Virtual GPU Software R535 for Microsoft Azure Stack HCI

Release Notes

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 Microsoft Azure Stack HCI.

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
16.5	538.33	538.46	535.161.08
16.4	538.33	538.33	535.161.07
16.3	538.15	538.15	535.154.05
16.2	537.70	537.70	535.129.03
16.1	537.13	537.13	535.104.05
16.0	536.22	536.25	535.54.03

For details of which Microsoft Azure Stack HCI 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. All releases in this release family of NVIDIA vGPU software are **incompatible** with all releases of the NVIDIA vGPU software 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 16.5 is used with guest drivers from release 13.1, the combination does **not** support Red Hat Enterprise Linux 8.1 because NVIDIA vGPU software release 16.5 does not support Red Hat Enterprise Linux 8.1.

The following table lists the specific software releases that are compatible with the components in the NVIDIA vGPU software 16 major release branch.

NVIDIA vGPU Software Component	Releases	Compatible Software Releases
NVIDIA vGPU Manager	16.0 through 16.5	▶ Guest VM driver releases 16.0 through 16.5 ▶ All guest VM driver 15.*x* releases
Guest VM drivers	16.0 through 16.5	NVIDIA vGPU Manager releases 16.0 through 16.5

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 16 major release branch.

NVIDIA vGPU Software Component	Releases	Incompatible Software Releases
NVIDIA vGPU Manager	16.0 through 16.5	All guest VM driver releases 14.*x* and earlier
Guest VM drivers	16.0 through 16.5	All NVIDIA vGPU Manager releases 15.*x* and earlier

1.3. Updates in Release 16.5

NVIDIA vGPU software 16.5 resolves an issue that affects graphics cards that are supported only by NVIDIA AI Enterprise.

1.4. Updates in Release 16.4

New Features in Release 16.4

- 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

Newly Supported Hardware and Software in Release 16.4

- Newly supported guest OS releases:
  - Microsoft Windows 11 23H2

1.5. Updates in Release 16.3

New Features in Release 16.3

- Miscellaneous bug fixes

Hardware and Software Support Introduced in Release 16.3

- Newly supported graphics cards:
  - NVIDIA L2
  - NVIDIA L20
- Newly supported guest OS releases:
  - Red Hat Enterprise Linux 8.9

1.6. Updates in Release 16.2

New Features in Release 16.2

- 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

Hardware and Software Support Introduced in Release 16.2

- Newly supported hypervisor software:
  - Microsoft Azure Stack HCI 23H2 preview

1.7. Updates in Release 16.1

New Features in Release 16.1

- New options in the NVML API and the `nvidia-smi` command for getting the scheduling behavior of time-sliced vGPUs
- Miscellaneous bug fixes

Hardware and Software Support Introduced in Release 16.1

- Support for the for the following GPUs:
  - NVIDIA L40S

1.8. Updates in Release 16.0

New Features in Release 16.0

- Support for the NVIDIA L4 and NVIDIA L40 graphics cards in DDA mode only
- Support for 4K displays with an aspect ratio of 16:10
- Options in the NVML API and the `nvidia-smi` command for controlling the scheduling behavior of time-sliced vGPUs
- Assignment of multiple fractional vGPUs to a single VM

  A fractional vGPU is allocated only a fraction of the physical GPU's frame buffer.
- 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

Newly Supported Hardware and Software in Release 16.0

- Newly supported guest OSes:
  - Red Hat Enterprise Linux 8.8

Feature Support Withdrawn in Release 16.0

- Graphics cards no longer supported:
    - Graphics cards that support only C-series vGPUs, namely:
        - NVIDIA H800 PCIe 80GB
        - NVIDIA H100 PCIe 80GB
        - NVIDIA A800 PCIe 80GB
        - NVIDIA A800 PCIe 80GB liquid cooled
        - NVIDIA A800 HGX 80GB
        - NVIDIA A100 PCIe 80GB
        - NVIDIA A100 PCIe 80GB liquid cooled
        - NVIDIA A100X
        - NVIDIA A100 HGX 80GB
        - NVIDIA A100 PCIe 40GB
        - NVIDIA A100 HGX 40GB
        - NVIDIA A30
        - NVIDIA A30X

  Instead, these graphics cards are supported with NVIDIA AI Enterprise.

Chapter 2. Validated Platforms

This release family of NVIDIA vGPU software provides support for several NVIDIA GPUs on validated server hardware platforms, Microsoft Azure Stack HCI hypervisor software versions, and guest operating systems. It also supports the version of NVIDIA CUDA Toolkit that is compatible with R535 drivers.

