Virtual GPU Software R450 for Linux with KVM

Release Notes
# Table of Contents

Chapter 1. Release Notes

1.1. NVIDIA vGPU Software Driver Versions......................................................... 1
1.2. Compatibility Requirements for the NVIDIA vGPU Manager and Guest VM Driver.................. 1
1.3. Updates in Release 11.0........................................................................... 3
1.4. Updates in Release 11.1........................................................................... 3
1.5. Updates in Release 11.2........................................................................... 3

Chapter 2. Validated Platforms

2.1. Supported NVIDIA GPUs and Validated Server Platforms......................... 4
2.2. Hypervisor Software Releases....................................................................... 5
2.3. Guest OS Support..................................................................................... 5
2.4. NVIDIA CUDA Toolkit Version Support...................................................... 5
2.5. Multiple vGPU Support............................................................................. 6
2.6. Peer-to-Peer CUDA Transfers over NVLink Support...................................... 8
2.7. Since 11.1: GPUDirect Technology Support................................................ 9

Chapter 3. Known Product Limitations

3.1. Issues occur when the channels allocated to a vGPU are exhausted............... 11
3.2. Virtual GPU hot plugging is not supported................................................ 12
3.3. Total frame buffer for vGPUs is less than the total frame buffer on the physical GPU... 12
3.4. Issues may occur with graphics-intensive OpenCL applications on vGPU types with limited frame buffer.................................................................................. 13
3.5. In pass through mode, all GPUs connected to each other through NVLink must be assigned to the same VM................................................................. 14
3.6. vGPU profiles with 512 Mbytes or less of frame buffer support only 1 virtual display head on Windows 10.............................................................................. 14
3.7. NVENC requires at least 1 Gbyte of frame buffer........................................ 15
3.8. VM running an incompatible NVIDIA vGPU guest driver fails to initialize vGPU when booted............................................................................................................... 15
3.9. Single vGPU benchmark scores are lower than pass-through GPU.................. 16
3.10. nvidia-smi fails to operate when all GPUs are assigned to GPU pass-through mode... 17

Chapter 4. Resolved Issues................................................................................. 18

Chapter 5. Known Issues.................................................................................. 20

5.1. Since 11.1: Guest VM frame buffer listed by nvidia-smi for vGPUs on GPUs that support SRIOV is incorrect................................................................. 20
5.2. 11.0 Only: Application responsiveness degrades over time........................... 21
5.3. 11.0 Only: On systems with more than 1 TiB of system memory, application performance degrades over time.

5.4. 11.1 Only: Licensing event logs indicate license renewal from unavailable primary server.

5.5. 11.0 Only: The NVIDIA license not present notification appears even for VMs for which the vGPU is licensed.

5.6. Driver upgrade in a Linux guest VM with multiple vGPUs might fail.

5.7. 11.0 Only: Failure to allocate resources causes VM failures or crashes.

5.8. NVIDIA Control Panel fails to start if launched too soon from a VM without licensing information.

5.9. Remoting solution session freezes with VGPU message 21 failed and VGPU message 14 failed errors.

5.10. On Linux, the frame rate might drop to 1 after several minutes.

5.11. DWM crashes randomly occur in Windows VMs.

5.12. Publisher not verified warning during Windows 7 driver installation.

5.13. RAPIDS cuDF merge fails on NVIDIA vGPU.

5.14. ECC memory settings for a vGPU cannot be changed by using NVIDIA X Server Settings.

5.15. Changes to ECC memory settings for a Linux vGPU VM by nvidia-smi might be ignored.

5.16. Vulkan applications crash in Windows 7 guest VMs configured with NVIDIA vGPU.

5.17. Host core CPU utilization is higher than expected for moderate workloads.

5.18. Frame capture while the interactive logon message is displayed returns blank screen.

5.19. RDS sessions do not use the GPU with some Microsoft Windows Server releases.

5.20. Even when the scheduling policy is equal share, unequal GPU utilization is reported.

5.21. When the scheduling policy is fixed share, GPU utilization is reported as higher than expected.

5.22. License is not acquired in Windows VMs.

5.23. nvidia-smi reports that vGPU migration is supported on all hypervisors.

5.24. Hot plugging and unplugging vCPUs causes a blue-screen crash in Windows VMs.

5.25. Luxmark causes a segmentation fault on an unlicensed Linux client.

5.26. A segmentation fault in DBus code causes nvidia-gridd to exit on Red Hat Enterprise Linux and CentOS.

5.27. No Manage License option available in NVIDIA X Server Settings by default.

5.28. Licenses remain checked out when VMs are forcibly powered off.

5.29. VM bug checks after the guest VM driver for Windows 10 RS2 is installed.

5.30. GNOME Display Manager (GDM) fails to start on Red Hat Enterprise Linux 7.2 and CentOS 7.0.
Chapter 1. Release Notes

These Release Notes summarize current status, information on validated platforms, and known issues with NVIDIA vGPU software and associated hardware on Linux with KVM.

Note: The most current version of the documentation for this release of NVIDIA vGPU software can be found online at NVIDIA Virtual GPU Software Documentation.

1.1. NVIDIA vGPU Software Driver Versions

Each release in this release family of NVIDIA vGPU software includes a specific version of the NVIDIA Virtual GPU Manager, NVIDIA Windows driver, and NVIDIA Linux driver.

<table>
<thead>
<tr>
<th>Software</th>
<th>11.0</th>
<th>11.1</th>
<th>11.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA Virtual GPU Manager for the Linux with KVM releases listed in Hypervisor Software Releases</td>
<td>450.55</td>
<td>450.80</td>
<td>450.89</td>
</tr>
<tr>
<td>NVIDIA Windows driver</td>
<td>451.48</td>
<td>452.39</td>
<td>452.57</td>
</tr>
<tr>
<td>NVIDIA Linux driver</td>
<td>450.51.05</td>
<td>450.80.02</td>
<td>450.89</td>
</tr>
</tbody>
</table>

1.2. Compatibility Requirements for the NVIDIA vGPU Manager and Guest VM Driver

The releases of the NVIDIA vGPU Manager and guest VM drivers that you install must be compatible. If you install the wrong guest VM driver release for the release of the vGPU Manager that you are using, the NVIDIA vGPU fails to load. See VM running an incompatible NVIDIA vGPU guest driver fails to initialize vGPU when booted.

Note: This requirement does not apply to the NVIDIA vGPU software license server. All releases in this release family of NVIDIA vGPU software are compatible with all releases of the license server.
Compatible NVIDIA vGPU Manager and Guest VM Driver Releases

The following combinations of NVIDIA vGPU Manager and guest VM driver releases are compatible with each other.

- NVIDIA vGPU Manager with guest VM drivers from the same release
- NVIDIA vGPU Manager with guest VM drivers from different releases within the same major release branch

In this situation, the combination supports only the features, hardware, and software (including guest OSes) that are supported on both releases.

- NVIDIA vGPU Manager from a later major release branch with guest VM drivers from the previous branch

In this situation, the combination supports only the features, hardware, and software (including guest OSes) that are supported on both releases.

The following table lists the specific software releases that are compatible with the components in the NVIDIA vGPU software 11 major release branch.

<table>
<thead>
<tr>
<th>NVIDIA vGPU Software Component</th>
<th>Releases</th>
<th>Compatible Software Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA vGPU Manager</td>
<td>11.0 through 11.2</td>
<td>◦ Guest VM driver releases 11.0 through 11.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>◦ All guest VM driver 10.x releases</td>
</tr>
<tr>
<td>Guest VM drivers</td>
<td>11.0 through 11.2</td>
<td>NVIDIA vGPU Manager releases 11.0 through 11.2</td>
</tr>
</tbody>
</table>

Incompatible NVIDIA vGPU Manager and Guest VM Driver Releases

The following combinations of NVIDIA vGPU Manager and guest VM driver releases are incompatible with each other.

- NVIDIA vGPU Manager from a later major release branch with guest VM drivers from two or more major releases before the release of the vGPU Manager
- NVIDIA vGPU Manager from an earlier major release branch with guest VM drivers from a later branch

The following table lists the specific software releases that are incompatible with the components in the NVIDIA vGPU software 11 major release branch.

<table>
<thead>
<tr>
<th>NVIDIA vGPU Software Component</th>
<th>Releases</th>
<th>Incompatible Software Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA vGPU Manager</td>
<td>11.0 through 11.2</td>
<td>All guest VM driver releases 9.x and earlier</td>
</tr>
<tr>
<td>Guest VM drivers</td>
<td>11.0 through 11.2</td>
<td>All NVIDIA vGPU Manager releases 10.x and earlier</td>
</tr>
</tbody>
</table>
1.3. Updates in Release 11.0

New Features in Release 11.0

‣ Cross-branch driver support
  This change allows a release of the Virtual GPU Manager from a later major release branch to be used with the NVIDIA vGPU software graphics drivers for the guest VMs from the previous branch.

‣ Licensing grace period for unlicensed virtual GPUs and physical GPUs
  An unlicensed virtual GPU or physical GPU initially operates at full capability but its performance is degraded over time if a license is not obtained.

‣ Miscellaneous bug fixes

1.4. Updates in Release 11.1

New Features in Release 11.1

‣ Support for Multi-Instance GPU (MIG)-backed vGPUs on GPUs that support MIG
‣ Support for GPUDirect® technology remote direct memory access (RDMA)
‣ Miscellaneous bug fixes

Hardware and Software Support Introduced in Release 11.1

‣ Support for the following GPUs:
  ‣ NVIDIA A100 PCIe 40GB
  ‣ NVIDIA A100 HGX 40GB

Feature Support Withdrawn in Release 11.1

‣ Windows Server 2012 R2 is no longer supported as a guest OS with GPUs based on architectures after NVIDIA Turing™ architecture.

1.5. Updates in Release 11.2

New Features in Release 11.2

‣ Miscellaneous bug fixes
Chapter 2. Validated Platforms

This release family of NVIDIA vGPU software provides support for several NVIDIA GPUs on validated server hardware platforms, Linux with KVM hypervisor software versions, and guest operating systems. It also supports the version of NVIDIA CUDA Toolkit that is compatible with R450 drivers.

2.1. Supported NVIDIA GPUs and Validated Server Platforms

For information about supported NVIDIA GPUs and the validated server hardware platforms on which they run, consult the documentation from your hypervisor vendor.

Note:

Tesla M60 and M6 GPUs support compute mode and graphics mode. NVIDIA vGPU requires GPUs that support both modes to operate in graphics mode.

Recent Tesla M60 GPUs and M6 GPUs are supplied in graphics mode. However, your GPU might be in compute mode if it is an older Tesla M60 GPU or M6 GPU, or if its mode has previously been changed.

To configure the mode of Tesla M60 and M6 GPUs, use the `gpumodeswitch` tool provided with NVIDIA vGPU software releases.

Even in compute mode, Tesla M60 and M6 GPUs do not support NVIDIA Virtual Compute Server vGPU types.
2.2. Hypervisor Software Releases

NVIDIA vGPU software supports only Linux with KVM hypervisor software from the vendors listed in the table.

Note: If a specific NVIDIA vGPU software release, even an update release, is not listed, it’s not supported. For information about which hypervisor software releases are supported, consult the documentation from your hypervisor vendor.

<table>
<thead>
<tr>
<th>Hypervisor Vendor</th>
<th>11.0</th>
<th>11.1</th>
<th>11.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat [1]</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>

Note:  
1. For more information, see Product Documentation for Red Hat OpenStack Platform.

# Release is supported.  
- Release is not supported.

2.3. Guest OS Support

For information about Windows releases and Linux distributions supported as a guest OS, consult the documentation from your hypervisor vendor.

Note:  
Use only a guest OS release that is listed as supported by NVIDIA vGPU software with your virtualization software. To be listed as supported, a guest OS release must be supported not only by NVIDIA vGPU software, but also by your virtualization software. NVIDIA cannot support guest OS releases that your virtualization software does not support. NVIDIA vGPU software supports only 64-bit guest operating systems. No 32-bit guest operating systems are supported.

2.4. NVIDIA CUDA Toolkit Version Support

The releases in this release family of NVIDIA vGPU software support NVIDIA CUDA Toolkit 11.0.

For more information about NVIDIA CUDA Toolkit, see CUDA Toolkit 11.0 Documentation.

Note:
If you are using NVIDIA vGPU software with CUDA on Linux, avoid conflicting installation methods by installing CUDA from a distribution-independent runfile package. Do not install CUDA from distribution-specific RPM or Deb package.

To ensure that the NVIDIA vGPU software graphics driver is not overwritten when CUDA is installed, deselect the CUDA driver when selecting the CUDA components to install.

For more information, see *NVIDIA CUDA Installation Guide for Linux*.

### 2.5. Multiple vGPU Support

To support applications and workloads that are compute or graphics intensive, multiple vGPUs can be added to a single VM. The assignment of more than one vGPU to a VM is supported only on a subset of vGPUs and Linux with KVM releases.

#### Supported vGPUs

Only Q-series and C-series time-sliced vGPUs that are allocated all of the physical GPU’s frame buffer are supported. MIG-backed vGPUs are not supported.

<table>
<thead>
<tr>
<th>GPU Architecture</th>
<th>Board</th>
<th>vGPU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Since 11.1: Ampere</strong></td>
<td>NVIDIA A100 PCIe 40GB</td>
<td>A100-40C See Note [1].</td>
</tr>
<tr>
<td></td>
<td>NVIDIA A100 HGX 40GB</td>
<td>A100X-40C See Note [1].</td>
</tr>
<tr>
<td><strong>Turing</strong></td>
<td>Tesla T4</td>
<td>T4-16Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T4-16C</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 6000</td>
<td>RTX6000-24Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTX6000-24C</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 6000 passive</td>
<td>RTX6000P-24Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTX6000P-24C</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 8000</td>
<td>RTX8000-48Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTX8000-48C</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 8000 passive</td>
<td>RTX8000P-48Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTX8000P-48C</td>
</tr>
<tr>
<td><strong>Volta</strong></td>
<td>Tesla V100 SXM2 32GB</td>
<td>V100DX-32Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V100D-32C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 PCIe 32GB</td>
<td>V100D-32Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V100D-32C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100S PCIe 32GB</td>
<td>V100S-32Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V100S-32C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 SXM2</td>
<td>V100X-16Q</td>
</tr>
<tr>
<td>GPU Architecture</td>
<td>Board</td>
<td>vGPU</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 PCIe</td>
<td>V100X-16C</td>
</tr>
<tr>
<td></td>
<td>V100-16Q</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V100-16C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tesla V100 FHHL</td>
<td>V100L-16Q</td>
</tr>
<tr>
<td></td>
<td>V100L-16C</td>
<td></td>
</tr>
<tr>
<td>Pascal</td>
<td>Tesla P100 SXM2</td>
<td>P100X-16Q</td>
</tr>
<tr>
<td></td>
<td>P100X-16C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tesla P100 PCIe 16GB</td>
<td>P100-16Q</td>
</tr>
<tr>
<td></td>
<td>P100-16C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tesla P100 PCIe 12GB</td>
<td>P100C-12Q</td>
</tr>
<tr>
<td></td>
<td>P100C-12C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tesla P40</td>
<td>P40-24Q</td>
</tr>
<tr>
<td></td>
<td>P40-24C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tesla P6</td>
<td>P6-16Q</td>
</tr>
<tr>
<td></td>
<td>P6-16C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tesla P4</td>
<td>P4-8Q</td>
</tr>
<tr>
<td></td>
<td>P4-8C</td>
<td></td>
</tr>
<tr>
<td>Maxwell</td>
<td>Tesla M60</td>
<td>M60-8Q</td>
</tr>
<tr>
<td></td>
<td>Tesla M10</td>
<td>M10-8Q</td>
</tr>
<tr>
<td></td>
<td>Tesla M6</td>
<td>M6-8Q</td>
</tr>
</tbody>
</table>

