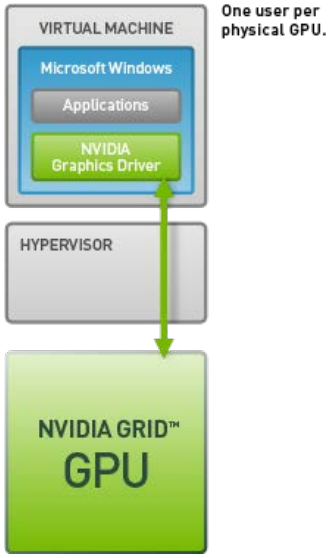



Figure 1 X9DRG-O-PCIE Daughterboard

# GPUS

NVIDIA Quadro® GPUs have the capabilities necessary for an enterprise implementation of a multi-user VR machine. In addition to large on-board memory capacities to handle larger VR datasets, Quadro GPUs have the inherent feature for NVIDIA® Multi-OS technology, which enables the GPU to run attached as passthrough to a virtual machine. NVIDIA GPU passthrough technology lets you create a virtual workstation that gives users all the benefits of a dedicated graphics processor, including the high frame rates and low latencies necessary for premium HMD VR. The multiple virtual machines in one system appear logically as several separate PCs on a network, so there is no impact to VR software development.




Use of the proven NVIDIA Quadro graphics driver in the virtual machine ensures full application compatibility with traditional non-virtualized workstation environments. It also ensures the availability of OpenGL 4.4, DirectX 11, and CUDA® 6.0. Availability of these versions ensures full support for VR-ready applications.

Table 1 GPU Comparison

| GPU | Code Name | Family | CUDA Parallel-Processor Cores | GPU Memory | Max Power Consumption |
|---|---|---|---|---|---|
| QUADRO GP100 | | Pascal | 3584 | 16 GB | 235 W |
| QUADRO P6000 | | Pascal | 3840 | 24 GB | 250 W |
| QUADRO P5000 | | Pascal | 2560 | 16 GB | 180 W |
| QUADRO P4000 | | Pascal | 1792 | 8 GB | 105 W |
| QUADRO M6000 24GB | | Maxwell | 3072 | 24 GB | 250 W |
| QUADRO M6000 | | Maxwell | 3072 | 12 GB | 250 W |
| QUADRO M5000 | | Maxwell | 2048 | 8 GB | 150 W |