2.1. Supported NVIDIA GPUs and Validated Server Platforms

This release of NVIDIA vGPU software on Microsoft Azure Stack HCI 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 Ada Lovelace Architecture

	Supported NVIDIA vGPU Software Products[1], [2], [3]	
GPU	**NVIDIA vGPU**	**DDA**
Since 16.1: NVIDIA L40S	N/A	vWS vApps
NVIDIA L40	N/A	vWS vApps
Since 16.3: NVIDIA L20	N/A	vWS vApps
NVIDIA L4	N/A	vWS vApps

GPU	Supported NVIDIA vGPU Software Products[1],[2],[3]	
	NVIDIA vGPU	DDA
Since 16.3: NVIDIA L2	N/A	vWS vApps

GPUs Based on the NVIDIA Ampere Architecture

GPU	Supported NVIDIA vGPU Software Products[1],[2],[3]	
	NVIDIA vGPU	DDA
NVIDIA A40[4]	vWS vPC vApps	vWS vApps
NVIDIA A16	vWS vPC vApps	vWS vApps
NVIDIA A10	vWS vPC vApps	vWS vApps
NVIDIA A2	vWS vPC vApps	vWS vApps

2.1.1. Support for a Mixture of Time-Sliced vGPU Types on the Same GPU

Microsoft Azure Stack HCI supports time-sliced vGPUs with the same amount of frame buffer from different virtual GPU series on the same physical GPU. A-series, B-series, and

[1] The supported products are as follows:

- vWS: NVIDIA RTX Virtual Workstation
- vPC: NVIDIA Virtual PC
- vApps: NVIDIA Virtual Applications

[2] N/A indicates that the deployment is not supported.

[3] vApps is supported only on Windows operating systems.

[4] This GPU is supported only in displayless mode. In displayless mode, local physical display connectors are disabled.

Q-series vGPUs with the same amount of frame buffer, for example A40-2B and A40-2Q, can reside on the same physical GPU simultaneously.

2.2. Hypervisor Software Releases

This release of NVIDIA vGPU software is supported on 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
Microsoft Azure Stack HCI	▶ **Since 16.2:** 23H2 preview ▶ 22H2	The following GPUs are supported in Microsoft DDA deployments only: ▶ **Since 16.1:** NVIDIA L40S ▶ NVIDIA L40 ▶ **Since 16.3:** NVIDIA L20 ▶ NVIDIA L4 ▶ **Since 16.3:** NVIDIA L2

2.3. Guest OS Support

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

NVIDIA vGPU software supports **only** the 64-bit Windows releases listed as a guest OS on Microsoft Azure Stack HCI. The releases of Microsoft Azure Stack HCI 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.

2.3.1.1. Windows Guest OS Support in Release 16.5

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Windows Server 2022	22H2, 23H2 preview	22H2, 23H2 preview
Windows Server 2019	22H2, 23H2 preview	22H2, 23H2 preview
Windows 11 23H2 and all Windows 11 releases supported by Microsoft up to and including this release	22H2, 23H2 preview	22H2, 23H2 preview
Windows 10 May 2020 Update (2004)	22H2, 23H2 preview	22H2, 23H2 preview

Note:

1. 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 in pass-through mode on GPUs based on later architectures.

2.3.1.2. Windows Guest OS Support in Release 16.4

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Windows Server 2022	22H2, 23H2 preview	22H2, 23H2 preview
Windows Server 2019	22H2, 23H2 preview	22H2, 23H2 preview
Windows 11 23H2 and all Windows 11 releases supported by Microsoft up to and including this release	22H2, 23H2 preview	22H2, 23H2 preview
Windows 10 May 2020 Update (2004)	22H2, 23H2 preview	22H2, 23H2 preview

Note:

1. 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 in pass-through mode on GPUs based on later architectures.

2.3.1.3. Windows Guest OS Support in Release 16.3

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Windows Server 2022	22H2, 23H2 preview	22H2, 23H2 preview

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Windows Server 2019	22H2, 23H2 preview	22H2, 23H2 preview
Windows 11 22H2 and all Windows 11 releases supported by Microsoft up to and including this release	22H2, 23H2 preview	22H2, 23H2 preview
Windows 10 May 2020 Update (2004)	22H2, 23H2 preview	22H2, 23H2 preview

Note:

1. 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 in pass-through mode on GPUs based on later architectures.

2.3.1.4. Windows Guest OS Support in Release 16.2

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Windows Server 2022	22H2, 23H2 preview	22H2, 23H2 preview
Windows Server 2019	22H2, 23H2 preview	22H2, 23H2 preview
Windows 11 22H2 and all Windows 11 releases supported by Microsoft up to and including this release	22H2, 23H2 preview	22H2, 23H2 preview
Windows 10 May 2020 Update (2004)	22H2, 23H2 preview	22H2, 23H2 preview

Note:

1. 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 in pass-through mode on GPUs based on later architectures.

2.3.1.5. Windows Guest OS Support in Release 16.1

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Windows Server 2022	22H2	22H2
Windows Server 2019	22H2	22H2
Windows 11 22H2 and all Windows 11 releases supported by Microsoft up to and including this release	22H2	22H2

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Windows 10 May 2020 Update (2004)	22H2	22H2

Note:

1. 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 in pass-through mode on GPUs based on later architectures.

2.3.1.6. Windows Guest OS Support in Release 16.0

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Windows Server 2022	22H2	22H2
Windows Server 2019	22H2	22H2
Windows 11 22H2 and all Windows 11 releases supported by Microsoft up to and including this release	22H2	22H2
Windows 10 May 2020 Update (2004)	22H2	22H2

Note:

1. 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 in pass-through mode on GPUs based on later architectures.

