TECHNICAL WHITE PAPER – **NOVEMBER 2017**

# DEPLOYING HARDWARE-ACCELERATED GRAPHICS WITH VMWARE HORIZON 7

Horizon 7 version 7.x

vmware®

## Table of Contents

## Introduction

Engineers, designers, and scientists have traditionally relied on dedicated graphics workstations to perform the most demanding tasks, such as manipulating 3D models and visually analyzing large data sets. These standalone workstations carry high acquisition and maintenance costs. In addition, in areas such as oil and gas, space exploration, aerospace, engineering, science, and manufacturing, individuals with these advanced requirements must be located in the same physical location as the workstation.

This paper describes hardware-accelerated graphics in VMware virtual desktops in VMware Horizon® 7. It begins with typical use cases and matches these use cases to the three types of graphics acceleration, explaining the differences. Later sections provide installation and configuration instructions, as well as best practices and troubleshooting.

### Purpose

Moving the graphics-acceleration hardware from the workstation to a server is a key architectural innovation. This shift changes the computing metaphor for graphics processing, putting the additional compute, memory, networking, and security advantages of the data center at the disposal of the user, so that complex models and very large data sets can be accessed and manipulated from virtually anywhere. With appropriate network bandwidth and suitable remote client devices, IT can now offer the most advanced users an immersive 3D-graphics experience while freeing them from the limitations of the old computing metaphor. Fewer physical resources are needed, the wait time to open complex models or run simulations is greatly reduced, and users are no longer tied to a single physical location. In addition to handling the most demanding graphical workloads, hardware acceleration can also reduce CPU usage for less demanding basic desktop or published application usage, and for video encoding or decoding, which includes the default Blast Extreme remote display protocol.

### What Is VMware Horizon 7?

VMware Horizon 7 provides a platform to deliver a virtual desktop solution as well as an enterprise-class application-publishing solution. Horizon 7 features and components, such as the Blast Extreme display protocol, instant-clone provisioning, VMware App Volumes™ application delivery, and VMware User Environment Manager™, are also integrated into RDSH to provide a seamless user experience and an easy-to-manage, scalable solution.

### Intended Audience

This white paper is for administrators deploying hardware-accelerated graphics in Horizon 7, or anyone interested in the technology.

## Types of Graphics Acceleration

There are three types of graphics acceleration for Horizon 7:

- Virtual Shared Graphics Acceleration
- Virtual Shared Pass-Through Graphics Acceleration
- Virtual Dedicated Graphics Acceleration

### Virtual Shared Graphics Acceleration

Virtual Shared Graphics Acceleration (vSGA) allows a GPU to be shared across multiple virtual desktops. It is an attractive solution for users who require the full potential of the GPU's capability during brief periods. However, vSGA can create bottlenecks, depending on which applications are used, and the resources these applications require from the GPU. Virtual Shared Graphics Acceleration is generally used for knowledge workers and, occasionally, for power users.

With vSGA, the physical GPUs in the host are virtualized and shared across multiple guest virtual machines. A vendor driver must be installed in the hypervisor. Each guest virtual machine uses a proprietary VMware vSGA 3D driver that communicates with the vendor driver in VMware vSphere®. Drawbacks of vSGA are that applications might need to be recertified to be supported, API support is limited, and support is restricted for the various versions of OpenGL and DirectX.

Some examples of supported vSGA cards for Horizon 7 version 7.x and vSphere 6.5 are

- Intel Iris Pro Graphics P580
- NVIDIA Tesla M10/M60/P40

For a full list of compatible Virtual Shared Graphics Acceleration cards, see the VMware Virtual Shared Graphics Acceleration Guide.

### Virtual Shared Pass-Through Graphics Acceleration

Virtual Shared Pass-Through Graphics Acceleration allows a graphical processing unit to be shared with multiple users instead of focused on only one user. The difference from vSGA is that the proprietary VMware 3D driver is not used, and most of the graphics card's features are supported.

You must install the appropriate vendor driver on the guest virtual machine, and all graphics commands are passed directly to the GPU without having to be translated by the hypervisor. On the hypervisor, a vSphere Installation Bundle (VIB) is installed, which aids or performs the scheduling. Depending on the card, up to 24 virtual machines can share a GPU, and some cards have multiple GPUs. Calculating the exact number of desktops or users per GPU depends on the type of card, application requirements, screen resolution, number of displays, and frame rate, measured in frames per second (FPS).

The amount of frame buffer (VRAM) per VM is fixed, and the GPU engines are shared between VMs. AMD has an option to also have a fixed amount of compute, which is called *predictable performance*.

Virtual shared pass-through technology provides better performance than vSGA and higher consolidation ratios than vDGA. It is a good technology to use for low-, mid-, or even advanced-level engineers and designers, as well as for power users with 3D application requirements. One drawback of shared pass-through is that it might require applications to be recertified for support. Another drawback is the lack of VMware vSphere vMotion® support.

Some examples of supported shared pass-through cards for Horizon 7.x and vSphere 6.5 are

- AMD FirePro S7100X/S7150/S7150X2 (multi-user graphics processing unit, or MxGPU)
- NVIDIA Tesla M10/M60/P40 (virtual graphics processing unit, or vGPU)

For a full list of compatible shared pass-through graphics cards, see the VMware Shared Pass-Through Graphics Guide.

### Virtual Dedicated Graphics Acceleration

Virtual Dedicated Graphics Acceleration (vDGA) technology provides each user with unrestricted, fully dedicated access to one of the GPUs within the host. Although consolidation and management trade-offs are associated with dedicated access, vDGA offers the highest level of performance for users with the most intensive graphics computing needs.

With vDGA, the hypervisor passes the GPUs directly to individual guest virtual machines. This technology is also known as *GPU pass-through.* No special drivers are required in the hypervisor. However, to enable graphics acceleration, the appropriate vendor driver must be installed on each guest virtual machine. The installation procedures are the same as for physical machines. One drawback of vDGA, however, is the lack of vMotion support.

Some examples of supported vDGA cards in Horizon 7 version 7.x and vSphere 6.5 are

- AMD FirePro S7100X/S7150/S7150X2
- Intel Iris Pro Graphics P580/P6300
- NVIDIA Quadro M5000/P6000, Tesla M10/M60/P40

For a list of partner servers that are compatible with specific vDGA devices, see the VMware Virtual Dedicated Graphics Acceleration (vDGA) Guide.

