NVIDIA Data Center GPU Driver version 575.57.08 (Linux)/ 576.57 (Windows)

Release Notes

RN-08625-565 _v1.0 | June 2025

Table of Contents

Chapter 1. Version Highlights

This section provides highlights of the NVIDIA Data Center GPU R575 Driver (version 575.57.08 Linux and 576.57 Windows).

For changes related to the 575 release of the NVIDIA display driver, review the file "NVIDIA_Changelog" available in the .run installer packages.

- Linux driver release date: 06/03/2025
- Windows driver release date: 06/03/2025

1.1. Software Versions

For this release, the software versions are as follows:

- CUDA Toolkit 12: 12.x

  Note that starting with CUDA 11, individual components of the toolkit are versioned independently. For a full list of the individual versioned components (for example, nvcc, CUDA libraries, and so on), see the CUDA Toolkit Release Notes.
- NVIDIA Data Center GPU Driver: 575.57.08 (Linux) / 576.57 (Windows)
- Fabric Manager: 575.57.08 (Use `nv-fabricmanager -v`)
- NVFlash: 5.791

For more information on getting started with the NVIDIA Fabric Manager on NVSwitch-based systems (for example, NVIDIA HGX A100), refer to the Fabric Manager User Guide.

1.2. Fixed Issues

- `nvidia-bug-report.sh` has been updated to collect logs into the temporary filesystem as designated by `TMPDIR` in accordance with POSIX standard.
- Channel Adapters are iterated in alphabetical order before attempting to pick one. This behavior mimics existing IB tools such as `ibnetdiscover`.
- Initialization of a field in the CSL context is moved to an earlier point to prevent decryption failures.

- A bug in some error handling for inband messaging can incompletely cleanup after an error, resulting in residual data causing an assertion to fail in a subsequent inband message. This fix implements correct code that frees the message buffer regardless.
- `PREAD/PWRITE` is only processed when a call is issued (i.e., memory is allocated).
- Thread ID was incorrectly being used instead of Thread Group ID to match against the saved Thread Group ID during PID translation; this has been resolved.
- The output print for `GPU FABRIC GUID` is now padded so that any print of the hex output will have 16 characters.
- Register definition containing one-show power limit values (PMU) has been fixed.
- Regression in PTXAS between CUDA 12.4 and CUDA 12.8 no longer causes an illegal memory access error on Hopper architectures.
- When the container blocks support to a GPU for any reason, that GPU will be skipped instead of checking against a particular error code. This puts the checkpointing path in alignment with the CUDA driver's initialization.
- Fixed an issue where GPU access to system memory, which is allocated using cudaMallocHost, cudaHostRegister, and so on might observe reduced performance.

1.3. Known Issues

- This version of the GPU driver will fail to initialize on systems with Hopper GPUs subrevision = 3 and VBIOS versions older than 96.00.68.00.xx. Please ensure the system is using a VBIOS version 96.00.68.00.xx or newer before upgrading to this version of the driver.
- When upgrading from ClosedRM to OpenRM, nvidia-smi may fail.

  Workaround

  Run the following commands:

```
sudo rpm -e  nvidia-open-driver-G06-kmp-default --nodeps
sudo zypper in nvidia-driver-G06-kmp-default
sudo  zypper install -y nvidia-open-570
```

Disable GPU initiated RO traffic on Ada Lovelace and older GPUs

Historically, for GPUDirect P2P over PCIe (i.e., not for NVLink where that may apply), Ada Lovelace and older GPU architectures rely on the host platform to keep the order of GPU-initiated posted PCIe transactions targeting a peer GPU, regardless of the Relaxed Ordering (RO) bit. That is due to a hardware issue.