**Note:** The sample build uses the Quadro P6000 GPU.

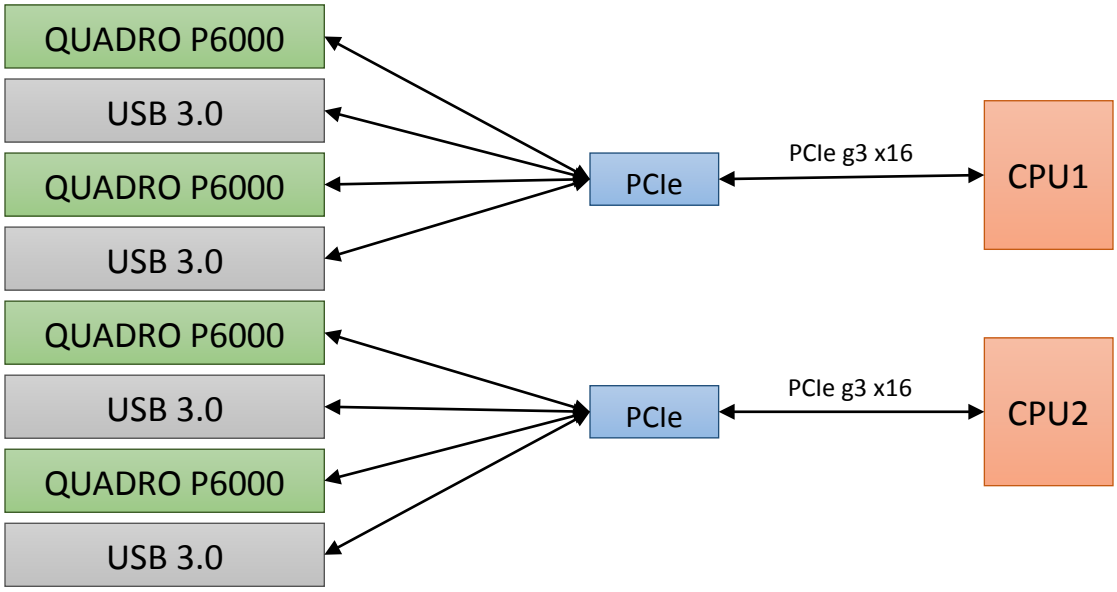



Figure 2 Topology of the Test System

# USB 3.0 CONTROLLERS

HMDs require dedicated USB 3.0 connections to provide the low latencies needed for full functionality of the display. For example, HTC VIVE requires one USB 3.0 port and Oculus Rift requires three USB 3.0 ports. To meet this requirement for dedicated USB 3.0 connections, a multiport USB controller must be passed through to the virtual machine.

**CAUTION:** Because of a limitation in the hypervisor, the entire USB root hub must be passed through to the virtual machine at the PCIe device address. You **cannot** pass through only the required USB 3.0 ports on the hub.

USB controllers are not currently supported by VMware vSphere ESXi.

To ensure support for the variety of HMD display requirements, the USB 3.0 controller must provide the full specified performance to all ports and must meet a minimum compliance with PCI-E rev 2.0 and xHCI 1.0.

Figure 3 Quadro GPUs and Inateck USB 3.0 Controllers in the Sample System

**Note:** The sample build uses the Inateck KTU3FR-502I US USB 3.0 controller, which is compatible with 32-bit and 64-bit Windows® 7, Windows 8, and Windows 10 operating systems. It provides five USB 3.0 ports and one 20-pin internal USB 3.0 port. It requires a 15-pin SATA power connector.

# CPU AND COOLING

The CPU must be compatible with the selected chipset while providing sufficient PCIe support. Enterprise CPUs are ideal because the application target is sensitive to failures and the multi-user VR system is acting as a server hosting multiple user sessions rather than a single workstation. Table 2 lists recommended CPUs for the sample build.

**Note:** Intel Core i5 and Core i7 CPUs do not support dual-CPU configurations and are unsuitable for this build. Enterprise CPUs, such as the Intel Xeon, are required.

Table 2 Recommended CPUs

| CPU | PCIe Configuration | Cores | Processor Base Frequency | TDP (W) |
|---|---|---|---|---|
| Xeon E5-2623v4 | 40 | 4 | 2.6 GHz | 85 W |
| Xeon E5-2637v4 | 40 | 4 | 3.5 GHz | 135 W |
| Xeon E5-2643v4 | 40 | 6 | 3.4 GHz | 135 W |
| Xeon E5-2620v4 | 40 | 8 | 2.1 GHz | 85 W |
| Xeon E5-2667v4 | 40 | 8 | 3.2 GHz | 135 W |

Because this system is a GPU system, the configuration must include proper GPU power and cooling kits. Meeting this requirement typically entails using a modified low-profile CPU heat sink and special power whips for the GPUs. Because the CPU heat sink is not full height, thermal considerations require that the power rating for the CPU meet the thermal specifications for the chassis. Consult your chassis manufacturer to determine the proper TDP wattage limits for GPU-enabled systems.

**Note:** The sample system uses the Xeon E5-2697v4 2.3GHz CPU, which does not provide optimum performance. It has 18 cores and is designed for highly dense server virtualization. It also consumes 145 W TDP, which is slightly high for a GPU system and runs slightly too hot for production use.

# MEMORY AND STORAGE

The Intel Xeon E5-2600v4 CPU supports a maximum of four memory channels and up to 1.54 TB of DDR4 2400/2133/1866/1600 MHz ECC SDRAM. The sample system has 256 GB of DDR4 1600 MHz ECC SDRAM.

The mass storage I/O subsystem must offer fast performance because the GPUs can consume data at rates far beyond the capabilities of a single mechanical drive. SSDs provide speed but increase system cost while limiting capacity. To deliver optimal performance for the VR experience, the sample system uses SSDs for VM storage, cache, and the boot device as follows:

- One 256-GB SSD is used for cache and the boot device.
- Four 256 GB-SSDs are used for VM storage, with each SSD dedicated to a single guest VM.

