TABLE OF CONTENTS

Chapter 1. Release Notes........................................................................................................... 1
  1.1. Updates in Release 9.0..................................................................................................... 2
  1.2. Updates in Release 9.1..................................................................................................... 2
  1.3. Updates in Release 9.2..................................................................................................... 2

Chapter 2. Validated Platforms................................................................................................. 4
  2.1. Supported NVIDIA GPUs and Validated Server Platforms.............................................. 4
  2.2. Hypervisor Software Releases.......................................................................................... 5
  2.3. Guest OS Support............................................................................................................. 6
      2.3.1. Windows Guest OS Support....................................................................................... 6
      2.3.2. Linux Guest OS Support........................................................................................... 7
  2.4. NVIDIA CUDA Toolkit Version Support........................................................................... 8
  2.5. Multiple vGPU Support................................................................................................... 9
  2.6. Peer-to-Peer CUDA Transfers over NVLink Support......................................................... 10

Chapter 3. Known Product Limitations..................................................................................... 12
  3.1. Total frame buffer for vGPUs is less than the total frame buffer on the physical GPU...... 12
  3.2. Issues may occur with graphics-intensive OpenCL applications on vGPU types with limited frame buffer.............................................................................................. 13
  3.3. In pass through mode, all GPUs connected to each other through NVLink must be assigned to the same VM................................................................................................. 13
  3.4. vGPU profiles with 512 Mbytes or less of frame buffer support only 1 virtual display head on Windows 10................................................................................................................. 14
  3.5. NVENC requires at least 1 Gbyte of frame buffer................................................................. 14
  3.6. VM running older NVIDIA vGPU drivers fails to initialize vGPU when booted.................. 15
  3.7. Single vGPU benchmark scores are lower than pass-through GPU.................................. 15
  3.8. 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. 9.0, 9.1 Only: ECC memory with NVIDIA vGPU is not supported on Tesla M60 and Tesla M6 20
  5.2. 9.0, 9.1 Only: Virtual GPU fails to start if ECC is enabled.................................................. 20
  5.3. 9.0 only: Hypervisor host hangs when persistence mode is set......................................... 22
  5.4. RAPIDS cuDF merge fails on NVIDIA vGPU................................................................. 23
  5.5. 9.0 only: Incorrect ECC error counts are reported for vGPUs on some GPUs................. 23
  5.6. ECC memory settings for a vGPU cannot be changed by using NVIDIA X Server Settings... 24
  5.7. Changes to ECC memory settings for a Linux vGPU VM by nvidia-smi might be ignored..... 25
  5.8. 9.0 only: VM crashes after the volatile ECC error count is reset........................................ 25
  5.9. 9.0 only: No vComputeServer option available in NVIDIA X Server Settings.................. 26
  5.10. 9.0 only: On Linux VMs, the license directory is not deleted when the guest driver is uninstalled......................................................................................................................... 28
  5.11. Vulkan applications crash in Windows 7 guest VMs configured with NVIDIA vGPU........ 28
  5.12. Host core CPU utilization is higher than expected for moderate workloads.................. 29
5.13. Frame capture while the interactive logon message is displayed returns blank screen…29
5.14. RDS sessions do not use the GPU with some Microsoft Windows Server releases........... 30
5.15. Even when the scheduling policy is equal share, unequal GPU utilization is reported…… 31
5.16. When the scheduling policy is fixed share, GPU utilization is reported as higher than expected..........................................................................................................................32
5.17. License is not acquired in Windows VMs........................................................................33
5.18. nvidia-smi reports that vGPU migration is supported on all hypervisors......................33
5.19. Hot plugging and unplugging vCPUs causes a blue-screen crash in Windows VMs........34
5.20. Luxmark causes a segmentation fault on an unlicensed Linux client..........................34
5.21. Resolution is not updated after a VM acquires a license and is restarted.....................35
5.22. A segmentation fault in DBus code causes nvidia-gridd to exit on Red Hat Enterprise Linux and CentOS.................................................................35
5.23. No Manage License option available in NVIDIA X Server Settings by default..............36
5.24. Licenses remain checked out when VMs are forcibly powered off.............................37
5.25. VM bug checks after the guest VM driver for Windows 10 RS2 is installed..................38
5.26. GNOME Display Manager (GDM) fails to start on Red Hat Enterprise Linux 7.2 and CentOS 7.0..................................................................................................................38
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 Red Hat Enterprise Linux with KVM.