## Comparison of the Types of Graphics Acceleration

The following table compares the features of the three types of graphics acceleration.

| TYPE | VIRTUAL SHARED GRAPHICS ACCELERATION | VIRTUAL SHARED PASS-THROUGH GRAPHICS ACCELERATION | VIRTUAL DEDICATED GRAPHICS ACCELERATION |
|---|---|---|---|
| Abbreviation | vSGA | vGPU/MxGPU | vDGA |
| Consolidation | High (limited by video memory) | Up to 1:32 | None (1:1) |
| Performance level | Lightweight | Lightweight or Workstation | Workstation |
| Compatibility | Limited | Full, but not all applications are certified | Maximum |
| DirectX level | 9.0c SM3 only | All supported versions | All supported versions |
| OpenGL version | 2.1 only | All supported versions | All supported versions |
| Video encoding and decoding | Software | Hardware | Hardware |
| OpenCL and/or CUDA compute | No | MxGPU: OpenCL only<br>GRID 1: No<br>GRID 2: 1:1 only | Yes |
| vMotion support | Yes | No | No |

**Table 1:** Feature Comparison for the Types of Graphics Acceleration

### Hardware Requirements for Hardware-Accelerated Graphics

The hardware requirements for graphics acceleration solutions are listed in Table 2.

| COMPONENT | DESCRIPTION |
|---|---|
| Physical space for graphics cards | Many high-end GPU cards are full height, full length, and double width, most taking up two slots on the motherboard, but using only a single PCIe x16 slot. Verify that the host has enough room internally to hold the chosen GPU card in the appropriate PCIe slot. |
| Host power supply unit (PSU) | Check the power requirements of the GPU to make sure that the PSU is powerful enough and contains the proper power cables to power the GPU. For example, a single NVIDIA K2 GPU can use as much as 225 watts of power and requires either an 8-pin PCIe power cord or a 6-pin PCIe power cord. |
| BIOS | For MxGPU, Single-Root IO Virtualization (SR-IOV) must be enabled. For vGPU, Intel Virtualization Technology support for Direct I/O (Intel VT-d) or AMD input–output memory management unit (IOMMU) must be enabled. To locate these settings in the server BIOS, contact the hardware vendor. |
| Two display adapters | If the host does not have an extra graphics adapter, VMware recommends that you install an additional low-end display adapter to act as the primary display adapter because the VMware ESXi™ console display adapter is not available to Xorg. If the GPU is set as the primary adapter, Xorg cannot use the GPU for rendering. If two GPUs are installed, the server BIOS might have an option to select which GPU is primary and which is secondary. |

**Table 2:** Hardware Requirements for Hardware-Accelerated Graphics

## Use Cases for Hardware-Accelerated Graphics

Following are the typical use cases for the different types of hardware-accelerated graphics.

- Knowledge workers

  Office workers and executives fall into the knowledge-worker category, typically using applications such as Microsoft Office, Adobe Photoshop, and other non-specialized end-user applications.

  Because the graphical load of these users is expected to be low, consolidation becomes important, which is why these types or users are best matched with one of the following types of graphics acceleration:

  - Virtual Shared Pass-Through Graphics Acceleration (MxGPU or vGPU) – When performance and features (hardware video encoding and decoding, or DirectX/OpenGL levels) matter most.
  - Virtual Shared Graphics Acceleration (vSGA) – When consolidation matters most.

- Power users

  Power users consume more complex visual data, but their requirements for manipulations of large datasets and specialized software are less intense than for designers, or they use only viewers like Autodesk DWG TrueView.

  Power users are best matched with

  - Virtual Shared Pass-Through Graphics Acceleration (MxGPU or vGPU).

- Designers

  Designers and advanced engineering and scientific users often create and work with large, complex datasets, and require graphics-intensive applications such as 3D design, molecular modeling, and medical diagnostics software from companies such as Dassault Systèmes, Enovia, Siemens NX, and Autodesk.

  Designers are best matched with one of the following:

  - Virtual Shared Pass-Through Graphics Acceleration (MxGPU or vGPU) – When availability or consolidation matters most.
  - Virtual Dedicated Graphics Acceleration (vDGA) – When every bit of performance counts.

Figure 1 summarizes the performance and consolidation profiles of the three types of graphics acceleration.

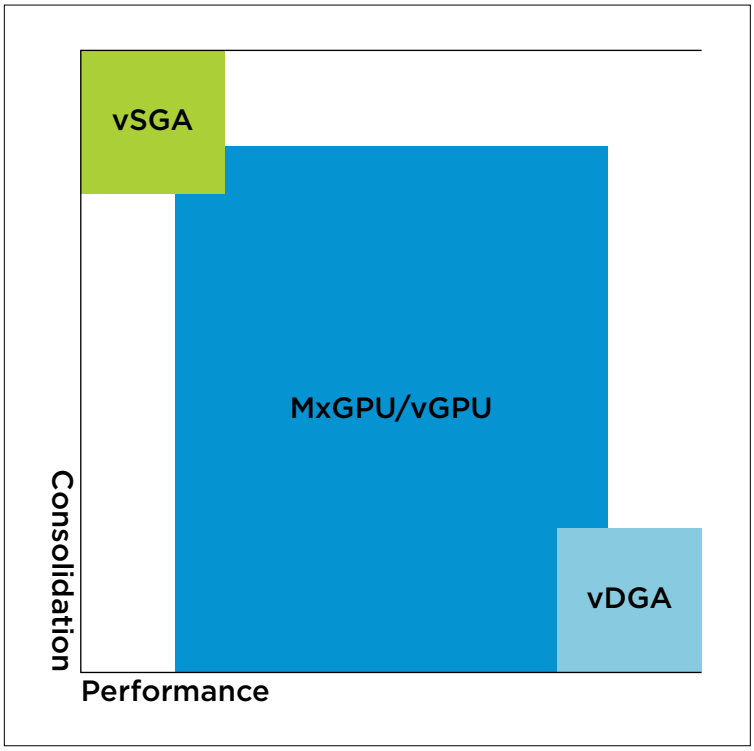



**Figure 1:** Consolidation and Performance Overview

## Installation, Configuration, and Setup

This section gives details on how to install and configure the following components for graphics acceleration:

- ESXi 6.x host
- Virtual machine
- Guest operating system
- Horizon 7 version 7.x desktop pool settings
- License server

### ESXi 6.x Host

The installation and configuration on the ESXi host varies by graphics-acceleration type.