2.3.2. Linux Guest OS Support

NVIDIA vGPU software supports **only** the Linux distributions listed as a guest OS on Microsoft Azure Stack HCI. The releases of Microsoft Azure Stack HCI 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.

2.3.2.1. Linux Guest OS Support in Release 16.5

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Red Hat Enterprise Linux 8.9	22H2, 23H2 preview	22H2, 23H2 preview

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Red Hat Enterprise Linux 8.8	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 8.6	22H2, 23H2 preview	22H2, 23H2 preview
Rocky Linux 8.4	22H2, 23H2 preview	22H2, 23H2 preview
Deprecated: CentOS Linux 8 (2105)	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 7.9	22H2, 23H2 preview	22H2, 23H2 preview
Ubuntu 20.04 LTS	22H2, 23H2 preview	22H2, 23H2 preview

2.3.2.2. Linux Guest OS Support in Release 16.4

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Red Hat Enterprise Linux 8.9	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 8.8	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 8.6	22H2, 23H2 preview	22H2, 23H2 preview
Rocky Linux 8.4	22H2, 23H2 preview	22H2, 23H2 preview
Deprecated: CentOS Linux 8 (2105)	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 7.9	22H2, 23H2 preview	22H2, 23H2 preview
Ubuntu 20.04 LTS	22H2, 23H2 preview	22H2, 23H2 preview

2.3.2.3. Linux Guest OS Support in Release 16.3

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Red Hat Enterprise Linux 8.9	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 8.8	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 8.6	22H2, 23H2 preview	22H2, 23H2 preview
Rocky Linux 8.4	22H2, 23H2 preview	22H2, 23H2 preview
Deprecated: CentOS Linux 8 (2105)	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 7.9	22H2, 23H2 preview	22H2, 23H2 preview
Ubuntu 20.04 LTS	22H2, 23H2 preview	22H2, 23H2 preview

2.3.2.4. Linux Guest OS Support in Release 16.2

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Red Hat Enterprise Linux 8.8	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 8.6	22H2, 23H2 preview	22H2, 23H2 preview
Rocky Linux 8.4	22H2, 23H2 preview	22H2, 23H2 preview
Deprecated: CentOS Linux 8 (2105)	22H2, 23H2 preview	22H2, 23H2 preview
Red Hat Enterprise Linux 7.9	22H2, 23H2 preview	22H2, 23H2 preview
Ubuntu 20.04 LTS	22H2, 23H2 preview	22H2, 23H2 preview

2.3.2.5. Linux Guest OS Support in Release 16.1

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Red Hat Enterprise Linux 8.8	22H2	22H2
Red Hat Enterprise Linux 8.6	22H2	22H2
Rocky Linux 8.4	22H2	22H2
Deprecated: CentOS Linux 8 (2105)	22H2	22H2
Red Hat Enterprise Linux 7.9	22H2	22H2
Ubuntu 20.04 LTS	22H2	22H2

2.3.2.6. Linux Guest OS Support in Release 16.0

Guest OS	NVIDIA vGPU - Microsoft Azure Stack HCI Releases	Pass-Through GPU - Microsoft Azure Stack HCI Releases
Red Hat Enterprise Linux 8.8	22H2	22H2
Red Hat Enterprise Linux 8.6	22H2	22H2
Rocky Linux 8.4	22H2	22H2
Deprecated: CentOS Linux 8 (2105)	22H2	22H2
Red Hat Enterprise Linux 7.9	22H2	22H2
Ubuntu 20.04 LTS	22H2	22H2

2.4. NVIDIA CUDA Toolkit Version Support

The releases in this release family of NVIDIA vGPU software support NVIDIA CUDA Toolkit 12.1.

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 12.1.0*.

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 12.1 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. vGPU Hibernation Support

NVIDIA vGPU software supports Advanced Configuration and Power Interface (ACPI) hibernation (Sx state S4) for VMs that are configured with GPU-P.

In Sx state S4, all contents of the main memory, including the runtime state of any vGPUs assigned to the VM, are saved to persistent storage and the VM is powered down. When the VM is woken up, all contents of the main memory are restored by the guest OS.

vGPU hibernation is supported on all supported GPUs, Microsoft Azure Stack HCI hypervisor software releases, Windows guest operating systems, and Ubuntu guest operating systems. vGPU hibernation is **not** supported on other Linux guest operating systems.

2.6. 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 hypervisor software releases.

2.6.1. vGPUs that Support Multiple vGPUs Assigned to a VM

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

You can assign multiple vGPUs with differing amounts of frame buffer to a single VM, provided the board type and the series of all the vGPUs is the same. For example, you can assign an A40-48Q vGPU and an A40-16Q vGPU to the same VM. However, you cannot assign an A30-8Q vGPU and an A16-8Q vGPU to the same VM.

Multiple vGPU Support on the NVIDIA Ampere GPU Architecture

Board	vGPU
NVIDIA A40	All Q-series vGPUs See Note (1).
NVIDIA A16	All Q-series vGPUs See Note (1).
NVIDIA A10	All Q-series vGPUs See Note (1).
NVIDIA A2	A2-16Q See Note (1).

