



# Virtual GPU Software R418 for Red Hat Enterprise Linux with KVM

Release Notes

# Table of Contents

<b>Chapter 1. Release Notes.....</b>	<b>1</b>
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 8.10.....	2
1.4. Updates in Release 8.9.....	3
1.5. Updates in Release 8.8.....	3
1.6. Updates in Release 8.7.....	3
1.7. Updates in Release 8.6.....	3
1.8. Updates in Release 8.5.....	4
1.9. Updates in Release 8.4.....	4
1.10. Updates in Release 8.3.....	4
1.11. Updates in Release 8.2.....	5
1.12. Updates in Release 8.1.....	5
1.13. Updates in Release 8.0.....	5
<b>Chapter 2. Validated Platforms.....</b>	<b>7</b>
2.1. Supported NVIDIA GPUs and Validated Server Platforms.....	7
2.2. Hypervisor Software Releases.....	8
2.3. Guest OS Support.....	9
2.3.1. Windows Guest OS Support.....	9
2.3.2. Linux Guest OS Support.....	12
2.4. NVIDIA CUDA Toolkit Version Support.....	14
2.5. Multiple vGPU Support.....	15
<b>Chapter 3. Known Product Limitations.....</b>	<b>17</b>
3.1. Issues occur when the channels allocated to a vGPU are exhausted.....	17
3.2. Virtual GPU hot plugging is not supported.....	18
3.3. Total frame buffer for vGPUs is less than the total frame buffer on the physical GPU....	18
3.4. Issues may occur with graphics-intensive OpenCL applications on vGPU types with limited frame buffer.....	19
3.5. vGPU profiles with 512 Mbytes or less of frame buffer support only 1 virtual display head on Windows 10.....	19
3.6. NVENC requires at least 1 Gbyte of frame buffer.....	20
3.7. VM running older NVIDIA vGPU drivers fails to initialize vGPU when booted.....	20
3.8. Virtual GPU fails to start if ECC is enabled.....	21
3.9. Single vGPU benchmark scores are lower than pass-through GPU.....	23
3.10. nvidia-smi fails to operate when all GPUs are assigned to GPU pass-through mode....	24

<b>Chapter 4. Resolved Issues.....</b>	<b>25</b>
<b>Chapter 5. Security Updates.....</b>	<b>28</b>
5.1. Restricting Access to GPU Performance Counters.....	28
5.1.1. Windows: Restricting Access to GPU Performance Counters for One User by Using NVIDIA Control Panel.....	28
5.1.2. Windows: Restricting Access to GPU Performance Counters Across an Enterprise by Using a Registry Key.....	29
5.1.3. Linux Guest VMs and Hypervisor Host: Restricting Access to GPU Performance Counters.....	29
<b>Chapter 6. Known Issues.....</b>	<b>31</b>
6.1. Since 8.9: NVENC does not work with Teradici Cloud Access Software on Windows.....	31
6.2. A licensed client might fail to acquire a license if a proxy is set.....	31
6.3. 8.0-8.6 Only: Rebooting a Windows 10 vGPU VM causes a host crash.....	32
6.4. 8.0-8.5 Only: Desktop sessions disconnect after the server becomes unresponsive.....	33
6.5. 8.0-8.4 Only: Application responsiveness degrades over time.....	33
6.6. On systems with more than 1 TiB of system memory, application performance degrades over time.....	34
6.7. 8.0-8.3 Only: When the VMs to which 16 vGPUs on a single GPU are assigned are started simultaneously, one VM fails to boot.....	35
6.8. 8.0, 8.1 Only: VMs fail to boot with failed assertions.....	35
6.9. NVIDIA Control Panel fails to start if launched too soon from a VM without licensing information.....	36
6.10. DWM crashes randomly occur in Windows VMs.....	37
6.11. 8.0 Only: Incorrect NVIDIA vGPU software Windows graphics driver version in the installer.....	37
6.12. Vulkan applications crash in Windows 7 guest VMs configured with NVIDIA vGPU.....	38
6.13. Host core CPU utilization is higher than expected for moderate workloads.....	39
6.14. Frame capture while the interactive logon message is displayed returns blank screen.	39
6.15. RDS sessions do not use the GPU with some Microsoft Windows Server releases.....	40
6.16. 8.0-8.2 Only: Even when the scheduling policy is equal share, unequal GPU utilization is reported.....	41
6.17. When the scheduling policy is fixed share, GPU utilization is reported as higher than expected.....	42
6.18. License is not acquired in Windows VMs.....	43
6.19. nvidia-smi reports that vGPU migration is supported on all hypervisors.....	43
6.20. Hot plugging and unplugging vCPUs causes a blue-screen crash in Windows VMs.....	44
6.21. Luxmark causes a segmentation fault on an unlicensed Linux client.....	44
6.22. Resolution is not updated after a VM acquires a license and is restarted.....	45

6.23. A segmentation fault in DBus code causes nvidia-gridd to exit on Red Hat Enterprise Linux and CentOS.....	45
6.24. No Manage License option available in NVIDIA X Server Settings by default.....	46
6.25. Licenses remain checked out when VMs are forcibly powered off.....	47
6.26. VM bug checks after the guest VM driver for Windows 10 RS2 is installed.....	48
6.27. GNOME Display Manager (GDM) fails to start on Red Hat Enterprise Linux 7.2 and CentOS 7.0.....	49

---

# 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.



**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.

NVIDIA vGPU Software Version	NVIDIA Virtual GPU Manager Version	NVIDIA Windows Driver Version	NVIDIA Linux Driver Version
8.10	418.240	427.71	418.240
8.9	418.226.00	427.60	418.226.00
8.8	418.213	427.48	418.211.00
8.7	418.196	427.33	418.197.02
8.6	418.181	427.11	418.181.07
8.5	418.165.01	426.94	418.165.01
8.4	418.149	426.72	418.149
8.3	418.130	426.52	418.130
8.2	418.109	426.26	418.109
8.1	418.92	426.04	418.92
8.0	418.66	425.31	418.70

For details of which Red Hat Enterprise 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 an incompatible guest VM driver release for the release of the vGPU Manager that you are using, the NVIDIA vGPU fails to load. See [VM running older NVIDIA vGPU drivers fails to initialize vGPU when booted](#).