**For vSGA or vGPU**

1. Install the graphics card on the ESXi host.
2. Put the host in maintenance mode.
3. If you are using an NVIDIA Tesla P card, make sure that ECC is disabled.
4. If you are using an NVIDIA Tesla M card, make sure that the card is set to `graphics` mode (`compute` is the default), with `GpuModeSwitch`, which comes as a bootable ISO, or a VIB:
   a. Install `GpuModeSwitch` without an NVIDIA driver installed:

      ```
      esxcli software vib install --no-sig-check -v
      /<path_to_vib>/NVIDIA-GpuModeSwitch-
      1OEM.xxx.0.0.xxxxxxx.x86_64.vib
      ```

   b. Reboot the host.
   c. Change all GPUs to graphics mode:

      ```
      gpumodeswitch --gpumode graphics
      ```

   d. Remove `GpuModeSwitch:`

      ```
      esxcli software vib remove -n NVIDIA-
      VMware_ESXi_xxx_GpuModeSwitch_Driver
      ```

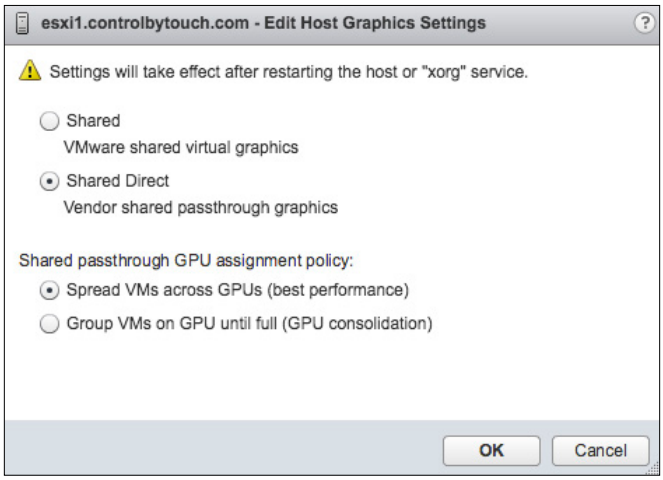
5. Install the GPU vSphere Installation Bundle (VIB):

   ```
   esxcli software vib install -v /<path_to_vib>\NVIDIA-
   VMware_ESXi_xxx_Host_Driver_xxx.xx-1OEM.xxx.0.0.xxxxxxx.vib
   ```

   If you are using ESXi 6.0, VIB files for vSGA and vGPU are separate; with ESXi 6.5, there is a single VIB for both vSGA and vGPU.
6. Reboot and take the host out of maintenance mode.

7. If you are using vSphere 6.5 or later, and an NVIDIA card:
   a. In the vSphere Web Client, navigate to **Host** > **Configure** > **Hardware** > **Graphics** > **Host Graphics** > **Edit**.

      The Edit Host Graphics Settings window appears.




   b. Select **Shared Direct** for vGPU, or **Shared** for vSGA.
   c. If you are using vGPU with different profiles per GPU, select **Group VMs on GPU until full (GPU consolidation)** because different profiles are only possible between GPUs, and you could run out of free GPUs.

      Example:

      The host has a single M60 card, which has two GPUs, each of which has 8 GB of memory. When trying to run two VMs with 4 GB of frame buffer and four VMs with 2 GB, and the first two machines started have the same profile, they would be started on different GPUs, and no GPU would be available for the other profile. With **Group VMs on GPU until full (GPU consolidation)**, the same profiles would start on the same GPU.

**For MxGPU**

1. Install the graphics card on the ESXi host.
2. Put the host in maintenance mode.
3. In the BIOS of the ESXi host, verify that the following is enabled:
   - Single-Root IO Virtualization (SR-IOV)

   and that one of the following is also enabled:
   - Intel Virtualization Technology support for Direct I/O (Intel VT-d)
   - AMD input–output memory management unit (IOMMU)
4. Browse to the location of the AMD FirePro VIB Driver and AMD VIB Install Utility:

   ```
   cd /<path_to_vib>
   ```
5. Make the VIB Install Utility executable, and execute it:

   ```
   chmod +x mxgpu-install.sh && sh mxgpu-install.sh –i
   ```
   - The script shows three options. Select the option that suits your environment.

     ```
     Enter the configuration
     mode([A]uto/[H]ybrid/[M]anual,default:A)A
     ```
   - The script asks for the number of Virtual Functions; indicate the number of users you want to run on a GPU:

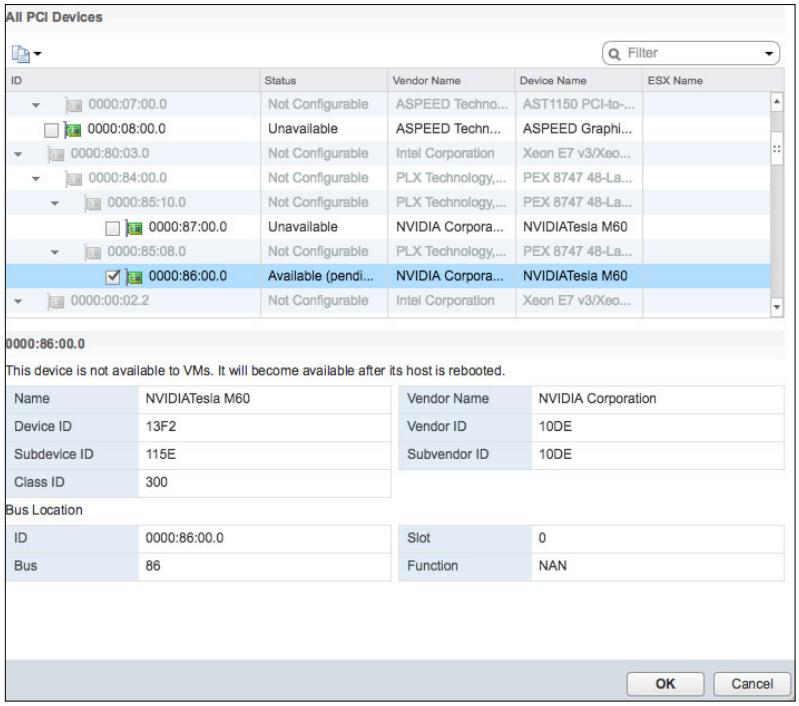
     ```
     Please enter number of VFs: (default:4): 8
     ```
   - The script asks if you want to keep performance fixed, independent of the number of active VMs. Indicate Yes or No according to your requirements.