Note:

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

2.6.2. Maximum Number of vGPUs Supported per VM

For Microsoft Azure Stack HCI, NVIDIA vGPU software supports up to a maximum of 8 vGPUs per VM.

2.6.3. Hypervisor Releases that Support Multiple vGPUs Assigned to a VM

All hypervisor releases that support NVIDIA vGPU software are supported.

2.7. 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 enabled by default. If you do not want to use unified memory, you must disable it individually for each vGPU by setting a vGPU plugin parameter. NVIDIA CUDA Toolkit profilers are supported and can be enabled on a VM for which unified memory is enabled.

2.7.1. vGPUs that Support Unified Memory

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

Unified Memory Support on the NVIDIA Ampere GPU Architecture

Board	vGPU
NVIDIA A40	A40-48Q
NVIDIA A16	A16-16Q
NVIDIA A10	A10-24Q
NVIDIA A2	A2-16Q

2.7.2. Guest OS Releases that Support Unified Memory

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

2.8. 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 L40
- **Since 16.1:** NVIDIA L40S

- **Since 16.3:** NVIDIA L20
- NVIDIA L4
- **Since 16.3:** NVIDIA L2
- NVIDIA A40
- NVIDIA A16
- NVIDIA A2
- NVIDIA A10

> **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. vCS is not supported on Microsoft Azure Stack HCI

NVIDIA Virtual Compute Server (vCS) is not supported on Microsoft Azure Stack HCI. C-series vGPU types are not available.

However, you can run compute workloads on physical GPUs in DDA deployments with Microsoft Azure Stack HCI.

3.3. 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.

3.4. 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.

3.5. 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.

For all vGPU types supported by Microsoft Azure Stack HCI, *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.

3.6. 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 Microsoft Azure Stack HCI 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 Microsoft Azure Stack HCI VM's 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 Microsoft Azure Stack HCI VM's 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.

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 Microsoft Azure Stack HCI .

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**.

4.1. Issues Resolved in Release 16.5

No resolved issues are reported in this release for Microsoft Azure Stack HCI.

4.2. Issues Resolved in Release 16.4

No resolved issues are reported in this release for Microsoft Azure Stack HCI.

4.3. Issues Resolved in Release 16.3

Bug ID	Summary and Description
4297231	16.0-16.2 Only: NVIDIA vGPU software does not work on Microsoft Azure Stack HCI 23H2 if kernel DMA protection is enabled NVIDIA vGPU software does not work on Microsoft Azure Stack HCI 23H2 if kernel direct memory access (DMA) protection is enabled. After the Virtual GPU Manager is installed, the GPU device status is reported as critical with the following error: `Windows stopped this device because it reported problems. (Code 43)`
4399699	16.0-16.2 Only: Black screens and display disconnection occur after reboot After the hypervisor host is rebooted, a black screen occurs if an attempt is made to connect to a VM configured with NVIDIA vGPU. This issue affects only GPUs based on the Ampere GPU architecture and later GPU architectures. When this issue occurs, the following error messages are written to the log files on the hypervisor host.

Bug ID	Summary and Description
	- XID error 38 - XID error 43 - XID error 109 - `vGPU Message 22` - Timeout detection and recovery (TDR) failures

4.4. Issues Resolved in Release 16.2

Bug ID	Summary and Description
4309888	**16.0, 16.1 Only: NVWMI functions for faking EDID have no effect** The NVIDIA Enterprise Management Toolkit (NVWMI) functions for faking Extended Display Identification Data (EDID), namely, `fakeEDID`, `fakeEDIDAll`, and `fakeEDIDOnPort` have no effect. This issue affects only Windows guest VMs and can prevent a VM from being enabled with multiple displays. When this issue occurs, `unable to fake EDID` events can be seen in **Event Viewer**.

4242693	**16.0, 16.1 Only: Windows Server 2022 VMs support only a maximum of nine RDP sessions**
	Windows Server 2022 guest VMs support only a maximum of nine Remote Desktop Protocol (RDP) sessions. An attempt to launch a 10th session on a Windows Server 2022 guest VM fails. When this issue occurs, the following error messages are logged.
	`2023-08-21T22:55:40.279Z Er(02) vthread-3390694 - vmiop_log: (0x0): Cannot use virtual context buffers in sysmem`
	`2023-08-21T22:55:40.279Z Er(02) vthread-3390694 - vmiop_log: (0x0): Invalid promote context input`
	`2023-08-21T22:55:40.279Z Er(02) vthread-3390694 - vmiop_log: (0x0): VGPU message 111 failed, result code: 0x1f`

4.5. Issues Resolved in Release 16.1

Bug ID	Summary and Description
4142288	**16.0 Only: The NVIDIA L40 GPU brand is incorrectly identified if GSP firmware is disabled** If GPU System Processor (GSP) firmware is disabled, the NVIDIA Virtual GPU Manager incorrectly identifies the brand of the NVIDIA L40 GPU. This incorrect identification of the GPU brand might cause performance degradation with some applications that are optimised for features of

Bug ID	Summary and Description
	the NVIDIA L40 that are not available in the incorrect brand. However, the output from the `nvidia-smi` command is **not** affected.
3641947	16.0 Only: Graphics applications are corrupted on some Windows vGPU VMs 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.