To provide the fault tolerance that enterprise systems require, the SSDs would be configured as follows:

- The 256-GB SSD for the system boot device would be in a RAID 0 mirror configuration.
- The four 256-GB SSDs for VM storage should be combined into a RAID 5 array.

# THERMAL AND ACOUSTIC CONSIDERATIONS

Acoustics and heat management are major considerations, especially if the sample multi-user VR system is to be deployed in a normal office environment. A chassis that separates the power supply and disks from the heat generated by the CPU and GPUs is ideal. Furthermore, effective cooling requires generating significant positive air pressure inside the chassis to draw air through the gaps between the closely spaced NVIDIA Quadro GPU cards. These cards include blower-style active cooling fans that direct airflow through the GPU heat sinks and out of the chassis to prevent internal heat buildup.

Spacing between the GPU cards is critical for consistent airflow. However, in a chassis where GPUs are mounted vertically spacing isn't usually a problem. Maintaining proper airflow reduces the thermal stress on the system and lowers the fan speed required to keep the system cool.

Even with all these features, GPU-based systems typically run hot and are much louder than systems without a GPU. To isolate users from the heat and the noise, these systems are designed for data center use, not office environments. Therefore, proper acoustic cabinets should be considered if these types of systems are to be used in an office environment rather than a data center.

**Note:** The sample system is loud. To protect their hearing, anybody in the same room as the system would need to use ear defenders.

# POWER

To provide a fully redundant power configuration that is enterprise ready and fault tolerant, the sample system has four 1600 W power supplies, only two of which are required to operate the system at full power with four HMDs in session.

Figure 4 Rear of the Sample System Showing Four 1600 W Power Supplies

# HYPERVISOR SYSTEM

Many factors influence the choice hypervisor, for example, personal preference, feature set, previous operating knowledge compared to training requirements, corporate enterprise standards, and so forth. Irrespective of these factors, a hypervisor used for an enterprise deployment must meet these requirements:

- The server must be on the supported hardware compatibility list for the hypervisor vendor.
- The hypervisor version must be on the supported software compatibility list for the NVIDIA GPUs.
- The hypervisor must support PCIe devices in passthrough.

For the sample system, the VMware vSphere Hypervisor 6.5 was chosen, despite it not meeting these requirements. Because the sample system is intended a proof of concept, an unsupported, untested, and unproven configuration was acceptable.

## HYPERVISOR INSTALLATION

After the hypervisor has been selected, the following software is needed to get started:

- The latest version of the hypervisor installer, which, for the sample system, is VMware vSphere at http://www.vmware.com/go/evaluate-vsphere-en
- The latest version of any virtual machine tools for the chosen hypervisor
- The latest version of any management tools needed to manage the server, for example:
  - RealVNC for ease of access and configuration
  - A secure shell (SSH) client, such as PuTTY
  - A secure copy client, such as PSCP, the PuTTY secure copy client

On a system enabled with IPMI, iLO, DRAC, or another remote management controller, the simplest way to the install the software is to use the management console to attach the ISO image of the hypervisor installer. Otherwise, installing the software entails creating a bootable USB flash drive from the ISO image, or burning the ISO image to an optical disc and attaching a USB optical drive to the system.

**Note:** If you plan to use DHCP in your network configuration, note the MAC addresses of your primary and secondary network controllers. You need these addresses either to manage a MAC reservation on your DHCP server, or to retrieve the IP address assigned by referencing the DHCP server.

For the sample system, no advanced configuration options were set during installation. All defaults were accepted, for example:

- Network configuration was left as DHCP.
- Storage was left as local standalone SSDs in the server for the virtual machines.

If you are developing a system like the sample system, install and configure the hypervisor software to meet your needs, including any advanced network configuration and storage configuration to prepare the system for virtual machines.

# CONFIGURING PASSTHROUGH DEVICES

After the hypervisor software is installed, it must be configured to enable the hardware for PCIe passthrough.

This example demonstrates the configuration of PCIe passthrough on the VMware vSphere 6.5 web client.