     ```
     Do you want to enable Predictable Performance?
     ([Y]es/[N]o,default:N)N
     …
     Done
     The configuration needs a reboot to take effect
     ```
   - Reboot and take the host out of maintenance mode.

For vDGA

1. Install the graphics card on the ESXi host.
2. Verify that Intel VT-d or AMD IOMMU is enabled in the BIOS of the ESXi host.
3. To enable pass-through for the GPU in the vSphere Web Client, navigate to **Host** > **Configure** > **Hardware** > **PCI Devices** > **Edit**.

   The All PCI Devices window appears.
4. Select the check box for the GPU, and reboot.



## Virtual Machine

Set up the virtual machine with general settings, as follows, and then further configure it according to the type of graphics acceleration you are using.

### General Settings for Virtual Machines

**Hardware level** – The recommended hardware level is the highest that all hosts support, with Version 11 as minimum.

**CPU** – The amount of CPU required depends on the usage and should be determined by actual workload. As a starting point, you might use these numbers:

Knowledge workers: 2
Power users: 4
Designers: 6

**Memory** – The amount of memory required depends on the usage and should be determined by actual workload. As a starting point, you might use these numbers:

Knowledge workers: 2
Power users: 4
Designers: 8

**Virtual network adapter** – The recommended virtual network adapter is VMXNET3.

**Virtual storage controller** – The recommended virtual disk is LSI Logic SAS, but the highest workloads using local flash-based storage might benefit from using VMware Paravirtual.

**Other devices** – We recommend removing devices that are not used, such as COM/LTP/DVD/Floppy.

Now that you have configured the general settings for the virtual machines, configure the settings for the type of graphics acceleration.

### Virtual Machine Settings for vSGA

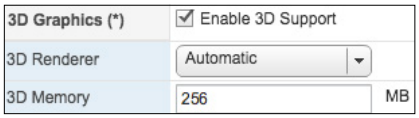
Configure the virtual machine as follows if you are using vSGA.

1. Enable 3D Graphics by selecting **Enable 3D Support**.
2. Set the `3D Renderer` to **Automatic** or **Hardware**.

   **Automatic** uses hardware acceleration if there is a capable, and available, hardware GPU in the host that the virtual machine is starting in. However, if a hardware GPU is not available, the virtual machine uses software 3D rendering for any 3D tasks. The Automatic option allows the virtual machine to be started on, or migrated (via vSphere vMotion) to any host (VMware vSphere version 5.0 or later), and to use the best solution available on that host.

   **Hardware** uses only hardware-accelerated GPUs. If a hardware GPU is not present in a host, the virtual machine will not start, or you will not be able to perform a live vSphere vMotion migration to that host. VMware vSphere vMotion is possible with the Hardware option as long as the host the virtual machine is being moved to has a capable and available hardware GPU. This setting can be used to guarantee that a virtual machine will always use hardware 3D rendering when a GPU is available, but that in turn limits the virtual machine to hosts that have hardware GPUs.
3. Select the amount of video memory (`3D Memory`).

   3D Memory has a default of 96 MB, a minimum of 64 MB, and a maximum of 512 MB.

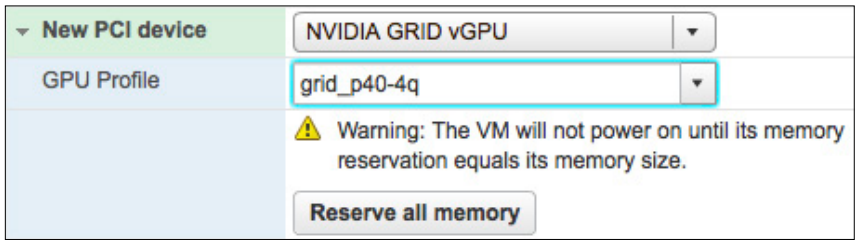


### Virtual Machine Settings for vGPU

Configure the virtual machine as follows if you are using vGPU.

1. On the vSphere console, select your virtual machine. Navigate to **Edit Settings**. Add a shared PCI device to the virtual machine and select the appropriate PCI device to enable GPU pass-through on the virtual machine.


After you add a shared PCI device, you see a list of all supported graphics profile types that are available from the GPU card on the ESXi host. Select the correct profile from the drop-down menu.




The last part of the GPU Profile string (4q, in this example) indicates the amount of frame buffer (VRAM) in gigabytes (0 meaning 512 MB, in this example 4 GB), and the type of GRID license required (q, in this example):

**b - GRID Virtual PC:** Virtual GPUs for business desktop computing

**a - GRID Virtual Application:** Virtual GPUs for Remote Desktop Session Hosts

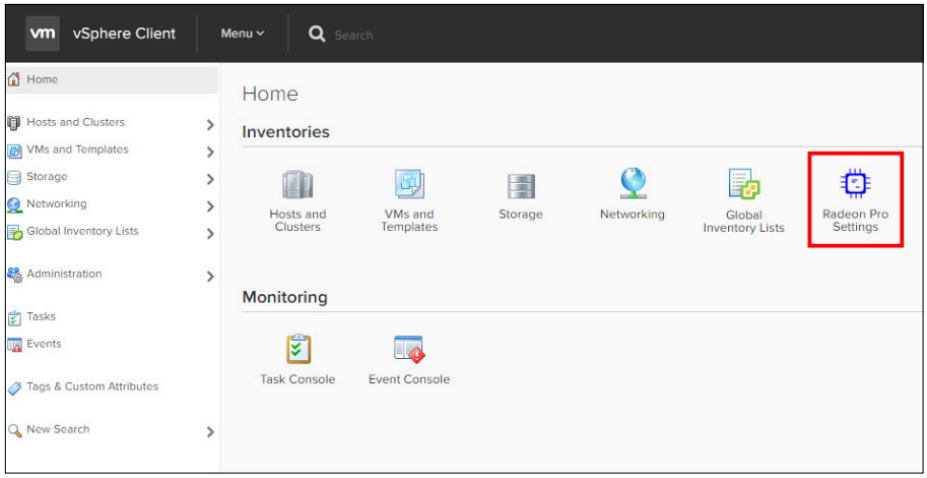
**q - Quadro Virtual Datacenter Workstation (vDWS):** Workstation-specific graphics features and accelerations, such as up to four 4K monitors and certified drivers for professional applications.

