Virtual GPU Software R460 for Linux with KVM

Release Notes
Table of Contents

Chapter 1. Release Notes.................................................................................................... 1
  1.1. NVIDIA vGPU Software Driver Versions........................................................................1
  1.2. Compatibility Requirements for the NVIDIA vGPU Manager and Guest VM Driver......2
  1.3. Updates in Release 12.0.................................................................................................3

Chapter 2. Validated Platforms............................................................................................4
  2.1. Supported NVIDIA GPUs and Validated Server Platforms........................................4
  2.2. Hypervisor Software Releases.......................................................................................4
  2.3. Guest OS Support.........................................................................................................4
  2.4. NVIDIA CUDA Toolkit Version Support.......................................................................5
  2.5. Multiple vGPU Support...............................................................................................5
  2.6. Peer-to-Peer CUDA Transfers over NVLink Support....................................................7
  2.7. GPUDirect Technology Support....................................................................................9
  2.8. NVIDIA NVSwitch On-Chip Memory Fabric Support..................................................10

Chapter 3. Known Product Limitations..............................................................................11
  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...................................................................................................19
  5.1. NVIDIA A100 HGX 80GB vGPU names shown as Graphics Device by nvidia-smi..........19
  5.2. Idle Teradici Cloud Access Software session disconnects from Linux VM....................20
  5.3. Idle NVIDIA A100 GPUs show 100% GPU utilization..................................................20
  5.4. Guest VM frame buffer listed by nvidia-smi for vGPUs on GPUs that support SRIOV is incorrect............................................................................................................21
5.5. Driver upgrade in a Linux guest VM with multiple vGPUs might fail ........................................... 22
5.6. NVIDIA Control Panel fails to start if launched too soon from a VM without licensing information ........................................................................................................................................ 23
5.7. Remoting solution session freezes with VGPU message 21 failed and VGPU message 14 failed errors ........................................................................................................................................ 23
5.8. On Linux, the frame rate might drop to 1 after several minutes .................................................. 24
5.9. DWM crashes randomly occur in Windows VMs .............................................................................. 25
5.10. Publisher not verified warning during Windows 7 driver installation ............................................. 25
5.11. RAPIDS cuDF merge fails on NVIDIA vGPU ................................................................................ 26
5.12. ECC memory settings for a vGPU cannot be changed by using NVIDIA X Server Settings .............. 26
5.13. Changes to ECC memory settings for a Linux vGPU VM by nvidia-smi might be ignored .............. 26
5.14. Vulkan applications crash in Windows 7 guest VMs configured with NVIDIA vGPU ............ 28
5.15. Host core CPU utilization is higher than expected for moderate workloads .............................. 28
5.16. Frame capture while the interactive logon message is displayed returns blank screen.............. 29
5.17. RDS sessions do not use the GPU with some Microsoft Windows Server releases ..................... 30
5.18. When the scheduling policy is fixed share, GPU utilization is reported as higher than expected ........................................................................................................................................ 30
5.19. License is not acquired in Windows VMs ...................................................................................... 31
5.20. nvidia-smi reports that vGPU migration is supported on all hypervisors ................................ 32
5.21. Hot-plugging and unplugging vCPUs causes a blue-screen crash in Windows VMs ................... 33
5.22. Luxmark causes a segmentation fault on an unlicensed Linux client ........................................ 33
5.23. A segmentation fault in DBus code causes nvidia-gridd to exit on Red Hat Enterprise Linux and CentOS ........................................................................................................................................ 34
5.24. No Manage License option available in NVIDIA X Server Settings by default .......................... 34
5.25. Licenses remain checked out when VMs are forcibly powered off ............................................. 35
5.26. VM bug checks after the guest VM driver for Windows 10 RS2 is installed ................................. 36
5.27. GNOME Display Manager (GDM) fails to start on Red Hat Enterprise Linux 7.2 and CentOS 7.0 ........................................................................................................................................ 37
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>NVIDIA vGPU Software Version</th>
<th>NVIDIA Virtual GPU Manager Version</th>
<th>NVIDIA Windows Driver Version</th>
<th>NVIDIA Linux Driver Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.0</td>
<td>460.32.04</td>
<td>461.09</td>
<td>460.32.03</td>
</tr>
</tbody>
</table>

For details of which Linux with KVM releases are supported, see Hypervisor Software Releases.
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 from a later major release branch with guest VM drivers from the previous branch

**Note:**

When NVIDIA vGPU Manager is used with guest VM drivers from the previous branch, the combination supports only the features, hardware, and software (including guest OSes) that are supported on both releases.

For example, if vGPU Manager from release 12.0 is used with guest drivers from release 11.2, the combination does not support Red Hat Enterprise Linux 7.6 because NVIDIA vGPU software release 12.0 does not support Red Hat Enterprise Linux 7.6.

The following table lists the specific software releases that are compatible with the components in the NVIDIA vGPU software 12 major release branch.

<table>
<thead>
<tr>
<th>NVIDIA vGPU Software Component</th>
<th>Release</th>
<th>Compatible Software Releases</th>
</tr>
</thead>
</table>
| NVIDIA vGPU Manager           | 12.0    | ➤ Guest VM driver release 12.0  
|                               |         | ➤ All guest VM driver 11.x releases  |
| Guest VM drivers              | 12.0    | NVIDIA vGPU Manager release 12.0 |
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 12 major release branch.

<table>
<thead>
<tr>
<th>NVIDIA vGPU Software Component</th>
<th>Release</th>
<th>Incompatible Software Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA vGPU Manager</td>
<td>12.0</td>
<td>All guest VM driver releases 10.x and earlier</td>
</tr>
<tr>
<td>Guest VM drivers</td>
<td>12.0</td>
<td>All NVIDIA vGPU Manager releases 11.x and earlier</td>
</tr>
</tbody>
</table>

1.3. Updates in Release 12.0

New Features in Release 12.0

- Support for NVIDIA® NVSwitch™ on-chip memory fabric
- 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 R460 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.

2.2. Hypervisor Software Releases

NVIDIA vGPU software is supported on Linux with KVM platforms only by specific hypervisor software vendors. For information about which NVIDIA vGPU software releases and hypervisor software releases are supported, consult the documentation from your hypervisor vendor.

<table>
<thead>
<tr>
<th>Hypervisor Vendor</th>
<th>Platform</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat</td>
<td>OpenStack Platform</td>
<td>Product Documentation for Red Hat OpenStack Platform</td>
</tr>
<tr>
<td>SUSE</td>
<td>Linux Enterprise Server</td>
<td></td>
</tr>
</tbody>
</table>

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.
2.4. NVIDIA CUDA Toolkit Version Support

The releases in this release family of NVIDIA vGPU software support NVIDIA CUDA Toolkit 11.2.

For more information about NVIDIA CUDA Toolkit, see CUDA Toolkit 11.2 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>Ampere [compute workloads only]</td>
<td>NVIDIA A100 HGX 80GB</td>
<td>A1000X-80C See Note [1].</td>
</tr>
<tr>
<td></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>Ampere [compute and graphics workloads]</td>
<td>NVIDIA A40</td>
<td>A40-48Q See Note [1].</td>
</tr>
<tr>
<td></td>
<td>NVIDIA RTX A6000</td>
<td>A6000-48Q See Note [1].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A6000-48C See Note [1].</td>
</tr>
<tr>
<td>Turing</td>
<td>Tesla T4</td>
<td>T4-16Q</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 6000</td>
<td>RTX6000-24Q</td>
</tr>
<tr>
<td>GPU Architecture</td>
<td>Board</td>
<td>vGPU</td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Quadro RTX 6000 passive</td>
<td>RTX6000-24C</td>
<td></td>
</tr>
<tr>
<td>Quadro RTX 8000</td>
<td>RTX6000P-24Q</td>
<td></td>
</tr>
<tr>
<td>Quadro RTX 8000 passive</td>
<td>RTX6000P-24C</td>
<td></td>
</tr>
<tr>
<td>Quadro RTX 6000 passive</td>
<td>RTX8000-48Q</td>
<td></td>
</tr>
<tr>
<td>Quadro RTX 8000</td>
<td>RTX8000-48C</td>
<td></td>
</tr>
<tr>
<td>Quadro RTX 8000 passive</td>
<td>RTX8000P-48Q</td>
<td></td>
</tr>
<tr>
<td>Quadro RTX 6000 passive</td>
<td>RTX8000P-48C</td>
<td></td>
</tr>
<tr>
<td>Volta</td>
<td>Tesla V100 SXM2 32GB</td>
<td>V100DX-32Q</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 PCIe 32GB</td>
<td>V100D-32Q</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 PCIe 32GB</td>
<td>V100D-32C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100S PCIe 32GB</td>
<td>V100S-32Q</td>
</tr>
<tr>
<td></td>
<td>Tesla V100S PCIe 32GB</td>
<td>V100S-32C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 PCIe</td>
<td>V100-16Q</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 PCIe</td>
<td>V100-16C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 FHHL</td>
<td>V100L-16Q</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 FHHL</td>
<td>V100L-16C</td>
</tr>
<tr>
<td>Pascal</td>
<td>Tesla P100 SXM2</td>
<td>P100X-16Q</td>
</tr>
<tr>
<td></td>
<td>Tesla P100 PCIe 16GB</td>
<td>P100-16Q</td>
</tr>
<tr>
<td></td>
<td>Tesla P100 PCIe 12GB</td>
<td>P100C-12Q</td>
</tr>
<tr>
<td></td>
<td>Tesla P40</td>
<td>P40-24Q</td>
</tr>
<tr>
<td></td>
<td>Tesla P6</td>
<td>P6-16Q</td>
</tr>
<tr>
<td></td>
<td>Tesla P6</td>
<td>P6-16C</td>
</tr>
<tr>
<td></td>
<td>Tesla P4</td>
<td>P4-8Q</td>
</tr>
<tr>
<td></td>
<td>Tesla P4</td>
<td>P4-8C</td>
</tr>
<tr>
<td>Maxwell</td>
<td>Tesla M60</td>
<td>M60-8Q</td>
</tr>
</tbody>
</table>
Validated Platforms