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

The releases in this release family of NVIDIA vGPU software include the software listed in the following table:

<table>
<thead>
<tr>
<th>Software</th>
<th>9.0</th>
<th>9.1</th>
<th>9.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA Virtual GPU Manager for the Red Hat Enterprise Linux with KVM releases listed in Hypervisor Software Releases</td>
<td>430.27</td>
<td>430.46</td>
<td>430.67</td>
</tr>
<tr>
<td>NVIDIA Windows driver</td>
<td>431.02</td>
<td>431.79</td>
<td>432.08</td>
</tr>
<tr>
<td>NVIDIA Linux driver</td>
<td>430.30</td>
<td>430.46</td>
<td>430.63</td>
</tr>
</tbody>
</table>

Caution

If you install the wrong NVIDIA vGPU software packages for the version of Red Hat Enterprise Linux with KVM you are using, NVIDIA Virtual GPU Manager will fail to load.

The releases of the vGPU Manager and guest VM drivers that you install must be compatible. Different versions of the vGPU Manager and guest VM driver from within the same main release branch can be used together. For example, you can use the vGPU Manager from release 9.1 with guest VM drivers from release 9.0. However, versions of the vGPU Manager and guest VM driver from different main release branches cannot be used together. For example, you cannot use the vGPU Manager from release 9.1 with guest VM drivers from release 7.2.
See VM running older NVIDIA vGPU drivers fails to initialize vGPU when booted.

This requirement does not apply to the NVIDIA vGPU software license sever. All releases of NVIDIA vGPU software are compatible with all releases of the license server.

1.1. Updates in Release 9.0

New Features in Release 9.0

- NVIDIA vComputeServer vGPUs for artificial intelligence, deep learning, and high-performance computing workloads
- Error correcting code (ECC) memory support
- Page retirement support
- Peer-to-peer CUDA transfers over NVLink support
- Configurable times slices for equal share schedulers and fixed share schedulers
- New configuration parameter to specify host ID of a licensed client
- Miscellaneous bug fixes

Hardware and Software Support Introduced in Release 9.0

- Support for Windows 10 May 2019 Update (1903) as a guest OS

1.2. Updates in Release 9.1

New Features in Release 9.1

- Support for NVIDIA vComputeServer vGPUs on the following GPUs:
  - Quadro RTX 6000
  - Quadro RTX 8000
- Security updates
- Miscellaneous bug fixes

1.3. Updates in Release 9.2

New Features in Release 9.2

- Miscellaneous bug fixes
- Security updates
- Limitation on the maximum number of NVIDIA vComputeServer vGPUs to eight vGPUs per physical GPU, irrespective of the available hardware resources of the physical GPU
Inclusion of two Virtual GPU Manager RPM packages for Red Hat Enterprise Linux with KVM 8.0:

- One package for the GA Linux KVM kernel
- One package for the updated z-stream kernel
Chapter 2.
VALIDATED PLATFORMS

This release family of NVIDIA vGPU software provides support for several NVIDIA GPUs on validated server hardware platforms, Red Hat Enterprise Linux with KVM hypervisor software versions, and guest operating systems. It also supports the version of NVIDIA CUDA Toolkit that is compatible with R430 drivers.