2. In the same New PCI Device window, reserve all memory when creating the virtual machine by clicking **Reserve all memory**.

### Virtual Machine Settings for MxGPU or vDGA

Configure the virtual machine as follows if you are using MxGPU or vDGA.

1. For devices with a large BAR size (for example, Tesla P40), vSphere 6.5 must be used, and the following Advanced Configuration Parameters must be set on the VM:
   - `firmware="efi"`
   - `pciPassthru.use64bitMMIO="TRUE"`
   - `pciPassthru.64bitMMIOSizeGB="64"`
2. Add a PCI device (virtual functions are also presented as PCI devices) to the virtual machine and select the appropriate PCI device to enable GPU pass-through on the virtual machine. With MxGPU, you can also do this by installing the Radeon Pro Settings for the VMware vSphere Client Plug-in:




Or you can do this for multiple machines at once from `ssh`, as follows.

a. Browse to the location of the AMD FirePro VIB driver and AMD VIB install utility:

```
cd /<path_to_vib
```

b. Edit `vms.cfg`:

```
vi vms.cfg
```

Press `I` and change `.*` to a regular expression (or multiple expressions on multiple lines) to match the names of your VMs that require a GPU. Following is an example to match `*MxGPU*` to VM names that include `MxGPU`, such as `WIN10-MxGPU-001` or `WIN8.1-MxGPU-002`:

```
.*MxGPU.*
```

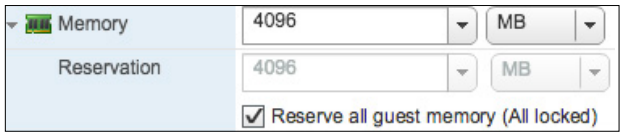
Press `Esc`, enter `:wq`, and press `Enter` to save and quit.

c. Assign the virtual functions to the VMs:

```
sh mxgpu-install.sh –a assign
Eligible VMs:
WIN10-MxGPU-001
WIN10-MxGPU-002
WIN8.1-MxGPU-001
WIN8.1-MxGPU-002
These VMs will be assigned a VF, is it OK?[Y/N]y
```

Press `Enter`.

3. Select **Reserve all guest memory (All locked)** when creating the virtual machine.




## Guest Operating System

For the guest operating system, perform the following installations and configurations.

### Installation and Configuration for Windows Guest Operating System

For a Windows guest operating system, install and configure as follows.

1. Install Windows 7, 10, or 2012 R2 and install all updates. The following installations are also recommended:
   a. Install common Microsoft runtimes and features:

      Before updating Windows in the VM, install all required versions of Microsoft runtimes that are patched by Windows Update and that can run side by side in the image. For example, install:

      • .NET Framework (3.5, 4.5, and so on)

      • Visual C++ Redistributables x86 / x64 (2005 SP1, 2008, 2012, and so on)

   b. Install Microsoft updates:

      Install all available updates to Microsoft Windows and other Microsoft products with Windows Update or Windows Server Update Service. You might need to first manually install Windows Update Client for Windows 8.1 and Windows Server 2012 R2: March 2016.

   c. Tune Windows with the VMware OS Optimization Tool.

      Run the VMware OS Optimization Tool with the default options.
2. If you are not using vSGA, obtain the GPU drivers from the GPU vendor (with vGPU, this is a matched pair with the VIB file) and install the GPU device drivers (with MxGPU, make sure the GPU Server option is selected) in the guest operating system of the virtual machine.
3. Install VMware Tools™ and Horizon Agent (select 3D RDSH feature for Windows 2012 R2 Remote Desktop Session Hosts) in the guest operating system, and reboot.

**Installation and Configuration for Red Hat Enterprise Linux Operating System (vGPU and vDGA)**

For a Red Hat Enterprise Linux guest operating system, install and configure as follows.

1. Install Red Hat Enterprise Linux 6.9 or 7.4 x64, install all updates, and reboot.
2. Install `gcc`, kernel `makefiles`, and headers:

   ```
   sudo yum install gcc-c++ kernel-devel-$(uname -r) kernel-headers-$(uname -r) -y
   ```
3. Disable `libvirt`:

   ```
   sudo systemctl disable libvirtd.service
   ```
4. Disable the open-source nouveau driver.
   a. Navigate to and open `vi` for the following configuration file:

      ```
      sudo vi /etc/default/grub
      ```

      or for RHEL 6.x:

      ```
      sudo vi /boot/grub/grub.conf
      ```

   b. Find the line for `GRUB_CMDLINE_LINUX`, and add `blacklist=nouveau` to the line.
   c. Navigate to and open `vi` for the following configuration file:

      ```
      sudo vi /etc/modprobe.d/blacklist.conf
      ```

   d. Add a `blacklist=nouveau` line anywhere in `blacklist.conf`.
5. Generate new `grub.cfg` and `initramfs` files:

   ```
   sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   sudo dracut /boot/initramfs-$(uname -r).img $(uname -r) -f
   ```
6. Reboot.
7. Install the NVIDIA driver:

   ```
   init 3
   chmod +x NVIDIA-Linux-x86_64-xxx.xx-grid.run
   ```

   `sudo ./NVIDIA-Linux-x86_64-xxx.xx-grid.run` (Acknowledge all questions.)
8. (Optional) Install the CUDA Toolkit (run file method recommended), but do not install the included driver.
9. Add license server information:

   ```
   sudo cp /etc/nvidia/gridd.conf.template
   /etc/nvidia/gridd.conf
   sudo vi /etc/nvidia/gridd.conf
   ```

