



GRID VGPU FOR VMWARE VSPHERE

DU-07354-001 | March 2015

Quick Start Guide



DOCUMENT CHANGE HISTORY

DU-07354-001

Version	Date	Authors	Description of Change
0.1	7/1/2014	AC	Initial draft for vGPU early access
0.3	7/18/2014	Avi	Initial release for vGPU early access
0.4	8/12/2014	Avi	Update release for vGPU early access
0.5	10/1/2014	AC	Update release for vGPU early access
0.6	1/14/2015	AC	Updated for production release
1.0	3/10/2015	AC	Production release

GETTING STARTED

This document provides a brief, step-by-step guide to booting a VM on VMware vSphere Hypervisor (ESXi) with NVIDIA Virtual GPU, and assumes familiarity with VMware vSphere. For comprehensive documentation on setting up vGPU with vSphere and VMware Horizon, refer to the GRID Virtual GPU Deployment Guide available at <http://www.nvidia.com/grid-vgpu-deployment-guide>.

1.1 PREREQUISITES

Before proceeding, ensure that you have these prerequisites:

- ▶ NVIDIA GRID K1 or K2 cards.
- ▶ A server platform capable of hosting VMware vSphere Hypervisor (ESXi) and the NVIDIA GRID cards. Refer to the release notes for a list of recommended servers.
- ▶ The NVIDIA GRID vGPU software package for VMware vSphere, consisting of the GRID Virtual GPU Manager for ESXi, and NVIDIA GRID vGPU drivers for Windows, 32- and 64-bit.
- ▶ VMware vSphere 2015 or later, obtainable from VMware.
- ▶ An installed Windows VM to be enabled with vGPU.

To run VMware Horizon with virtual machines running NVIDIA Virtual GPU, you will also need:

- ▶ VMware Horizon 6.1 or later, obtainable from VMware.



Note: Earlier versions of VMware vSphere and Horizon are not supported for use with NVIDIA Virtual GPU.

Review the release notes and known issues for GRID Virtual GPU before proceeding with installation.

1.2 INSTALLING VMWARE VSPHERE

Install VMware vSphere Hypervisor (ESXi), following VMware's installation instructions. Install VMware vCenter Server, following VMware's installation instructions.

1.3 INSTALLING THE NVIDIA VIRTUAL GPU MANAGER FOR VSPHERE

The NVIDIA Virtual GPU Manager runs on ESXi host. It is provided as a VIB file, which must be copied to the ESXi host and then installed.

1.3.1 Package installation

To install the vGPU Manager VIB you need to access the ESXi host via the ESXi Shell or SSH. Refer to VMware's documentation on how to enable ESXi Shell or SSH for an ESXi host.



Note: Before proceeding with the vGPU Manager installation make sure that all VM's are powered off and the ESXi host is placed in maintenance mode. Refer to VMware's documentation on how to place an ESXi host in maintenance mode.

Use the `esxcli` command to install the vGPU Manager package:

```
[root@esxi:~] esxcli software vib install -v /NVIDIA-vgx-
VMware_ESXi_6.0_Host_Driver_346.42-1OEM.600.0.0.2159203.vib
Installation Result
  Message: Operation finished successfully.
  Reboot Required: false
  VIBs Installed: NVIDIA_bootbank_NVIDIA-vgx-
VMware_ESXi_6.0_Host_Driver_346.42-1OEM.600.0.0.2159203
  VIBs Removed:
  VIBs Skipped:
```

Reboot the ESXi host and remove it from maintenance mode.

1.3.2 Verifying installation

After the ESXi host has rebooted, verify that the GRID package installed and loaded correctly by checking for the NVIDIA kernel driver in the list of kernel loaded modules.

```
[root@esxi:~] vmkload_mod -l | grep nvidia
nvidia                5      8420
```

If the `nvidia` driver is not listed in the output, check `dmesg` for any load-time errors reported by the driver.