1. After logging in to the VMware 6.5 vSphere web client navigate to the **Host →Management → Hardware** tab.

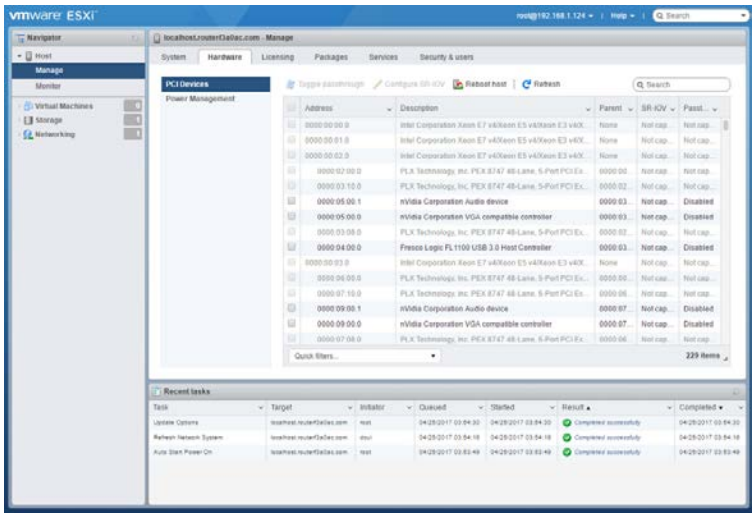


Figure 5 VMware 6.5 vSphere Host Hardware Management Tab

2. In the body of the **Hardware** tab, scroll through and select these hardware items:
   - **nVidia Corporation Audio device**
   - **nVidia Corporation VGA compatible controller**
   - Your USB 3.0 Host controller

   After these items selected, they are marked with a check mark, and the **Toggle Passthrough** button is enabled at the top of the body of the **Hardware** tab.
3. After selecting all the hardware that you want to enable, click **Toggle Passthrough**.

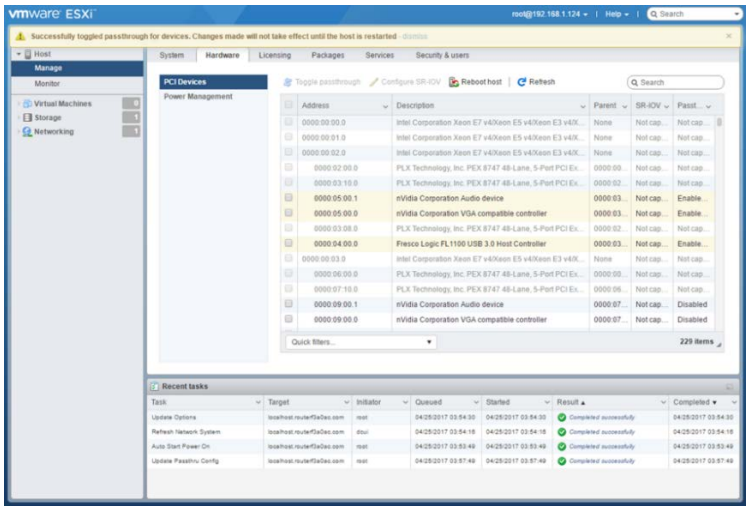


Figure 6 VMware 6.5 vSphere Host Hardware Management Tab – Toggle Passthrough enabled.

4. Reboot the system to finalize and activate the hardware passthrough.

After the reboot, the USB 3.0 controller is ready to be assigned to a virtual machine.

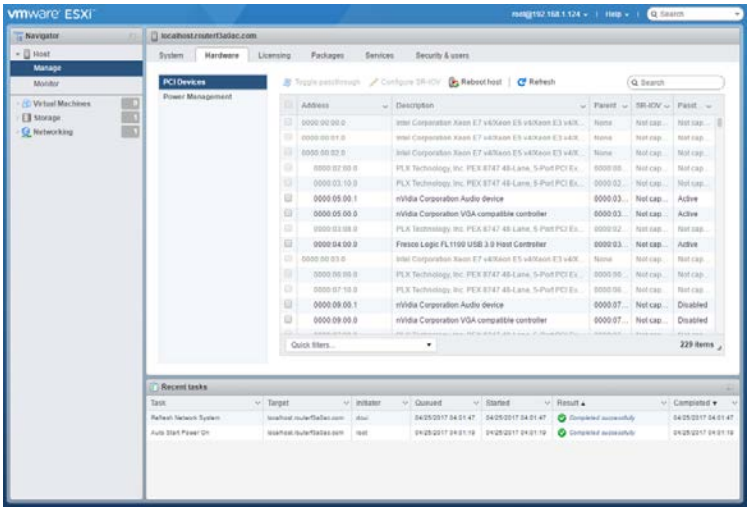



Figure 7 VMware 6.5 vSphere Host Hardware Management Tab - Active Hardware Passthrough

# VM GUEST SYSTEMS

After configuring the device hardware in the hypervisor to allow it to be used in PCIe passthrough, create the virtual machines so that you can add the PCIe hardware to the virtual machine configuration.