It was later noted that some data center platforms, like those based on Intel Xeon (codenamed Sapphire Rapids) and later, do not provide that guarantee. Therefore, using GPUDirect P2P may lead to run-time silent data corruption. For example, see below for the data validation errors possibly detected by simpleP2P :

```
$ cuda-samples/Samples/0_Introduction/simpleP2P/simpleP2P
...
Checking for multiple GPUs...
CUDA-capable device count: 3

Checking GPU(s) for support of peer to peer memory access...
> Peer access from NVIDIA A2 (GPU0) -> NVIDIA A2 (GPU1) : Yes
```

```
> Peer access from NVIDIA A2 (GPU0) -> NVIDIA A2 (GPU2) : Yes
> Peer access from NVIDIA A2 (GPU1) -> NVIDIA A2 (GPU0) : Yes
> Peer access from NVIDIA A2 (GPU1) -> NVIDIA A2 (GPU2) : Yes
> Peer access from NVIDIA A2 (GPU2) -> NVIDIA A2 (GPU0) : Yes
> Peer access from NVIDIA A2 (GPU2) -> NVIDIA A2 (GPU1) : Yes
Enabling peer access between GPU0 and GPU1...
Allocating buffers (64MB on GPU0, GPU1 and CPU Host)...
Creating event handles...
cudaMemcpyPeer / cudaMemcpy between GPU0 and GPU1: 9.66GB/s
Preparing host buffer and memcpy to GPU0...
Run kernel on GPU1, taking source data from GPU0 and writing to GPU1...
Run kernel on GPU0, taking source data from GPU1 and writing to GPU0...
Copy data back to host from GPU0 and verify results...
Verification error @ element 0: val = 5888.000000, ref = 0.000000
Verification error @ element 1: val = 5892.000000, ref = 4.000000
Verification error @ element 2: val = 5896.000000, ref = 8.000000
Verification error @ element 3: val = 5900.000000, ref = 12.000000
Verification error @ element 4: val = 5904.000000, ref = 16.000000
Verification error @ element 5: val = 5908.000000, ref = 20.000000
Verification error @ element 6: val = 5912.000000, ref = 24.000000
Verification error @ element 7: val = 5916.000000, ref = 28.000000
Verification error @ element 8: val = 5920.000000, ref = 32.000000
Verification error @ element 9: val = 5924.000000, ref = 36.000000
Verification error @ element 10: val = 5928.000000, ref = 40.000000
Verification error @ element 11: val = 5932.000000, ref = 44.000000
Disabling peer access...
Shutting down...
Test failed!
```

In GPU drivers 525 and newer, the issue is mitigated. The mitigation relies on disabling Relaxed Ordering traffic for all GPU-initiated PCIe transactions, including toward host memory. At load time, the GPU kernel-mode driver enables the mitigation based on the vendor and device IDs of the PCIe host bridge.

Note that other host platforms may be affected by the same issue, and that its occurrence may be influenced by the specific platform configuration; for example, whether the IOMMU is enabled, or whether the GPU-to-GPU traffic runs over the inter-socket bus.

More recently it has been noted that since the exact platform PCIe topology may not always be exposed to the GPU driver — for example, when running on the guest OS within a Virtual Machine (VM) — the mitigation might not be applied even when necessary. This is currently tracked as a known issue.

Workaround

When in doubt, consider forcefully disabling all GPU initiated Relaxed Ordering PCIe transactions. As an example, see the sequence below:

1. Enable persistence mode, using the NVIDIA persistence daemon. As a fallback, use nvidia-smi -pm 1 .
2. Disable Relaxed Ordering in the GPU PCIe config space as shown below.
3. Run the applications.

The config space change:

```
# Take note of the current value:
$ setpci -s <GPU BDF> CAP_EXP+8.w
# Write back the original value after resetting bit 4 to 0
$ setpci -s <GPU BDF> CAP_EXP+8.w=<modified value>
```

Alternatively, that can be done in a single invocation:

```
$ setpci -s <GPU BDF> CAP_EXP+8.w=0x0000:0x0010
```

For reference, before applying that change:

```
$ sudo lspci -s 09:00.0 -vv
09:00.0 3D controller: NVIDIA Corporation Device 2235 (rev a1)
...
           Capabilities: [78] Express (v2) Legacy Endpoint, MSI 00
                DevCap: MaxPayload 256 bytes, PhantFunc 0, Latency L0s unlimited, L1
 <64us
                        ExtTag+ AttnBtn- AttnInd- PwrInd- RBE+ FLReset+
             DevCtl: CorrErr- NonFatalErr- FatalErr- UnsupReq-
                        RlxdOrd+ ExtTag+ PhantFunc- AuxPwr- NoSnoop+ FLReset-
                        MaxPayload 256 bytes, MaxReadReq 512 bytes
...
$ sudo setpci -s 09:00.0 CAP_EXP+8.w
2930
```