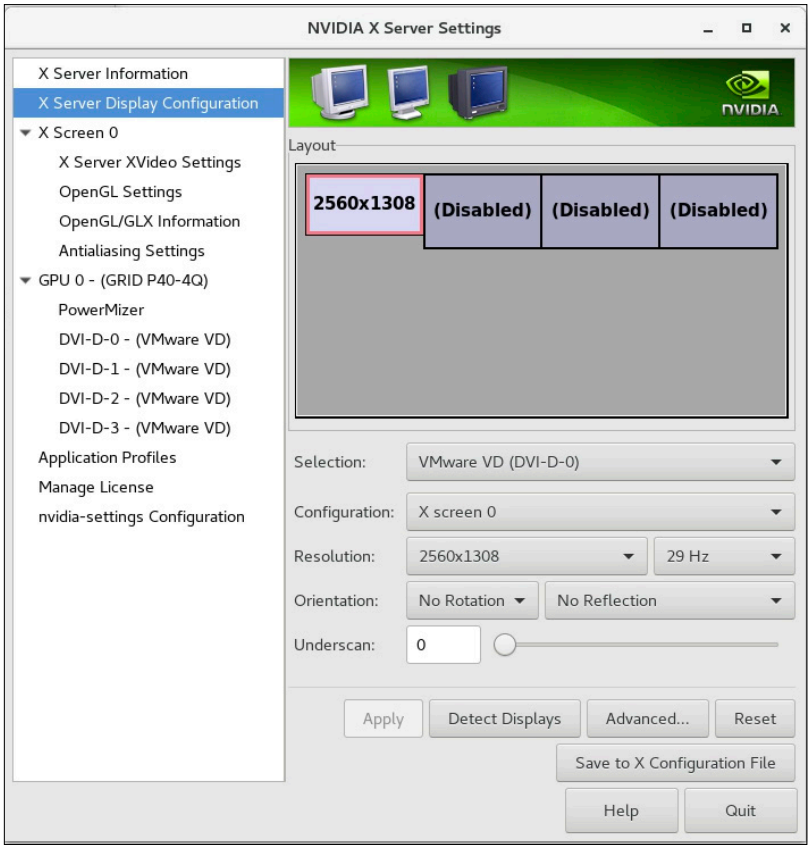
   Set `ServerAddress` and `BackupServerAddress` to the DNS names or IPs of your license servers and `FeatureType` to `1` for vGPU and `2` for vDGA.

10. Install the Horizon Agent:

    `tar -zxvf VMware-horizonagent-linux-x86_64-7.3.0-6604962.tar.gz`

    `cd VMware-horizonagent-linux-x86_64-7.3.0-6604962`

    `sudo ./install_viewagent.sh` (Acknowledge all questions.)

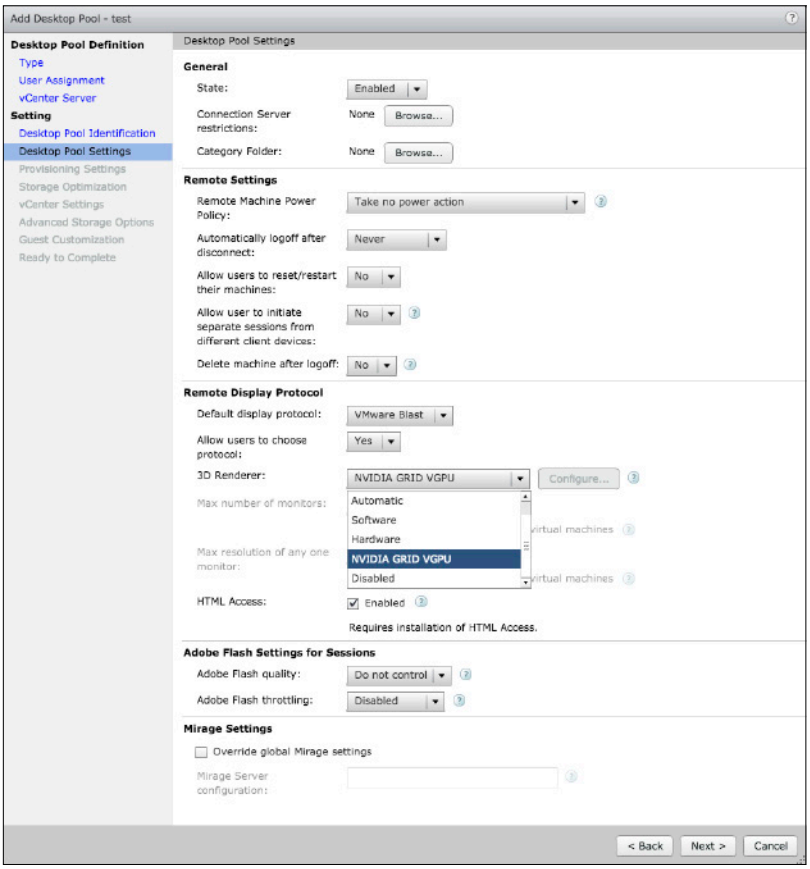


### Horizon 7 version 7.x Pool and Farm Settings

During the creation of a new Farm in Horizon 7, configuring a 3D Farm is the same as a normal Farm.

During the creation of a new View desktop pool in Horizon 7, configure the pool as normal until you reach the Desktop Pool Settings section.

1. Scroll down the Add Desktop Pool window until you reach the **Remote Display Protocol** section. In this section, you see the **3D Renderer** option.



2. Select either **Hardware** or **Automatic** as the 3D rendering option for vSGA, **Hardware** for vDGA or MxGPU, or **NVIDIA GRID VGPU** for vGPU.

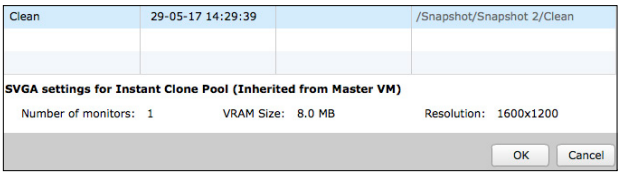
   **Automatic** uses hardware acceleration if there is a capable, and available, hardware GPU in the host that the virtual machine is starting in. However, if a hardware GPU is not available, the virtual machine uses software 3D rendering for any 3D tasks. The Automatic option allows the virtual machine to be started on, or migrated (via vSphere vMotion) to any host (VMware vSphere version 5.0 or later), and to use the best solution available on that host.