2.1. Supported NVIDIA GPUs and Validated Server Platforms

This release of NVIDIA vGPU software provides support for the following NVIDIA GPUs on Red Hat Enterprise Linux with KVM, running on validated server hardware platforms:

- GPUs based on the NVIDIA Maxwell™ graphic architecture:
  - Tesla P4
  - Tesla P6
  - Tesla P40
  - Tesla P100 PCIe 16 GB
  - Tesla P100 SXM2 16 GB
  - Tesla P100 PCIe 12GB
- GPUs based on the NVIDIA Pascal™ architecture:
  - Tesla P100 SXM2
  - Tesla P100 SXM2 32GB
  - Tesla V100 PCIe
  - Tesla V100 PCIe 32GB
- ...
Validated Platforms

- Tesla V100 FHHL
- GPUs based on the NVIDIA Turing™ architecture:
  - Tesla T4
  - Quadro RTX 6000 in displayless mode (GRID Virtual PC and GRID Virtual Applications are **not** supported. vComputeServer is supported **only** since release 9.1.)
  - Quadro RTX 8000 in displayless mode (GRID Virtual PC and GRID Virtual Applications are **not** supported. vComputeServer is supported **only** since release 9.1.)

In displayless mode, local physical display connectors are disabled.

For a list of validated server platforms, refer to NVIDIA GRID Certified Servers.

---

**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 vComputeServer vGPU types.

---

### 2.2. Hypervisor Software Releases

This release supports **only** the hypervisor software release listed in the table.

---

**If a specific release, even an update release, is not listed, it’s **not** supported.**

<table>
<thead>
<tr>
<th>Software</th>
<th>Releases Supported</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux with KVM</td>
<td>8.0</td>
<td>All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux with KVM</td>
<td>7.6, 7.5</td>
<td>All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux with KVM</td>
<td>7.2 through 7.4</td>
<td>All NVIDIA GPUs that NVIDIA vGPU software supports are supported in pass-through mode only.</td>
</tr>
</tbody>
</table>
## 2.3. Guest OS Support

NVIDIA vGPU software supports several Windows releases and Linux distributions as a guest OS. The supported guest operating systems depend on the hypervisor software version.

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.

### 2.3.1. Windows Guest OS Support

Red Hat Enterprise Linux with KVM and Red Hat Virtualization (RHV) support Windows guest operating systems only under specific Red Hat subscription programs. For details, see:

- Certified guest operating systems for Red Hat Enterprise Linux with KVM
- Certified Guest Operating Systems in Red Hat OpenStack Platform and Red Hat Enterprise Virtualization

NVIDIA vGPU software supports only the 64-bit Windows releases listed in the table as a guest OS on Red Hat Enterprise Linux with KVM. The releases of Red Hat Enterprise Linux with KVM for which a Windows release is supported depend on whether NVIDIA vGPU or pass-through GPU is used.

If a specific release, even an update release, is not listed, it’s not supported.
<table>
<thead>
<tr>
<th>Guest OS</th>
<th>NVIDIA vGPU - Red Hat Enterprise Linux with KVM Releases</th>
<th>Pass-Through GPU - Red Hat Enterprise Linux with KVM Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2019</td>
<td>RHEL KVM 8.0, 7.6, 7.5 RHV 4.3, 4.2</td>
<td>RHEL KVM 8.0, 7.6, 7.5, 7.4, 7.3, 7.2, 7.1, 7.0 RHV 4.3, 4.2, 4.1</td>
</tr>
<tr>
<td>Windows Server 2016 1709, 1607</td>
<td>RHEL KVM 8.0, 7.6, 7.5 RHV 4.3, 4.2</td>
<td>RHEL KVM 8.0, 7.6, 7.5, 7.4, 7.3, 7.2, 7.1, 7.0 RHV 4.3, 4.2, 4.1</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>RHEL KVM 8.0, 7.6, 7.5 RHV 4.3, 4.2</td>
<td>RHEL KVM 8.0, 7.6, 7.5, 7.4, 7.3, 7.2, 7.1, 7.0 RHV 4.3, 4.2, 4.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>RHEL KVM 8.0, 7.6, 7.5 RHV 4.3, 4.2</td>
<td>RHEL KVM 8.0, 7.6, 7.5, 7.4, 7.3, 7.2, 7.1, 7.0 RHV 4.3, 4.2, 4.1 Supported only on GPUs based on the Maxwell architecture</td>
</tr>
<tr>
<td>Windows 10:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>‣ May 2019 Update (1903)</td>
<td>RHV 4.3, 4.2</td>
<td>RHV 4.3, 4.2, 4.1</td>
</tr>
<tr>
<td>‣ October 2018 Update (1809)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>‣ Spring Creators Update (1803)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>‣ Fall Creators Update (1709)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>‣ Creators Update (1703)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>‣ Anniversary Update (1607)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>‣ November Update (1511)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>‣ RTM (1507)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows 8.1 Update</td>
<td>RHV 4.3, 4.2</td>
<td>RHV 4.3, 4.2, 4.1</td>
</tr>
<tr>
<td>Windows 7</td>
<td>RHV 4.3, 4.2</td>
<td>RHV 4.3, 4.2, 4.1                                      Supported only on GPUs based on the Maxwell architecture</td>
</tr>
</tbody>
</table>