After applying the suggested change:

```
$ sudo setpci -s 09:00.0 CAP_EXP+8.w=0x0000:0x0010
$ sudo setpci -s 09:00.0 CAP_EXP+8.w
2920
$ sudo lspci -s 09:00.0 -vv
 09:00.0 3D controller: NVIDIA Corporation Device 2235 (rev a1)
...
          Capabilities: [78] Express (v2) Legacy Endpoint, MSI 00
                DevCap: MaxPayload 256 bytes, PhantFunc 0, Latency L0s unlimited, L1
 <64us
                        ExtTag+ AttnBtn- AttnInd- PwrInd- RBE+ FLReset+
                DevCtl: CorrErr- NonFatalErr- FatalErr- UnsupReq-
                        RlxdOrd- ExtTag+ PhantFunc- AuxPwr- NoSnoop+ FLReset-
                        MaxPayload 256 bytes, MaxReadReq 512 bytes
...
```

Note the RlxdOrd bit of the DevCtl register flipping its value.

Chapter 2. Virtualization

To make use of GPU passthrough with virtual machines running Windows and Linux, the hardware platform must support the following features:

- A CPU with hardware-assisted instruction set virtualization: Intel VT-x or AMD-V.
- Platform support for I/O DMA remapping.
- On Intel platforms, the DMA remapper technology is called Intel VT-d.
- On AMD platforms, it is called AMD IOMMU.

Support for these features varies by processor family, product, and system, and should be verified at the manufacturer's website.

The following hypervisors are supported for virtualization:

Hypervisor	Notes
Citrix XenServer	Version 6.0 and later
VMware vSphere (ESX / ESXi)	Version 5.1 and later.
Red Hat KVM	Red Hat Enterprise Linux 7 with KVM
Microsoft Hyper-V	Windows Server 2019 Hyper-V Generation 2

Data Center products now support one display of up to 2560x1600 resolution.

The following GPUs are supported for device passthrough for virtualization:

GPU Family	Boards Supported
NVIDIA Blackwell	NVIDIA HGX GB200 NVL, NVIDIA HGX B200
NVIDIA Grace Hopper	NVIDIA GH200
NVIDIA Hopper	NVIDIA H100, NVIDIA H800
NVIDIA Ada Lovelace	NVIDIA L40, L4, L2, L20
NVIDIA Ampere GPU Architecture	NVIDIA A800, A100, A40, A30, A16, A10, A10G, A2, AX800
NVIDIA Turing	NVIDIA T4, NVIDIA T4G
NVIDIA Volta	NVIDIA V100

GPU Family	Boards Supported
NVIDIA Pascal	Quadro: P2000, P4000, P5000, P6000, GP100 Tesla: P100, P40, P4
NVIDIA Maxwell	Quadro: K2200, M2000, M4000, M5000, M6000, M6000 24GB Tesla: M60, M40, M6, M4

Chapter 3. Hardware and Software Support

Support for these features varies by processor family, product, and system, and should be verified at the manufacturer's website.

Supported Operating Systems for NVIDIA Data Center GPUs

The Release 575 driver is supported on the following operating systems:

- Windows x86_64 operating systems:
  - Microsoft Windows® Server 2025 24H2
  - Microsoft Windows® Server 2022 21H2
  - Microsoft Windows® 11 24H2 - SV4
  - Microsoft Windows® 11 23H2
  - Microsoft Windows® 11 22H2 - SV2
  - Microsoft Windows® 10 22H2
- The following table summarizes the supported Linux 64-bit distributions. For a complete list of distributions, kernel versions supported, see the CUDA Linux System Requirements documentation.

Distribution	x86_64	Arm64 Server
Debian 12.x (where x <= 10)	Yes	No
OpenSUSE Leap 15.x (where y = 6)	Yes	No
Fedora 41	Yes	No
Red Hat Enterprise Linux 9.y (where y <= 5)	Yes	Yes
Rocky Linux 9.y (where y <= 5)	Yes	No
Red Hat Enterprise Linux 8.y (where y <= 10)	Yes	Yes
Rocky Linux 8.y (where y <= 10)	Yes	No

Distribution	x86_64	Arm64 Server
SUSE Linux Enterprise Server 15.y (where y = 6)	Yes	Yes
Ubuntu 24.04.z LTS (where z <= 2)	Yes	Yes
Ubuntu 22.04.z LTS (where z <= 5)	Yes	Yes
Ubuntu 20.04.z LTS (where z <= 6)	Yes	Yes
KylinOS V10 SP3 2403	Yes	Yes
Amazon Linux AL2023	Yes	Yes
Microsoft Azure Linux 3.0	No	Yes
Microsoft Azure Linux 2.0	Yes	No
Oracle Linux 8	Yes	No
Oracle Linux 9	Yes	No