Verify that the NVIDIA kernel driver can successfully communicate with the GRID physical GPUs in your system by running the `nvidia-smi` command, which should produce a listing of the GPUs in your platform:

```
[root@esxi:~] nvidia-smi
Tue Mar 10 17:56:22 2015
+-----+
| NVIDIA-SMI 346.42      Driver Version: 346.42      |
+-----+-----+-----+-----+-----+-----+
| GPU  Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+-----+
|   0   GRID K2                On          | 0000:04:00.0   Off  |           Off        |
| N/A   27C   P8     27W / 117W |  11MiB / 4095MiB |      0%    Default   |
+-----+-----+-----+-----+-----+-----+
|   1   GRID K2                On          | 0000:05:00.0   Off  |           Off        |
| N/A   27C   P8     27W / 117W |  10MiB / 4095MiB |      0%    Default   |
+-----+-----+-----+-----+-----+-----+
|   2   GRID K2                On          | 0000:08:00.0   Off  |           Off        |
| N/A   32C   P8     27W / 117W |  10MiB / 4095MiB |      0%    Default   |
+-----+-----+-----+-----+-----+-----+
|   3   GRID K2                On          | 0000:09:00.0   Off  |           Off        |
| N/A   32C   P8     27W / 117W |  10MiB / 4095MiB |      0%    Default   |
+-----+-----+-----+-----+-----+-----+
|   4   GRID K2                On          | 0000:86:00.0   Off  |           Off        |
| N/A   24C   P8     27W / 117W |  10MiB / 4095MiB |      0%    Default   |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage    |
+-----+-----+-----+-----+-----+-----+
| No running processes found                                     |
+-----+-----+-----+-----+-----+-----+
```

If `nvidia-smi` fails to report the expected output, check `dmesg` for NVIDIA kernel driver messages.



Note: Information and debug messages from the NVIDIA kernel driver are logged in `dmesg`, prefixed with “NVRM” or ‘nvidia’. You can view NVIDIA kernel driver messages using:

```
[root@esxi:~] dmesg | grep -E "NVRM|nvidia"
```

1.4 CONFIGURING A VM WITH VIRTUAL GPU



Note: VMware vSphere does not support VM console in vSphere Web Client for VMs configured with vGPU. Make sure that you have installed an alternate means of accessing the VM (such as VMware Horizon or a VNC server) before you configure vGPU.

VM console in vSphere Web Client should become active again once the vGPU parameters are removed from the VM’s configuration.

To configure vGPU for a VM:

- ▶ Select `Edit Settings` after right-clicking on the VM in the vCenter Web UI
- ▶ Select the `Virtual Hardware` tab
- ▶ In the `New device selection`, select `Shared PCI Device` and hit `Add`

This should auto-populate `NVIDIA GRID vGPU` in the PCI device field, as shown in Figure 1. In the GPU Profile dropdown menu, select the type of vGPU you wish to configure. The supported vGPU types are listed in Table 1.