4.6. Issues Resolved in Release 16.0

Bug ID	Summary and Description
4096848	**Optical Flow object allocation fails on VMs configured with vGPUs based on the NVIDIA Ampere architecture** Optical Flow object allocation fails on VMs configured with vGPUs that reside on GPUs based on the NVIDIA Ampere GPU architecture. This issue has been observed as the failure of the Omniverse Kit container on a VM configured with NVIDIA vGPU.
3334310	**NVIDIA Control Panel is started only for the RDP user that logs on first** On all supported Windows Server guest OS releases, **NVIDIA Control Panel** is started only for the RDP user that logs on first. Other users cannot start **NVIDIA Control Panel**. If more than one RDP user is logged on when **NVIDIA Control Panel** is started, it always opens in the session of the RDP user that logged on first, irrespective of which user started **NVIDIA Control Panel**. Furthermore, on Windows Server 2016, **NVIDIA Control Panel** crashes if a user session is disconnected and then reconnected while **NVIDIA Control Panel** is open.

Chapter 5. Known Issues

5.1. 16.0-16.2 Only: Black screens and display disconnection occur after reboot

Description

After the hypervisor host is rebooted, a black screen occurs if an attempt is made to connect to a VM configured with NVIDIA vGPU. This issue affects only GPUs based on the Ampere GPU architecture and later GPU architectures. When this issue occurs, the following error messages are written to the log files on the hypervisor host.

- XID error 38
- XID error 43
- XID error 109
- `vGPU Message 22`
- Timeout detection and recovery (TDR) failures

Status

Resolved in NVIDIA vGPU software 16.3

Ref.

4399699

5.2. 16.0-16.2 Only: NVIDIA vGPU software does not work on Microsoft Azure Stack HCI 23H2 if kernel DMA protection is enabled

Description

NVIDIA vGPU software does not work on Microsoft Azure Stack HCI 23H2 if kernel direct memory access (DMA) protection is enabled. After the Virtual GPU Manager is installed, the GPU device status is reported as critical with the following error:

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

Version

This issue affects only Microsoft Azure Stack HCI 23H2.

Workaround

Disable the kernel DMA protection option in the system BIOS.

Status

Resolved in NVIDIA vGPU software 16.3

Ref.

4297231

5.3. **NVIDIA Control Panel** is not available in multiuser environments

Description

After the NVIDIA vGPU software graphics driver for Windows is installed, the **NVIDIA Control Panel** app might be missing from the system. This issue typically occurs in the following situations:

- Multiple users connect to virtual machines by using remote desktop applications such as Microsoft RDP, VMware Horizon, and Citrix Virtual Apps and Desktops.

- VM instances are created by using Citrix Machine Creation Services (MCS) or VMware Instant Clone technology.
- Roaming user desktop profiles are deployed.

This issue occurs because the **NVIDIA Control Panel** app is now distributed through the **Microsoft Store**. The **NVIDIA Control Panel** app might fail to be installed when the NVIDIA vGPU software graphics driver for Windows is installed if the **Microsoft Store** app is disabled, the system is not connected to the Internet, or installation of apps from the **Microsoft Store** is blocked by your system settings.

To determine whether the **NVIDIA Control Panel** app is installed on your system, use the **Windows Settings** app or the `Get-AppxPackage` Windows PowerShell command.

- To use the **Windows Settings** app:
  1. From the Windows **Start** menu, choose **Settings** > **Apps** > **Apps & feautures** .
  2. In the **Apps & features** window, type **`nvidia control panel`** in the search box and confirm that the **NVIDIA Control Panel** app is found.
- To use the `Get-AppxPackage`Windows PowerShell command:
  1. Run **Windows PowerShell** as Administrator.
  2. Determine whether the **NVIDIA Control Panel** app is installed for the current user.

     ```
     PS C:\> Get-AppxPackage -Name NVIDIACorp.NVIDIAControlPanel
     ```
  3. Determine whether the **NVIDIA Control Panel** app is installed for all users.

     ```
     PS C:\> Get-AppxPackage -AllUsers -Name NVIDIACorp.NVIDIAControlPanel
     ```

     This example shows that the **NVIDIA Control Panel** app is installed for the users `Administrator`, `pliny`, and `trajan`.

     ```
     PS C:\> Get-AppxPackage -AllUsers -Name NVIDIACorp.NVIDIAControlPanel

     Name                   : NVIDIACorp.NVIDIAControlPanel
     Publisher              : CN=D6816951-877F-493B-B4EE-41AB9419C326
     Architecture           : X64
     ResourceId             :
     Version                : 8.1.964.0
     PackageFullName        :
      NVIDIACorp.NVIDIAControlPanel_8.1.964.0_x64__56jybvy8sckqj
     InstallLocation        : C:\Program Files\WindowsApps
     \NVIDIACorp.NVIDIAControlPanel_8.1.964.0_x64__56jybvy8sckqj
     IsFramework            : False
     PackageFamilyName      : NVIDIACorp.NVIDIAControlPanel_56jybvy8sckqj
     PublisherId            : 56jybvy8sckqj
     PackageUserInformation :
      {S-1-12-1-530092550-1307989247-1105462437-500 [Administrator]: Installed,

      S-1-12-1-530092550-1307989247-1105462437-1002 [pliny]: Installed,

      S-1-12-1-530092550-1307989247-1105462437-1003 [trajan]: Installed}
     IsResourcePackage      : False
     IsBundle               : False
     IsDevelopmentMode      : False
     NonRemovable           : False
     IsPartiallyStaged      : False
     SignatureKind          : Store
     Status                 : Ok
     ```

Preventing this Issue

Since 16.3: If your system does not allow the installation apps from the **Microsoft Store**, download and run the standalone **NVIDIA Control Panel** installer that is available from NVIDIA Licensing Portal. For instructions, refer to *Virtual GPU Software User Guide*.

