# Virtual GPU Software R460 for Red Hat Enterprise Linux with KVM

Release Notes

RN-08687-001 _v12.0 through 12.4 Revision 02 | December 2021

# Table of Contents

# 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 |
|---|---|---|---|
| 12.4 | 460.107 | 463.15 | 460.106.00 |
| 12.3 | 460.91.03 | 462.96 | 460.91.03 |
| 12.2 | 460.73.02 | 462.31 | 460.73.01 |
| 12.1 | 460.32.04 | 461.33 | 460.32.03 |
| 12.0 | 460.32.04 | 461.09 | 460.32.03 |

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 an incompatible NVIDIA vGPU guest driver fails to initialize vGPU when booted.

**Note:** This requirement does not apply to the NVIDIA vGPU software license server. All releases in this release family of NVIDIA vGPU software are compatible with **all** releases of the license server.

## Compatible NVIDIA vGPU Manager and Guest VM Driver Releases

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

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

**Note:**

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

For example, if vGPU Manager from release 12.4 is used with guest drivers from release 11.2, the combination does **not** support Red Hat Enterprise Linux 7.6 because NVIDIA vGPU software release 12.4 does not support Red Hat Enterprise Linux 7.6.

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

| NVIDIA vGPU Software Component | Releases | Compatible Software Releases |
|---|---|---|
| NVIDIA vGPU Manager | 12.0 through 12.4 | ▶ Guest VM driver releases 12.0 through 12.4<br>▶ All guest VM driver 11.*x* releases |

| NVIDIA vGPU Software Component | Releases | Compatible Software Releases |
|---|---|---|
| Guest VM drivers | 12.0 through 12.4 | NVIDIA vGPU Manager releases 12.0 through 12.4 |

### Incompatible NVIDIA vGPU Manager and Guest VM Driver Releases

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

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

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

| NVIDIA vGPU Software Component | Releases | Incompatible Software Releases |
|---|---|---|
| NVIDIA vGPU Manager | 12.0 through 12.4 | All guest VM driver releases 10.*x* and earlier |
| Guest VM drivers | 12.0 through 12.4 | All NVIDIA vGPU Manager releases 11.*x* and earlier |

## 1.3. Updates in Release 12.4

### New Features in Release 12.4

- 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

## 1.4. Updates in Release 12.3

### New Features in Release 12.3

- Security updates - see Security Bulletin: NVIDIA GPU Display Driver - July 2021
- Miscellaneous bug fixes

### Feature Support Withdrawn in Release 12.3

- Red Hat Enterprise Linux with KVM hypervisor 8.3 is no longer supported.
- Red Hat Enterprise Linux 8.3 is no longer supported as a guest OS.

# 1.5. Updates in Release 12.2

## New Features in Release 12.2

- Support for unified memory with NVIDIA vGPU
- Security updates - see Security Bulletin: NVIDIA GPU Display Driver - April 2021
- Miscellaneous bug fixes

## Hardware and Software Support Introduced in Release 12.2

- Support for the following GPUs:
  - NVIDIA® A10
  - NVIDIA RTX A5000
- Support for Red Hat Enterprise Linux with KVM hypervisor 8.4
- Support for Red Hat Enterprise Linux 8.4 as a guest OS
- Support for NVIDIA GPU Operator with Red Hat Enterprise Linux with KVM 8.2, Red Hat OpenShift 4.7, and Red Hat CoreOS 4.7

# 1.6. Updates in Release 12.1

## New Features in Release 12.1

- Miscellaneous bug fixes

# 1.7. Updates in Release 12.0

## New Features in Release 12.0

- Support for NVIDIA® GPU Operator
- Support for NVIDIA® NVSwitch™ on-chip memory fabric
- Miscellaneous bug fixes

## Hardware and Software Support Introduced in Release 12.0

- Support for the following GPUs:
  - NVIDIA A100 HGX 80GB
  - NVIDIA A40
  - NVIDIA RTX A6000

- Support for Windows 10 October 2020 Update (20H2) as a guest OS

  Windows 10 May 2021 Update (21H1), which is a bug fix release for Windows 10 October 2020 Update (20H2), is also supported.
- Support for Red Hat CoreOS 4.7 as a guest OS

# 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 R460 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 (NVIDIA Virtual Compute Server (vCS) is **not** supported.)
  - Tesla M10 (vCS is **not** supported.)
  - Tesla M60 (vCS is **not** supported.)
- 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 V100S PCIe 32GB

- Tesla V100 FHHL

- GPUs based on the NVIDIA Turing™ architecture:
  - Tesla T4
  - Quadro RTX 6000 in displayless mode
  - Quadro RTX 6000 passive in displayless mode
  - Quadro RTX 8000 in displayless mode
  - Quadro RTX 8000 passive in displayless mode

  In displayless mode, local physical display connectors are disabled.

- GPUs based on the NVIDIA Ampere architecture:
  - NVIDIA A100 HGX 80GB (supports **only** compute workloads on Linux with NVIDIA Virtual Compute Server and GPU pass through; graphics acceleration is **not** supported)
  - NVIDIA A100 PCIe 40GB (supports **only** compute workloads on Linux with NVIDIA Virtual Compute Server and GPU pass through; graphics acceleration is **not** supported)
  - NVIDIA A100 HGX 40GB (supports **only** compute workloads on Linux with NVIDIA Virtual Compute Server and GPU pass through; graphics acceleration is **not** supported)
  - NVIDIA A40 in displayless mode
  - **Since 12.2:** NVIDIA A10
  - NVIDIA RTX A6000 in displayless mode
  - **Since 12.2:** NVIDIA RTX A5000 in displayless mode

  In displayless mode, local physical display connectors are disabled.

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

### 2.1.1. Switching the Mode of a GPU that Supports Multiple Display Modes

Some GPUs support displayless and display-enabled modes but must be used in NVIDIA vGPU software deployments in displayless mode.

The GPUs listed in the following table support multiple display modes. As shown in the table, some GPUs are supplied from the factory in displayless mode, but other GPUs are supplied in a display-enabled mode.

| GPU | Mode as Supplied from the Factory |
|---|---|
| NVIDIA A40 | Displayless |
| NVIDIA RTX A5000 | Display enabled |
| NVIDIA RTX A6000 | Display enabled |

A GPU that is supplied from the factory in displayless mode, such as the NVIDIA A40 GPU, might be in a display-enabled mode if its mode has previously been changed.

To change the mode of a GPU that supports multiple display modes, use the `displaymodeselector` tool, which you can request from the NVIDIA Display Mode Selector Tool page on the NVIDIA Developer website.

**Note:**

Only the following GPUs support the `displaymodeselector` tool:

- NVIDIA A40
- NVIDIA RTX A5000
- NVIDIA RTX A6000

Other GPUs that support NVIDIA vGPU software do not support the `displaymodeselector` tool and, unless otherwise stated, do not require display mode switching.

### 2.1.2. Switching the Mode of a Tesla M60 or M6 GPU

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. If you are unsure which mode your GPU is in, use the `gpumodeswitch` tool to find out the mode.

**Note:**

Only Tesla M60 and M6 GPUs support the `gpumodeswitch` tool. Other GPUs that support NVIDIA vGPU do not support the `gpumodeswitch` tool and, except as stated in Switching the Mode of a GPU that Supports Multiple Display Modes, do not require mode switching.

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

For more information, refer to *gpumodeswitch User Guide*.

## 2.2. Hypervisor Software Releases

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

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

| Software | Releases Supported | Notes |
|---|---|---|
| **Since 12.2:** 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. |

| Software | Releases Supported | Notes |
|---|---|---|
| **12.0-12.2 only:** Red Hat Enterprise Linux with KVM | 8.3 | All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode. |
| 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. |
| Red Hat Enterprise Linux with KVM | 8.1 | The following GPUs are supported in GPU pass through mode **only**:<br>▶ NVIDIA RTX A6000<br>▶ NVIDIA RTX A5000<br>▶ NVIDIA A40<br>▶ NVIDIA A10<br>▶ NVIDIA A100 HGX 80GB<br>▶ NVIDIA A100 PCIe 40GB<br>▶ NVIDIA A100 HGX 40GB<br>All other NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode. |
| Red Hat Enterprise Linux with KVM | 7.9, 7.8, 7.7 | The following GPUs are supported in GPU pass through mode **only**:<br>▶ NVIDIA RTX A6000<br>▶ NVIDIA RTX A5000<br>▶ NVIDIA A40<br>▶ NVIDIA A10<br>▶ NVIDIA A100 HGX 80GB<br>▶ NVIDIA A100 PCIe 40GB<br>▶ NVIDIA A100 HGX 40GB<br>All other NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode. |
| Red Hat Virtualization (RHV) | 4.4 | The following GPUs are supported in GPU pass through mode **only**:<br>▶ NVIDIA A100 HGX 80GB<br>▶ NVIDIA A100 PCIe 40GB<br>▶ NVIDIA A100 HGX 40GB<br>All other NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode. |

| Software | Releases Supported | Notes |
|---|---|---|
| Red Hat Virtualization (RHV) | 4.3, 4.2 | Not supported on the following GPUs:<br>▶ NVIDIA RTX A6000<br>▶ NVIDIA RTX A5000<br>▶ NVIDIA A40<br>▶ NVIDIA A10<br>The following GPUs are supported in GPU pass through mode **only**:<br>▶ NVIDIA A100 HGX 80GB<br>▶ NVIDIA A100 PCIe 40GB<br>▶ NVIDIA A100 HGX 40GB<br>All other NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode. |

## 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 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8,3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 |
| Windows Server 2016 1709, 1607 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 |
| Windows Server 2012 R2 (not supported on GPUs based on architectures after the NVIDIA Turing™ architecture) | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 |
| Windows 10 May 2021 Update (21H1) and all Windows 10 releases supported by Microsoft up to and including this release<br>The hardware-accelerated GPU scheduling feature introduced in Windows 10 May 2020 Update (2004) is **not** supported on GPUs based on the Maxwell architecture and is supported only | RHV 4.4, 4.3, 4.2 | RHV 4.4, 4.3, 4.2 |

| Guest OS | NVIDIA vGPU - Red Hat Enterprise Linux with KVM Releases | Pass-Through GPU - Red Hat Enterprise Linux with KVM Releases |
|---|---|---|
| in pass-through mode on GPUs based on later architectures. | | |

## 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 |
|---|---|---|
| Red Hat CoreOS 4.7 | RHEL KVM 8.2 | RHEL KVM 8.2 |
| **Since 12.2:** Red Hat Enterprise Linux 8.4 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1 |
| **12.0-12.2 only:** Red Hat Enterprise Linux 8.3 | **12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 | **12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 |
| Red Hat Enterprise Linux 8.2 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 |
| Red Hat Enterprise Linux 8.1 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 |

| Guest OS | NVIDIA vGPU - Red Hat Enterprise Linux with KVM Releases | Pass-Through GPU - Red Hat Enterprise Linux with KVM Releases |
|---|---|---|
| CentOS Linux 8 (1911) | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 |
| CentOS 8.0 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1 |
| Red Hat Enterprise Linux 7.7-7.9 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 |
| CentOS 7.6-7.8 See Note (1) | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 |
| Red Hat Enterprise Linux 6.6 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 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 |
|---|---|---|
| | RHV 4.4, 4.3, 4.2 | RHV 4.4, 4.3, 4.2 |
| CentOS 6.6 See Note (1) | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 | **Since 12.3:** RHEL KVM 8.4, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.2 only:** RHEL KVM 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>**12.0, 12.1 only:** RHEL KVM 8.3, 8.2, 8.1, 7.9, 7.8, 7.7<br>RHV 4.4, 4.3, 4.2 |

**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 11.2.

For more information about NVIDIA CUDA Toolkit, see CUDA Toolkit 11.2 Documentation.

**Note:**

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

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 time-sliced vGPUs that are allocated all of the physical GPU's frame buffer are supported. MIG-backed vGPUs are **not** supported.

| GPU Architecture | Board | vGPU |
|---|---|---|
| Ampere (compute workloads only) | NVIDIA A100 HGX 80GB | A100DX-80C See Note (1). |
| | NVIDIA A100 PCIe 40GB | A100-40C See Note (1). |
| | NVIDIA A100 HGX 40GB | A100X-40C See Note (1). |
| Ampere (compute and graphics workloads) | NVIDIA A40 | A40-48Q See Note (1). |
| | | A40-48C See Note (1). |
| | NVIDIA A10 | A10-24Q See Note (1). |
| | | A10-24C See Note (1). |
| | NVIDIA RTX A6000 | A6000-48Q See Note (1). |
| | | A6000-48C See Note (1). |
| | NVIDIA RTX A5000 | A5000-24Q See Note (1). |
| | | A5000-24C See Note (1). |
| Turing | Tesla T4 | T4-16Q |
| | | T4-16C |
| | Quadro RTX 6000 | RTX6000-24Q |
| | | RTX6000-24C |
| | Quadro RTX 6000 passive | RTX6000P-24Q |
| | | RTX6000P-24C |
| | Quadro RTX 8000 | RTX8000-48Q |
| | | RTX8000-48C |
| | Quadro RTX 8000 passive | RTX8000P-48Q |
| | | RTX8000P-48C |
| Volta | Tesla V100 SXM2 32GB | V100DX-32Q |
| | | V100D-32C |
| | Tesla V100 PCIe 32GB | V100D-32Q |

| GPU Architecture | Board | vGPU |
|---|---|---|
| | | V100D-32C |
| | Tesla V100S PCIe 32GB | V100S-32Q |
| | | V100S-32C |
| | Tesla V100 SXM2 | V100X-16Q |
| | | V100X-16C |
| | Tesla V100 PCIe | V100-16Q |
| | | V100-16C |
| | Tesla V100 FHHL | V100L-16Q |
| | | V100L-16C |
| Pascal | Tesla P100 SXM2 | P100X-16Q |
| | | P100X-16C |
| | Tesla P100 PCIe 16GB | P100-16Q |
| | | P100-16C |
| | Tesla P100 PCIe 12GB | P100C-12Q |
| | | P100C-12C |
| | Tesla P40 | P40-24Q |
| | | P40-24C |
| | Tesla P6 | P6-16Q |
| | | P6-16C |
| | Tesla P4 | P4-8Q |
| | | P4-8C |
| Maxwell | Tesla M60 | M60-8Q |
| | Tesla M10 | M10-8Q |
| | Tesla M6 | M6-8Q |

**Note:**

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

## Maximum vGPUs per VM

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

### Supported Hypervisor Releases

**Since 12.3:** Red Hat Enterprise Linux with KVM 8.4, 8.2, 8.1, 7,9, 7.8, and 7.7 only.

**12.2 only:** Red Hat Enterprise Linux with KVM 8.4, 8.3, 8.2, 8.1, 7,9, 7.8, and 7.7 only.

**12.0, 12.1 only:** Red Hat Enterprise Linux with KVM 8.3, 8.2, 8.1, 7,9, 7.8, and 7.7 only.

RHV 4.4, 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 are 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 time-sliced vGPUs that are allocated all of the physical GPU's frame buffer on physical GPUs that support NVLink are supported.

| GPU Architecture | Board | vGPU |
|---|---|---|
| Ampere (compute workloads only) | NVIDIA A100 HGX 80GB | A100DX-80C See Note (1). |
| | NVIDIA A100 PCIe 40GB | A100-40C |
| | NVIDIA A100 HGX 40GB | A100X-40C See Note (1). |
| Ampere (compute and graphics workloads) | NVIDIA A40 | A40-48Q |
| | | A40-48C |
| | NVIDIA A10 | A10-24Q |
| | | A10-24C |
| | NVIDIA RTX A6000 | A6000-48Q |
| | | A6000-48C |
| | NVIDIA RTX A5000 | A5000-24Q |
| | | A5000-24C |
| Turing | Quadro RTX 6000 | RTX6000-24Q |
| | | RTX6000-24C |
| | Quadro RTX 6000 passive | RTX6000P-24Q |
| | | RTX6000P-24C |

| GPU Architecture | Board | vGPU |
|---|---|---|
| | Quadro RTX 8000 | RTX8000-48Q |
| | | RTX8000-48C |
| | Quadro RTX 8000 passive | RTX8000P-48Q |
| | | RTX8000P-48C |
| Volta | Tesla V100 SXM2 32GB | V100DX-32Q |
| | | V100DX-32C |
| | Tesla V100 SXM2 | V100X-16Q |
| | | V100X-16C |
| Pascal | Tesla P100 SXM2 | P100X-16Q |
| | | P100X-16C |

**Note:**

1. Supported only on the following hardware:
   - NVIDIA HGX™ A100 4-GPU baseboard with four fully connected GPUs
   - NVIDIA HGX A100 8-GPU baseboards with eight fully connected GPUs

## Supported Hypervisor Releases

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

## Supported Guest OS Releases

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

## Limitations

- NVIDIA NVSwitch is supported only on the hardware platforms, vGPUs, and Red Hat Enterprise Linux with KVM releases listed in NVIDIA NVSwitch On-Chip Memory Fabric Support. Otherwise, only direct connections are supported.
- Only time-sliced vGPUs are supported. MIG-backed vGPUs are **not** supported.
- PCIe is not supported.
- SLI is not supported.

# 2.7. GPUDirect Technology Support

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

## Supported vGPUs

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

| GPU Architecture | Board | vGPU |
|---|---|---|
| Ampere (time-sliced and MIG-backed vGPUs) | NVIDIA A100 HGX 80GB | A100DX-80C |
| | | A100DX-7-80C |
| | NVIDIA A100 PCIe 40GB | A100-40C |
| | | A100-7-40C |
| | NVIDIA A100 HGX 40GB | A100X-40C |
| | | A100X-7-40C |
| Ampere (time-sliced vGPUs only) | NVIDIA A40 | A40-48C |
| | NVIDIA A10 | A10-24C |
| | NVIDIA RTX A6000 | A6000-48C |
| | NVIDIA RTX A5000 | A5000-24C |

## Supported Guest OS Releases

Linux only. GPUDirect technology is **not** supported on Windows.

## Supported Network Interface Cards

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

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

## Limitations

Only GPUDirect technology RDMA is supported. GPUDirect technology storage is not supported.

# 2.8. NVIDIA NVSwitch On-Chip Memory Fabric Support

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

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

## Supported Hardware Platforms

- NVIDIA HGX A100 8-GPU baseboard

## Supported vGPUs

Only C-series time-sliced vGPUs that are allocated all of the physical GPU's frame buffer on NVIDIA A100 HGX physical GPUs are supported.

| GPU Architecture | Board | vGPU |
|---|---|---|
| Ampere | NVIDIA A100 HGX 80GB | A100DX-80C |
| | NVIDIA A100 HGX 40GB | A100X-40C |

## Supported Hypervisor Releases

Red Hat Enterprise Linux with KVM 8.2 only.

## Supported Guest OS Releases

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

## Limitations

- Only time-sliced vGPUs are supported. MIG-backed vGPUs are **not** supported.
- PCIe is not supported.
- SLI is not supported.
- All vGPUs that are communicating peer-to-peer must be assigned to the same VM.

# 2.9. Since 12.2: Unified Memory Support

Unified memory is a single memory address space that is accessible from any CPU or GPU in a system. It creates a pool of managed memory that is shared between the CPU and GPU to provide a simple way to allocate and access data that can be used by code running on any CPU

or GPU in the system. Unified memory is supported only on a subset of vGPUs and guest OS releases.

**Note:** Unified memory is disabled by default. If used, you must enable unified memory individually for each vGPU that requires it by setting a vGPU plugin parameter.

### Supported vGPUs

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

| GPU Architecture | Board | vGPU |
|---|---|---|
| Ampere | NVIDIA A40 | A40-48Q |
| | | A40-48C |
| | NVIDIA A10 | A10-24Q |
| | | A10-24C |
| | NVIDIA RTX A6000 | A6000-48Q |
| | | A6000-48C |
| | NVIDIA RTX A5000 | A5000-24Q |
| | | A5000-24C |

### Supported Guest OS Releases

Linux only. Unified memory is **not** supported on Windows.

### Limitations

- Only time-sliced vGPUs are supported. MIG-backed vGPUs are **not** supported.

## 2.10. NVIDIA GPU Operator Support

NVIDIA GPU Operator simplifies the deployment of NVIDIA vGPU software with software container platforms on immutable operating systems. An immutable operating system does not allow the installation of the NVIDIA vGPU software graphics driver directly on the operating system. NVIDIA GPU Operator is supported only on specific combinations of Red Hat Enterprise Linux with KVM release, container platform, and guest OS release.

| Red Hat Enterprise Linux with KVM Release | Container Platform | Guest OS |
|---|---|---|
| **Since 12.2:** Red Hat Enterprise Linux with KVM 8.2 | Red Hat Openshift 4.7 with Red Hat Enterprise Linux CoreOS and the CRI-O container runtime | Red Hat CoreOS 4.7 |

| Red Hat Enterprise Linux with KVM Release | Container Platform | Guest OS |
|---|---|---|
| Red Hat Enterprise Linux with KVM 8.2 | Red Hat Openshift 4.6 with Red Hat Enterprise Linux CoreOS and the CRI-O container runtime | Red Hat CoreOS 4.6 |

# Chapter 3. Known Product Limitations

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

## 3.1. NVENC does not support resolutions greater than 4096×4096

### Description

The NVIDIA hardware-based H.264/HEVC video encoder (NVENC) does not support resolutions greater than 4096×4096. This restriction applies to all NVIDIA GPU architectures and is imposed by the GPU encoder hardware itself, not by NVIDIA vGPU software. The maximum supported resolution for each encoding scheme is listed in the documentation for NVIDIA Video Codec SDK. This limitation affects any remoting tool where H.264 encoding is used with a resolution greater than 4096×4096. Most supported remoting tools fall back to software encoding in such scenarios.

### Workaround

Use H.265 encoding. H.265 is more efficient than H.264 encoding and has a maximum resolution of 8192×8192.

## 3.2. 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.

## 3.3. 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.

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

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

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

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

support ECC memory are affected, even GPUs that have HBM2 memory or for which ECC memory is disabled.

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

*max-reserved-fb* = *vgpu-profile-size-in-mb*÷16 + 16 + *ecc-adjustments* + *page-retirement-allocation* + *compression-adjustment*

***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.

***ecc-adjustments***
The amount of frame buffer in Mbytes that is not usable by vGPUs when ECC is enabled on a physical GPU that does not have HBM2 memory.

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

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

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

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

***compression-adjustment***

The amount of frame buffer in Mbytes that is reserved for the higher compression overhead in vGPU types with 12 Gbytes or more of frame buffer on GPUs based on the Turing architecture.

*compression-adjustment* depends on the vGPU type as shown in the following table.

| vGPU Type | Compression Adjustment (MB) |
|---|---|
| T4-16Q<br>T4-16C<br>T4-16A | 28 |
| RTX6000-12Q<br>RTX6000-12C | 32 |

| vGPU Type | Compression Adjustment (MB) |
|---|---|
| RTX6000-12A | |
| RTX6000-24Q<br>RTX6000-24C<br>RTX6000-24A | 104 |
| RTX6000P-12Q<br>RTX6000P-12C<br>RTX6000P-12A | 32 |
| RTX6000P-24Q<br>RTX6000P-24C<br>RTX6000P-24A | 104 |
| RTX8000-12Q<br>RTX8000-12C<br>RTX8000-12A | 32 |
| RTX8000-16Q<br>RTX8000-16C<br>RTX8000-16A | 64 |
| RTX8000-24Q<br>RTX8000-24C<br>RTX8000-24A | 96 |
| RTX8000-48Q<br>RTX8000-48C<br>RTX8000-48A | 238 |
| RTX8000P-12Q<br>RTX8000P-12C<br>RTX8000P-12A | 32 |
| RTX8000P-16Q<br>RTX8000P-16C<br>RTX8000P-16A | 64 |
| RTX8000P-24Q<br>RTX8000P-24C<br>RTX8000P-24A | 96 |
| RTX8000P-48Q<br>RTX8000P-48C | 238 |

| vGPU Type | Compression Adjustment (MB) |
|---|---|
| RTX8000P-48A | |

For all other vGPU types, *compression-adjustment* is 0.

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

# 3.5. 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.

# 3.6. 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.

## 3.7. 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.

## 3.8. 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.

## 3.9. VM running an incompatible NVIDIA vGPU guest driver fails to initialize vGPU when booted

### Description

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

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

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

  In this situation, the Red Hat Enterprise Linux with KVM VM's `/var/log/messages` log file reports the following error:

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

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

  In this situation, the Red Hat Enterprise Linux with KVM VM's `/var/log/messages` log file reports the following error:

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

In either situation, the VM boots in standard VGA mode with reduced resolution and color depth. The NVIDIA virtual GPU is present in **Windows Device Manager** but displays a warning sign, and the following device status:

```
Windows has stopped this device because it has reported problems. (Code 43)
```

### Resolution

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

# 3.10. 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
```

# 3.11. 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 12.4

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

## Issues Resolved in Release 12.3

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

## Issues Resolved in Release 12.2

| Bug ID | Summary and Description |
|---|---|
| 3184762 | **12.0, 12.1 Only: Rebooting a Windows 10 vGPU VM causes a host crash**<br>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. |

## Issues Resolved in Release 12.1

| Bug ID | Summary and Description |
|---|---|
| 3230997 | **12.0 Only: Sessions freeze when Adobe Premiere with the Adobe Mercury Engine is used**<br>Sessions freeze when Adobe Premiere with the Adobe Mercury Engine is used. The session can freeze after 15-20 minutes of use or when a project is being exported. |
| 3225521 | **12.0 Only: Issues occur when Blackmagic Design DaVinci Resolve is used**<br>Multiple issues, such as application crashes, application instability, and session freezes, occur when Blackmagic Design DaVinci Resolve is used. CUDA error 702 might also be observed. |

## Issues Resolved in Release 12.0

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

# Chapter 5. Known Issues

## 5.1. Since 12.3: 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

## 5.2. 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

## 5.3. Session connection fails with four 4K displays and NVENC enabled on a 2Q, 3Q, or 4Q vGPU

### Description

Desktop session connections fail for a 2Q, 3Q, or 4Q vGPU that is configured with four 4K displays and for which the NVIDIA hardware-based H.264/HEVC video encoder (NVENC) is enabled. This issue affects only Teradici Cloud Access Software sessions on Linux guest VMs.

This issue is accompanied by the following error message:

```
This Desktop has no resources available or it has timed out
```

This issue is caused by insufficient frame buffer.

### Workaround

Ensure that sufficient frame buffer is available for all the virtual displays that are connected to a vGPU by changing the configuration in one of the following ways:

- Reducing the number of virtual displays. The number of 4K displays supported with NVENC enabled depends on the vGPU.

| vGPU | 4K Displays Supported with NVENC Enabled |
|---|---|
| 2Q | 1 |
| 3Q | 2 |
| 4Q | 3 |

- Disabling NVENC. The number of 4K displays supported with NVENC disabled depends on the vGPU.

| vGPU | 4K Displays Supported with NVENC Disabled |
|---|---|
| 2Q | 2 |
| 3Q | 2 |
| 4Q | 4 |

- Using a vGPU type with more frame buffer. Four 4K displays with NVENC enabled on any Q-series vGPU with at least 6144 MB of frame buffer are supported.

### Status

Not an NVIDIA bug

### Ref. #

200701959

## 5.4. 12.0 Only: Sessions freeze when Adobe Premiere with the Adobe Mercury Engine is used

### Description

Sessions freeze when Adobe Premiere with the Adobe Mercury Engine is used. The session can freeze after 15-20 minutes of use or when a project is being exported.

### Workaround

### Status

Resolved in NVIDIA vGPU software 12.1

### Ref. #

3230997

## 5.5. 12.0 Only: Issues occur when Blackmagic Design DaVinci Resolve is used

### Description

Multiple issues, such as application crashes, application instability, and session freezes, occur when Blackmagic Design DaVinci Resolve is used. CUDA error 702 might also be observed.

### Status

Resolved in NVIDIA vGPU software 12.1

### Ref. #

3225521

## 5.6. 12.0, 12.1 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 12.2.

### Ref. #

3184762

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

### Description

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

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

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

### Status

Open

### Ref. #

200691204

## 5.8. Idle Teradici Cloud Access Software session disconnects from Linux VM

### Description

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

### Status

Open

### Ref. #

200689126

## 5.9. GPU Operator doesn't support vGPU on GPUs based on architectures before NVIDIA Turing

### Description

NVIDIA GPU Operator doesn't support vGPU deployments on GPUs based on architectures before the NVIDIA Turing™ architecture. This issue is caused by the omission of version information for the vGPU manager from the configuration information that GPU Operator requires. Without this information, GPU Operator does not deploy the NVIDIA driver container because the container cannot determine if the driver is compatible with the vGPU manager.

### Status

Open

### Ref. #

3227576

# 5.10. Idle NVIDIA A100, NVIDIA A40, and NVIDIA A10 GPUs show 100% GPU utilization

## Description

The `nvidia-smi` command shows 100% GPU utilization for NVIDIA A100, NVIDIA A40, and NVIDIA A10 GPUs even if no vGPUs have been configured or no VMs are running. A GPU is affected by this issue only if the `sriov-manage` script has **not** been run to enable the virtual function for the GPU in the `sysfs` file system.

```
[root@host ~]# nvidia-smi
Fri Oct 29 11:45:28 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.107     Driver Version: 460.107    CUDA Version:  11.2       |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100-PCIE-40GB      On   | 00000000:5E:00.0 Off |                    0 |
| N/A   50C    P0    97W / 250W |      0MiB / 40537MiB |     100%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

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

## Workaround

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

After this workaround has been completed, the `nvidia-smi` command shows 0% GPU utilization for affected GPUs when they are idle.

```
root@host ~]# nvidia-smi
Fri Oct 29 11:47:38 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.107     Driver Version: 460.107    CUDA Version:  11.2       |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100-PCIE-40GB      On   | 00000000:5E:00.0 Off |                    0 |
| N/A   50C    P0    97W / 250W |      0MiB / 40537MiB |      0%       Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
```

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

### Status

Open

### Ref. #

200605527

## 5.11. Guest VM frame buffer listed by `nvidia-smi` for vGPUs on GPUs that support SRIOV is incorrect

### Description

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

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

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

### Status

Open

### Ref. #

200524749

# 5.12. VMs fail to boot on RHV 4.4

## Description

On RHV 4.4, VMs fail to boot with the error `Host doesn't support passthru of host PCI device`. This issue affects GPU pass through deployments with all supported GPUs and NVIDIA vGPU deployments with GPUs based on the NVIDIA Ampere architecture. This issue occurs because the `intel_iommu` parameter and the `nouveau.modeset` parameter are not set correctly.

## Version

This issue affects RHV 4.4.

## Workaround

Perform this workaround on the hypervisor host. This workaround requires root user privileges on the hypervisor host.

1. In a plain-text editor, edit the file `/boot/loader/entries/rhvh-4.4.1.1-0.20200722.0+1-4.18.0-193.13.2.el8_2.x86_64.conf` to add the following options to the boot options.
   - `nouveau.modeset=0`
   - `intel_iommu=on`

   **Note:** Line breaks have been added to this example to enhance readability.

```
title rhvh-4.4.1.1-0.20200722.0 (4.18.0-193.13.2.el8_2.x86_64)
version 4.18.0-193.13.2.el8_2.x86_64
linux //rhvh-4.4.1.1-0.20200722.0+1/vmlinuz-4.18.0-193.13.2.el8_2.x86_64
initrd //rhvh-4.4.1.1-0.20200722.0+1/initramfs-4.18.0-193.13.2.el8_2.x86_64.img
options crashkernel=auto resume=/dev/mapper/rhvh00-swap \
rd.lvm.lv=rhvh00/rhvh-4.4.1.1-0.20200722.0+1 rd.lvm.lv=rhvh00/swap \
root=/dev/rhvh00/rhvh-4.4.1.1-0.20200722.0+1 \
boot=UUID=38ff2175-b761-403d-8a91-d7ec9f7ec2f7 rootflags=discard \
img.bootid=rhvh-4.4.1.1-0.20200722.0+1 intel_iommu=on nouveau.modeset=0
id rhel-20200825140238-4.18.0-193.13.2.el8_2.x86_64
grub_users $grub_users
grub_arg --unrestricted
grub_class kernel
```

2. Reboot the hypervisor host machine.

## Status

Not an NVIDIA bug

### Ref. #

200653675

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

### Description

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

### Workaround

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

### Status

Open

### Ref. #

200633548

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

### Description

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

### Workaround

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

### Status

Open

### Ref. #

200623179

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

### Description

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

### Workaround

1. If necessary, stop the Xorg server.
   ```
   # /etc/init.d/xorg stop
   ```
2. In a plain text editor, edit the `/etc/X11/xorg.conf file` to set the options to disable DPMS and disable the screen saver.
   a). In the `Monitor` section, set the `DPMS` option to `false`.
   ```
   Option "DPMS" "false"
   ```
   b). At the end of the file, add a `ServerFlags` section that contains option to disable the screen saver.
   ```
   Section "ServerFlags"
       Option "BlankTime" "0"
     EndSection
   ```
   c). Save your changes to `/etc/X11/xorg.conf file` and quit the editor.
3. Start the Xorg server.
   ```
   # etc/init.d/xorg start
   ```

### Status

Open

### Ref. #

200605900

## 5.16. DWM crashes randomly occur in Windows VMs

### Description

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

### Version

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

### Status

Not an NVIDIA bug

### Ref. #

2730037

## 5.17. Publisher not verified warning during Windows 7 driver installation

### Description

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

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

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

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

### Version

Windows 7

### Workaround

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

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

### Status

Not a bug

## 5.18. 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

# 5.19. 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

# 5.20. 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

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

### Description

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

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

### Status

Open

### Ref. #

200381348

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

### Description

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

### Workaround

Disable monitoring of the following GPU performance statistics:

- vGPU engine usage by applications across multiple vGPUs
- Encoder session statistics
- Frame buffer capture (FBC) session statistics
- Statistics gathered by performance counters in guest VMs

### Status

Open

### Ref. #

2414897

## 5.23. 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

## 5.24. 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.

# 5.25. 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

# 5.26. 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

# 5.27. `nvidia-smi` reports that vGPU migration is supported on all hypervisors

## Description

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

### Status

Closed

### Ref. #

200407230

## 5.28. 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

## 5.29. 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

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

### Description

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

The `nvidia-gridd` service uses DBus for communication with **NVIDIA X Server Settings** to display licensing information through the **Manage License** page. Disabling the GUI for licensing resolves this issue.

To prevent this issue, the GUI for licensing is disabled by default. You might encounter this issue if you have enabled the GUI for licensing and are using Red Hat Enterprise Linux 6.8 or 6.9, or CentOS 6.8 and 6.9.

### Version

Red Hat Enterprise Linux 6.8 and 6.9

CentOS 6.8 and 6.9

### Status

Open

### Ref. #

- 200358191
- 200319854
- 1895945

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

## Description

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

## Workaround

This workaround requires `sudo` privileges.

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

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

1. If **NVIDIA X Server Settings** is running, shut it down.
2. If the `/etc/nvidia/gridd.conf` file does not already exist, create it by copying the supplied template file `/etc/nvidia/gridd.conf.template`.
3. As root, edit the `/etc/nvidia/gridd.conf` file to set the `EnableUI` option to `TRUE`.
4. Start the `nvidia-gridd` service.

```
# sudo service nvidia-gridd start
```

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

## Status

Open

# 5.32. 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

# 5.33. 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

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

## VESA DisplayPort

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

## HDMI

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

## OpenCL

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

## Trademarks

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



NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com