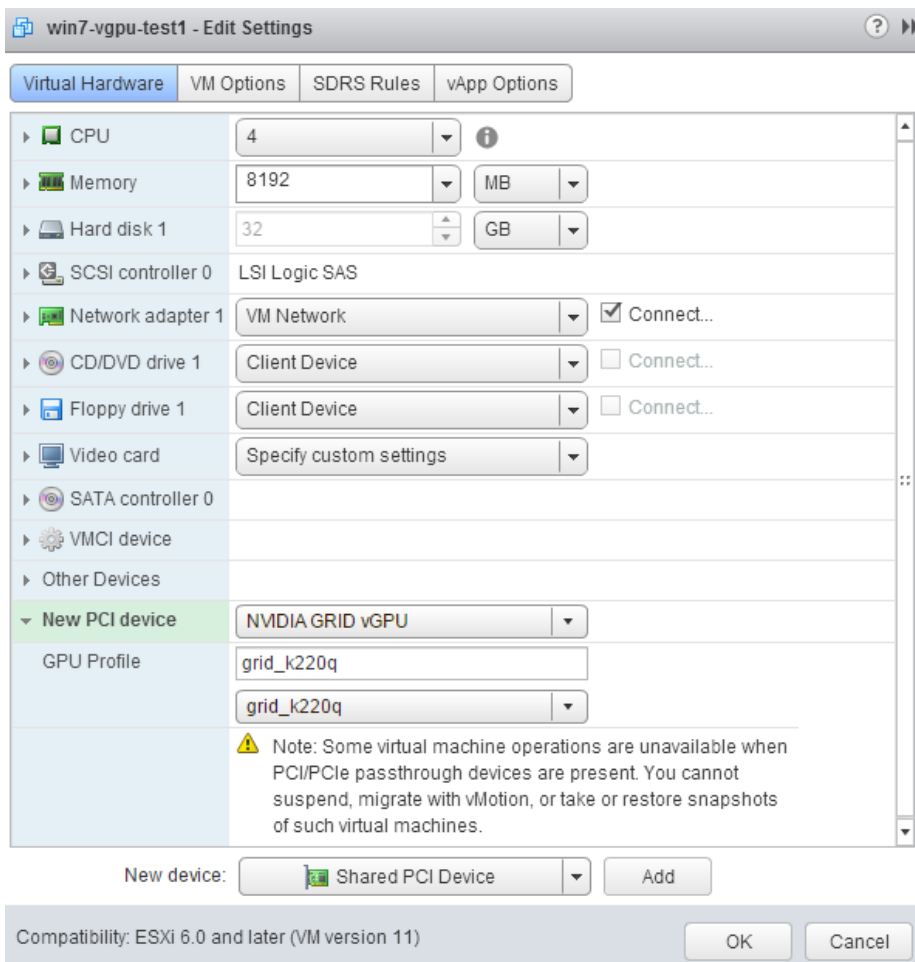


Figure 1 VM settings for vGPU

VM's running vGPU should have all their memory reserved, to do the same,

- ▶ Select Edit virtual machine settings from vCenter Web UI
- ▶ Expand Memory section and click Reserve all guest memory (All locked)

Card	Virtual GPU	"GPU Profile" name	Frame Buffer (Mbytes)	Virtual Display Heads	Max Resolution per Display Head
GRID K1	GRID K180Q	grid_k180q	4096	4	2560x1600
	GRID K160Q	grid_k160q	2048	4	2560x1600
	GRID K140Q	grid_k140q	1024	2	2560x1600
	GRID K120Q	grid_k120q	512	2	2560x1600
	GRID K100	grid_k100	256	2	1920x1200

GRID K2	GRID K280Q	grid_k280q	4096	4	2560x1600
	GRID K260Q	grid_k260q	2048	4	2560x1600
	GRID K240Q	grid_k240q	1024	2	2560x1600
	GRID K220Q	grid_k220q	512	2	2560x1600
	GRID K200	grid_k200	256	2	1920x1200

Table 1 Virtual GPU types

1.5 BOOTING THE VM AND INSTALLING DRIVERS

Once you have configured a VM with a vGPU, start the VM. VM console in vSphere Web Client is not supported in this vGPU release, use VMware Horizon or VNC to access the VM's desktop.

The VM should boot to a standard Windows desktop in VGA mode at 800x600 resolution. The Windows screen resolution control panel may be used to increase the resolution to other standard resolutions, but to fully enable vGPU operation, as for a physical NVIDIA GPU, the NVIDIA driver must be installed.

- ▶ Copy the 32- or 64-bit NVIDIA Windows driver package to the guest VM and execute it to unpack and run the driver installer.
- ▶ Click through the license agreement
- ▶ Select `Express Installation`
- ▶ Once driver installation completes, the installer may prompt you to restart the platform. Select `Restart Now` to reboot the VM, or exit the installer and reboot the VM when ready.

Once the VM restarts, it will boot to a Windows desktop. Verify that the NVIDIA driver is running by right-clicking on the desktop. The NVIDIA Control Panel will be listed in the menu; select it to open the control panel. Selecting "System Information" in the NVIDIA control panel will report the Virtual GPU that the VM is using, its capabilities, and the NVIDIA driver version that is loaded.