> **Note:** The sample system uses only 64-bit Windows 10 virtual machines. However, other Windows operating systems are supported by HMDs and software such as Steam.

## CREATING A VM FOR PCIE PASSTHROUGH HARDWARE ON VMWARE VSPHERE 6.5

1. Use the VMware vSphere web client to connect to your hypervisor through a web browser.
2. Create a Windows VM either by using your standard template or manually.

   In the **Customize Settings** dialog box, the following settings affect system performance:

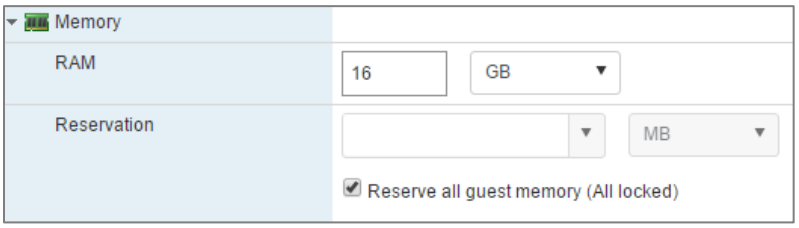
   - **CPU Socket Pinning**

     For optimum performance, allocate all cores assigned to your VM from the same CPU socket. Allocating the cores this way ensures that instruction traffic does not have to traverse the front side bus between CPU0 and CPU1 during normal VM operations.

     Ensure that the number cores per socket noted on the drop-down list matches the total number of cores being allocated. Additionally, ensure that you allocate cores to the proper topology for your CPU. To ensure the proper sizing allocation, consult the documentation for the CPU and the VMware hypervisor.


- **Reserve all memory**

  PCIe passthrough requires that all allocated memory be 100% reserved.




- **Thick Provisioning HDD resources**

  This setting has only minimal impact on the execution of the VM. However, thick provisioned eager zero allocation provides the fastest I/O performance. Ensure you consider the I/O requirements of your application before deciding how to set this option.

- **USB Controller set to USB 3.0**

  This setting applies to the virtual controller, **not** the PCIe passthrough device. The point to note here is to ensure that all USB passthrough devices and the virtual drivers are set to USB 3.0.


- **Virtual Network VMXNET 3**

  For best performance, 64-bit operating systems should use the VMXNET 3 virtual NIC hardware from VMware.

# ADDING THE PCIE DEVICES

After creating your virtual machine and saving the initial configuration, edit the settings of the virtual machine again to add the USB 3.0 controller, NVIDIA VGA compatible controller, and NVIDIA audio controller as PCIe devices.

1. Add each device to the virtual machine:
   a) In the Edit settings dialog box, click **Add other device** and from the menu that opens, choose **PCI device**.

      A **New PCI device** item is added to the bottom of the list of devices in the Edit settings dialog box.

   b) From the drop-down list, select the appropriate device.

c) With more devices in the list, be sure to check the hardware address listed next to the device name and select devices that are associated with the same PCIe host bus.