   **Hardware** uses only hardware-accelerated GPUs. If a hardware GPU is not present in a host, the virtual machine will not start, or you will not be able to perform a live vSphere vMotion migration to that host. VMware vSphere vMotion is possible with the Hardware option as long as the host the virtual machine is being moved to has a capable and available hardware GPU. This setting can be used to guarantee that a virtual machine will always use hardware 3D rendering when a GPU is available, but that in turn limits the virtual machine to hosts that have hardware GPUs.

   For Horizon 7 version 7.0 or 7.1, you need to configure the amount of VRAM you want each virtual desktop to have, and when using vGPU, select the profile you would like to use. With Horizon 7 version 7.1, vGPU can be used with instant clones. However the profile must match the profile set on the parent VM with the vSphere Web Client.

   **3D Memory** has a default of 96 MB, a minimum of 64 MB, and a maximum of 512 MB.

   With Horizon 7 version 7.2 and later, the video memory and vGPU profile are inherited from the VM or VM snapshot:




### License Server

For vGPU with GRID 2.0, you must install a license server. See the GRID Virtual GPU User Guide included with your NVIDIA driver download.

## Resource Monitoring

Following are some useful tools to monitor resources when employing graphics acceleration.

### gpuvm

To better manage the GPU resources available on an ESXi host, examine the current GPU resource allocation. The ESXi command-line query utility `gpuvm` lists the GPUs installed on an ESXi host and displays the amount of GPU memory that is allocated to each virtual machine on that host.

```
gpuvm

Xserver unix:0, GPU maximum memory 2076672KB

pid 118561, VM "Test-VM-001", reserved 131072KB of GPU memory pid
664081, VM "Test-VM-002", reserved 261120KB of GPU memory GPU memory left 1684480KB
```

### nvidia-smi

To get a summary of the vGPUs currently running on each physical GPU in the system, run `nvidia-smi` without additional arguments.

```
Thu Oct  5 09:28:05 2017

+-----------------------------------------------------------------------------+
| NVIDIA-SMI 384.73                 Driver Version: 384.73                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P40           On   | 00000000:84:00.0 Off |                  Off |
| N/A   38C    P0    60W / 250W |  12305MiB / 24575MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0    135930  M+C+G   manual                                     4084MiB |
|    0    223606  M+C+G   centos3D004                                4084MiB |
|    0    223804  M+C+G   centos3D003                                4084MiB |
+-----------------------------------------------------------------------------+
```

To monitor vGPU engine usage across multiple vGPUs, run `nvidia-smi vgpu` with the `-u` or `--utilization` option:

```
nvidia-smi vgpu -u
```

For each vGPU, the usage statistics in the following table are reported once every second.

| #GPU | VGPU | SM | MEM | ENC | DEC |
|---|---|---|---|---|---|
| #IDX | ID | % | % | % | % |
| 0 | 11924 | 6 | 3 | 0 | 0 |
| 1 | 11903 | 8 | 3 | 0 | 0 |
| 2 | 11908 | 10 | 4 | 0 | 0 |

**Key:**
gpu – GPU ID
vgpu – vGPU ID
sm – Compute
mem – Memory controller bandwidth
enc – Video encoder
dec – Video decoder

## Troubleshooting

Troubleshooting graphics acceleration can be for general problems or for a specific symptom.

### General Troubleshooting for Graphics Acceleration

If an issue arises with vSGA, vGPU, or vDGA, or if Xorg fails to start, try one or more of the following solutions, in any order that you choose.

#### Verify That the GPU Driver Loads

To verify that the GPU VIB is installed, run one of the following commands:

- For AMD-based GPUs:

```
#     esxcli software vib list | grep fglrx
```

- For NVIDIA-based GPUs:

```
#     esxcli software vib list | grep NVIDIA
```

If the VIB is installed correctly, the output resembles the following example:

```
NVIDIA-VMware  304.59-1-OEM.510.0.0.799733 NVIDIA
VMwareAccepted 2012-11-14
```

To verify that the GPU driver loads, run the following command:

- For AMD-based GPUs:

```
#     esxcli system module load –m fglrx
```

- For NVIDIA-based GPUs:

```
#     esxcli system module load –m nvidia
```

If the driver loads correctly, the output resembles the following example:

```
Unable to load module /usr/lib/vmware/vmkmod/nvidia: Busy
```

If the GPU driver does not load, check the `vmkernel.log`:

```
# vi /var/log/vmkernel.log
```

Search for `FGLRX` on AMD hardware or `NVRM` on NVIDIA hardware. Often, an issue with the GPU is identified in the `vmkernel.log`.

#### Verify That Display Devices Are Present in the Host

To make sure that the graphics adapter is installed correctly, run the following command on the ESXi host:

```
# esxcli hardware pci list –c 0x0300 –m 0xff
```

The output should resemble the following example, even if some of the particulars differ:

```
000:001:00.0
Address: 000:001:00.0
Segment: 0x0000
Bus: 0x01
Slot: 0x00
Function: 0x00
VMkernel Name:
Vendor Name: NVIDIA Corporation
Device Name: NVIDIA Quadro 6000
Configured Owner: Unknown
Current Owner: VMkernel
Vendor ID: 0x10de
Device ID: 0x0df8
SubVendor ID: 0x103c
SubDevice ID: 0x0835
Device Class: 0x0300
Device Class Name: VGA compatible controller
Programming Interface: 0x00
Revision ID: 0xa1
Interrupt Line: 0x0b
IRQ: 11
Interrupt Vector: 0x78
PCI Pin: 0x69
```

#### Check the PCI Bus Slot Order

If you installed a second lower-end GPU in the server, it is possible that the order of the cards in the PCIe slots will choose the higher-end card for the ESXi console session. If this occurs, swap the two GPUs between PCIe slots, or change the Primary GPU settings in the server BIOS.

#### Check Xorg Logs

If the correct devices are present in the previous troubleshooting methods, view the Xorg log file to see if there is an obvious issue.

```
#       vi /var/log/Xorg.log
```

### Troubleshooting Specific Issues in Graphics Acceleration

This section describes specific issues that may arise in graphics acceleration deployments, and presents probable solutions.