If your system can allow the installation apps from the **Microsoft Store**, ensure that:

- The Microsoft Store app is enabled.
- Installation of Microsoft Store apps is not blocked by your system settings.
- No local or group policies are set to block Microsoft Store apps.

Workaround

If the **NVIDIA Control Panel** app is missing, install it separately from the graphics driver.

- **Since 16.3:** You can install the **NVIDIA Control Panel** app by downloading and running the standalone **NVIDIA Control Panel** installer that is available from NVIDIA Licensing Portal. For instructions, refer to *Virtual GPU Software User Guide*.
- **16.0-16.2 only:** For a system that is running Windows 11 or a modern version of Windows 10, you can install the **NVIDIA Control Panel** app by using the `winget` command-line tool of **Windows Package Manager**.

  > **Note:** The `winget` command-line tool is not available on the Windows Server OS.

  Before using the `winget` command-line tool to install the **NVIDIA Control Panel** app, ensure that the following prerequisites are met:

  - Your system is connected to the Internet.
  - The Microsoft Store app is enabled.
  - Packages on which `winget` depends, such as `Microsoft.UI.Xaml` and `Microsoft.VCLibs.x64`, are installed.

  To use the `winget` command-line tool to install the **NVIDIA Control Panel** app, run the following command:

  ```
  PS C:\> winget install "NVIDIA Control Panel" --id 9NF8H0H7WMLT -s msstore
  --accept-package-agreements --accept-source-agreements
  ```

  For information about how to download and use the latest `winget` version, refer to Use the winget tool to install and manage applications on the Microsoft documentation site.

If the issue persists, contact NVIDIA Enterprise Support for further assistance.

Status

Open

Ref.

3999308

5.4. 16.0, 16.1 Only: NVWMI functions for faking EDID have no effect

Description

The NVIDIA Enterprise Management Toolkit (NVWMI) functions for faking Extended Display Identification Data (EDID), namely, `fakeEDID`, `fakeEDIDAll`, and `fakeEDIDOnPort` have no effect. This issue affects only Windows guest VMs and can prevent a VM from being enabled with multiple displays. When this issue occurs, `unable to fake EDID` events can be seen in **Event Viewer**.

Status

Resolved in NVIDIA vGPU software 16.2

Ref.

4309888

5.5. 16.0, 16.1 Only: Windows Server 2022 VMs support only a maximum of nine RDP sessions

Description

Windows Server 2022 guest VMs support only a maximum of nine Remote Desktop Protocol (RDP) sessions. An attempt to launch a 10th session on a Windows Server 2022 guest VM fails. When this issue occurs, the following error messages are logged.

```
2023-08-21T22:55:40.279Z Er(02) vthread-3390694 - vmiop_log: (0x0): Cannot use
 virtual context buffers in sysmem
2023-08-21T22:55:40.279Z Er(02) vthread-3390694 - vmiop_log: (0x0): Invalid promote
 context input
2023-08-21T22:55:40.279Z Er(02) vthread-3390694 - vmiop_log: (0x0): VGPU message 111
 failed, result code: 0x1f
```

Version

This issue affects only Windows Server 2022 guest VMs that are configured with NVIDIA vGPU.

Status

Resolved in NVIDIA vGPU software16.2

Resolution of this issue increases the maximum number of RDP sessions to 16. Issues similar to this issue might still occur if the channels allocated to a vGPU are exhausted. For more information, refer to Issues occur when the channels allocated to a vGPU are exhausted.

Ref.

4242693

5.6. 16.0 Only: The NVIDIA L40 GPU brand is incorrectly identified if GSP firmware is disabled

Description

If GPU System Processor (GSP) firmware is disabled, the NVIDIA Virtual GPU Manager incorrectly identifies the brand of the NVIDIA L40 GPU. This incorrect identification of the GPU brand might cause performance degradation with some applications that are optimised for features of the NVIDIA L40 that are not available in the incorrect brand. However, the output from the `nvidia-smi` command is **not** affected.

This issue occurs only if GPU System Processor (GSP) firmware is disabled. It does not occur if GSP firm is enabled.

Status

Resolved in NVIDIA vGPU software 16.1

Ref.

4142288

5.7. 16.0 Only: 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

Resolved in NVIDIA vGPU software 16.1

Ref.