Supported Operating Systems and CPU Configurations for NVIDIA RTX Pro 6000 Blackwell Server Edition

The Release 575 driver is validated with NVIDIA RTX Pro 6000 Blackwell Server Edition on the following operating systems and CPU configurations:

- Linux 64-bit distributions:
  - RedHat Enterprise Linux 9.4
  - RedHat Enterprise Linux 9.3
  - Rocky Linux 9.4
  - Rocky Linux 9.3
  - RedHat Enterprise Linux 8.10
  - OpenSUSE Leap 15.6
  - Fedora 41
  - CentOS Stream 9
  - Ubuntu 24.04
  - Ubuntu 22.04
- Windows 64-bit distributions:
  - Microsoft Windows® Server 2025
  - Microsoft Windows® 11 24H2 - SV4
  - Microsoft Windows® 11 23H2 - SV3

- Microsoft Windows® 10 22H2
- Microsoft Windows® 10 21H2

Supported Operating Systems and CPU Configurations for NVIDIA HGX GB200 NVL

- NVIDIA Grace Arm Linux 64-bit distributions:
  - Ubuntu 24.04 LTS (in 36/72 GPU configurations)
  - Ubuntu 22.04 LTS (in 36/72 GPU configurations)

Supported Operating Systems and CPU Configurations for NVIDIA HGX B200

- Linux 64-bit distributions:
  - Red Hat Enterprise Linux 9.5
  - Red Hat Enterprise Linux 8.10
  - Amazon Linux AL2023
  - Ubuntu 24.04 with NVIDIA HWE kernel
  - Ubuntu 22.04 with NVIDIA HWE kernel
- Windows 64-bit distributions:
  - Windows Server 2022

Supported Operating Systems and CPU Configurations for NVIDIA HGX H20

- Linux 64-bit distributions:
  - Red Hat Enterprise Linux 9.5
  - Ubuntu 24.04 with NVIDIA HWE kernel
  - Ubuntu 22.04 with NVIDIA HWE kernel
- Windows 64-bit distributions:
  - Windows Server 2025
  - Windows Server 2022

Supported Operating Systems and CPU Configurations for NVIDIA HGX GH200

- Linux 64-bit distributions:
  - Red Hat Enterprise Linux 9.5
  - SUSE Linux Enterprise Server 15.6

- Ubuntu 24.04 with NVIDIA HWE kernel
- Ubuntu 22.04 with NVIDIA HWE kernel

  RHEL and SLES feature parity with NVIDIA HWE Kernels. The latest RHEL 9 and SLES 15 SP6 kernels support bare metal.

Supported Operating Systems and CPU Configurations for NVIDIA HGX H200

The Release 575 driver is validated with NVIDIA HGX H200 on the following operating systems and CPU configurations:

- Linux 64-bit distributions:
  - Red Hat Enterprise Linux 9.5 (in 4/8/16-GPU configurations)
  - Ubuntu 24.04.2 LTS (in 4/8/16-GPU configurations)
  - Ubuntu 22.04.5 LTS (in 4/8/16-GPU configurations)
- Windows 64-bit distributions:
  - Windows Server 2025
  - Windows Server 2022
  - Windows is supported only in shared NVSwitch virtualization configurations.

Supported Operating Systems and CPU Configurations for NVIDIA HGX H100/H800

The Release 575 driver is validated with NVIDIA HGX H100 on the following operating systems and CPU configurations:

- Linux 64-bit distributions:
  - Red Hat Enterprise Linux 8.10 (in 4/8/16-GPU configurations)
  - Red Hat Enterprise Linux 9.5 (in 4/8/16-GPU configurations)
  - SUSE Linux Enterprise Server 15.6 (in 4/8/16-GPU configurations)
  - Ubuntu 24.04.2 LTS (in 4/8/16-GPU configurations)
  - Ubuntu 22.04.5 LTS (in 4/8/16-GPU configurations)
- Windows 64-bit distributions:
  - Windows Server 2025
  - Windows Server 2022
  - Windows is supported only in shared NVSwitch virtualization configurations.