**Problem:**

`sched.mem.min` error when starting virtual machine.

**Solution:**

Check `sched.mem.min`.

If you get a vSphere error about `sched.mem.min`, add the following parameter to the `VMX` file of the virtual machine:

```
sched.mem.min = "4096"
```

**Note:** The number in quotes, `4096` in the previous example, must match the amount of configured virtual machine memory. The example is for a virtual machine with 4 GB of RAM.

**Problem:**

Only able to use one display in Windows 10 with vGPU -0B or -0Q profiles.

**Solution:**

Use a profile that supports more than one virtual display head and has at least 1 GB of frame buffer.

To reduce the possibility of memory exhaustion, vGPU profiles with 512 MB or less of frame buffer support only one virtual display head on a Windows 10 guest OS.

**Problem:**

Unable to use NVENC with vGPU -0B or -0Q profiles.

**Solution:**

If you require NVENC to be enabled, use a profile that has at least 1 GB of frame buffer.

Using the frame buffer for the NVIDIA hardware-based H.264 / HEVC video encoder (NVENC) may cause memory exhaustion with vGPU profiles that have 512 MB or less of frame buffer. To reduce the possibility of memory exhaustion, NVENC is disabled on profiles that have 512 MB or less of frame buffer.

**Problem:**

Unable to load vGPU driver in guest operating system.

Depending on the versions of drivers in use, the VMware vSphere VM's log file reports one of the following errors:

- A version mismatch between guest and host drivers:

```
vthread-10| E105: vmiop_log: Guest VGX version(2.0) and Host VGX
version(2.1) do not match
```

- A signature mismatch:

```
vthread-10| E105: vmiop_log: VGPU message signature mismatch.
```

**Solution:**

Install the latest NVIDIA vGPU release drivers matching the installed VIB on ESXi in the VM.

**Problem:**

Tesla-based Virtual GPU fails to start.

**Solution:**

Ensure that error-correcting code (ECC) is disabled on all GPUs.

Tesla GPUs support ECC, but NVIDIA GRID vGPU does not support ECC memory. If ECC memory is enabled, the NVIDIA GRID vGPU fails to start. The following error is logged in the VMware vSphere VM's log file:

```
vthread10|E105: Initialization: VGX not supported with ECC Enabled.
```

1. Use `nvidia-smi` to list the status of all GPUs, and check for ECC noted as enabled on GPUs.
2. Change the ECC status to **Off** on each GPU for which ECC is enabled by executing the following command:

   `nvidia-smi -i id -e 0` (`id` is the index of the GPU as reported by `nvidia-smi`)
3. Reboot the host.

**Problem:**

Single vGPU benchmark scores are lower than pass-through GPU.

**Solution:**

Disable the Frame Rate Limiter by adding the configuration parameter `pciPassthru0.cfg.frame_rate_limiter` with a value of `0` in the VM's advanced configuration options.

A vGPU incorporates a performance-balancing feature known as Frame Rate Limiter (FRL), which is enabled on all vGPUs. FRL is used to ensure balanced performance across multiple vGPUs that are resident on the same physical GPU. The FRL setting is designed to give a good interactive remote graphics experience but may reduce scores in benchmarks that depend on measuring frame-rendering rates, as compared to the same benchmarks running on a pass-through GPU.

**Problem:**

VMs configured with large memory fail to initialize vGPU when booted.

When starting multiple VMs configured with large amounts of RAM (typically more than 32 GB per VM), a VM may fail to initialize vGPU. The NVIDIA GRID GPU is present in Windows Device Manager but displays a warning sign, and the following device status:

```
Windows has stopped this device because it has reported problems. (Code 43)
```

The VMware vSphere VM's log file contains these error messages:

```
vthread10|E105: NVOS status 0x29
vthread10|E105: Assertion Failed at 0x7620fd4b:179
vthread10|E105: 8 frames returned by backtrace
...
vthread10|E105: VGPU message 12 failed, result code: 0x29
...
vthread10|E105: NVOS status 0x8
vthread10|E105: Assertion Failed at 0x7620c8df:280
vthread10|E105: 8 frames returned by backtrace
...
vthread10|E105: VGPU message 26 failed, result code: 0x8
```

**Solution:**

A vGPU reserves a portion of the VM's frame buffer for use in GPU mapping of VM system memory. The default reservation is sufficient to support up to 32 GB of system memory. You can accommodate up to 64 GB by adding the configuration parameter

```
pciPassthru0.cfg.enable_large_sys_mem
```

with a value of `1` in the VM's advanced configuration options.

## Summary

With VMware Horizon 7, there are three available technologies for hardware-accelerated graphics, each with their own advantages.

- Virtual Shared Pass-Through Graphics Acceleration (MxGPU or vGPU) – Best match for nearly all use cases.
- Virtual Shared Graphics Acceleration (vSGA) – For light graphical workloads that use only DirectX9 or OpenGL 2.1 and require the maximum level of consolidation.
- Virtual Dedicated Graphics Acceleration (vDGA) – For heavy graphical workloads that require the maximum level of performance.

With the information in this paper, you can install, configure, and manage your 3D workloads for Horizon 7 version 7.x on vSphere 6.x.

## Additional Resources

*Setting Up Graphics for Linux Desktops* in Setting Up Horizon 7 for Linux Desktops

*Configuring Desktop Pools > Configuring 3D Rendering for Desktops* in Setting Up Virtual Desktops in Horizon 7

## About the Authors

This paper was updated by Hilko Lantinga, who is an End-User-Computing Architect in VMware Technical Marketing with a focus on 3D, Horizon 7 Windows desktops, and RDSH, Linux, and applications. Previously, he was a senior consultant in VMware Professional Services, leading large-scale EUC deployments in EMEA. Hilko has 18 years of experience in end-user computing.

The original version of this paper was written by Stéphane Asselin and Gary Sloane.

To comment on this paper, contact VMware End-User-Computing Technical Marketing at euc_tech_content_feedback@vmware.com

**VMware, Inc.** 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
 Item No: 5199-VMW-WP-HORIZON7-DEPLOYING-HARDWARE-ACCELERATED-GRAPHICS-7_3_1-USLET-20171103
11/17