3641947

5.8. CUDA profilers cannot gather hardware metrics on NVIDIA vGPU

Description

NVIDIA CUDA Toolkit profilers cannot gather hardware metrics on NVIDIA vGPU. This issue affects only traces that gather hardware metrics. Other traces are not affected by this issue and work normally.

Version

This issue affects NVIDIA vGPU software releases starting with 15.2.

Status

Open

Ref.

4041169

5.9. NVIDIA vGPU software graphics driver for Windows sends a remote call to `ngx.download.nvidia.com`

Description

After the NVIDIA vGPU software graphics for windows has been installed in the guest VM, the driver sends a remote call to `ngx.download.nvidia.com` to download and install additional components. Such a remote call might be a security issue.

Workaround

Before running the NVIDIA vGPU software graphics driver installer, disable the remote call to `ngx.download.nvidia.com` by setting the following Windows registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\NGXCore]
"EnableOTA"=dword:00000000
```

Note: If this Windows registry key is set to 1 or deleted, the remote call to `ngx.download.nvidia.com` is enabled again.

Status

Open

Ref.

4031840

5.10. Multiple RDP session reconnections on Windows Server 2022 can consume all frame buffer

Description

Multiple RDP session reconnections in a Windows Server 2022 guest VM can consume all the frame buffer of a vGPU or physical GPU. When this issue occurs, users' screens becomes black, their sessions are disconnected but left intact, and they cannot log on again. The following error message is written to the event log on the hypervisor host:

```
The Desktop Window Manager process has exited.
(Process exit code: 0xe0464645, Restart count: 1, Primary display device ID: )
```

Version

This issue affects only the Windows Server 2022 guest OS.

Workaround

Periodically restart the Windows Server 2022 guest VM to prevent all frame buffer from being consumed.

Status

Open

Ref.

3583766

5.11. 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

5.12. With multiple active sessions, **NVIDIA Control Panel** incorrectly shows that the system is unlicensed

Description

In an environment with multiple active desktop sessions, the **Manage License** page of **NVIDIA Control Panel** shows that a licensed system is unlicensed. However, the `nvidia-smi` command and the management interface of the NVIDIA vGPU software license server correctly show that the system is licensed. When an active session is disconnected and reconnected, the **NVIDIA Display Container** service crashes.

The **Manage License** page incorrectly shows that the system is unlicensed because of stale data in **NVIDIA Control Panel** in an environment with multiple sessions. The data is stale because **NVIDIA Control Panel** fails to get and update the settings for remote sessions when multiple sessions or no sessions are active in the VM. The **NVIDIA Display Container** service crashes when a session is reconnected because the session is not active at the moment of reconnection.

Status

Open

Ref.

3761243

5.13. 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 and later supported releases. 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

5.14. After an upgrade of the Linux graphics driver from a Debian package, the driver is not loaded into the VM

Description

After the NVIDIA vGPU software graphics driver for Linux is upgraded from a Debian package, the driver is not loaded into the VM.

Workaround

Use one of the following workarounds to load the driver into the VM:

- Reboot the VM.

- Remove the `nvidia` module from the Linux kernel and reinsert it into the kernel.
  1. Remove the `nvidia` module from the Linux kernel.
     ```
     $ sudo rmmod nvidia
     ```
  2. Reinsert the `nvidia` module into the Linux kernel.
     ```
     $ sudo modprobe nvidia
     ```

Status

Not a bug

Ref.

200748806

5.15. The reported NVENC frame rate is double the actual frame rate

Description

The frame rate in frames per second (FPS) for the NVIDIA hardware-based H.264/HEVC video encoder (NVENC) reported by the `nvidia-smi encodersessions` command and NVWMI is double the actual frame rate. Only the reported frame rate is incorrect. The actual encoding of frames is **not** affected.

This issue affects only Windows VMs that are configured with NVIDIA vGPU.

Status

Open

Ref.

2997564

5.16. 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

5.17. 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

5.18. 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

5.19. Disconnected sessions cannot be reconnected or might be reconnected very slowly with NVWMI installed

Description

Disconnected sessions cannot be reconnected or might be reconnected very slowly when the NVIDIA Enterprise Management Toolkit (NVWMI) is installed. This issue affects Citrix Virtual Apps and Desktops and VMware Horizon sessions on Windows guest VMs.

Workaround

Uninstall NVWMI.

Status

Open

Ref.

3262923

5.20. 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

5.21. 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.

```
[root@host ~]# nvidia-smi
Fri Apr 12 11:45:28 2024
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.161.05   Driver Version: 535.161.05   CUDA Version:  12.1    |
|-------------------------------+----------------------+----------------------+
| 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

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 Apr 12 11:47:38 2024
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.161.05   Driver Version: 535.161.05   CUDA Version:  12.1    |
|-------------------------------+----------------------+----------------------+
| 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

5.22. 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 NVDA A100 PCIe 40GB or NVDA 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

5.23. 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

5.24. **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

5.25. 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

5.26. Microsoft DDA fails with some GPUs

Description

Microsoft Discrete Device Assignment (DDA) fails with GPUs that have more than 16 GB of GPU memory. After the NVIDIA vGPU software graphics driver is installed in the guest VM, a second display device appears on the GPU and the driver prompts for a reboot. After the reboot, the device disappears and the Microsoft Hyper-V Video device appears.