### 2.3.2. Linux Guest OS Support

NVIDIA vGPU software supports **only** the 64-bit Linux distributions listed in the table as a guest OS on Red Hat Enterprise Linux with KVM. The releases of Red Hat Enterprise Linux with KVM for which a Linux release is supported depend on whether NVIDIA vGPU or pass-through GPU is used.

> **If a specific release, even an update release, is not listed, it's not supported.**
2.4. NVIDIA CUDA Toolkit Version Support

The releases in this release family of NVIDIA vGPU software support NVIDIA CUDA Toolkit 10.1 Update 1.

For more information about NVIDIA CUDA Toolkit, see CUDA Toolkit 10.1 Documentation.

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.
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 Red Hat Enterprise Linux with KVM releases.

Supported vGPUs

Only Q-series and C-series vGPUs that are allocated all of the physical GPU’s frame buffer are supported.

<table>
<thead>
<tr>
<th>GPU Architecture</th>
<th>Board</th>
<th>vGPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></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>Quadro RTX 8000</td>
<td>RTX8000-48Q</td>
</tr>
<tr>
<td>Volta</td>
<td></td>
<td></td>
</tr>
<tr>
<td></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 V100 SXM2</td>
<td>V100X-16Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V100X-16C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 PCIe</td>
<td>V100-16Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V100-16C</td>
</tr>
<tr>
<td></td>
<td>Tesla V100 FHHL</td>
<td>V100L-16Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V100L-16C</td>
</tr>
<tr>
<td>Pascal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tesla P100 SXM2</td>
<td>P100X-16Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P100X-16C</td>
</tr>
<tr>
<td></td>
<td>Tesla P100 PCIe 16GB</td>
<td>P100-16Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P100-16C</td>
</tr>
<tr>
<td></td>
<td>Tesla P100 PCIe 12GB</td>
<td>P100C-12Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P100C-12C</td>
</tr>
</tbody>
</table>
### GPU Architecture

<table>
<thead>
<tr>
<th>Board</th>
<th>vGPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tesla P40</td>
<td>P40-24Q</td>
</tr>
<tr>
<td></td>
<td>P40-24C</td>
</tr>
<tr>
<td>Tesla P6</td>
<td>P6-16Q</td>
</tr>
<tr>
<td></td>
<td>P6-16C</td>
</tr>
<tr>
<td>Tesla P4</td>
<td>P4-8Q</td>
</tr>
<tr>
<td></td>
<td>P4-8C</td>
</tr>
<tr>
<td>Maxwell</td>
<td></td>
</tr>
<tr>
<td>Tesla M60</td>
<td>M60-8Q</td>
</tr>
<tr>
<td>Tesla M10</td>
<td>M10-8Q</td>
</tr>
<tr>
<td>Tesla M6</td>
<td>M6-8Q</td>
</tr>
</tbody>
</table>

### Maximum vGPUs per VM

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

### Supported Hypervisor Releases

Red Hat Enterprise Linux with KVM 8.0, 7.6, and 7.5 only.

RHV 4.3 and 4.2 only.

### 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, Red Hat Enterprise Linux with KVM releases, and guest OS releases.