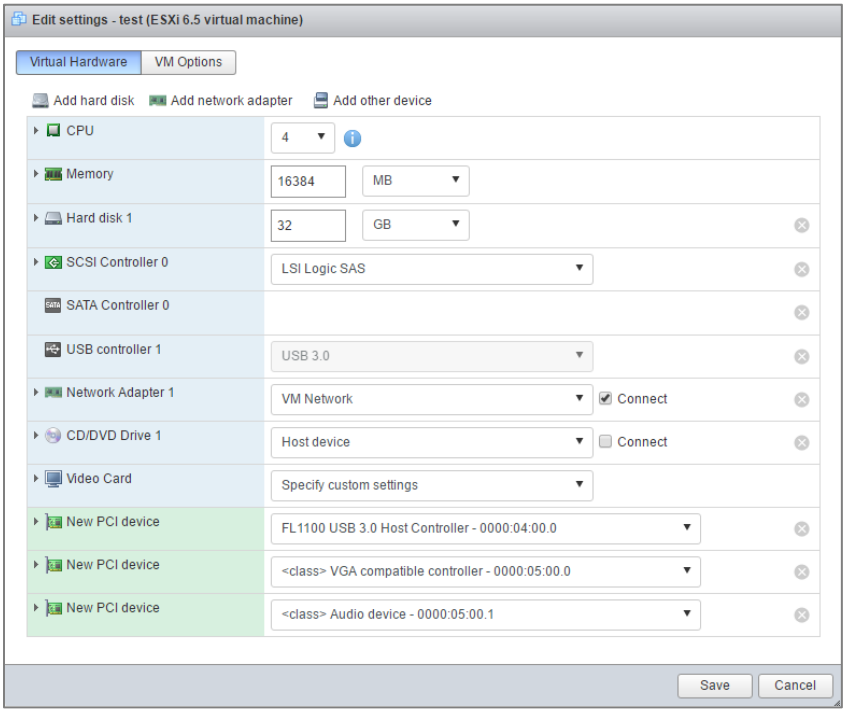


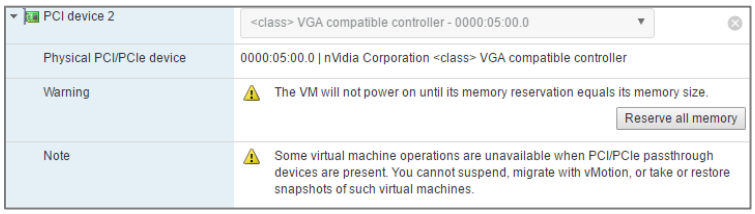

Figure 8 VMware Edit Setting Dialogue - Adding New PCIe Devices to Virtual Machine

2. After adding the devices, confirm that the virtual machine is installed.

   If the virtual machine has not been installed, attach your ISO file from your ISO library before you boot the virtual machine.

3. Boot the virtual machine.

   If your VM fails to boot with a warning about reserved memory, ensure you have allocated all assigned memory and the reservation matches the allocated amount.



# RESOLVING UNIDENTIFIED DEVICES

After the Windows operating system has been installed on the virtual machine, the Device Manager shows that a display adapter and the PCIe passthrough USB 3.0 controller are unidentified.

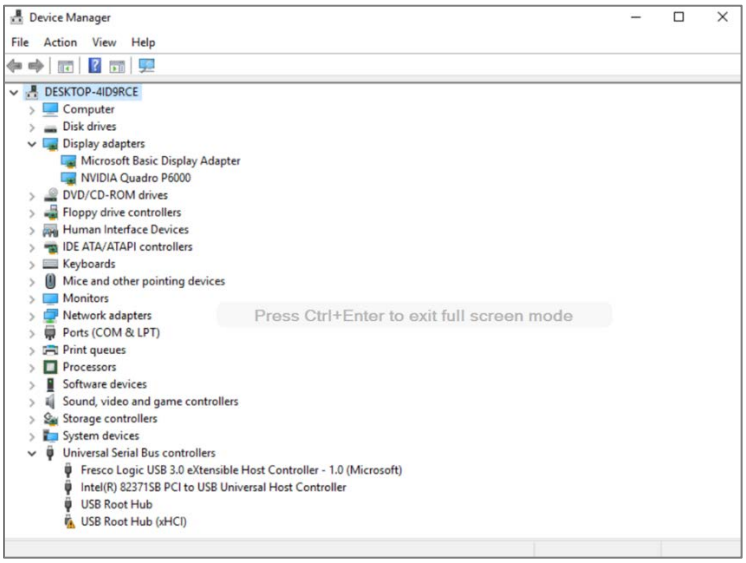



Figure 9 VMware Guest VM – PCIe Passthrough USB 3.0 Controller and NVIDIA Quadro P6000 Visible in Device Manager

To enable the USB 3.0 controller and NVIDIA graphics adapter on the VM to work, these unidentified devices in the Device Manager must be resolved.