<table>
<thead>
<tr>
<th>GPU Architecture</th>
<th>Board</th>
<th>vGPU</th>
</tr>
</thead>
<tbody>
<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>Ampere [compute workloads only]</td>
<td>NVIDIA A100 HGX 80GB</td>
<td>A100DX-80C See Note [1].</td>
</tr>
<tr>
<td></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>Ampere [compute and graphics workloads]</td>
<td>NVIDIA A40</td>
<td>A40-48Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A40-48C</td>
</tr>
<tr>
<td></td>
<td>NVIDIA RTX A6000</td>
<td>A6000-48Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A6000-48C</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>GPU Architecture</td>
<td>Board</td>
<td>vGPU</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td>---------------------</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 following hardware:
   - NVIDIA HGX™ A100 4-GPU baseboard with four fully connected GPUs
   - NVIDIA HGX A100 8-GPU baseboards with eight 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**

- NVIDIA NVSwitch is supported only on the hardware platforms, vGPUs, and Linux with KVM releases listed in [NVIDIA NVSwitch On-Chip Memory Fabric Support](#). Otherwise, only direct connections are supported.
- Only time-sliced vGPUs are supported. MIG-backed vGPUs are **not** supported.
- PCIe is supported **only** with the NVIDIA A100 PCIe 40GB GPU. Otherwise, PCIe is not supported.
- SLI is not supported.
2.7. 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 (time-sliced and MIG-backed vGPUs)</td>
<td>NVIDIA A100 HGX 80GB</td>
<td>A100DX-80C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A100DX-7-80C</td>
</tr>
<tr>
<td></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></td>
<td>NVIDIA A100 HGX 40GB</td>
<td>A100X-40C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A100X-7-40C</td>
</tr>
<tr>
<td>Ampere (time-sliced vGPUs only)</td>
<td>NVIDIA A40</td>
<td>A40-48C</td>
</tr>
<tr>
<td></td>
<td>NVIDIA RTX A6000</td>
<td>A6000-48C</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.
2.8. NVIDIA NVSwitch On-Chip Memory Fabric Support

NVIDIA® NVSwitch™ on-chip memory fabric enables peer-to-peer vGPU communication within a single node over the NVLink fabric. NVSwitch on-chip memory fabric is supported only on a subset of hardware platforms, vGPUs, Linux with KVM releases, and guest OS releases.

For information about how to use the NVSwitch on-chip memory fabric, see Fabric Manager for NVIDIA NVSwitch Systems User Guide (PDF).