### Supported vGPUs

Only Q-series and C-series 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>Turing</td>
<td>Quadro RTX 6000</td>
<td>RTX6000-24Q</td>
</tr>
<tr>
<td></td>
<td>Quadro RTX 8000</td>
<td>RTX8000-48Q</td>
</tr>
</tbody>
</table>
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.
- PCIe is not supported.
- SLI 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. 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.

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

\[
\text{reserved-fb} = \frac{\text{vgpu-profile-size-in-mb}}{16 + 64 + \text{ecc-adjustments}}
\]

- \(\text{reserved-fb}\): The total amount of reserved frame buffer in Mbytes that is not available for vGPUs.
- \(\text{vgpu-profile-size-in-mb}\): 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.
- \(\text{ecc-adjustments}\): 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 disabled or the GPU has HBM2 memory, \(\text{ecc-adjustments}\) is 0.
3.2. 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.

3.3. 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.
3.4. 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.

3.5. 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.

3.6. VM running older NVIDIA vGPU drivers fails to initialize vGPU when booted

Description

A VM running a version of the NVIDIA guest VM drivers from a previous main release branch, for example release 4.4, will fail to initialize vGPU when booted on a Red Hat Enterprise Linux with KVM platform running the current release of Virtual GPU Manager.

In this scenario, 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)

Depending on the versions of drivers in use, the Red Hat Enterprise Linux with KVM VM’s /var/log/messages log file reports one of the following errors:

- An error message:
  ```
  vmiop_log: error: Unable to fetch Guest NVIDIA driver information
  ```
- A version mismatch between guest and host drivers:
  ```
  vmiop_log: error: Guest VGX version(1.1) and Host VGX version(1.2) do not match
  ```
- A signature mismatch:
  ```
  vmiop_log: error: VGPU message signature mismatch.
  ```

Resolution

Install the current NVIDIA guest VM driver in the VM.

3.7. 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:

```
# 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
```