**Note:**

1. This type of vGPU cannot be assigned with other types of vGPU to the same VM.

**Maximum vGPUs per VM**

NVIDIA vGPU software supports up to a maximum of 16 vGPUs per VM on Linux with KVM.

**Supported Hypervisor Releases**

For information about which hypervisor software releases support the assignment of more than one vGPU device to a VM, consult the documentation from your hypervisor vendor.
2.6. Peer-to-Peer CUDA Transfers over NVLink Support

Peer-to-peer CUDA transfers enable device memory between vGPUs on different GPUs that are assigned to the same VM to be accessed from within the CUDA kernels. NVLink is a high-bandwidth interconnect that enables fast communication between such vGPUs. Peer-to-Peer CUDA Transfers over NVLink is supported only on a subset of vGPUs, Linux with KVM releases, and guest OS releases.

Supported vGPUs

Only Q-series and C-series time-sliced vGPUs that are allocated all of the physical GPU’s frame buffer on physical GPUs that support NVLink are supported.

<table>
<thead>
<tr>
<th>GPU Architecture</th>
<th>Board</th>
<th>vGPU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Since 11.1: Ampere</strong></td>
<td>NVIDIA A100 PCIe 40GB</td>
<td>A100-40C</td>
</tr>
<tr>
<td></td>
<td>NVIDIA A100 HGX 40GB</td>
<td>A100X-40C See Note [1].</td>
</tr>
<tr>
<td>Turing</td>
<td>Quadro RTX 6000</td>
<td>RTX6000-24Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTX6000-24C</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 6000 passive</td>
<td>RTX6000P-24Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTX6000P-24C</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 8000</td>
<td>RTX8000-48Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTX8000-48C</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 8000 passive</td>
<td>RTX8000P-48Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTX8000P-48C</td>
</tr>
<tr>
<td>Volta</td>
<td>Tesla V100 SXM2 32GB</td>
<td>V100DX-32Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V100DX-32C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 SXM2</td>
<td>V100X-16Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V100X-16C</td>
</tr>
<tr>
<td>Pascal</td>
<td>Tesla P100 SXM2</td>
<td>P100X-16Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P100X-16C</td>
</tr>
</tbody>
</table>

**Note:**

1. Supported only on the NVIDIA HGX A100 4-GPU baseboard with four fully connected GPUs.
Supported Hypervisor Releases

Peer-to-Peer CUDA Transfers over NVLink are supported on all hypervisor releases that support the assignment of more than one vGPU to a VM. For details, see Multiple vGPU Support.

Supported Guest OS Releases

Linux only. Peer-to-Peer CUDA Transfers over NVLink are not supported on Windows.

Limitations

- Only direct connections are supported. NVSwitch is not supported.
- Only time-sliced vGPUs are supported. MIG-backed vGPUs are not supported.
- Since 11.1: PCIe is supported only with the NVIDIA A100 PCIe 40GB GPU. Otherwise, PCIe is not supported.
- SLI is not supported.

2.7. Since 11.1: GPUDirect Technology Support

GPUDirect® technology remote direct memory access (RDMA) enables network devices to directly access vGPU frame buffer, bypassing CPU host memory altogether. GPUDirect technology is supported only on a subset of vGPUs and guest OS releases.

Supported vGPUs

Only C-series vGPUs that are allocated all of the physical GPU’s frame buffer on physical GPUs based on the NVIDIA Ampere architecture are supported. Both time-sliced and MIG-backed vGPUs that meet these requirements are supported.

<table>
<thead>
<tr>
<th>GPU Architecture</th>
<th>Board</th>
<th>vGPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ampere</td>
<td>NVIDIA A100 PCIe 40GB</td>
<td>A100-40C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A100-7-40C</td>
</tr>
<tr>
<td>Ampere</td>
<td>NVIDIA A100 HGX 40GB</td>
<td>A100X-40C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A100X-7-40C</td>
</tr>
</tbody>
</table>

Supported Guest OS Releases

Linux only. GPUDirect technology is not supported on Windows.
Supported Network Interface Cards

GPUDirect technology RDMA is supported on the following network interface cards:

- Mellanox Connect-X® 6 SmartNIC
- Mellanox Connect-X 5 Ethernet adapter card

Limitations

Only GPUDirect technology RDMA is supported. GPUDirect technology storage is not supported.
Chapter 3. Known Product Limitations

Known product limitations for this release of NVIDIA vGPU software are described in the following sections.

3.1. Issues occur when the channels allocated to a vGPU are exhausted

Description

Issues occur when the channels allocated to a vGPU are exhausted and the guest VM to which the vGPU is assigned fails to allocate a channel to the vGPU. A physical GPU has a fixed number of channels and the number of channels allocated to each vGPU is inversely proportional to the maximum number of vGPUs allowed on the physical GPU.

When the channels allocated to a vGPU are exhausted and the guest VM fails to allocate a channel, the following errors are reported on the hypervisor host or in an NVIDIA bug report:

Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0): Guest attempted to allocate channel above its max channel limit 0xfb
Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0): VGPU message 6 failed, result code: 0xa
Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0): 0x1000b000, 0xff0400fb, 0xc36f,
Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0): , 0x0

Workaround

Use a vGPU type with more frame buffer, thereby reducing the maximum number of vGPUs allowed on the physical GPU. As a result, the number of channels allocated to each vGPU is increased.
3.2. **Virtual GPU hot plugging is not supported**

NVIDIA vGPU software does not support the addition of virtual function I/O (VFIO) mediated device (mdev) devices after the VM has been started by QEMU. All mdev devices must be added before the VM is started.

3.3. **Total frame buffer for vGPUs is less than the total frame buffer on the physical GPU**

Some of the physical GPU’s frame buffer is used by the hypervisor on behalf of the VM for allocations that the guest OS would otherwise have made in its own frame buffer. The frame buffer used by the hypervisor is not available for vGPUs on the physical GPU. In NVIDIA vGPU deployments, frame buffer for the guest OS is reserved in advance, whereas in bare-metal deployments, frame buffer for the guest OS is reserved on the basis of the runtime needs of applications.

If error-correcting code (ECC) memory is enabled on a physical GPU that does not have HBM2 memory, the amount of frame buffer that is usable by vGPUs is further reduced. All types of vGPU are affected, not just vGPUs that support ECC memory.

On all GPUs that support ECC memory and, therefore, dynamic page retirement, additional frame buffer is allocated for dynamic page retirement. The amount that is allocated is inversely proportional to the maximum number of vGPUs per physical GPU. All GPUs that support ECC memory are affected, even GPUs that have HBM2 memory or for which ECC memory is disabled.

The approximate amount of frame buffer that NVIDIA vGPU software reserves can be calculated from the following formula:

\[ \text{max-reserved-fb} = \frac{\text{vgpu-profile-size-in-mb}}{16} + 16 + \text{ecc-adjustments} + \text{page-retirement-allocation} \]

- **max-reserved-fb**
  The maximum total amount of reserved frame buffer in Mbytes that is not available for vGPUs.
- **vgpu-profile-size-in-mb**
  The amount of frame buffer in Mbytes allocated to a single vGPU. This amount depends on the vGPU type. For example, for the T4-16Q vGPU type, \( \text{vgpu-profile-size-in-mb} = 16384 \).
- **ecc-adjustments**
  The amount of frame buffer in Mbytes that is not usable by vGPUs when ECC is enabled on a physical GPU that does not have HBM2 memory.
If ECC is enabled on a physical GPU that does not have HBM2 memory, \textit{ecc-adjustments} is \textit{fb-without-ecc}/16, which is equivalent to 64 Mbytes for every Gbyte of frame buffer assigned to the vGPU. \textit{fb-without-ecc} is total amount of frame buffer with ECC disabled.

If ECC is disabled or the GPU has HBM2 memory, \textit{ecc-adjustments} is 0.

\textbf{page-retirement-allocation}

The amount of frame buffer in Mbytes that is reserved for dynamic page retirement.

- On GPUs based on the NVIDIA Maxwell GPU architecture, \textit{page-retirement-allocation} = $4 \div \text{max-vgpus-per-gpu}$.
- On GPUs based on NVIDIA GPU architectures after the Maxwell architecture, \textit{page-retirement-allocation} = $128 \div \text{max-vgpus-per-gpu}$

\textbf{max-vgpus-per-gpu}

The maximum number of vGPUs that can be created simultaneously on a physical GPU. This number varies according to the vGPU type. For example, for the T4-16Q vGPU type, \textit{max-vgpus-per-gpu} is 1.

\textit{Note:} In VMs running Windows Server 2012 R2, which supports Windows Display Driver Model (WDDM) 1.x, an additional 48 Mbytes of frame buffer are reserved and not available for vGPUs.

3.4. Issues may occur with graphics-intensive OpenCL applications on vGPU types with limited frame buffer

\textbf{Description}

Issues may occur when graphics-intensive OpenCL applications are used with vGPU types that have limited frame buffer. These issues occur when the applications demand more frame buffer than is allocated to the vGPU.

For example, these issues may occur with the Adobe Photoshop and LuxMark OpenCL Benchmark applications:

- When the image resolution and size are changed in Adobe Photoshop, a program error may occur or Photoshop may display a message about a problem with the graphics hardware and a suggestion to disable OpenCL.
- When the LuxMark OpenCL Benchmark application is run, XID error 31 may occur.

\textbf{Workaround}

For graphics-intensive OpenCL applications, use a vGPU type with more frame buffer.
3.5. In pass through mode, all GPUs connected to each other through NVLink must be assigned to the same VM

Description
In pass through mode, all GPUs connected to each other through NVLink must be assigned to the same VM. If a subset of GPUs connected to each other through NVLink is passed through to a VM, unrecoverable error XID 74 occurs when the VM is booted. This error corrupts the NVLink state on the physical GPUs and, as a result, the NVLink bridge between the GPUs is unusable.

Workaround
Restore the NVLink state on the physical GPUs by resetting the GPUs or rebooting the hypervisor host.

3.6. vGPU profiles with 512 Mbytes or less of frame buffer support only 1 virtual display head on Windows 10

Description
To reduce the possibility of memory exhaustion, vGPU profiles with 512 Mbytes or less of frame buffer support only 1 virtual display head on a Windows 10 guest OS.

The following vGPU profiles have 512 Mbytes or less of frame buffer:

- Tesla M6-0B, M6-0Q
- Tesla M10-0B, M10-0Q
- Tesla M60-0B, M60-0Q

Workaround
Use a profile that supports more than 1 virtual display head and has at least 1 Gbyte of frame buffer.
3.7. **NVENC requires at least 1 Gbyte of frame buffer**

**Description**

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 Mbytes or less of frame buffer. To reduce the possibility of memory exhaustion, NVENC is disabled on profiles that have 512 Mbytes or less of frame buffer. Application GPU acceleration remains fully supported and available for all profiles, including profiles with 512 MBytes or less of frame buffer. NVENC support from both Citrix and VMware is a recent feature and, if you are using an older version, you should experience no change in functionality.

The following vGPU profiles have 512 Mbytes or less of frame buffer:

- Tesla M6-0B, M6-0Q
- Tesla M10-0B, M10-0Q
- Tesla M60-0B, M60-0Q

**Workaround**

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

3.8. **VM running an incompatible NVIDIA vGPU guest driver fails to initialize vGPU when booted**

**Description**

A VM running a version of the NVIDIA guest VM driver that is incompatible with the current release of Virtual GPU Manager will fail to initialize vGPU when booted on a Linux with KVM platform running that release of Virtual GPU Manager.

A guest VM driver is incompatible with the current release of Virtual GPU Manager in either of the following situations:

- The guest driver is from a release in a branch two or more major releases before the current release, for example release 9.4.

  In this situation, the Linux with KVM VM's `/var/log/messages` log file reports the following error:
Known Product Limitations

Virtual GPU Software R450 for Linux with KVM

---

vmiop_log: (0x0): Incompatible Guest/Host drivers: Guest VGX version is older than the minimum version supported by the Host. Disabling vGPU.

- The guest driver is from a later release than the Virtual GPU Manager.

In this situation, the Linux with KVM VM’s /var/log/messages log file reports the following error:

vmiop_log: (0x0): Incompatible Guest/Host drivers: Guest VGX version is newer than the maximum version supported by the Host. Disabling vGPU.

In either situation, the VM boots in standard VGA mode with reduced resolution and color depth. The NVIDIA virtual 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)

**Resolution**

Install a release of the NVIDIA guest VM driver that is compatible with current release of Virtual GPU Manager.

---

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

**Description**

A single vGPU configured on a physical GPU produces lower benchmark scores than the physical GPU run in pass-through mode.

Aside from performance differences that may be attributed to a vGPU’s smaller frame buffer size, vGPU incorporates a performance balancing feature known as Frame Rate Limiter (FRL). On vGPUs that use the best-effort scheduler, FRL is enabled. On vGPUs that use the fixed share or equal share scheduler, FRL is disabled.

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 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.

**Resolution**

FRL is controlled by an internal vGPU setting. On vGPUs that use the best-effort scheduler, NVIDIA does not validate vGPU with FRL disabled, but for validation of benchmark performance, FRL can be temporarily disabled by setting `frame_rate_limiter=0` in the vGPU configuration file.

```
# echo "frame_rate_limiter=0" > /sys/bus/mdev/devices/vgpu-id/nvidia/vgpu_params
```

For example:
Known Product Limitations

Virtual GPU Software R450 for Linux with KVM
RN-09065-001 _v11.0 through 11.2 Revision 02   |   17

# echo "frame_rate_limiter=0" > /sys/bus/mdev/devices/aa618089-8b16-4d01-a136-25a0f3c73123/nvidia/vgpu_params

The setting takes effect the next time any VM using the given vGPU type is started.

With this setting in place, the VM’s vGPU will run without any frame rate limit.

The FRL can be reverted back to its default setting as follows:

1. Clear all parameter settings in the vGPU configuration file.
   # echo " " > /sys/bus/mdev/devices/vgpu-id/nvidia/vgpu_params

   **Note:** You cannot clear specific parameter settings. If your vGPU configuration file contains other parameter settings that you want to keep, you must reinstate them in the next step.

2. Set frame_rate_limiter=1 in the vGPU configuration file.
   # echo "frame_rate_limiter=1" > /sys/bus/mdev/devices/vgpu-id/nvidia/vgpu_params

   If you need to reinstate other parameter settings, include them in the command to set frame_rate_limiter=1. For example:
   # echo "frame_rate_limiter=1 disable_vnc=1" > /sys/bus/mdev/devices/aa618089-8b16-4d01-a136-25a0f3c73123/nvidia/vgpu_params

### 3.10. **nvidia-smi** fails to operate when all GPUs are assigned to GPU pass-through mode

**Description**

If all GPUs in the platform are assigned to VMs in pass-through mode, **nvidia-smi** will return an error:

```
[root@vgx-test ~]# nvidia-smi
Failed to initialize NVML: Unknown Error
```

This is because GPUs operating in pass-through mode are not visible to **nvidia-smi** and the NVIDIA kernel driver operating in the Linux with KVM host.

To confirm that all GPUs are operating in pass-through mode, confirm that the **vfio-pci** kernel driver is handling each device.

```
# lspci -s 05:00.0 -k
05:00.0 VGA compatible controller: NVIDIA Corporation GM204GL [Tesla M60] (rev a1)
  Subsystem: NVIDIA Corporation Device 113a
  Kernel driver in use: vfio-pci
```

**Resolution**

N/A
Chapter 4. Resolved Issues

Only resolved issues that have been previously noted as known issues or had a noticeable user impact are listed. The summary and description for each resolved issue indicate the effect of the issue on NVIDIA vGPU software before the issue was resolved.

Issues Resolved in Release 11.0

<table>
<thead>
<tr>
<th>Bug ID</th>
<th>Summary and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200275925</td>
<td>Resolution is not updated after a VM acquires a license and is restarted</td>
</tr>
<tr>
<td></td>
<td>In a Red Enterprise Linux 7.3 guest VM, an increase in resolution from 1024×768 to 2560×1600 is not applied after a license is acquired and the gridd service is restarted. This issue occurs if the multimonitor parameter is added to the xorg.conf file.</td>
</tr>
</tbody>
</table>

Issues Resolved in Release 11.1

<table>
<thead>
<tr>
<th>Bug ID</th>
<th>Summary and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200626446</td>
<td>11.0 Only: Failure to allocate resources causes VM failures or crashes</td>
</tr>
<tr>
<td></td>
<td>Failure to allocate resources causes VM failures or crashes. When the error occurs, the error message NVOS status 0x19 is written to the log file on the hypervisor host. Depending on the resource and the underlying cause of the failure, VGPU message 52 failed, VGPU message 4 failed, VGPU message 21 failed, and VGPU message 10 failed might also be written to the log file on the hypervisor host.</td>
</tr>
<tr>
<td>3051614</td>
<td>11.0 Only: Application responsiveness degrades over time</td>
</tr>
</tbody>
</table>
|          | Application responsiveness degrades over time, causing slow application performance and stutter when users switch between applications. This issue occurs because the GPU driver is not setting the Linux kernel PCI coherent_dma_mask for NVIDIA GPU devices. If the coherent_dma_mask is not set, IOMMU IOVA space is restricted to the default size of 32 bits for DMA allocations performed in the NVIDIA GPU device context. Furthermore, for hosts on which iommu=pt is set, the default coherent_dma_mask causes IOMMU mappings to always be created. When IOMMU mappings are always
## Resolved Issues

<table>
<thead>
<tr>
<th>Bug ID</th>
<th>Summary and Description</th>
</tr>
</thead>
</table>
| 3063042  | **11.0 Only:** On systems with more than 1 TiB of system memory, application performance degrades over time  
On systems with more than 1 TiB of system memory, application performance degrades over time. As a result, application performance is slow and stutter occurs when users switch between applications. This issue occurs because the virtual GPU manager temporarily limits the `dma_mask` and the `coherent_dma_mask` to 40 bits while the vGPU is being initialized. On systems with more than 1 TiB of system memory, the `coherent_dma_mask` addressing capability is less than the amount of system memory. As a result, IOMMU mappings are always created, which can cause performance degradation because all host to device accesses require translation by hardware IOMMU. |
| 3087984  | **11.0 Only:** The NVIDIA license not present notification appears even for VMs for which the vGPU is licensed  
After starting and logging on to a Windows VM that is configured with NVIDIA vGPU, users see the *NVIDIA license not present* notification even when the vGPU is already licensed. This notification is misleading and should be ignored. This issue occurs because the NVIDIA driver is not storing the correct NVIDIA vGPU software license state. |

### Issues Resolved in Release 11.2

<table>
<thead>
<tr>
<th>Bug ID</th>
<th>Summary and Description</th>
</tr>
</thead>
</table>
| 200658253| **11.1 Only:** Licensing event logs indicate license renewal from unavailable primary server  
Licensing event logs for the guest VM indicate that a license is renewed from primary license server even when primary license server is unavailable and the license is renewed from the secondary server. |
Chapter 5. Known Issues

5.1. Since 11.1: Guest VM frame buffer listed by `nvidia-smi` for vGPUs on GPUs that support SRIOV is incorrect

Description

The amount of frame buffer listed in a guest VM by the `nvidia-smi` command for vGPUs on GPUs that support Single Root I/O Virtualization (SR-IOV) is incorrect. Specifically, the amount of frame buffer listed is the amount of frame buffer allocated for the vGPU type minus the size of the VMMU segment (`vmmu_page_size`). Examples of GPUs that support SRIOV are GPUs based on the NVIDIA Ampere architecture, such as NVIDIA A100 PCIe 40GB or NVIDIA A100 HGX 40GB.

For example, frame buffer for -4C and -20C vGPU types is listed as follows:

- For -4C vGPU types, frame buffer is listed as 3963 MB instead of 4096 MB.
- For -20C vGPU types, frame buffer is listed as 20347 MB instead of 20480 MB.

Status

Open

Ref. #

200524749
5.2. 11.0 Only: Application responsiveness degrades over time

Description

Application responsiveness degrades over time, causing slow application performance and stutter when users switch between applications. This issue occurs because the GPU driver is not setting the Linux kernel PCI `coherent_dma_mask` for NVIDIA GPU devices. If the `coherent_dma_mask` is not set, IOMMU IOVA space is restricted to the default size of 32 bits for DMA allocations performed in the NVIDIA GPU device context. Furthermore, for hosts on which `iommu=pt` is set, the default `coherent_dma_mask` causes IOMMU mappings to always be created. When IOMMU mappings are always created, performance degradation can occur because all host to device accesses require translation by hardware IOMMU.

Status

Resolved in NVIDIA vGPU software 11.1

Note: On systems with more than 1 TiB of system memory and GPUs based on GPU architectures earlier than the NVIDIA Ampere architecture, a related issue might still cause application performance to degrade over time. For details, see 11.0 Only: On systems with more than 1 TiB of system memory, application performance degrades over time.

Ref. #

3051614

5.3. 11.0 Only: On systems with more than 1 TiB of system memory, application performance degrades over time

Description

On systems with more than 1 TiB of system memory, application performance degrades over time. As a result, application performance is slow and stutter occurs when users switch between applications. This issue occurs because the virtual GPU manager temporarily limits the `dma_mask` and the `coherent_dma_mask` to 40 bits while the vGPU is being initialized. On systems with more than 1 TiB of system memory, the `coherent_dma_mask` addressing capability is less than the amount of system memory. As a result, IOMMU mappings are...
always created, which can cause performance degradation because all host to device accesses require translation by hardware IOMMU.

**Workaround**
Reduce the amount of system memory to 1 TiB or less.

**Status**
Resolved in NVIDIA vGPU software 11.1

**Ref. #**
3063042

5.4. **11.1 Only: Licensing event logs indicate license renewal from unavailable primary server**

**Description**
Licensing event logs for the guest VM indicate that a license is renewed from primary license server even when primary license server is unavailable and the license is renewed from the secondary server.

**Workaround**
None. However, these incorrect event log entries are benign and can be ignored.

**Status**
Resolved in NVIDIA vGPU software 11.2

**Ref. #**
200658253
5.5. 11.0 Only: The **NVIDIA license not present** notification appears even for VMs for which the vGPU is licensed

**Description**

After starting and logging on to a Windows VM that is configured with NVIDIA vGPU, users see the **NVIDIA license not present** notification even when the vGPU is already licensed. This notification is misleading and should be ignored. This issue occurs because the NVIDIA driver is not storing the correct NVIDIA vGPU software license state.

Users can confirm that their vGPU is licensed in one of the following ways:

- Opening **NVIDIA Control Panel** and checking the **Licensing > Manage License** page.
- Running the following command in a **Command Prompt** window:

  ```
  C:\Program Files\NVIDIA Corporation\NVSMI\nvidia-smi -q
  ```

**Version**

This issue affects Windows server and Windows desktop operating systems.

**Workaround**

While the misleading notifications can be ignored, this workaround can be used to suppress the notifications if they cause confusion.

**CAUTION:** This workaround suppresses **all notifications**, **even valid notifications**, from **NVIDIA Container**. Therefore, use this workaround only if immediate resolution is needed.

1. When the notification appears, grab it by right-clicking in the notification window.
2. In **Action Center**, go to Windows notifications for **NVIDIA Container**, right-click the notification, and click **Go to notification settings**.
3. In the notification settings for **NVIDIA Container**, set **Show notification banners** to **Off**.
If necessary, you can reenable these notifications in one of the following ways:

▶ Use **System settings** as follows:

1. In **Notifications & actions**, under **Get notifications from these senders**, select **NVIDIA Container** to see more settings.

2. In the notification settings for **NVIDIA Container**, set the **Show notification banners** option to **On**.

▶ Set the Windows registry key `HKEY_CURRENT_USER\Software\Microsoft\Windows \CurrentVersion\Notifications\Settings\Microsoft.Explorer.Notification. {D38519CE-37BE-5DFF-CF14-CC1848376559}\ShowBanner` to **1**.
5.6. Driver upgrade in a Linux guest VM with multiple vGPUs might fail

Description
Upgrading the NVIDIA vGPU software graphics driver in a Linux guest VM with multiple vGPUs might fail. This issue occurs if the driver is upgraded by overinstalling the new release of the driver on the current release of the driver while the `nvidia-gridd` service is running in the VM.

Workaround
1. Stop the `nvidia-gridd` service.
2. Try again to upgrade the driver.

Status
Open

Ref. #
200633548
5.7. 11.0 Only: Failure to allocate resources causes VM failures or crashes

Description
Failure to allocate resources causes VM failures or crashes. When the error occurs, the error message NVOS status 0x19 is written to the log file on the hypervisor host. Depending on the resource and the underlying cause of the failure, VGPU message 52 failed, VGPU message 4 failed, VGPU message 21 failed, and VGPU message 10 failed might also be written to the log file on the hypervisor host.

Status
Resolved in NVIDIA vGPU software 11.1

Ref. #
200626446

5.8. NVIDIA Control Panel fails to start if launched too soon from a VM without licensing information

Description
If NVIDIA licensing information is not configured on the system, any attempt to start NVIDIA Control Panel by right-clicking on the desktop within 30 seconds of the VM being started fails.

Workaround
Restart the VM and wait at least 30 seconds before trying to launch NVIDIA Control Panel.

Status
Open

Ref. #
200623179
5.9. Remoting solution session freezes with VGPU message 21 failed and VGPU message 14 failed errors

Description
The remoting solution session sometimes freezes while a window is being resized. For a Windows guest VM, the error message VGPU message 21 failed is written to the log file on the hypervisor host. For a Linux guest VM, the error messages VGPU message 21 failed and VGPU message 14 failed are written to the log file on the hypervisor host.

Workaround
Try resizing the window again.

Status
Open

Ref. #
200627445

5.10. On Linux, the frame rate might drop to 1 after several minutes

Description
On Linux, the frame rate might drop to 1 frame per second (FPS) after NVIDIA vGPU software has been running for several minutes. Only some applications are affected, for example, glxgears. Other applications, such as Unigine Heaven, are not affected. This behavior occurs because Display Power Management Signaling (DPMS) for the Xorg server is enabled by default and the display is detected to be inactive even when the application is running. When DPMS is enabled, it enables power saving behavior of the display after several minutes of inactivity by setting the frame rate to 1 FPS.

Workaround
1. If necessary, stop the Xorg server.

   # /etc/init.d/xorg stop
2. In a plain text editor, edit the `/etc/X11/xorg.conf` file to set the options to disable DPMS and disable the screen saver.

   a). In the Monitor section, set the DPMS option to false.
      ```
      Option "DPMS" "false"
      ```

   b). At the end of the file, add a ServerFlags section that contains option to disable the screen saver.
      ```
      Section "ServerFlags"
      Option "BlankTime" "0"
      EndSection
      ```

   c). Save your changes to `/etc/X11/xorg.conf` file and quit the editor.

3. Start the Xorg server.
   ```
   # etc/init.d/xorg start
   ```

   Status
   Open

   Ref. #
   200605900

5.11. DWM crashes randomly occur in Windows VMs

Description
Desktop Windows Manager (DWM) crashes randomly occur in Windows VMs, causing a blue-screen crash and the bug check CRITICAL_PROCESS_DIED. Computer Management shows problems with the primary display device.

Version
This issue affects Windows 10 1809, 1903 and 1909 VMs.

Status
Not an NVIDIA bug

Ref. #
2730037
5.12. Publisher not verified warning during Windows 7 driver installation

Description

During installation of the NVIDIA vGPU software graphics driver for Windows on Windows 7, Windows warns that it can’t verify the publisher of the driver software. If Device Manager is used to install the driver, Device Manager warns that the driver is not digitally signed. If you install the driver, error 52 (CM_PROB_UNSIGNED_DRIVER) occurs.

This issue occurs because Microsoft is no longer dual signing WHQL-tested software binary files by using the SHA-1 and SHA-2 hash algorithms. Instead, WHQL-tested software binary files are signed only by using the SHA-2 hash algorithm. All NVIDIA vGPU software graphics drivers for Windows are WHQL tested.

By default, Windows 7 systems cannot recognize signatures that were created by using the SHA-2 hash algorithm. As a result, software binary files that are signed only by using the SHA-2 hash algorithm are considered unsigned.

For more information, see 2019 SHA-2 Code Signing Support requirement for Windows and WSUS on the Microsoft Windows support website.

Version

Windows 7

Workaround

If you experience this issue, install the following updates and restart the VM or host before installing the driver:

- Servicing stack update (SSU) [KB4490628]
- SHA-2 update [KB4474419]

Status

Not a bug
5.13. **RAPIDS cuDF merge fails on NVIDIA vGPU**

**Description**

The merge function of the RAPIDS cuDF GPU data frame library fails on NVIDIA vGPU. This function fails because RAPIDS uses the Unified Memory feature of CUDA, which NVIDIA vGPU does not support.

**Status**

Open

**Ref. #**

2642134

5.14. **ECC memory settings for a vGPU cannot be changed by using NVIDIA X Server Settings**

**Description**

The ECC memory settings for a vGPU cannot be changed from a Linux guest VM by using **NVIDIA X Server Settings**. After the ECC memory state has been changed on the **ECC Settings** page and the VM has been rebooted, the ECC memory state remains unchanged.

**Workaround**

Use the `nvidia-smi` command in the guest VM to enable or disable ECC memory for the vGPU as explained in **Virtual GPU Software User Guide**.

If the ECC memory state remains unchanged even after you use the `nvidia-smi` command to change it, use the workaround in **Changes to ECC memory settings for a Linux vGPU VM by nvidia-smi might be ignored**.

**Status**

Open
5.15. Changes to ECC memory settings for a Linux vGPU VM by `nvidia-smi` might be ignored

Description
After the ECC memory state for a Linux vGPU VM has been changed by using the `nvidia-smi` command and the VM has been rebooted, the ECC memory state might remain unchanged. This issue occurs when multiple NVIDIA configuration files in the system cause the kernel module option for setting the ECC memory state `RMGuestECCState` in `/etc/modprobe.d/nvidia.conf` to be ignored.

When the `nvidia-smi` command is used to enable ECC memory, the file `/etc/modprobe.d/nvidia.conf` is created or updated to set the kernel module option `RMGuestECCState`. Another configuration file in `/etc/modprobe.d/` that contains the keyword `NVreg_RegisteryDwordsPerDevice` might cause the kernel module option `RMGuestECCState` to be ignored.

Workaround
This workaround requires administrator privileges.

1. Move the entry containing the keyword `NVreg_RegisteryDwordsPerDevice` from the other configuration file to `/etc/modprobe.d/nvidia.conf`.
2. Reboot the VM.

Status
Open

Ref. #
200505777
5.16. **Vulkan applications crash in Windows 7 guest VMs configured with NVIDIA vGPU**

**Description**

In Windows 7 guest VMs configured with NVIDIA vGPU, applications developed with Vulkan APIs crash or throw errors when they are launched. Vulkan APIs require sparse texture support, but in Windows 7 guest VMs configured with NVIDIA vGPU, sparse textures are not enabled.

In Windows 10 guest VMs configured with NVIDIA vGPU, sparse textures are enabled and applications developed with Vulkan APIs run correctly in these VMs.

**Status**

Open

**Ref. #**

200381348

5.17. **Host core CPU utilization is higher than expected for moderate workloads**

**Description**

When GPU performance is being monitored, host core CPU utilization is higher than expected for moderate workloads. For example, host CPU utilization when only a small number of VMs are running is as high as when several times as many VMs are running.

**Workaround**

Disable monitoring of the following GPU performance statistics:

- vGPU engine usage by applications across multiple vGPUs
- Encoder session statistics
- Frame buffer capture (FBC) session statistics
- Statistics gathered by performance counters in guest VMs
5.18. **Frame capture while the interactive logon message is displayed returns blank screen**

**Description**
Because of a known limitation with NvFBC, a frame capture while the interactive logon message is displayed returns a blank screen.

An NvFBC session can capture screen updates that occur after the session is created. Before the logon message appears, there is no screen update after the message is shown and, therefore, a black screen is returned instead. If the NvFBC session is created after this update has occurred, NvFBC cannot get a frame to capture.

**Workaround**
Press **Enter** or wait for the screen to update for NvFBC to capture the frame.

**Status**
Not a bug

**Ref. #**
2115733
5.19. **RDS sessions do not use the GPU with some Microsoft Windows Server releases**

**Description**

When some releases of Windows Server are used as a guest OS, Remote Desktop Services (RDS) sessions do not use the GPU. With these releases, the RDS sessions by default use the Microsoft Basic Render Driver instead of the GPU. This default setting enables 2D DirectX applications such as Microsoft Office to use software rendering, which can be more efficient than using the GPU for rendering. However, as a result, 3D applications that use DirectX are prevented from using the GPU.

**Version**

- Windows Server 2019
- Windows Server 2016
- Windows Server 2012

**Solution**

Change the local computer policy to use the hardware graphics adapter for all RDS sessions.

2. Set the *Use the hardware default graphics adapter for all Remote Desktop Services sessions* option.

5.20. **Even when the scheduling policy is equal share, unequal GPU utilization is reported**

**Description**

When the scheduling policy is equal share, unequal GPU engine utilization can be reported for the vGPUs on the same physical GPU.
For example, GPU engine usage for three P40-8Q vGPUs on a Tesla P40 GPU might be reported as follows:

```
[root@localhost:~] nvidia-smi vgpu
Wed Jun 27 10:33:18 2018
+-----------------------------------------------------------------------------+
<table>
<thead>
<tr>
<th>NVIDIA-SMI 390.59</th>
<th>Driver Version: 390.59</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU Name</td>
<td>vGPU ID</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Tesla P40</td>
<td>00000000:81:00.0</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122661</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122663</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122659</td>
</tr>
<tr>
<td>Tesla P40</td>
<td>00000000:85:00.0</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122662</td>
</tr>
<tr>
<td>2122658 GRID P40-8Q</td>
<td>2122667 centos7.4-xmpl-211...</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122660</td>
</tr>
</tbody>
</table>
```

The vGPU utilization of the vGPU 2122658 is reported as 59%. However, the expected vGPU utilization should not exceed 33%.

This behavior is a result of the mechanism that is used to measure GPU engine utilization.

**Status**

Open

**Ref. #**

2175888

### 5.21. When the scheduling policy is fixed share, GPU utilization is reported as higher than expected

**Description**

When the scheduling policy is fixed share, GPU engine utilization can be reported as higher than expected for a vGPU.

For example, GPU engine usage for six P40-4Q vGPUs on a Tesla P40 GPU might be reported as follows:

```
[root@localhost:~] nvidia-smi vgpu
Mon Aug 20 10:33:18 2018
+-----------------------------------------------------------------------------+
<table>
<thead>
<tr>
<th>NVIDIA-SMI 390.42</th>
<th>Driver Version: 390.42</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU Name</td>
<td>vGPU ID</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Tesla P40</td>
<td>00000000:81:00.0</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122661</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122663</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122659</td>
</tr>
<tr>
<td>Tesla P40</td>
<td>00000000:85:00.0</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122662</td>
</tr>
<tr>
<td>2122658 GRID P40-8Q</td>
<td>2122667 centos7.4-xmpl-211...</td>
</tr>
<tr>
<td>GRID P40-8Q</td>
<td>2122660</td>
</tr>
</tbody>
</table>
```

Virtual GPU Software R450 for Linux with KVM
The vGPU utilization of vGPU 85109 is reported as 32%. For vGPU 87195, vGPU utilization is reported as 39%. And for 88095, it is reported as 26%. However, the expected vGPU utilization of any vGPU should not exceed approximately 16.7%.

This behavior is a result of the mechanism that is used to measure GPU engine utilization.

**Status**

Open

**Ref. #**

2227591

### 5.22. License is not acquired in Windows VMs

**Description**

When a windows VM configured with a licensed vGPU is started, the VM fails to acquire a license.

Error messages in the following format are written to the NVIDIA service logs:

```
[000000020.860152600 sec] - [Logging.lib]   ERROR: [nvGridLicensing.FlexUtility] 353@FlexUtility::LogFneError : Error: Failed to add trusted storage. Server URL : license-server-url -
[1,7E2,2,1[7000003F,0,9B00A7]]

System machine type does not match expected machine type.
```

**Workaround**

This workaround requires administrator privileges.

1. Stop the **NVIDIA Display Container LS** service.
2. Delete the contents of the folder `%SystemDrive%\Program Files\NVIDIA Corporation\Grid Licensing`.
3. Start the **NVIDIA Display Container LS** service.
5.23. **nvidia-smi** reports that vGPU migration is supported on all hypervisors

**Description**

The command `nvidia-smi vgpu -m` shows that vGPU migration is supported on all hypervisors, even hypervisors or hypervisor versions that do not support vGPU migration.

**Status**

Closed

**Ref. #**

200407230

5.24. Hot plugging and unplugging vCPUs causes a blue-screen crash in Windows VMs

**Description**

Hot plugging or unplugging vCPUs causes a blue-screen crash in Windows VMs that are running NVIDIA vGPU software graphics drivers.

When the blue-screen crash occurs, one of the following error messages may also be seen:

- `SYSTEM_SERVICE_EXCEPTION(nvlddmkm.sys)`
- `DRIVER_IRQL_NOT_LESS_OR_EQUAL(nvlddmkm.sys)`

NVIDIA vGPU software graphics drivers do not support hot plugging and unplugging of vCPUs.
5.25. Luxmark causes a segmentation fault on an unlicensed Linux client

Description
If the Luxmark application is run on a Linux guest VM configured with NVIDIA vGPU that is booted without acquiring a license, a segmentation fault occurs and the application core dumps. The fault occurs when the application cannot allocate a CUDA object on NVIDIA vGPUs where CUDA is disabled. On NVIDIA vGPUs that can support CUDA, CUDA is disabled in unlicensed mode.

Status
Not an NVIDIA bug.

Ref. #
200330956

5.26. A segmentation fault in DBus code causes nvidia-gridd to exit on Red Hat Enterprise Linux and CentOS

Description
On Red Hat Enterprise Linux 6.8 and 6.9, and CentOS 6.8 and 6.9, a segmentation fault in DBus code causes the nvidia-gridd service to exit.

The nvidia-gridd service uses DBus for communication with NVIDIA X Server Settings to display licensing information through the Manage License page. Disabling the GUI for licensing resolves this issue.
To prevent this issue, the GUI for licensing is disabled by default. You might encounter this issue if you have enabled the GUI for licensing and are using Red Hat Enterprise Linux 6.8 or 6.9, or CentOS 6.8 and 6.9.

Version
Red Hat Enterprise Linux 6.8 and 6.9
CentOS 6.8 and 6.9

Status
Open

Ref. #
- 200358191
- 200319854
- 1895945

5.27. No **Manage License** option available in **NVIDIA X Server Settings** by default

Description
By default, the **Manage License** option is not available in **NVIDIA X Server Settings**. This option is missing because the GUI for licensing on Linux is disabled by default to work around the issue that is described in *A segmentation fault in DBus code causes nvidia-gridd to exit on Red Hat Enterprise Linux and CentOS*.

Workaround
This workaround requires `sudo` privileges.

**Note:** Do not use this workaround with Red Hat Enterprise Linux 6.8 and 6.9 or CentOS 6.8 and 6.9. To prevent a segmentation fault in DBus code from causing the `nvidia-gridd` service from exiting, the GUI for licensing must be disabled with these OS versions.

If you are licensing a physical GPU for vCS, you **must** use the configuration file `/etc/nvidia/gridd.conf`.

1. If **NVIDIA X Server Settings** is running, shut it down.
2. If the `/etc/nvidia/gridd.conf` file does not already exist, create it by copying the supplied template file `/etc/nvidia/gridd.conf.template`.

3. As root, edit the `/etc/nvidia/gridd.conf` file to set the `EnableUI` option to `TRUE`.

4. Start the `nvidia-gridd` service.
   ```
   # sudo service nvidia-gridd start
   ```
   When **NVIDIA X Server Settings** is restarted, the **Manage License** option is now available.

**Status**

Open

### 5.28. Licenses remain checked out when VMs are forcibly powered off

**Description**

NVIDIA vGPU software licenses remain checked out on the license server when non-persistent VMs are forcibly powered off.

The NVIDIA service running in a VM returns checked out licenses when the VM is shut down. In environments where non-persistent licensed VMs are not cleanly shut down, licenses on the license server can become exhausted. For example, this issue can occur in automated test environments where VMs are frequently changing and are not guaranteed to be cleanly shut down. The licenses from such VMs remain checked out against their MAC address for seven days before they time out and become available to other VMs.

**Resolution**

If VMs are routinely being powered off without clean shutdown in your environment, you can avoid this issue by shortening the license borrow period. To shorten the license borrow period, set the `LicenseInterval` configuration setting in your VM image. For details, refer to [Virtual GPU Client Licensing User Guide](#).

**Status**

Closed

**Ref. #**

1694975
5.29. VM bug checks after the guest VM driver for Windows 10 RS2 is installed

Description

When the VM is rebooted after the guest VM driver for Windows 10 RS2 is installed, the VM bug checks. When Windows boots, it selects one of the standard supported video modes. If Windows is booted directly with a display that is driven by an NVIDIA driver, for example a vGPU on Citrix Hypervisor, a blue screen crash occurs.

This issue occurs when the screen resolution is switched from VGA mode to a resolution that is higher than 1920×1200.

Fix

Download and install Microsoft Windows Update KB4020102 from the Microsoft Update Catalog.

Workaround

If you have applied the fix, ignore this workaround.

Otherwise, you can work around this issue until you are able to apply the fix by not using resolutions higher than 1920×1200.

1. Choose a GPU profile in Citrix XenCenter that does not allow resolutions higher than 1920×1200.
2. Before rebooting the VM, set the display resolution to 1920×1200 or lower.

Status

Not an NVIDIA bug

Ref. #

200310861
5.30. **GNOME Display Manager (GDM) fails to start on Red Hat Enterprise Linux 7.2 and CentOS 7.0**

**Description**

GDM fails to start on Red Hat Enterprise Linux 7.2 and CentOS 7.0 with the following error:

```
Oh no! Something has gone wrong!
```

**Workaround**

Permanently enable permissive mode for Security Enhanced Linux (SELinux).

1. As root, edit the `/etc/selinux/config` file to set SELINUX to permissive.
   ```bash
   SELINUX=permissive
   ```
2. Reboot the system.
   ```bash
   ~]# reboot
   ```

For more information, see [Permissive Mode](#) in *Red Hat Enterprise Linux 7 SELinux User's and Administrator's Guide*.

**Status**

Not an NVIDIA bug

**Ref. #**

200167868
Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete. NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND 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. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

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, the NVIDIA logo, NVIDIA GRID, NVIDIA GRID vGPU, NVIDIA Maxwell, NVIDIA Pascal, NVIDIA Turing, NVIDIA Volta, GPU Direct, Quadro, and Tesla 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

© 2013-2020 NVIDIA Corporation. All rights reserved.