This issue occurs because less memory-mapped input/output (MMIO) space is configured for the operating system than the device requires.

Workaround

Perform this workaround in a **Windows Power Shell** window on the hypervisor host.

Set the upper MMIO space to the amount that the device requires to allow all of the MMIO to be mapped. Upper MMIO space starts at approximately 64 GB in address space.

```
Set-VM –HighMemoryMappedIoSpace mmio-space –VMName vm-name
```

mmio-space

The amount of MMIO space that the device requires, appended with the appropriate unit of measurement, for example, **`64GB`** for 64 GB of MMIO space.

The required amount of MMIO space depends on the amount of BAR1 memory on the installed GPUs and the number of GPUs assigned to the VM as follows:

mmio-space = 2 # *gpu-bar1-memory* # *assigned-gpus*

gpu-bar1-memory

The amount of BAR1 memory on one of the installed GPUs. For example, in a server in which eight GPUs are installed and each GPU has 32 GB of BAR1 memory, *gpu-bar1-memory* is 32 GB.

assigned-gpus

The number of GPUs assigned to the VM.

vm-name

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

The following example sets the upper MMIO space to 64 GB for the VM named `mygpuvm`, to which one GPU with 32 GB of BAR1 memory is assigned.

```
Set-VM –HighMemoryMappedIoSpace 64GB –VMName mygpuvm
```

For more information, see Deploy graphics devices using Discrete Device Assignment on the Microsoft technical documentation site.

Status

Not an NVIDIA bug

Ref.

2812853

5.27. 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

5.28. Citrix Virtual Apps and Desktops session freezes when the desktop is unlocked

Description

When a Citrix Virtual Apps and Desktops session that is locked is unlocked by pressing **Ctrl**+**Alt**+**Del**, the session freezes. This issue affects only VMs that are running Microsoft Windows 10 1809 as a guest OS.

Version

Microsoft Windows 10 1809 guest OS

Workaround

Restart the VM.

Status

Not an NVIDIA bug

Ref.

2767012

5.29. NVIDIA vGPU software graphics driver fails after Linux kernel upgrade with DKMS enabled

Description

After the Linux kernel is upgraded (for example by running `sudo apt full-upgrade`) with Dynamic Kernel Module Support (DKMS) enabled, the `nvidia-smi` command fails to run. If DKMS is enabled, an upgrade to the Linux kernel triggers a rebuild of the NVIDIA vGPU software graphics driver. The rebuild of the driver fails because the compiler version is incorrect. Any attempt to reinstall the driver fails because the kernel fails to build.

When the failure occurs, the following messages are displayed:

```
-> Installing DKMS kernel module:
        ERROR: Failed to run `/usr/sbin/dkms build -m nvidia -v  535.54.03 -k
 5.3.0-28-generic`:
        Kernel preparation unnecessary for this kernel. Skipping...
        Building module:
        cleaning build area...
        'make' -j8 NV_EXCLUDE_BUILD_MODULES='' KERNEL_UNAME=5.3.0-28-generic
 IGNORE_CC_MISMATCH='' modules...(bad exit status: 2)
        ERROR (dkms apport): binary package for nvidia:  535.54.03 not found
        Error! Bad return status for module build on kernel: 5.3.0-28-generic
 (x86_64)
        Consult /var/lib/dkms/nvidia/ 535.54.03/build/make.log for more information.
        -> error.
        ERROR: Failed to install the kernel module through DKMS. No kernel module
 was installed;
        please try installing again without DKMS, or check the DKMS logs for more
 information.
        ERROR: Installation has failed. Please see the file '/var/log/nvidia-
installer.log' for details.
        You may find suggestions on fixing installation problems in the README
 available on the Linux driver download page at www.nvidia.com.
```

Workaround

When installing the NVIDIA vGPU software graphics driver with DKMS enabled, use one of the following workarounds:

- Before running the driver installer, install the `dkms` package, then run the driver installer with the `-dkms` option.
- Run the driver installer with the `--no-cc-version-check` option.

Status

Not a bug.

Ref.

2836271

5.30. Blue screen crash occurs or no devices are found after VM reset

Description

If a VM on Microsoft Windows Server with Hyper-V role is reset from the hypervisor host, a blue screen crash (BSOD) occurs on Windows VMs and the `nvidia-smi` command reports `No devices were found` on Linux VMs. This issue occurs only on Windows Server 2019 with Tesla T4 GPUs with SRIOV enabled, Quadro RTX 8000 passive GPUs, and Quadro RTX 6000 passive GPUs.

Workaround

Contact NVIDIA Enterprise Support for a workaround for this issue, referencing the knowledge base article *Workaround for Blue Screen Crashes On Hyper-V DDA With SRIOV-Enabled GPUs*. This article is available only to NVIDIA Enterprise Support personnel.

Status

Not an NVIDIA bug

Ref.

200567935

5.31. 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

5.32. 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

5.33. 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

5.34. 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

Status

Not a bug

Ref.

2115733

5.35. 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.

5.36. 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

5.37. `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

5.38. 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

5.39. 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

5.40. 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

5.41. 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

5.42. 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