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
```
3.8. *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 Red Hat Enterprise 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
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 9.0

<table>
<thead>
<tr>
<th>Bug ID</th>
<th>Summary and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Virtual GPU fails to start if ECC is enabled</td>
</tr>
<tr>
<td></td>
<td>NVIDIA vGPU does not support error correcting code (ECC) memory. If ECC memory is enabled, NVIDIA vGPU fails to start.</td>
</tr>
<tr>
<td></td>
<td>Starting with NVIDIA vGPU software release 9.0, NVIDIA vGPU supports ECC memory on GPUs and hypervisor software versions that support ECC.</td>
</tr>
</tbody>
</table>

### Issues Resolved in Release 9.1

<table>
<thead>
<tr>
<th>Bug ID</th>
<th>Summary and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200534988</td>
<td>Error XID 47 followed by multiple XID 32 errors</td>
</tr>
<tr>
<td></td>
<td>After disconnecting Citrix Virtual Apps and Desktops and clicking the power button in the VM, error XID 47 occurs followed by multiple XID 32 errors. When these errors occur, the hypervisor host becomes unusable.</td>
</tr>
<tr>
<td>200537078</td>
<td>9.0 only: Hypervisor host hangs when persistence mode is set</td>
</tr>
<tr>
<td></td>
<td>When persistence mode is set through the <code>nvidia-smi</code> command, a very large number of <code>nv_queue</code> worker threads is spawned, causing the hypervisor host to hang.</td>
</tr>
<tr>
<td>200526633</td>
<td>9.0 only: VM crashes after the volatile ECC error count is reset</td>
</tr>
</tbody>
</table>
### Resolved Issues

<table>
<thead>
<tr>
<th>Bug ID</th>
<th>Summary and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>After the command <code>nvidia-smi -p 0</code> is run from a guest VM to reset the volatile ECC error count, the VM crashes.</td>
</tr>
<tr>
<td>200525006</td>
<td>9.0 only: Incorrect ECC error counts are reported for vGPUs on some GPUs</td>
</tr>
<tr>
<td></td>
<td>Incorrect ECC error counts are reported for vGPUs on some GPUs when the command <code>nvidia-smi -q</code> is run from a guest VM.</td>
</tr>
<tr>
<td>200524555</td>
<td>9.0 only: On Linux VMs, the license directory is not deleted when the guest driver is uninstalled</td>
</tr>
<tr>
<td></td>
<td>On Linux guest VMs, the license directory <code>/etc/nvidia/license</code> is not deleted when the NVIDIA vGPU software graphics driver is uninstalled.</td>
</tr>
<tr>
<td>200522255</td>
<td>9.0 only: No vComputeServer option available in NVIDIA X Server Settings</td>
</tr>
<tr>
<td></td>
<td>The <code>vComputeServer</code> option is missing from the Manage License section in the NVIDIA X Server Settings window.</td>
</tr>
</tbody>
</table>

### Issues Resolved in Release 9.2

<table>
<thead>
<tr>
<th>Bug ID</th>
<th>Summary and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.0, 9.1 Only: ECC memory with NVIDIA vGPU is not supported on Tesla M60 and Tesla M6</td>
</tr>
<tr>
<td></td>
<td>Error-correcting code (ECC) memory with NVIDIA vGPU is not supported on Tesla M60 and Tesla M6 GPUs. The effect of starting NVIDIA vGPU when it is configured on a Tesla M60 or Tesla M6 GPU on which ECC memory is enabled depends on your NVIDIA vGPU software release.</td>
</tr>
<tr>
<td></td>
<td>‣ 9.0 only: The hypervisor host fails.</td>
</tr>
<tr>
<td></td>
<td>‣ 9.1 only: The VM fails to start.</td>
</tr>
<tr>
<td></td>
<td>9.0, 9.1 Only: Virtual GPU fails to start if ECC is enabled</td>
</tr>
<tr>
<td></td>
<td>NVIDIA vGPU does not support ECC memory with the following GPUs:</td>
</tr>
<tr>
<td></td>
<td>‣ Tesla M60 GPUs</td>
</tr>
<tr>
<td></td>
<td>‣ Tesla M6 GPUs</td>
</tr>
<tr>
<td></td>
<td>If ECC memory is enabled and your GPU does not support ECC, NVIDIA vGPU fails to start.</td>
</tr>
</tbody>
</table>
Chapter 5.
KNOWN ISSUES

5.1. 9.0, 9.1 Only: ECC memory with NVIDIA vGPU is not supported on Tesla M60 and Tesla M6

Description

Error-correcting code (ECC) memory with NVIDIA vGPU is not supported on Tesla M60 and Tesla M6 GPUs. The effect of starting NVIDIA vGPU when it is configured on a Tesla M60 or Tesla M6 GPU on which ECC memory is enabled depends on your NVIDIA vGPU software release.

- 9.0 only: The hypervisor host fails.
- 9.1 only: The VM fails to start.

Workaround

Ensure that ECC memory is disabled on Tesla M60 and Tesla M6 GPUs. For more information, see 9.0, 9.1 Only: Virtual GPU fails to start if ECC is enabled.

Status

Resolved in NVIDIA vGPU software 9.2

5.2. 9.0, 9.1 Only: Virtual GPU fails to start if ECC is enabled

Description

Tesla M60, Tesla M6, and GPUs based on the Pascal GPU architecture, for example Tesla P100 or Tesla P4, support error correcting code (ECC) memory for improved data
integrity. Tesla M60 and M6 GPUs in graphics mode are supplied with ECC memory disabled by default, but it may subsequently be enabled using `nvidia-smi`. GPUs based on the Pascal GPU architecture are supplied with ECC memory enabled.

NVIDIA vGPU does not support ECC memory with the following GPUs:

- Tesla M60 GPUs
- Tesla M6 GPUs

If ECC memory is enabled and your GPU does not support ECC, NVIDIA vGPU fails to start.

The following error is logged in the Red Hat Enterprise Linux with KVM host’s `/var/log/messages` log file:

```
vmiop_log: error: Initialization: VGX not supported with ECC Enabled.
```

**Workaround**

If you are using Tesla M60 or Tesla M6 GPUs, ensure that ECC is disabled on all GPUs. Before you begin, ensure that NVIDIA Virtual GPU Manager is installed on your hypervisor.

1. Use `nvidia-smi` to list the status of all GPUs, and check for ECC noted as enabled on GPUs.

   ```
   # nvidia-smi -q
   ================NVSMI LOG===============
   Timestamp : Tue Dec 19 18:36:45 2017
   Driver Version : 384.99
   Attached GPUs : 1
   GPU 0000:02:00.0
   [...]
   Ecc Mode
   Current : Enabled
   Pending : Enabled
   [...]
   ```

2. Change the ECC status to off on each GPU for which ECC is enabled.

   - If you want to change the ECC status to off for all GPUs on your host machine, run this command:

     ```
     # nvidia-smi -e 0
     ```

   - If you want to change the ECC status to off for a specific GPU, run this command:

     ```
     # nvidia-smi -i id -e 0
     
     id is the index of the GPU as reported by `nvidia-smi`.```
This example disables ECC for the GPU with index 0000:02:00.0.

```
# nvidia-smi -i 0000:02:00.0 -e 0
```

3. Reboot the host.

4. Confirm that ECC is now disabled for the GPU.

```
# nvidia-smi -q
```

```
==============NVSMI LOG==============
Timestamp                           : Tue Dec 19 18:37:53 2017
Driver Version                      : 384.99
Attached GPUs                       : 1
GPU 0000:02:00.0
[...]
  Ecc Mode
  Current                     : Disabled
  Pending                     : Disabled
[...]
```

If you later need to enable ECC on your GPUs, run one of the following commands:

- If you want to change the ECC status to on for all GPUs on your host machine, run this command:

  ```
  # nvidia-smi -e 1
  ```

- If you want to change the ECC status to on for a specific GPU, run this command:

  ```
  # nvidia-smi -i id -e 1
  ```

  *id* is the index of the GPU as reported by nvidia-smi.

This example enables ECC for the GPU with index 0000:02:00.0.

```
# nvidia-smi -i 0000:02:00.0 -e 1
```

After changing the ECC status to on, reboot the host.

**Status**

Resolved in NVIDIA vGPU software 9.2

5.3. 9.0 only: Hypervisor host hangs when persistence mode is set

**Description**

When persistence mode is set through the nvidia-smi command, a very large number of nv_queue worker threads is spawned, causing the hypervisor host to hang.
When this issue occurs, the hypervisor host creates a kernel stack trace similar to the following example.

Jun 24 04:42:16 example-02 kernel: kernel BUG at /build/linux-Ue9GXV/linux-4.15.0/drivers/pci/msi.c:352!
...  
Jun 24 04:42:16 example-02 kernel: Call Trace:
Jun 24 04:42:16 example-02 kernel:  pci_disable_msi+0x100/0x120 
Jun 24 04:42:16 example-02 kernel:  nv_open_device+0x411/0x8e0 [nvidia] 
Jun 24 04:42:16 example-02 kernel:  ? kmem_cache_alloc+0x158/0x1b0 
Jun 24 04:42:16 example-02 kernel:  nvidia_open+0x339/0x550 [nvidia] 
Jun 24 04:42:16 example-02 kernel:  ? lookup_fast+0xcc/0x320 
Jun 24 04:42:16 example-02 kernel:  nvidia_frontend_open+0x58/0xa0 [nvidia] 
Jun 24 04:42:16 example-02 kernel:  chrdev_open+0xc4/0x1b0

Status

Resolved in NVIDIA vGPU software 9.1

Ref. #
200537078

5.4. 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

5.5. 9.0 only: Incorrect ECC error counts are reported for vGPUs on some GPUs

Description

Incorrect ECC error counts are reported for vGPUs on some GPUs when the command `nvidia-smi -q` is run from a guest VM.
This issue affects only vGPUs that reside on physical GPUs based on the NVIDIA Volta GPU architecture. For vGPUs on GPUs based on other architectures, the ECC error count is correct.

**Status**

Resolved in NVIDIA vGPU software 9.1

**Ref. #**

200525006

**5.6. 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
5.7. 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.

Status
Open

Ref. #
200505777

5.8. 9.0 only: VM crashes after the volatile ECC error count is reset

Description
After the command nvidia-smi -p 0 is run from a guest VM to reset the volatile ECC error count, the VM crashes.

This issue does not occur if the EEC state in the VM is set to off.
Status
Resolved in NVIDIA vGPU software 9.1

Ref. #
200526633

5.9. 9.0 only: No vComputeServer option available in NVIDIA X Server Settings

Description
The vComputeServer option is missing from the Manage License section in the NVIDIA X Server Settings window.

As a result of this missing option, the NVIDIA X Server Settings window incorrectly states that the system is licensed for Quadro vDWS when, in fact, the system is licensed for vComputeServer.
Workaround

If you are licensing a physical GPU for vComputeServer, you must use the configuration file /etc/nvidia/gridd.conf. See Virtual GPU Client Licensing User Guide.

Status

Resolved in NVIDIA vGPU software 9.1

Ref. #

200522255
5.10. 9.0 only: On Linux VMs, the license directory is not deleted when the guest driver is uninstalled

Description
On Linux guest VMs, the license directory /etc/nvidia/license is not deleted when the NVIDIA vGPU software graphics driver is uninstalled.

The following error message is written to the nvidia-uninstaller log file:

Failed to delete the directory '/etc/nvidia' (Directory not empty).

Workaround
As root, remove the /etc/nvidia/license directory after the NVIDIA vGPU software graphics driver is uninstalled.

Status
Resolved in NVIDIA vGPU software 9.1

Ref. #
200524555

5.11. 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
5.12. 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

5.13. 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

5.14. 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 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.
5.15. 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:

```bash
[root@localhost:-] nvidia-smi vgpu
Wed Jun 27 10:33:18 2018 +-----------------------------------------------------------------------------+ | NVIDIA-SMI 390.59                 Driver Version: 390.59                     | |-------------------------------+--------------------------------+------------| | GPU  Name                     | Bus-Id                         | GPU-Util   | | vGPU ID    Name          | VM ID    VM Name               | vGPU-Util  | |===============================+================================+============| |   0  Tesla P40                | 00000000:81:00.0               |  52%       | |      2122661    GRID P40-8Q   | 2122682  centos7.4-xmpl-211... |     19%    | |      2122663    GRID P40-8Q   | 2122692  centos7.4-xmpl-211... |      0%    | |      2122659    GRID P40-8Q   | 2122664  centos7.4-xmpl-211... |     25%    | |   1  Tesla P40                | 00000000:85:00.0               |  58%       | |      2122662    GRID P40-8Q   | 2122689  centos7.4-xmpl-211... |      0%    | |      2122658    GRID P40-8Q   | 2122667  centos7.4-xmpl-211... |     59%    | |      2122660    GRID P40-8Q   | 2122670  centos7.4-xmpl-211... |      0%    | +-------------------------------+--------------------------------+------------+
```

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
5.16. 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%    |
|      89170      GRID P40-4Q   | 89171    win7-xmpl-146048-4    |      0%    |
|      90475      GRID P40-4Q   | 90476    win7-xmpl-146048-5    |      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
5.17. 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.

Status

Closed

Ref. #

200407287

5.18. 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
5.19. 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.

Status
Closed

Ref. #
2101499

5.20. 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

5.21. Resolution is not updated after a VM acquires a license and is restarted

Description
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.

Version
Red Enterprise Linux 7.3

Status
Open

Ref. #
200275925

5.22. 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.
5.23. 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.

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 vComputeServer, 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.
When NVIDIA X Server Settings is restarted, the Manage License option is now available.

Status
Open

5.24. 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
5.25. 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

5.26. 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

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

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

HDMI

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

OpenCL

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

Trademarks

NVIDIA, the NVIDIA logo, NVIDIA GRID, vGPU, Pascal, 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

© 2013-2019 NVIDIA Corporation. All rights reserved.