1. Install the VMware Xen-Tools on your virtual and reboot the virtual machine.
2. From the NVIDIA Driver Downloads page at http://www.nvidia.com/drivers, download and cleanly install the latest NVIDIA graphics driver for the Quadro card installed in the server.
3. If necessary, download and install the device driver for your specific USB 3.0 controller.

   For example, for the sample system to work, it was necessary to download and install the latest Inateck driver.
4. Reboot your virtual machine.
5. Confirm that the USB 3.0 controller and NVIDIA graphics adapter on the VM are working.
   a) Plug in a USB flash drive and ensure that it loads the native drivers for the drive.
   b) Connect an external monitor to the external ports of the NVIDIA Quadro graphics cards and confirm that video is being displayed correctly.

# LOADING A DISPLAY EDID ON WINDOWS

Whether you are using a KVM switch or manually switching between GPUs, you should prevent the Windows OS from receiving a hot-plug event when a real monitor is moved between display outputs. To achieve this result, load a standard Extended Display Identification Data (EDID) on one designated display connector for each GPU in the system. This Quadro driver feature ensures that the Windows OS on each VM will always appear to be connected to a primary display through HDMI.

For more information, refer to Managing a Display EDID on Windows in the NVIDIA Support knowledgebase.

# SIMPLIFYING ON-SITE SETUP OF A MULTI-USER VR SYSTEM

To simplify on-site setup of a multi-user VR system, use a small display to confirm that the server boots correctly and to verify the functionality of each virtual machine. To ensure that you can perform both tasks with a single display, use a display that accepts both HDMI input (for monitoring the VMs) and VGA input (for monitoring the server console).

# VMWARE PURPLE SCREEN CRASH ERROR DURING VM REBOOT OR SHUTDOWN

One significant problem with the sample system relates to how VMware vSphere 6.x fails to handle PCIe device deallocation during the reboot or shutdown of a virtual machine. This failure typically causes crash of the host machine with a purple screen on the console as shown in Figure 10. Only a hard reset of the power can recover the system.

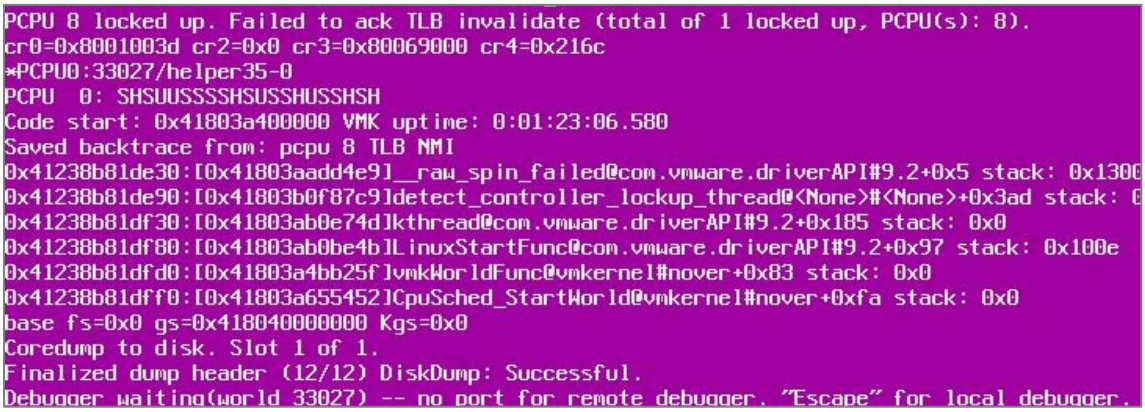



Figure 10 VMware Purple Screen Error

## Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

## HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

## ROVI Compliance Statement

NVIDIA Products that support Rovi Corporation's Revision 7.1.L1 Anti-Copy Process (ACP) encoding technology can only be sold or distributed to buyers with a valid and existing authorization from ROVI to purchase and incorporate the device into buyer's products.

This device is protected by U.S. patent numbers 6,516,132; 5,583,936; 6,836,549; 7,050,698; and 7,492,896 and other intellectual property rights. The use of ROVI Corporation's copy protection technology in the device must be authorized by ROVI Corporation and is intended for home and other limited pay-per-view uses only, unless otherwise authorized in writing by ROVI Corporation. Reverse engineering or disassembly is prohibited.

## OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

## Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright