Virtual GPU Software R470 for Ubuntu

Release Notes

RN-10660-001 _v13.0 through 13.10 | February 2024

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 Ubuntu.

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
13.10	470.239.01	474.82	470.239.06
13.9	470.223.02	474.64	470.223.02
13.8	470.199.03	474.44	470.199.02
13.7	470.182.02	474.30	470.182.03
13.6	470.161.02	474.14	470.161.03
13.5	470.161.02	474.04	470.161.03
13.4	470.141.05	473.81	470.141.03
13.3	470.129.04	473.47	470.129.06
13.2	470.103.02	472.98	470.103.01
13.1	470.82	472.39	470.82.01
13.0	Not supported	Not supported	Not supported

For details of which Ubuntu 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: You must use NVIDIA License System with every release in this release family of NVIDIA vGPU software. The legacy NVIDIA vGPU software license server has reached end of life (EOL) and is no longer supported.

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
- NVIDIA vGPU Manager from a later long-term support branch with guest VM drivers from the previous long-term support 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 13.10 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 13.10 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 13 major release branch.

NVIDIA vGPU Software Component	Releases	Compatible Software Releases
NVIDIA vGPU Manager	13.0 through 13.10	▶ Guest VM driver releases 13.0 through 13.10 ▶ All guest VM driver 12.*x* releases ▶ All guest VM driver 11.*x* releases
Guest VM drivers	13.0 through 13.10	NVIDIA vGPU Manager releases 13.0 through 13.10

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 a production branch 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 13 major release branch.

NVIDIA vGPU Software Component	Releases	Incompatible Software Releases
NVIDIA vGPU Manager	13.0 through 13.10	All guest VM driver releases 10.*x* and earlier
Guest VM drivers	13.0 through 13.10	All NVIDIA vGPU Manager releases 12.*x* and earlier

1.3. Updates in Release 13.10

New Features in Release 13.10

- Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - February 2024*, 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 13.9

New Features in Release 13.9

- Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - October 2023*, 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 13.8

New Features in Release 13.8

- Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - June 2023*, which is posted shortly after the release date of this software and is listed on the NVIDIA Product Security page
- Miscellaneous bug fixes

1.6. Updates in Release 13.7

New Features in Release 13.7

- Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - March 2023*, which is posted shortly after the release date of this software and is listed on the NVIDIA Product Security page
- Miscellaneous bug fixes

1.7. Updates in Release 13.6

New Features in Release 13.6

- Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - November 2022*, which is updated shortly after the release date of this software and is listed on the NVIDIA Product Security page

1.8. Updates in Release 13.5

New Features in Release 13.5

- Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - November 2022*, which is posted shortly after the release date of this software and is listed on the NVIDIA Product Security page
- Support for non-transparent local proxy servers when NVIDIA vGPU software is served licenses by a Cloud License Service (CLS) instance
- Miscellaneous bug fixes

1.9. Updates in Release 13.4

New Features in Release 13.4

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

1.10. Updates in Release 13.3

New Features in Release 13.3

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

1.11. Updates in Release 13.2

New Features in Release 13.2

- 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

1.12. Updates in Release 13.1

New Features in Release 13.1

- Support for CUDA profilers on vGPUs on the following GPUs:
  - NVIDIA A40
  - NVIDIA A16
  - NVIDIA A10
  - NVIDIA RTX A6000
  - NVIDIA RTX A5000
- NVIDIA Deep Learning Super Sampling (DLSS) support on NVIDIA RTX Virtual Workstation
- Security updates - see *Security Bulletin: NVIDIA GPU Display Driver - October 2021*, which is available on 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 13.1

- Support for the following releases of Ubuntu as a hypervisor:
  - Ubuntu 20.04 LTS
  - Ubuntu 18.04 LTS

1.13. Updates in Release 13.0

This release is not supported on Ubuntu.

Chapter 2. Validated Platforms

This release family of NVIDIA vGPU software provides support for several NVIDIA GPUs on validated server hardware platforms, Ubuntu hypervisor software versions, and guest operating systems. It also supports the version of NVIDIA CUDA Toolkit that is compatible with R470 drivers.

2.1. Supported NVIDIA GPUs and Validated Server Platforms

This release of NVIDIA vGPU software on Ubuntu provides support for several NVIDIA GPUs running on validated server hardware platforms. For a list of validated server platforms, refer to NVIDIA Virtual GPU Certified Servers.

The supported products for each type of NVIDIA vGPU software deployment depend on the GPU.

GPUs Based on the NVIDIA Ampere Architecture

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
NVIDIA A100 PCIe 80GB	vCS	vCS	vCS
NVIDIA A100 HGX 80GB	vCS	vCS	vCS
NVIDIA A100 PCIe 40GB	vCS	vCS	vCS
NVIDIA A100 HGX 40GB	vCS	vCS	vCS
NVIDIA A40[5]	▶ vCS ▶ vWS	N/A	▶ vCS ▶ vWS

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
	▶ vPC ▶ vApps		▶ vApps
NVIDIA A30	vCS	vCS	vCS
NVIDIA A16	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
NVIDIA A10	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
NVIDIA RTX A6000[5]	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
NVIDIA RTX A5000[5]	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps

GPUs Based on the NVIDIA Turing™ Architecture

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
Tesla T4	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Quadro RTX 6000[5]	▶ vCS	N/A	▶ vCS

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
	▶ vWS ▶ vPC ▶ vApps		▶ vWS ▶ vApps
Quadro RTX 6000 passive[5]	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Quadro RTX 8000 [5]	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Quadro RTX 8000 passive[5]	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps

GPUs Based on the NVIDIA Volta Architecture

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
Tesla V100 SXM2	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Tesla V100 SXM2 32GB	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Tesla V100 PCIe	▶ vCS	N/A	▶ vCS

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
	▶ vWS ▶ vPC ▶ vApps		▶ vWS ▶ vApps
Tesla V100 PCIe 32GB	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Tesla V100S PCIe 32GB	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Tesla V100 FHHL	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps

GPUs Based on the NVIDIA Pascal™ Architecture

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
Tesla P4	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Tesla P6	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Tesla P40	▶ vCS	N/A	▶ vCS

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
	▶ vWS ▶ vPC ▶ vApps		▶ vWS ▶ vApps
Tesla P100 PCIe 16 GB	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Tesla P100 SXM2 16 GB	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps
Tesla P100 PCIe 12GB	▶ vCS ▶ vWS ▶ vPC ▶ vApps	N/A	▶ vCS ▶ vWS ▶ vApps

GPUs Based on the NVIDIA Maxwell™ Graphic Architecture

Note: NVIDIA Virtual Compute Server (vCS) is **not** supported on GPUs based on the NVIDIA Maxwell graphic architecture.

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
Tesla M6	▶ vWS ▶ vPC ▶ vApps	N/A	▶ vWS ▶ vApps
Tesla M10	▶ vWS ▶ vPC ▶ vApps	N/A	▶ vWS ▶ vApps

GPU	Supported NVIDIA vGPU Software Products[1,2,3,4]		
	Time-Sliced NVIDIA vGPU	MIG-Backed NVIDIA vGPU	GPU Pass Through
Tesla M60	▶ vWS ▶ vPC ▶ vApps	N/A	▶ vWS ▶ vApps

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

[1] The supported products are as follows:

- vCS: NVIDIA Virtual Compute Server
- vWS: NVIDIA RTX Virtual Workstation
- vPC: NVIDIA Virtual PC
- vApps: NVIDIA Virtual Applications

[2] N/A indicates that the deployment is not supported.
[3] vCS is supported only on Linux operating systems.
[4] vApps is supported only on Windows operating systems.
[5] This GPU is supported only in displayless mode. In displayless mode, local physical display connectors are disabled.

- 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
Ubuntu	20.04 LTS	All NVIDIA GPUs that NVIDIA vGPU software supports are supported with vGPU and in pass-through mode, **except** on systems that are based on NVIDIA® NVSwitch™ on-chip memory fabric.
Ubuntu	18.04 LTS	Support is limited to HWE kernels 5.4.0-77 and later. All NVIDIA GPUs that NVIDIA vGPU software supports are supported

Software	Releases Supported	Notes
		with vGPU and in pass-through mode, **except** on systems that are based on NVIDIA NVSwitch on-chip memory fabric.

2.3. Guest OS Support

NVIDIA vGPU software supports several 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. Linux Guest OS Support

NVIDIA vGPU software supports **only** the 64-bit Linux distributions listed in the table as a guest OS on Ubuntu. The releases of Ubuntu 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 - Ubuntu Releases	Pass-Through GPU - Ubuntu Releases
Ubuntu 20.04 LTS	20.04, 18.04	20.04, 18.04
Ubuntu 18.04 LTS	20.04, 18.04	20.04, 18.04

2.4. NVIDIA CUDA Toolkit Version Support

The releases in this release family of NVIDIA vGPU software support NVIDIA CUDA Toolkit 11.4.

To build a CUDA application, the system must have the NVIDIA CUDA Toolkit and the libraries required for linking. For details of the components of NVIDIA CUDA Toolkit, refer to *NVIDIA CUDA Toolkit Release Notes for CUDA 11.4*.

To run a CUDA application, the system must have a CUDA-enabled GPU and an NVIDIA display driver that is compatible with the NVIDIA CUDA Toolkit release that was used to build the application. If the application relies on dynamic linking for libraries, the system must also have the correct version of these libraries.

For more information about NVIDIA CUDA Toolkit, refer to CUDA Toolkit 11.4 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 Ubuntu 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 PCIe 80GB	A100D-80C See Note (1).
	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).

GPU Architecture	Board	vGPU
	NVIDIA A30	A30-24C See Note (1).
Ampere (compute and graphics workloads)	NVIDIA A40	A40-48Q See Note (1).
		A40-48C See Note (1).
	NVIDIA A16	A16-16Q See Note (1).
		A16-16C 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
		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

GPU Architecture	Board	vGPU
		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 Ubuntu.

Supported Hypervisor Releases

Ubuntu 20.04 LTS, 18.04 LTS

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, Ubuntu 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 PCIe 80GB	A100D-80C
	NVIDIA A100 HGX 80GB	A100DX-80C See Note (1).
	NVIDIA A100 PCIe 40GB	A100-40C
	NVIDIA A100 HGX 40GB	A100X-40C See Note (1).
	NVIDIA A30	A30-24C
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
	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

   Fully connected means that each GPU is connected to every other GPU on the baseboard.

Supported Hypervisor Releases

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

Supported Guest OS Releases

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

Limitations

- Only direct connections are supported. NVSwitch is not supported.
- Only time-sliced vGPUs are supported. MIG-backed vGPUs are **not** supported.
- 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 PCIe 80GB	A100D-80C
		A100D-7-80C
	NVIDIA A100 HGX 80GB	A100DX-80C
		A100DX-7-80C
	NVIDIA A100 PCIe 40GB	A100-40C
		A100-7-40C
	NVIDIA A100 HGX 40GB	A100X-40C

GPU Architecture	Board	vGPU
		A100X-7-40C
	NVIDIA A30	A30-4-24C
		A30-24C
Ampere (time-sliced vGPUs only)	NVIDIA A40	A40-48C
	NVIDIA A16	A16-16C
	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. 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 A16	A16-16Q
		A16-16C
	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.
- When unified memory is enabled for a VM, NVIDIA CUDA Toolkit profilers are disabled.

2.9. Since 13.1: NVIDIA Deep Learning Super Sampling (DLSS) Support

NVIDIA vGPU software supports NVIDIA DLSS on NVIDIA RTX Virtual Workstation.

Supported DLSS versions: 2.0. Version 1.0 is **not** supported.

Supported GPUs:

- NVIDIA A40
- NVIDIA A16
- NVIDIA A10
- NVIDIA RTX A6000
- NVIDIA RTX A5000
- Tesla T4
- Quadro RTX 8000
- Quadro RTX 8000 passive
- Quadro RTX 6000

- Quadro RTX 6000 passive

Note: NVIDIA graphics driver components that DLSS requires are installed only if a supported GPU is detected during installation of the driver. Therefore, if the creation of VM templates includes driver installation, the template should be created from a VM that is configured with a supported GPU while the driver is being installed.

Supported applications: only applications that use `nvngx_dlss.dll` version 2.0.18 or newer

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 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

If your GPU is based on a GPU architecture later than the NVIDIA Maxwell® architecture, use H.265 encoding. H.265 is more efficient than H.264 encoding and has a maximum resolution of 8192×8192. On GPUs based on the NVIDIA Maxwell architecture, H.265 has the same maximum resolution as H.264, namely 4096×4096.

Note: Resolutions greater than 4096×4096 are supported only by the H.265 decoder that 64-bit client applications use. The H.265 decoder that 32-bit applications use supports a maximum resolution of 4096×4096.

3.2. Nested Virtualization Is Not Supported by NVIDIA vGPU

NVIDIA vGPU deployments do not support nested virtualization, that is, running a hypervisor in a guest VM. For example, enabling the Hyper-V role in a guest VM running the Windows Server OS is **not** supported because it entails enabling nested virtualization. Similarly, enabling Windows Hypervisor Platform is not supported because it requires the Hyper-V role to be enabled.

3.3. 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.

3.4. 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.

3.5. 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 T4-16C T4-16A	28
RTX6000-12Q RTX6000-12C RTX6000-12A	32
RTX6000-24Q RTX6000-24C RTX6000-24A	104
RTX6000P-12Q RTX6000P-12C RTX6000P-12A	32
RTX6000P-24Q RTX6000P-24C RTX6000P-24A	104
RTX8000-12Q	32

vGPU Type	Compression Adjustment (MB)
RTX8000-12C RTX8000-12A	
RTX8000-16Q RTX8000-16C RTX8000-16A	64
RTX8000-24Q RTX8000-24C RTX8000-24A	96
RTX8000-48Q RTX8000-48C RTX8000-48A	238
RTX8000P-12Q RTX8000P-12C RTX8000P-12A	32
RTX8000P-16Q RTX8000P-16C RTX8000P-16A	64
RTX8000P-24Q RTX8000P-24C RTX8000P-24A	96
RTX8000P-48Q RTX8000P-48C RTX8000P-48A	238

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

3.6. 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.

3.7. 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.

3.8. 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 Ubuntu 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 Ubuntu 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 Ubuntu 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 Ubuntu 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 13.10

No resolved issues are reported in this release for Ubuntu.

Issues Resolved in Release 13.9

No resolved issues are reported in this release for Ubuntu.

Issues Resolved in Release 13.8

Bug ID	Summary and Description
3596327	13.0-13.7 Only: Remote desktop connection is lost and the NVIDIA vGPU software graphics driver is unloaded The remote desktop connection is lost and the NVIDIA vGPU software graphics driver is unloaded after an attempt to access a VM over RDP and VMware Horizon agent direct connect. After an attempt to log in again, a black screen is displayed.

Issues Resolved in Release 13.7

No resolved issues are reported in this release for Ubuntu.

Issues Resolved in Release 13.6

No resolved issues are reported in this release for Ubuntu.

Issues Resolved in Release 13.5

Bug ID	Summary and Description
3658686	13.0-13.4 Only: VMs configured with a vGPU based on the NVIDIA Ampere architecture can become slow to respond

Bug ID	Summary and Description
	VMs configured with a vGPU on a GPU that is based on the NVIDIA Ampere GPU architecture can become slow to respond. When this error occurs, multiple `XID error 62` and `XID error 45` messages are written to the log file on the hypervisor host.

Issues Resolved in Release 13.4

No resolved issues are reported in this release for Ubuntu.

Issues Resolved in Release 13.3

Bug ID	Summary and Description
200756399	**13.0-13.2 Only: Linux VM might fail to return a license after shutdown if the license server is specified by its name** If the license server is specified by its fully qualified domain name, a Linux VM might fail to return its license when the VM is shut down. This issue occurs if the `nvidia-gridd` service cannot resolve the fully qualified domain name of the license server because `systemd-resolved.service` is not available when the service attempts to return the license. When this issue occurs, the `nvidia-gridd` service writes the following message to the `systemd` journal: `General data transfer failure. Couldn't resolve host name`
200724807	**13.0-13.2 Only: Memory leaks in the vGPU manager plugin cause the VM to hang** Applications running in a VM request memory to be allocated and freed by the vGPU manager plugin, which runs on the hypervisor host. When an application requests the vGPU manager plugin to free previously allocated memory, some of the memory is not freed. Some applications request memory more frequently than other applications. If such applications run for a long period of time, for example for two or more days, the failure to free all allocated memory might cause the hypervisor host to run out of memory. As a result, memory allocation for applications running in the VM might fail, causing the applications and, sometimes, the VM to hang.

Issues Resolved in Release 13.2

Bug ID	Summary and Description
3513019	**13.1 Only: Hypervisor host randomly freezes when multiple vGPU VMs are running**

Bug ID	Summary and Description
	The hypervisor host randomly freezes when multiple VMs configured with vGPUs on GPUs based on the NVIDIA Ampere architecture are running. When the host freezes, CPU usage inreases sharply. To recover from the freeze, the host must be rebooted.

Issues Resolved in Release 13.1

No resolved issues are reported in this release for Ubuntu.

Issues Resolved in Release 13.0

No resolved issues are reported in this release for Ubuntu.

Chapter 5. Known Issues

5.1. Frame buffer seems to be missing from GPUs

Description

On a host on which the Virtual GPU Manager is installed, GPU management tools, such as the `nvidia-smi` command, give the impression that some portion of a GPU's frame buffer is missing. For example, the NVIDIA A16 GPU has 16 GB of frame buffer, but total frame buffer is shown as 15.745 GB. This issue occurs because the Virtual GPU Manager does not report frame buffer that it has reserved for its own purposes, only the frame buffer that is available for applications.

Version

This issue affects only releases in the NVIDIA vGPU software 13 branch.

Status

Closed

Ref.

4266954

5.2. Graphics applications are corrupted on some Windows vGPU VMs

Description

Graphics applications are corrupted on Windows VMs that are configured with one or more vGPUs that are based on the NVIDIA Ampere or NVIDIA Ada Lovelace GPU architecture.

Status

Open

Ref.

3641947

5.3. 13.0-13.7 Only: Remote desktop connection is lost and the NVIDIA vGPU software graphics driver is unloaded

Description

The remote desktop connection is lost and the NVIDIA vGPU software graphics driver is unloaded after an attempt to access a VM over RDP and VMware Horizon agent direct connect. After an attempt to log in again, a black screen is displayed.

When this issue occurs, the following errors are written to the log files on the guest VM:

- A timeout detection and recovery (TDR) error:

```
vmiop_log: (0x0): Timeout occurred, reset initiated.
vmiop_log: (0x0): TDR_DUMP:0x52445456 0x006907d0 0x000001cc 0x00000001
```

- XID error 43:

```
vmiop_log: (0x0): XID 43 detected on physical_chid
```

- vGPU error 22:

```
vmiop_log: (0x0): VGPU message 22 failed
```

- Guest driver unloaded error:

```
vmiop_log: (0x0): Guest driver unloaded!
```

Workaround

To recover from this issue, reboot the VM.

Since 13.7: To prevent this issue from occurring, disable translation lookaside buffer (TLB) invalidation by setting the vGPU plugin parameter `tlb_invalidate_enabled` to 0.

Status

Resolved in NVIDIA vGPU software 13.8

Ref.

3596327

5.4. 13.0-13.4 Only: VMs configured with a vGPU based on the NVIDIA Ampere architecture can become slow to respond

Description

VMs configured with a vGPU on a GPU that is based on the NVIDIA Ampere GPU architecture can become slow to respond. When this error occurs, multiple `XID error 62` and `XID error 45` messages are written to the log file on the hypervisor host.

Status

Resolved in NVIDIA vGPU software 13.5

Ref.

3658686

5.5. NLS client fails to acquire a license with the error `The allowed time to process response has expired`

Description

A licensed client of NVIDIA License System (NLS) fails to acquire a license with the error `The allowed time to process response has expired`. This error can affect clients of a Cloud License Service (CLS) instance or a Delegated License Service (DLS) instance.

This error occurs when the time difference between the system clocks on the client and the server that hosts the CLS or DLS instance is greater than 10 minutes. A common cause of this error is the failure of either the client or the server to adjust its system clock when daylight savings time begins or ends. The failure to acquire a license is expected to prevent clock windback from causing licensing errors.

Workaround

Ensure that system clock time of the client and any server that hosts a DLS instance match the current time in the time zone where they are located.

To prevent this error from occurring when daylight savings time begins or ends, enable the option to automatically adjust the system clock for daylight savings time:

- **Windows:** Set the **Adjust for daylight saving time automatically** option.
- **Linux:** Use the `hwclock` command.

Status

Not a bug

Ref.

3859889

5.6. NVIDIA vGPU software graphics driver fails to load on KVM-based hypervsiors

Description

The NVIDIA vGPU software graphics driver fails to load on hypervsiors based on Linux with KVM. This issue affects UEFI VMs configured with a vGPU or pass-through GPU that requires a large BAR address space. This issue does not affect VMs that are booted in legacy BIOS mode. The issue occurs because BAR resources are not mapped into the VM.

Workaround

1. In `virsh`, open for editing the XML document of the VM to which the vGPU or GPU is assigned.

   ```
   # virsh edit vm-name
   ```

   vm-name

   The name of the VM to which the vGPU or GPU is assigned.

2. Declare the custom `libvirt` XML namespace that supports command-line pass through of QEMU arguments.

   Declare this namesapce by modifying the start tag of the top-level `domain` element in the first line of the XML document.

   ```
   <domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
   ```

3. At the end of the XML document, between the `</devices>` end tag and the `</domain>` end tag, add the highlighted `qemu` elements.

   These elements pass the QEMU arguments for mapping the required BAR resources into the VM.

   ```
   </devices>
     <qemu:commandline>
       <qemu:arg value='-fw_cfg'/>
       <qemu:arg value='opt/ovmf/X-PciMmio64Mb,string=262144'/>
     </qemu:commandline>
   </domain>
   ```

4. Start the VM to which the vGPU or GPU is assigned.

   ```
   # virsh start vm-name
   ```

   vm-name

   The name of the VM to which the vGPU or GPU is assigned.

Status

Not an NVIDIA bug

Ref.

200719557

5.7. VP9 and AV1 decoding with web browsers are not supported on Microsoft Windows Server 2019

Description

VP9 and AV1 decoding with web browsers are not supported on Microsoft Windows Server 2019. This issue occurs because starting with Windows Server 2019, the required codecs are not included with the OS and are not available through the **Microsoft Store** app. As a result, hardware decoding is not available for viewing YouTube videos or using collaboration tools such as Google Meet in a web browser.

Version

This issue affects Microsoft Windows Server releases starting with Windows Server 2019.

Status

Not an NVIDIA bug

Ref.

200756564

5.8. 13.0-13.2 Only: Linux VM might fail to return a license after shutdown if the license server is specified by its name

Description

If the license server is specified by its fully qualified domain name, a Linux VM might fail to return its license when the VM is shut down. This issue occurs if the `nvidia-gridd` service cannot resolve the fully qualified domain name of the license server because `systemd-resolved.service` is not available when the service attempts to return the

license. When this issue occurs, the `nvidia-gridd` service writes the following message to the `systemd` journal:

```
General data transfer failure. Couldn't resolve host name
```

Status

Resolved in NVIDIA vGPU software 13.3

Ref.

200756399

5.9. 13.0-13.2 Only: Memory leaks in the vGPU manager plugin cause the VM to hang

Description

Applications running in a VM request memory to be allocated and freed by the vGPU manager plugin, which runs on the hypervisor host. When an application requests the vGPU manager plugin to free previously allocated memory, some of the memory is not freed. Some applications request memory more frequently than other applications. If such applications run for a long period of time, for example for two or more days, the failure to free all allocated memory might cause the hypervisor host to run out of memory. As a result, memory allocation for applications running in the VM might fail, causing the applications and, sometimes, the VM to hang.

When memory allocation fails, the error messages that are written to the log file on the hypervisor host depend on the hypervisor.

- For VMware vSphere ESXi, the following error messages are written to `vmware.log`:

```
2021-10-05T04:57:35.547Z| vthread-2329002| E110: vmiop_log: Fail to create the
 buffer for translate pte rpc node

2021-06-05T10:48:33.007Z| vcpu-3| E105: PANIC: Unrecoverable memory allocation
 failure
```

- For Citrix Hypervisor and hypervisors based on Linux KVM, the following messages are written to the standard activity log in the `/var/log` directory (`/var/log/messages` or `/var/log/syslog`):

```
Feb 15 09:27:48 bkrzxen1 kernel: [1278743.170072] Out of memory: Kill process
 20464 (vgpu) score 9 or sacrifice child

Feb 15 09:27:48 bkrzxen1 kernel: [1278743.170111] Killed process 20464 (vgpu)
 total-vm:305288kB, anon-rss:56508kB, file-rss:30828kB, shmem-rss:0kB

Feb 15 09:27:48 bkrzxen1 kernel: [1278743.190484] oom_reaper: reaped process
 20464 (vgpu), now anon-rss:0kB, file-rss:27748kB, shmem-rss:4kB".
```

Workaround

If an application or a VM hangs after a long period of usage, restart the VM every couple of days to prevent the hypervisor host from running out of memory.

Status

Resolved in NVIDIA vGPU software 13.3

Ref.

200724807

5.10. 13.1 Only: Hypervisor host randomly freezes when multiple vGPU VMs are running

Description

The hypervisor host randomly freezes when multiple VMs configured with vGPUs on GPUs based on the NVIDIA Ampere architecture are running. When the host freezes, CPU usage inreases sharply. To recover from the freeze, the host must be rebooted.

Status

Resolved in NVIDIA vGPU software 13.2

Ref.

3513019

5.11. 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

5.12. 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

5.13. 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

5.14. 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

5.15. 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

5.16. 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 Feb 23 11:45:28 2024
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.239.01   Driver Version: 470.239.01   CUDA Version:  11.4     |
|-------------------------------+----------------------+----------------------+
| 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 Feb 23 11:47:38 2024
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.239.01   Driver Version: 470.239.01   CUDA Version:  11.4     |
|-------------------------------+----------------------+----------------------+
| 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

5.17. 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

5.18. 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

5.19. 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

5.20. 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

5.21. 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

5.22. 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. 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

5.25. `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

5.26. 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

5.27. 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

5.28. 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

5.29. 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

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