Supported Operating Systems and CPU Configurations for NVIDIA HGX A100/A800

The Release 575 driver is validated with NVIDIA HGX A100 on the following operating systems and CPU configurations:

- Linux 64-bit distributions:
  - Debian 12.10
  - Red Hat Enterprise Linux 8.10 (in 4/8/16-GPU configurations)
  - Rocky Linux 8.10 (in 4/8/16-GPU configurations)
  - Red Hat Enterprise Linux 9.5 (in 4/8/16-GPU configurations)
  - Ubuntu 24.04.2 LTS (in 4/8/16-GPU configurations)
  - Ubuntu 22.04.5 LTS (in 4/8/16-GPU configurations)
  - Ubuntu 20.04.6 LTS (in 4/8/16-GPU configurations)
  - SUSE SLES 15.6 (in 4/8/16-GPU configurations)
  - KylinOS V10 SP3 2403
- Windows 64-bit distributions:
  - Windows Server 2025
  - Windows Server 2022
  - Windows is supported only in shared NVSwitch virtualization configurations.
- CPU Configurations:
  - AMD Rome in PCIe Gen4 mode
  - Intel Skylake/Cascade Lake (4-socket) in PCIe Gen3 mode

Supported Virtualization Configurations

The Release 575 driver is validated with NVIDIA HGX A100, HGX A800, H100, and H800 on the following configurations:

- Passthrough (full visibility of GPUs and NVSwitches to guest VMs):
  - 8-GPU configurations with Ubuntu 20.04.6 and 22.04.5
- Shared NVSwitch (guest VMs only have visibility of GPUs and full NVLink bandwidth between GPUs in the same guest VM):
  - 1/2/4/8/16-GPU configurations with Ubuntu 20.04.6 LTS

API Support

This release supports the following APIs:

- NVIDIA® CUDA® 12.x for NVIDIA® Maxwell™, Pascal™, Volta™, Turing™, Hopper™, NVIDIA Ampere architecture, and NVIDIA Ada Lovelace architecture GPUs
- OpenGL® 4.6
- Vulkan® 1.3
- DirectX 11
- DirectX 12 (Windows 10)
- Open Computing Language (OpenCL™ software) 3.0

Note that for using graphics APIs on Windows (such as OpenGL, Vulkan, DirectX 11, and DirectX 12) or any WDDM 2.0+ based functionality on Data Center GPUs, vGPU is required. See the vGPU documentation for more information.

Supported NVIDIA Data Center GPUs

The NVIDIA Data Center GPU driver package is designed for systems that have one or more Data Center GPU products installed. This release of the driver supports CUDA C/C++ applications and libraries that rely on the CUDA C Runtime and/or CUDA Driver API.

Attention: Release 470 was the last driver branch to support Data Center GPUs based on the NVIDIA Kepler architecture. This includes discontinued support for the following compute capabilities:

- sm_30 (NVIDIA Kepler)
- sm_32 (NVIDIA Kepler)
- sm_35 (NVIDIA Kepler)
- sm_37 (NVIDIA Kepler)

For more information on GPU products and compute capability, see https://developer.nvidia.com/cuda-gpus.

NVIDIA Server Platforms	
Product	**Architecture**
NVIDIA HGX GB200 NVL	GB200 and NVLink
NVIDIA HGX B200 8-GPU	B200 and NVSwitch
NVIDIA HGX H20-3e 8-GPU	H20 and NVSwitch
NVIDIA HGX H20 8-GPU	H20 and NVSwitch
NVIDIA HGX H200 8-GPU	H200 and NVSwitch
NVIDIA HGX H100 8-GPU	H100 and NVSwitch
NVIDIA HGX H800 8-GPU	H800 and NVSwitch
NVIDIA HGX H100 4-GPU	H100 and NVLink
NVIDIA HGX A800 8-GPU	A800 and NVSwitch

NVIDIA Server Platforms	
Product	**Architecture**
NVIDIA HGX A100 8-GPU	A100 and NVSwitch
NVIDIA HGX A100 4-GPU	A100 and NVLink
NVIDIA HGX-2	V100 and NVSwitch

Data Center H-Series Products	
Product	**GPU Architecture**
NVIDIA H100 PCIe	NVIDIA Hopper
NVIDIA H100 NVL	NVIDIA Hopper
NVIDIA H200 NVL	NVIDIA Hopper
NVIDIA H800 PCIe	NVIDIA Hopper
NVIDIA H800 NVL	NVIDIA Hopper