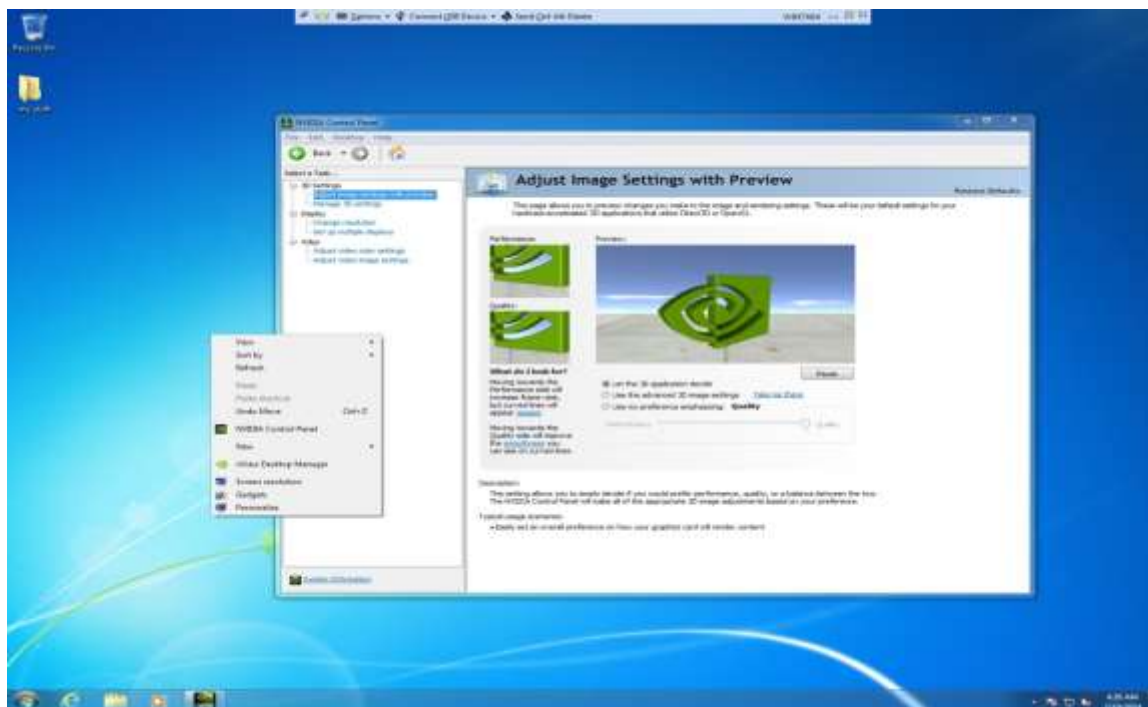


Figure 2 Verifying NVIDIA driver operation using NVIDIA Control Panel

This completes the process of setting up a single VM to use GRID vGPU. The VM is now capable of running the full range of DirectX and OpenGL graphics applications.

1.6 REMOVING A VM'S VGPU CONFIGURATION

To remove a vGPU configuration from a VM:

- ▶ Select `Edit settings` after right-clicking on the VM in the vCenter Web UI
- ▶ Select the `Virtual Hardware` tab

- ▶ Mouse over the `PCI Device` entry showing `NVIDIA GRID vGPU` and click on the (X) icon to mark the device for removal.
- ▶ Click `OK` to remove the device and update the VM settings.

1.7 MODIFYING GPU ASSIGNMENT FOR VGPU-ENABLED VMS

VMware vSphere Hypervisor (ESXi) by default uses a *breadth-first* allocation scheme for vGPU-enabled VMs; allocating new vGPU-enabled VMs on an available, least loaded physical GPU. This policy generally leads to higher performance because it attempts to minimize sharing of physical GPUs, but in doing so it may artificially limit the total number of vGPUs that can run.

ESXi also provides a *depth-first* allocation scheme for vGPU-enabled VMs. The depth-first allocation policy attempts to maximize the number of vGPUs running on each physical GPU, by placing newly-created vGPUs on the physical GPU that can support the new vGPU and that has the most number of vGPUs already resident. This policy generally leads to higher density of vGPUs, particularly when different types of vGPUs are being run, but may result in lower performance because it attempts to maximize sharing of physical GPUs.

To switch to depth-first allocation scheme add the following parameter to `/etc/vmware/config`:

```
vGPU.consolidation = true
```

1.8 FILING A BUG REPORT

When filing a bug report with NVIDIA, capture relevant configuration data from the platform exhibiting the bug, using `nvidia-bug-report.sh`.

Run `nvidia-bug-report.sh` from the ESXi Shell or SSH to capture debug information from your ESXi installation into a gzipped log file on the server:

```
[root@esxi:~] nvidia-bug-report.sh

nvidia-bug-report.sh will now collect information about your
system and create the file 'nvidia-bug-report.log.gz' in the current
directory. It may take several seconds to run. In some
cases, it may hang trying to capture data generated dynamically
by the ESXi kernel and/or the NVIDIA kernel module. While
the bug report log file will be incomplete if this happens, it
may still contain enough data to diagnose your problem.
```

Please include the 'nvidia-bug-report.log.gz' log file when reporting problems to NVIDIA.

Running nvidia-bug-report.sh... complete.

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.

OpenCL

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

Trademarks

NVIDIA and the NVIDIA logo are trademarks 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

© 2014 NVIDIA Corporation. All rights reserved.