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 8.1 with guest VM drivers from release 8.0. 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 8.1 with guest VM drivers from release 7.2.



**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.

## 1.3. Updates in Release 8.10

### New Features in Release 8.10

- ▶ Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - February 2022*, which is posted shortly after the release date of this software and is listed on the [NVIDIA Product Security](#) page
- ▶ Miscellaneous bug fixes

### Hardware and Software Support Introduced in Release 8.10

- ▶ Support for Red Hat Enterprise Linux with KVM hypervisor 8.5
- ▶ Support for Red Hat Enterprise Linux 8.5 as a guest OS

### Feature Support Withdrawn in Release 8.10

- ▶ Red Hat Enterprise Linux with KVM hypervisor 8.1, 7.8, and 7.7 are no longer supported.
- ▶ Red Hat Enterprise Linux 8.1 is no longer supported as a guest OS.
- ▶ Red Hat Enterprise Linux 7.8. and 7.7 are no longer supported as a guest OS.

## 1.4. Updates in Release 8.9

### New Features in Release 8.9

- ▶ Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - October 2021*, which is posted shortly after the release date of this software and is listed on the [NVIDIA Product Security](#) page
- ▶ Miscellaneous bug fixes

## 1.5. Updates in Release 8.8

### New Features in Release 8.8

- ▶ Security updates - see [Security Bulletin: NVIDIA GPU Display Driver - July 2021](#)
- ▶ Miscellaneous bug fixes

### Hardware and Software Support Introduced in Release 8.8

- ▶ Support for Red Hat Enterprise Linux with KVM hypervisor release 8.4
- ▶ Support for Red Hat Enterprise Linux 8.4 as a guest OS

## 1.6. Updates in Release 8.7

### New Features in Release 8.7

- ▶ Security updates - see [Security Bulletin: NVIDIA GPU Display Driver - April 2021](#)
- ▶ Miscellaneous bug fixes

## 1.7. Updates in Release 8.6

### New Features in Release 8.6

- ▶ Security updates - see [Security Bulletin: NVIDIA GPU Display Driver - January 2021](#)
- ▶ Miscellaneous bug fixes

### Feature Support Withdrawn in Release 8.6

- ▶ Red Hat Enterprise Linux with KVM hypervisor 7.6 is no longer supported
- ▶ Red Hat Enterprise Linux 7.6 is no longer supported as a guest OS

## 1.8. Updates in Release 8.5

### New Features in Release 8.5

- ▶ Security updates- see [Security Bulletin: NVIDIA GPU Display Driver - September 2020](#)
- ▶ Miscellaneous bug fixes

## 1.9. Updates in Release 8.4

### New Features in Release 8.4

- ▶ Miscellaneous bug fixes
- ▶ Security updates - see [Security Bulletin: NVIDIA GPU Display Driver - June 2020](#)

### Hardware and Software Support Introduced in Release 8.4

- ▶ Support for Red Hat Enterprise Linux with KVM hypervisor release 8.2 and 7.8
- ▶ Support for the following guest OS releases
  - ▶ Red Hat Enterprise Linux 8.2 and 7.8
  - ▶ CentOS 7.8

### Feature Support Withdrawn in Release 8.4

- ▶ The following Red Hat hypervisor software releases are no longer supported:
  - ▶ Red Hat Enterprise Linux with KVM hypervisor releases 7.0-7.5
  - ▶ Red Hat Virtualization (RHV) 4.1
- ▶ The following OS releases are no longer supported as a guest OS:
  - ▶ Red Hat Enterprise Linux 7.5
  - ▶ CentOS 7.5

## 1.10. Updates in Release 8.3

### New Features in Release 8.3

- ▶ Miscellaneous bug fixes
- ▶ Security updates - see [Security Bulletin: NVIDIA GPU Display Driver - February 2020](#)



## Hardware and Software Support Introduced in Release 8.3

- ▶ Support for Red Hat Enterprise Linux with KVM hypervisor release 8.1
- ▶ Support for the following guest OS releases
  - ▶ Red Hat Enterprise Linux 8.1
  - ▶ CentOS Linux 8 (1911)

## Feature Support Withdrawn in Release 8.3

- ▶ Red Hat Enterprise Linux with KVM hypervisor releases 7.0-7.4 are no longer supported.
- ▶ The following OS releases are no longer supported as a guest OS:
  - ▶ Windows Server 2008 R2
  - ▶ Red Hat Enterprise Linux 7.0-7.4
  - ▶ CentOS 7.0-7.4

# 1.11. Updates in Release 8.2

## New Features in Release 8.2

- ▶ Miscellaneous bug fixes
- ▶ Security updates - see [Security Updates](#)

# 1.12. Updates in Release 8.1

## New Features in Release 8.1

- ▶ Security updates
- ▶ Miscellaneous bug fixes

## Hardware and Software Support Introduced in Release 8.1

- ▶ Support for Red Hat Enterprise Linux with KVM 7.7 hypervisor
- ▶ Support for Red Hat Enterprise Linux 7.7 as a guest OS

# 1.13. Updates in Release 8.0

## New Features in Release 8.0

- ▶ New -1B4 virtual GPU types

- ▶ Support for multiple vGPUs of the following types in a single VM:
  - ▶ RTX6000-24Q
  - ▶ RTX8000-48Q

For the complete list of vGPU types that support multiple vGPUs in a single VM, see [Multiple vGPU Support](#).

- ▶ Support for multiple vGPUs in a single VM on RHV 4.2 and RHV 4.3
- ▶ Increase in the number of vGPUs supported in a single VM from four to 16
- ▶ Security updates
- ▶ Miscellaneous bug fixes

### Hardware and Software Support Introduced in Release 8.0

- ▶ Support for the following GPUs:
  - ▶ Quadro RTX 6000
  - ▶ Quadro RTX 8000
- ▶ Support for RHV 4.3
- ▶ Support for the following OS releases as a guest OS:
  - ▶ Windows 10 October 2018 Update (1809)
  - ▶ Windows Server 2019

---

# 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 R418 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 M6
  - ▶ Tesla M10
  - ▶ Tesla M60
- ▶ GPUs based on the NVIDIA Pascal™ 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 Volta architecture:
  - ▶ Tesla V100 SXM2
  - ▶ Tesla V100 SXM2 32GB
  - ▶ Tesla V100 PCIe
  - ▶ Tesla V100 PCIe 32GB
  - ▶ 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.)
  - ▶ Quadro RTX 8000 in displayless mode (GRID Virtual PC and GRID Virtual Applications are **not** supported.)

In displayless mode, local physical display connectors are disabled.

For a list of validated server platforms, refer to [NVIDIA GRID Certified Servers](#).



**Note:**

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

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

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

## 2.2. Hypervisor Software Releases

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



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

Software	Releases Supported	Notes
<b>Since 8.10:</b> Red Hat Enterprise Linux with KVM	8.5	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.
<b>Since 8.8:</b> Red Hat Enterprise Linux with KVM	8.4	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.
<b>Since 8.4:</b> Red Hat Enterprise Linux with KVM	8.2	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.
<b>8.3-8.9 only:</b> Red Hat Enterprise Linux with KVM	8.1	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.

Software	Releases Supported	Notes
<b>8.4-8.9 only:</b> Red Hat Enterprise Linux with KVM	7.8	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.
<b>8.1-8.9 only:</b> Red Hat Enterprise Linux with KVM	7.7	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.
<b>8.0-8.6 only:</b> Red Hat Enterprise Linux with KVM	7.6	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.
<b>8.0-8.4 only:</b> Red Hat Enterprise Linux with KVM	7.5	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.
Red Hat Virtualization (RHV)	4.3, 4.2	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode.
<b>8.0-8.3 only:</b> Red Hat Virtualization (RHV)	4.1	All NVIDIA GPUs that NVIDIA vGPU software supports are supported in pass-through mode only.

## 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.



### Note:

Use only a guest OS release that is listed as supported by NVIDIA vGPU software with your virtualization software. To be listed as supported, a guest OS release must be supported not only by NVIDIA vGPU software, but also by your virtualization software. NVIDIA **cannot** support guest OS releases that your virtualization software does not support.

NVIDIA vGPU software supports **only** 64-bit guest operating systems. No 32-bit guest operating systems are supported.

### 2.3.1. Windows Guest OS Support



**Note:** 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.

**Note:**

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

Guest OS	NVIDIA vGPU - Red Hat Enterprise Linux with KVM Releases	Pass-Through GPU - Red Hat Enterprise Linux with KVM Releases
Windows Server 2019	<p><b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2</p> <p><b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p><b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7</p> <p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2</p>	<p><b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2</p> <p><b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p><b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7</p> <p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 <b>Since 8.4:</b> RHV 4.3, 4.2 <b>8.0-8.3 only:</b> RHV 4.3, 4.2, 4.1</p>
Windows Server 2016 1709, 1607	<p><b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2</p> <p><b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p><b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7</p> <p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2</p>	<p><b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2</p> <p><b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p><b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7</p> <p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 <b>Since 8.4:</b> RHV 4.3, 4.2 <b>8.0-8.3 only:</b> RHV 4.3, 4.2, 4.1</p>

Guest OS	NVIDIA vGPU - Red Hat Enterprise Linux with KVM Releases	Pass-Through GPU - Red Hat Enterprise Linux with KVM Releases
Windows Server 2012 R2	<p><b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2</p> <p><b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p><b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7</p> <p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2</p>	<p><b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2</p> <p><b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p><b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7</p> <p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 <b>Since 8.4:</b> RHV 4.3, 4.2 <b>8.0-8.3 only:</b> RHV 4.3, 4.2, 4.1</p>
<b>8.0-8.2 only:</b> Windows Server 2008 R2	<p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2</p>	<p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2, 4.1</p> <p>Supported only on GPUs based on the Maxwell architecture</p>
Windows 10 October 2018 Update (1809) and all Windows 10 releases supported by Microsoft up to and including this release	RHV 4.3, 4.2	<p><b>Since 8.4:</b> RHV 4.3, 4.2</p> <p><b>8.0-8.3 only:</b> RHV 4.3, 4.2, 4.1</p>
Windows 8.1 Update	RHV 4.3, 4.2	<p><b>Since 8.4:</b> RHV 4.3, 4.2</p> <p><b>8.0-8.3 only:</b> RHV 4.3, 4.2, 4.1</p>
Windows 7	RHV 4.3, 4.2	<p><b>Since 8.4:</b> RHV 4.3, 4.2</p> <p><b>8.0-8.3 only:</b> RHV 4.3, 4.2, 4.1</p> <p>Supported only on GPUs based on the Maxwell architecture</p>

## 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.



### Note:

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

Guest OS	NVIDIA vGPU - Red Hat Enterprise Linux with KVM Releases	Pass-Through GPU - Red Hat Enterprise Linux with KVM Releases
Since 8.10: Red Hat Enterprise Linux 8.5	RHEL KVM 8.5, 8.4, 8.2	RHEL KVM 8.5, 8.4, 8.2
Since 8.8: Red Hat Enterprise Linux 8.4	<p>Since 8.10: RHEL KVM 8.5, 8.4, 8.2</p> <p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1</p>	<p>Since 8.10: RHEL KVM 8.5, 8.4, 8.2</p> <p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1</p>
Since 8.4: Red Hat Enterprise Linux 8.2	<p>Since 8.10: RHEL KVM 8.5, 8.4, 8.2</p> <p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1</p> <p>8.4-8.7 only: RHEL KVM 8.2, 8.1</p>	<p>Since 8.10: RHEL KVM 8.5, 8.4, 8.2</p> <p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1</p> <p>8.4-8.7 only: RHEL KVM 8.2, 8.1</p>
8.3-8.9 only: Red Hat Enterprise Linux 8.1	<p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1</p> <p>8.4-8.7 only: RHEL KVM 8.2, 8.1</p> <p>8.3 only: RHEL KVM 8.1</p>	<p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1</p> <p>8.4-8.7 only: RHEL KVM 8.2, 8.1</p> <p>8.3 only: RHEL KVM 8.1</p>
Since 8.3: CentOS Linux 8 (1911)	<p>Since 8.10: RHEL KVM 8.5, 8.4, 8.2</p> <p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1</p> <p>8.4-8.7 only: RHEL KVM 8.2, 8.1</p> <p>8.3 only: RHEL KVM 8.1</p>	<p>Since 8.10: RHEL KVM 8.5, 8.4, 8.2</p> <p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1</p> <p>8.4-8.7 only: RHEL KVM 8.2, 8.1</p> <p>8.3 only: RHEL KVM 8.1</p>
8.6-8.9 only: Red Hat Enterprise Linux 7.7-7.8	<p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p>8.6, 8.7 only: RHEL KVM 8.2, 8.1, 7.8, 7.7</p>	<p>8.8, 8.9 only: RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p>8.6, 8.7 only: RHEL KVM 8.2, 8.1, 7.8, 7.7</p>



Guest OS	NVIDIA vGPU - Red Hat Enterprise Linux with KVM Releases	Pass-Through GPU - Red Hat Enterprise Linux with KVM Releases
	RHV 4.3, 4.2	RHV 4.3, 4.2
<b>8.4, 8.5 only:</b> Red Hat Enterprise Linux 7.6-7.8	RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6 RHV 4.3, 4.2	RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6 RHV 4.3, 4.2
<b>8.3 only:</b> Red Hat Enterprise Linux 7.5-7.7	RHEL KVM 8.1, 7.7, 7.6, 7.5 RHV 4.3, 4.2	RHEL KVM 8.1, 7.7, 7.6, 7.5 RHV 4.3, 4.2, 4.1
<b>8.1, 8.2 only:</b> Red Hat Enterprise Linux 7.0-7.7	RHEL KVM 7.7, 7.6, 7.5 RHV 4.3, 4.2	RHEL KVM 7.7, 7.6, 7.5 RHV 4.3, 4.2, 4.1
<b>8.0 only:</b> Red Hat Enterprise Linux 7.0-7.6	RHEL KVM 7.6, 7.5 RHV 4.3, 4.2	RHEL KVM 7.6, 7.5, RHV 4.3, 4.2, 4.1
<b>Since 8.6:</b> CentOS 7.7-7.8 See Note <a href="#">[1]</a>	<b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2 <b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7 <b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7 RHV 4.3, 4.2	<b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2 <b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7 <b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7 RHV 4.3, 4.2
<b>8.4, 8.5 only:</b> CentOS 7.6-7.8 See Note <a href="#">[1]</a>	RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6 RHV 4.3, 4.2	RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6 RHV 4.3, 4.2
<b>8.3 only:</b> CentOS 7.5, 7.6 See Note <a href="#">[1]</a>	RHEL KVM 8.1, 7.7, 7.6, 7.5 RHV 4.3, 4.2	RHEL KVM 8.1, 7.7, 7.6, 7.5 RHV 4.3, 4.2, 4.1
<b>8.0-8.2 only:</b> CentOS 7.0-7.6 See Note <a href="#">[1]</a>	<b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5 <b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2	<b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5 <b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2, 4.1
Red Hat Enterprise Linux 6.6	<b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2 <b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7 <b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7	<b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2 <b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7 <b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7

Guest OS	NVIDIA vGPU - Red Hat Enterprise Linux with KVM Releases	Pass-Through GPU - Red Hat Enterprise Linux with KVM Releases
	<p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2</p>	<p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 <b>Since 8.4:</b> RHV 4.3, 4.2</p> <p><b>8.0-8.3 only:</b> RHV 4.3, 4.2, 4.1</p>
CentOS 6.6 See Note <a href="#">[1]</a>	<p><b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2</p> <p><b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p><b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7</p> <p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 RHV 4.3, 4.2</p>	<p><b>Since 8.10:</b> RHEL KVM 8.5, 8.4, 8.2</p> <p><b>8.8, 8.9 only:</b> RHEL KVM 8.4, 8.2, 8.1, 7.8, 7.7</p> <p><b>8.6, 8.7 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7</p> <p><b>8.4, 8.5 only:</b> RHEL KVM 8.2, 8.1, 7.8, 7.7, 7.6</p> <p><b>8.3 only:</b> RHEL KVM 8.1, 7.7, 7.6, 7.5</p> <p><b>8.1, 8.2 only:</b> RHEL KVM 7.7, 7.6, 7.5</p> <p><b>8.0:</b> RHEL KVM 7.6, 7.5 <b>Since 8.4:</b> RHV 4.3, 4.2</p> <p><b>8.0-8.3 only:</b> RHV 4.3, 4.2, 4.1</p>