Supported Hardware Platforms

- NVIDIA HGX A100 8-GPU baseboard

Supported vGPUs

Only C-series time-sliced vGPUs that are allocated all of the physical GPU’s frame buffer on NVIDIA A100 HGX physical GPUs 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 HGX 80GB</td>
<td>A100DX-80C</td>
</tr>
<tr>
<td></td>
<td>NVIDIA A100 HGX 40GB</td>
<td>A100X-40C</td>
</tr>
</tbody>
</table>

Supported Hypervisor Releases

For information about which hypervisor software releases support NVIDIA NVSwitch on-chip memory fabric, consult the documentation from your hypervisor vendor.

Supported Guest OS Releases

Linux only. NVIDIA NVSwitch on-chip memory fabric is not supported on Windows.

Limitations

- Only time-sliced vGPUs are supported. MIG-backed vGPUs are not supported.
- PCIe is not supported.
- SLI is not supported.
- All vGPUs that are communicating peer-to-peer must be assigned to the same VM.
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: 0x1a
Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0): 0xc1d004a1, 0xff0e0000, 0xff0400fb, 0xc36f,
Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0):         0x1, 0xff1fe314, 0xff1fe038, 0x100b6f000, 0x1000,
Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0):         0x80000000, 0xff0e0200, 0x0, 0x0, (Not logged),
Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0):         0x1, 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} is 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.
Known Product Limitations

- If ECC is enabled on a physical GPU that does not have HBM2 memory, ecc-adjustments is \( fb\-without\-ecc/16 \), which is equivalent to 64 Mbytes for every Gbyte of frame buffer assigned to the vGPU. \( fb\-without\-ecc \) is total amount of frame buffer with ECC disabled.
- If ECC is disabled or the GPU has HBM2 memory, ecc-adjustments is 0.

**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, page-retirement-allocation = \( 4\div \max\-vgpus\-per\-gpu \).
- On GPUs based on NVIDIA GPU architectures after the Maxwell architecture, page-retirement-allocation = \( 128\div \max\-vgpus\-per\-gpu \).

**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, \( \max\-vgpus\-per\-gpu \) is 1.

**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

**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.

**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:
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 R460 for Linux with KVM

RN-09065-001 _v12.0 Revision 03   |   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 12.0

No resolved issues are reported in this release for Linux with KVM.
Chapter 5. Known Issues

5.1. NVIDIA A100 HGX 80GB vGPU names shown as Graphics Device by nvidia-smi

Description

The names of vGPUs that reside on the NVIDIA A100 80GB GPU are incorrectly shown as Graphics Device by the nvidia-smi command. The correct names indicate the vGPU type, for example, A100DX-40C.

```
$ nvidia-smi
Mon Jan 25 02:52:57 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.04    Driver Version: 460.32.04    CUDA Version: 11.2    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M.  |
|===============================+======================+======================|
|   0  Graphics Device On   | 00000000:07:00.0 Off |                    0 |
| N/A  N/A    P0    N/A /  N/A | 6053MiB / 81915MiB |      0%      Default |
|                               |                      |             Disabled |
|   1  Graphics Device On   | 00000000:08:00.0 Off |                    0 |
| N/A  N/A    P0    N/A /  N/A | 6053MiB / 81915MiB |      0%      Default |
|                               |                      |             Disabled |
+-----------------------------------------------------------------------------+

Processes:

<table>
<thead>
<tr>
<th>GPU</th>
<th>GI</th>
<th>CI</th>
<th>PID</th>
<th>Type</th>
<th>Process name</th>
<th>GPU Memory Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ID</td>
<td>ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No running processes found
```

Status

Open
5.2. Idle Teradici Cloud Access Software session disconnects from Linux VM

Description
After a Teradici Cloud Access Software session has been idle for a short period of time, the session disconnects from the VM. When this issue occurs, the error messages NVOS status 0x19 and vGPU Message 21 failed are written to the log files on the hypervisor host. This issue affects only Linux guest VMs.

Status
Open

Ref. #
200689126

5.3. Idle NVIDIA A100 GPUs show 100% GPU utilization

Description
The `nvidia-smi` command shows 100% GPU utilization for NVIDIA A100 GPUs even if no vGPUs have been configured or no VMs are running. This issue affects only NVIDIA A100 GPUs on which the `sriov-manage` script has not been run to enable the virtual function for the GPU in the sysfs file system.

```
[root@host ~]# nvidia-smi
Fri Jan 22 11:45:28 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.04  Driver Version: 460.32.03  CUDA Version: 11.2       |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100-PCIE-40GB      On   | 00000000:5E:00.0 Off |                     0 |
| N/A   50C    P0    97W / 250W |      0MiB / 40537MiB |          100%  Default |
|                               |                      |            Disabled |
+-------------------------------+----------------------+----------------------+
```


Known Issues

Virtual GPU Software R460 for Linux with KVM

RN-09065-001 _v12.0 Revision 03   |   21

| Processes: | GPU   GI   CI        PID   Type   Process name | GPU Memory | Usage |
|------------|---------|---------------------|------------|-------|
|            | GPU ID  | ID                  | PID ID     | Type   | Process name                  | GPU Memory | Usage |
| No running processes found | | | | | | | |

Workaround

Run the `sriov-manage` script to enable the virtual function for the GPU in the `sysfs` file system as explained in Virtual GPU Software User Guide.

After the script has been run, the `nvidia-smi` command shows 0% GPU utilization for idle NVIDIA A100 GPUs.

```
root@host ~]# nvidia-smi
Fri Jan 22 11:47:38 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.04   Driver Version: 460.32.03  CUDA Version:  11.2       |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap| Memory-Usage | GPU-Util  Compute M. | MIG M. |
|                               |                      |               MIG M. |                  |
+-----------------------------+----------------------+
|   0  A100-PCIE-40GB      On   | 00000000:5E:00.0 Off |                    0 |
| N/A   50C    P0    97W / 250W |      0MiB / 40537MiB |          0% Default |
|                               |                      |             Disabled |                  |
+-----------------------------+

Processes:

| Processes: | GPU   GI   CI        PID   Type   Process name | GPU Memory | Usage |
|------------|---------|---------------------|------------|-------|
|            | GPU ID  | ID                  | PID ID     | Type   | Process name                  | GPU Memory | Usage |
| No running processes found | | | | | | | |

Status

Open

Ref. #

200605527

5.4. 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

### 5.5. 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
5.6. **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

5.7. **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
5.8. **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. #**

200627445
5.9. 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

5.10. 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

5.11. 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

5.12. 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

Ref. #

200523086

5.13. 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_RegistryDwordsPerDevice` might cause the kernel module option `RMGuestECCState` to be ignored.

Workaround

This workaround requires administrator privileges.

1. Move the entry containing the keyword `NVreg_RegistryDwordsPerDevice` from the other configuration file to `/etc/modprobe.d/nvidia.conf`.
2. Reboot the VM.
5.14. **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. #**

200505777

5.15. **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

Status

Open

Ref. #

2414897

5.16. 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
5.17. 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.

5.18. 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
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 390.42                 Driver Version: 390.42                    |
|-------------------------------+--------------------------------+------------|
| GPU  Name                     | Bus-Id                         | GPU-Util   |
|      vGPU ID    Name          | VM ID    VM Name               | vGPU-Util  |
|===============================+================================+============|
|   0  Tesla P40                | 00000000:81:00.0               |  99%       |
| 85109      GRID P40-4Q   | 85110    win7-xmpl-146048-1    |     32%   |
| 87195      GRID P40-4Q   | 87196    win7-xmpl-146048-2    |     39%   |
| 88095      GRID P40-4Q   | 88096    win7-xmpl-146048-3    |     26%   |
| 90475      GRID P40-4Q   | 90476    win7-xmpl-146048-4    |      0%   |
| 93363      GRID P40-4Q   | 93364    win7-xmpl-146048-6    |      0%   |
|   1  Tesla P40                | 00000000:85:00.0               |   0%       |
+-------------------------------+--------------------------------+------------+
```

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

### 5.19. 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:

```
[0000000020.860152600 sec] - [Logging.lib] ERROR: [nvGridLicensing.FlexUtility] 3538FlexUtility::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.

**Status**

Closed

**Ref. #**

200407287

---

**5.20. 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
5.21. **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:

- SYSTEMSERVICE_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.

**Status**

Closed

**Ref. #**

2101499

---

5.22. **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
5.23. 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

5.24. 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

5.25. 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

5.26. 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.
Known Issues

Status
Not an NVIDIA bug

Ref. #
200310861

5.27. 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.

   SELINUX=permissive

2. Reboot the system.

   ~]# 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