Data Center L-Series Products	
Product	**GPU Architecture**
NVIDIA L2	NVIDIA Ada Lovelace
NVIDIA L20	NVIDIA Ada Lovelace
NVIDIA L40	NVIDIA Ada Lovelace
NVIDIA L40S	NVIDIA Ada Lovelace
NVIDIA L4	NVIDIA Ada Lovelace

RTX-Series / T-Series Products	
Product	**GPU Architecture**
NVIDIA RTX Pro 6000 Blackwell Server Edition	NVIDIA Blackwell
NVIDIA RTX 6000 Ada Generation	NVIDIA Ada Lovelace
NVIDIA RTX 5880 Ada Generation	NVIDIA Ada Lovelace
NVIDIA RTX 5000 Ada Generation	NVIDIA Ada Lovelace
NVIDIA RTX 4500 Ada Generation	NVIDIA Ada Lovelace
NVIDIA RTX 4000 Ada Generation	NVIDIA Ada Lovelace
NVIDIA RTX 4000 SFF Ada Generation	NVIDIA Ada Lovelace
NVIDIA RTX 2000 Ada Generation	NVIDIA Ada Lovelace
NVIDIA RTX 2000E Ada Generation	NVIDIA Ada Lovelace
NVIDIA RTX A6000	NVIDIA Ampere architecture

RTX-Series / T-Series Products	
Product	**GPU Architecture**
NVIDIA RTX A5500	NVIDIA Ampere architecture
NVIDIA RTX A5000	NVIDIA Ampere architecture
NVIDIA RTX A4500	NVIDIA Ampere architecture
NVIDIA RTX A4000H	NVIDIA Ampere architecture
NVIDIA RTX A4000	NVIDIA Ampere architecture
NVIDIA RTX A2000 12GB	NVIDIA Ampere architecture
NVIDIA RTX A2000	NVIDIA Ampere architecture
NVIDIA RTX A1000	NVIDIA Ampere architecture
NVIDIA RTX A400	NVIDIA Ampere architecture
NVIDIA RTX A800 40GB Active	NVIDIA Ampere architecture
Quadro RTX 8000	NVIDIA Turing
Quadro RTX 6000	NVIDIA Turing
Quadro RTX A6000	NVIDIA Turing
Quadro RTX 5000	NVIDIA Turing
Quadro RTX A5000	NVIDIA Turing
Quadro RTX 4000	NVIDIA Turing
Quadro RTX A4000	NVIDIA Turing
NVIDIA T1000 8GB	NVIDIA Turing
NVIDIA T600	NVIDIA Turing
NVIDIA T400 4GB	NVIDIA Turing
NVIDIA T400	NVIDIA Turing
NVIDIA T400E	NVIDIA Turing

Data Center A-Series Products	
Product	**GPU Architecture**
NVIDIA A2	NVIDIA Ampere architecture
NVIDIA A800, AX800	NVIDIA Ampere architecture
NVIDIA A100X	NVIDIA Ampere architecture
NVIDIA A100 NVIDIA A100 80 GB PCIe	NVIDIA Ampere architecture
NVIDIA A40	NVIDIA Ampere architecture

Data Center A-Series Products	
Product	**GPU Architecture**
NVIDIA A30, A30X	NVIDIA Ampere architecture
NVIDIA A16	NVIDIA Ampere architecture
NVIDIA A10, A10M, A10G	NVIDIA Ampere architecture

Data Center T-Series Products	
Product	**GPU Architecture**
NVIDIA T4, T4G	NVIDIA Turing

Data Center V-Series Products	
Product	**GPU Architecture**
NVIDIA V100	Volta

Data Center P-Series Products	
Product	**GPU Architecture**
NVIDIA Tesla P100	NVIDIA Pascal
NVIDIA Tesla P40	NVIDIA Pascal
NVIDIA Tesla P4	NVIDIA Pascal

Data Center M-Class Products	
Product	**GPU Architecture**
NVIDIA Tesla M60	Maxwell
NVIDIA Tesla M40 24 GB	Maxwell
NVIDIA Tesla M40	Maxwell
NVIDIA Tesla M6	Maxwell
NVIDIA Tesla M4	Maxwell

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.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the Unites States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2025 NVIDIA Corporation & affiliates. All rights reserved.

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