**Note:**

1. CentOS is not a certified guest OS for Red Hat Enterprise Linux with KVM or RHV.

## 2.4. NVIDIA CUDA Toolkit Version Support

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

For more information about NVIDIA CUDA Toolkit, see [CUDA Toolkit 10.1 Documentation](#). This documentation applies to the base NVIDIA CUDA Toolkit 10.1 release and updates to the base release.



**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 a 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

The assignment of more than one vGPU device to a VM is supported only on a subset of vGPUs and Red Hat Enterprise Linux with KVM releases.

### Supported vGPUs

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

GPU Architecture	Board	vGPU
Turing	Tesla T4	T4-16Q
	Quadro RTX 6000	RTX6000-24Q
	Quadro RTX 8000	RTX8000-48Q
Volta	Tesla V100 SXM2 32GB	V100DX-32Q
	Tesla V100 PCIe 32GB	V100D-32Q
	Tesla V100 SXM2	V100X-16Q
	Tesla V100 PCIe	V100-16Q
	Tesla V100 FHHL	V100L-16Q
Pascal	Tesla P100 SXM2	P100X-16Q
	Tesla P100 PCIe 16GB	P100-16Q
	Tesla P100 PCIe 12GB	P100C-12Q
	Tesla P40	P40-24Q
	Tesla P6	P6-8Q
	Tesla P4	P4-8Q
Maxwell	Tesla M60	M60-8Q
	Tesla M10	M10-8Q

GPU Architecture	Board	vGPU
	Tesla M6	M6-8Q

### 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

Only these Red Hat Enterprise Linux with KVM releases:

- ▶ **Since 8.8:** 8.4, 8.2, 8.1, and 7.6
- ▶ **8.4-8.7 only:** 8.2, 8.1, and 7.6
- ▶ **8.3 only:** 8.1, 7.6, and 7.5
- ▶ **8.0-8.2 only:** 7.6, and 7.5

RHV 4.3 and 4.2 only.

---

# 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
Jun 26 08:01:25 srvxen06f vgpu-3[14276]: error: vmiop_log: (0x0): , 0x0
```

### Workaround

Use a vGPU type with more frame buffer, thereby reducing the maximum number of vGPUs allowed on the physical GPU. As a result, the number of channels allocated to each vGPU is increased.

## 3.2. Virtual GPU hot plugging is not supported

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

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

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

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

$$\text{max-reserved-fb} = \text{vgpu-profile-size-in-mb} \div 16 + 16$$

### **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, *vgpu-profile-size-in-mb* is 16384.



**Note:** In VMs running a Windows guest OS that supports Windows Display Driver Model (WDDM) 1.x, namely, Windows 7, Windows 8.1, Windows Server 2008, and Windows Server 2012, 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. 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.

## 3.6. 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.

## 3.7. 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.

## 3.8. 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.

However, NVIDIA vGPU does not support ECC memory. If ECC memory is enabled, 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.
```

### Resolution

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.

## 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:

```
# 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 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

---

# 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 8.10

No resolved issues are reported in this release for Red Hat Enterprise Linux with KVM.

## Issues Resolved in Release 8.9

No resolved issues are reported in this release for Red Hat Enterprise Linux with KVM.

## Issues Resolved in Release 8.8

No resolved issues are reported in this release for Red Hat Enterprise Linux with KVM.

## Issues Resolved in Release 8.7

Bug ID	Summary and Description
3184762	<p><b><u>8.0-8.6 Only: Rebooting a Windows 10 vGPU VM causes a host crash</u></b></p> <p>When a Windows 10 VM that is configured with NVIDIA vGPU is rebooted, the hypervisor host crashes. This issue is caused by the failure of the Virtual GPU Manger to honor a particular notifier request from the kernel, which causes the kernel to crash.</p>

## Issues Resolved in Release 8.6

Bug ID	Summary and Description
2925629	<p><b><u>8.0-8.5 Only: Desktop sessions disconnect after the server becomes unresponsive</u></b></p> <p>Desktop sessions disconnect after the server becomes unresponsive. Before the sessions disconnect, the error message <code>vGPU message 32 failed, result code: 0x59</code> is written to the log files on the hypervisor host.</p>

## Issues Resolved in Release 8.5

Bug ID	Summary and Description
3051614	<p><b><u>8.0-8.4 Only: Application responsiveness degrades over time</u></b></p> <p>Application responsiveness degrades over time, causing slow application performance and stutter when users switch between applications. This issue occurs because the GPU driver is not setting the Linux kernel PCI <code>coherent_dma_mask</code> for NVIDIA GPU devices. If the <code>coherent_dma_mask</code> is not set, IOMMU IOVA space is restricted to the default size of 32 bits for DMA allocations performed in the NVIDIA GPU device context. Furthermore, for hosts on which <code>iommu=pt</code> is set, the default <code>coherent_dma_mask</code> causes IOMMU mappings to always be created. When IOMMU mappings are always created, performance degradation can occur because all host to device accesses require translation by hardware IOMMU.</p>

## Issues Resolved in Release 8.4

Bug ID	Summary and Description
200594274	<p><b><u>8.0-8.3 Only: When the VMs to which 16 vGPUs on a single GPU are assigned are started simultaneously, one VM fails to boot</u></b></p> <p>When the VMs to which 16 vGPUs on a single GPU (for example 16 T4-1Q vGPUs on a Tesla T4 GPU) are assigned are started simultaneously, only 15 VMs boot and the remaining VM fails to start. When the other VMs are shut down, the VM that failed to start boots successfully.</p>

## Issues Resolved in Release 8.3

Bug ID	Summary and Description
2175888	<p><b><u>8.0-8.2 Only: Even when the scheduling policy is equal share, unequal GPU utilization is reported</u></b></p> <p>When the scheduling policy is equal share, unequal GPU engine utilization can be reported for the vGPUs on the same physical GPU.</p>

## Issues Resolved in Release 8.2

Bug ID	Summary and Description
2644858	<p><b><u>8.0, 8.1 Only: VMs fail to boot with failed assertions</u></b></p> <p>In some scenarios with heavy workloads running on multiple VMs configured with NVIDIA vGPUs on a single physical GPU, additional VMs configured with NVIDIA vGPU on the same GPU fail to boot. The failure of the VM to boot is</p>

Bug ID	Summary and Description
	followed by failed assertions. This issue affects GPUs based on the NVIDIA Volta GPU architecture and later architectures.

### Issues Resolved in Release 8.1

Bug ID	Summary and Description
200534988	<p><b>Error XID 47 followed by multiple XID 32 errors</b></p> <p>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.</p>
-	<p><b><u>8.0 Only: Incorrect NVIDIA vGPU software Windows graphics driver version in the installer</u></b></p> <p>The NVIDIA vGPU software Windows graphics driver version in the installer is incorrect. The driver version incorrectly appears as 325.31 instead of 425.31 in the <b>Extraction path</b> field and the title of the <b>NVIDIA Graphics Driver (325.31) Package Window</b>.</p>

### Issues Resolved in Release 8.0

Bug ID	Summary and Description
1971698	<p><b>NVIDIA vGPU encoder and process utilization counters don't work with Windows Performance Counters</b></p> <p>GPU encoder and process utilization counter groups are listed in Windows Performance Counters, but no instances of the counters are available. The counters are disabled by default and must be enabled.</p>

---

# Chapter 5. Security Updates

## 5.1. Restricting Access to GPU Performance Counters

The NVIDIA graphics driver contains a vulnerability (CVE-2018-6260) that may allow access to application data processed on the GPU through a side channel exposed by the GPU performance counters. To address this vulnerability, update the driver and restrict access to GPU performance counters to allow access only by administrator users and users who need to use CUDA profiling tools.

The GPU performance counters that are affected by this vulnerability are the hardware performance monitors used by the CUDA profiling tools such as CUPTI, Nsight Graphics, and Nsight Compute. These performance counters are exposed on the hypervisor host and in guest VMs only as follows:

- ▶ On the hypervisor host, they are always exposed. However, the Virtual GPU Manager does not access these performance counters and, therefore, is not affected.
- ▶ In Windows and Linux guest VMs, they are exposed **only** in VMs configured for GPU pass through. They are not exposed in VMs configured for NVIDIA vGPU.

### 5.1.1. Windows: Restricting Access to GPU Performance Counters for One User by Using NVIDIA Control Panel

Perform this task from the guest VM to which the GPU is passed through.

Ensure that you are running **NVIDIA Control Panel** version 8.1.950.

1. Open **NVIDIA Control Panel**:
  - ▶ Right-click on the Windows desktop and select **NVIDIA Control Panel** from the menu.
  - ▶ Open **Windows Control Panel** and double-click the **NVIDIA Control Panel** icon.
2. In **NVIDIA Control Panel**, select the **Manage GPU Performance Counters** task in the **Developer** section of the navigation pane.



- Complete the task by following the instructions in the **Manage GPU Performance Counters > Developer** topic in the **NVIDIA Control Panel** help.

## 5.1.2. Windows: Restricting Access to GPU Performance Counters Across an Enterprise by Using a Registry Key

You can use a registry key to restrict access to GPU Performance Counters for all users who log in to a Windows guest VM. By incorporating the registry key information into a script, you can automate the setting of this registry for all Windows guest VMs across your enterprise.

Perform this task from the guest VM to which the GPU is passed through.



**CAUTION:** Only enterprise administrators should perform this task. Changes to the Windows registry must be made with care and system instability can result if registry keys are incorrectly set.

- Set the `RmProfilingAdminOnly` Windows registry key to 1.

```
[HKLM\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global\NVTweak]
Value: "RmProfilingAdminOnly"
Type: DWORD
Data: 00000001
```

The data value 1 restricts access, and the data value 0 allows access, to application data processed on the GPU through a side channel exposed by the GPU performance counters.

- Restart the VM.

## 5.1.3. Linux Guest VMs and Hypervisor Host: Restricting Access to GPU Performance Counters

On systems where unprivileged users don't need to use GPU performance counters, restrict access to these counters to system administrators, namely users with the `CAP_SYS_ADMIN` capability set. By default, the GPU performance counters are not restricted to users with the `CAP_SYS_ADMIN` capability.

Perform this task from the guest VM to which the GPU is passed through or from your hypervisor host machine.

In Linux guest VMs, this task requires `sudo` privileges. On your hypervisor host machine, this task must be performed as the root user on the machine.

- Log in to the guest VM or open a command shell on your hypervisor host machine.
- Set the kernel module parameter `NVreg_RestrictProfilingToAdminUsers` to 1 by adding this parameter to the `/etc/modprobe.d/nvidia.conf` file.

- ▶ If you are setting only this parameter, add an entry for it to the `/etc/modprobe.d/nvidia.conf` file as follows:

```
options nvidia NVreg_RegistryDwords="NVreg_RestrictProfilingToAdminUsers=1"
```

- ▶ If you are setting multiple parameters, set them in a single entry as in the following example:

```
options nvidia NVreg_RegistryDwords="RmPVMRL=0x0"  
"NVreg_RestrictProfilingToAdminUsers=1"
```

If the `/etc/modprobe.d/nvidia.conf` file does not already exist, create it.

3. Restart the VM or reboot your hypervisor host machine.

---

## Chapter 6. Known Issues

### 6.1. Since 8.9: NVENC does not work with Teradici Cloud Access Software on Windows

#### Description

The NVIDIA hardware-based H.264/HEVC video encoder (NVENC) does not work with Teradici Cloud Access Software on Windows. This issue affects NVIDIA vGPU and GPU pass through deployments.

This issue occurs because the check that Teradici Cloud Access Software performs on the DLL signer name is case sensitive and NVIDIA recently changed the case of the company name in the signature certificate.

#### Status

Not an NVIDIA bug

This issue is resolved in the latest 21.07 and 21.03 Teradici Cloud Access Software releases.

#### Ref. #

200749065

### 6.2. A licensed client might fail to acquire a license if a proxy is set

#### Description

If a proxy is set with a system environment variable such as `HTTP_PROXY` or `HTTPS_PROXY`, a licensed client might fail to acquire a license.

## Workaround

Perform this workaround on each affected licensed client.

1. Add the address of the NVIDIA vGPU software license server to the system environment variable `NO_PROXY`.

The address must be specified exactly as it is specified in the client's license server settings either as a fully-qualified domain name or an IP address. If the `NO_PROXY` environment variable contains multiple entries, separate the entries with a comma (,).

If high availability is configured for the license server, add the addresses of the primary license server and the secondary license server to the system environment variable `NO_PROXY`.

2. Restart the NVIDIA driver service that runs the core NVIDIA vGPU software logic.
  - ▶ On Windows, restart the **NVIDIA Display Container** service.
  - ▶ On Linux, restart the `nvidia-gridd` service.

## Status

Closed

## Ref. #

200704733

## 6.3. 8.0-8.6 Only: Rebooting a Windows 10 vGPU VM causes a host crash

### Description

When a Windows 10 VM that is configured with NVIDIA vGPU is rebooted, the hypervisor host crashes. This issue is caused by the failure of the Virtual GPU Manger to honor a particular notifier request from the kernel, which causes the kernel to crash.

### Status

Resolved in NVIDIA vGPU software 8.7.

### Ref. #

3184762

## 6.4. 8.0-8.5 Only: Desktop sessions disconnect after the server becomes unresponsive

### Description

Desktop sessions disconnect after the server becomes unresponsive. Before the sessions disconnect, the error message `vGPU message 32 failed, result code: 0x59` is written to the log files on the hypervisor host.

### Status

Resolved in NVIDIA vGPU software 8.6

### Ref. #

2925629

## 6.5. 8.0-8.4 Only: Application responsiveness degrades over time

### Description

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

### Status

Resolved in NVIDIA vGPU software 8.5



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

**Ref. #**

3051614

## 6.6. On systems with more than 1 TiB of system memory, application performance degrades over time

**Description**

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

**Workaround**

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

**Status**

Open

**Ref. #**

3063042

## 6.7. 8.0-8.3 Only: When the VMs to which 16 vGPUs on a single GPU are assigned are started simultaneously, one VM fails to boot

### Description

When the VMs to which 16 vGPUs on a single GPU (for example 16 T4-1Q vGPUs on a Tesla T4 GPU) are assigned are started simultaneously, only 15 VMs boot and the remaining VM fails to start. When the other VMs are shut down, the VM that failed to start boots successfully.

The log file on the hypervisor host contains these error messages:

```
2020-03-04T16:01:13.626Z| vmx| E110: vmiop_log: NVOS status 0x51
2020-03-04T16:01:13.626Z| vmx| E110: vmiop_log: Assertion Failed at 0x5e530d8c:303
...
2020-03-04T16:01:13.626Z| vmx| E110: vmiop_log: (0x0): Failed to alloc guest FB
memory
2020-03-04T16:01:13.626Z| vmx| E110: vmiop_log: (0x0): init_device_instance failed
for inst 0 with error 2 (vmiop-display: error allocating framebuffer)
2020-03-04T16:01:13.626Z| vmx| E110: vmiop_log: (0x0): Initialization:
init_device_instance failed error 2
2020-03-04T16:01:13.629Z| vmx| E110: vmiop_log: display_init failed for inst: 0
```

### Status

Resolved in NVIDIA vGPU software 8.4

### Ref. #

200594274

## 6.8. 8.0, 8.1 Only: VMs fail to boot with failed assertions

### Description

In some scenarios with heavy workloads running on multiple VMs configured with NVIDIA vGPUs on a single physical GPU, additional VMs configured with NVIDIA vGPU on the same GPU fail to boot. The failure of the VM to boot is followed by failed assertions. This issue affects GPUs based on the NVIDIA Volta GPU architecture and later architectures.

When this error occurs, error messages similar to the following examples are logged to the Red Hat Enterprise Linux with KVM log file:

```
nvidia-vgpu-mgr[31526]: error: vmiop_log: NVOS status 0x1e
```

```

nvidia-vgpu-mgr[31526]: error: vmiop_log: Assertion Failed at 0xb2d3e4d7:96
nvidia-vgpu-mgr[31526]: error: vmiop_log: 12 frames returned by backtrace
nvidia-vgpu-mgr[31526]: error: vmiop_log: /usr/lib64/libnvidia-vgpu.so(_nv003956vgpu
+0x18) [0x7f4bb2cfb338] vmiop_dump_stack
nvidia-vgpu-mgr[31526]: error: vmiop_log: /usr/lib64/libnvidia-vgpu.so(_nv004018vgpu
+0xd4) [0x7f4bb2d09ce4] vmiopd_alloc_pb_channel
nvidia-vgpu-mgr[31526]: error: vmiop_log: /usr/lib64/libnvidia-vgpu.so(_nv002878vgpu
+0x137) [0x7f4bb2d3e4d7] vgpufceInitCopyEngine_GK104
nvidia-vgpu-mgr[31526]: error: vmiop_log: /usr/lib64/libnvidia-vgpu.so(+0x80e27)
[0x7f4bb2cd0e27]
nvidia-vgpu-mgr[31526]: error: vmiop_log: /usr/lib64/libnvidia-vgpu.so(+0x816a7)
[0x7f4bb2cd16a7]
nvidia-vgpu-mgr[31526]: error: vmiop_log: vgpu() [0x413820]
nvidia-vgpu-mgr[31526]: error: vmiop_log: vgpu() [0x413a8d]
nvidia-vgpu-mgr[31526]: error: vmiop_log: vgpu() [0x40e11f]
nvidia-vgpu-mgr[31526]: error: vmiop_log: vgpu() [0x40bb69]
nvidia-vgpu-mgr[31526]: error: vmiop_log: vgpu() [0x40b51c]
nvidia-vgpu-mgr[31526]: error: vmiop_log: /lib64/libc.so.6(__libc_start_main+0x100)
[0x7f4bb2feed20]
nvidia-vgpu-mgr[31526]: error: vmiop_log: vgpu() [0x4033ea]
nvidia-vgpu-mgr[31526]: error: vmiop_log: (0x0): Alloc Channel(Gpfifo) for device
failed error: 0x1e
nvidia-vgpu-mgr[31526]: error: vmiop_log: (0x0): Failed to allocate FCE channel
nvidia-vgpu-mgr[31526]: error: vmiop_log: (0x0): init_device_instance failed for
inst 0 with error 2 (init frame copy engine)
nvidia-vgpu-mgr[31526]: error: vmiop_log: (0x0): Initialization:
init_device_instance failed error 2
nvidia-vgpu-mgr[31526]: error: vmiop_log: display_init failed for inst: 0
nvidia-vgpu-mgr[31526]: error: vmiop_env_log: (0x0): vmiop_process_configuration:
plugin registration error
nvidia-vgpu-mgr[31526]: error: vmiop_env_log: (0x0): vmiop_process_configuration
failed with 0x1a
kernel: [858113.083773] [nvidia-vgpu-vfio] ace3f3bb-17d8-4587-920e-199b8fed532d:
start failed. status: 0x1

```

## Status

Resolved in NVIDIA vGPU software 8.2

## Ref. #

2644858

# 6.9. 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

Wait at least 30 seconds before trying to launch **NVIDIA Control Panel**.



### Status

Open

### Ref. #

200623179

## 6.10. 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

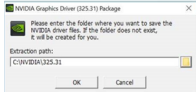
### Ref. #

2730037

## 6.11. 8.0 Only: Incorrect NVIDIA vGPU software Windows graphics driver version in the installer

### Description

The NVIDIA vGPU software Windows graphics driver version in the installer is incorrect. The driver version incorrectly appears as 325.31 instead of 425.31 in the **Extraction path** field and the title of the **NVIDIA Graphics Driver (325.31) Package Window**.



## Version

NVIDIA vGPU software Windows graphics driver version 425.31 in NVIDIA vGPU software release 8.0

## Workaround

To simplify future administration of your system, you can correct the driver version in the folder name when the installer prompts you for the extraction path. However, if you do not change the name of the folder in the extraction path, the installation succeeds and the driver functions correctly.

## Status

Resolved in NVIDIA vGPU software release 8.1

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

## Description

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

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

## Status

Open

## Ref. #

200381348

## 6.13. 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

## 6.14. 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

# 6.15. 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.

1. Choose **Local Computer Policy > Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Remote Session Environment** .
2. Set the **Use the hardware default graphics adapter for all Remote Desktop Services sessions** option.

## 6.16. 8.0-8.2 Only: Even when the scheduling policy is equal share, unequal GPU utilization is reported

### Description

When the scheduling policy is equal share, unequal GPU engine utilization can be reported for the vGPUs on the same physical GPU.

For example, GPU engine usage for three P40-8Q vGPUs on a Tesla P40 GPU might be reported as follows:

```
[root@localhost:~] nvidia-smi vgpu
Wed Jun 27 10:33:18 2018
+-----+-----+
| 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

Resolved in NVIDIA vGPU software 8.3

### Ref. #

2175888

## 6.17. 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

## 6.18. 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

## 6.19. `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

# 6.20. 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

# 6.21. 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

## 6.22. 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

## 6.23. 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

# 6.24. 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.

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

# 6.25. 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

## 6.26. 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.

### Status

Not an NVIDIA bug

### Ref. #

200310861

## 6.27. 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

## Notice

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

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

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

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

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

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

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

## VESA DisplayPort

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

## HDMI

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

## OpenCL

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

## Trademarks

NVIDIA, the NVIDIA logo, NVIDIA GRID, NVIDIA GRID vGPU, NVIDIA Maxwell, NVIDIA Pascal, NVIDIA Turing, NVIDIA Volta, GPUDirect, 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

© 2013-2022 NVIDIA Corporation. All rights reserved.

