NVIDIA Data Center GPU Driver version 510.47.03 (Linux) / 511.65 (Windows)

Release Notes

RN-08625-510 _v1.0 | February 2022

Table of Contents

Chapter 1. Version Highlights

This section provides highlights of the NVIDIA Data Center GPU R510 Driver (version 510.47.03 Linux and 511.65 Windows).

For changes related to the 510 release of the NVIDIA display driver, review the file "NVIDIA_Changelog" available in the .run installer packages.

- Linux driver release date: 2/1/2022
- Windows driver release date: 2/1/2022

1.1. Software Versions

For this release, the software versions are listed below.

- CUDA Toolkit 11: 11.6

  Note that starting with CUDA 11, individual components of the toolkit are versioned independently. For a full list of the individual versioned components (e.g. nvcc, CUDA libraries etc.), see the CUDA Toolkit Release Notes
- NVIDIA Data Center GPU Driver: 510.47.03 (Linux) / 511.65 (Windows)
- Fabric Manager: 510.47.03 (Use `nv-fabricmanager -v`)
- GPU VBIOS:
  - 92.00.19.00.01 (NVIDIA A100 SKU200 with heatsink for HGX A100 8-way and 4-way)
  - 92.00.19.00.02 (NVIDIA A100 SKU202 w/o heatsink for HGX A100 4-way)
- NVSwitch VBIOS: 92.10.14.00.01
- NVFlash: 5.641

Due to a revision lock between the VBIOS and driver, VBIOS versions >= 92.00.18.00.00 must use corresponding drivers >= 450.36.01. Older VBIOS versions will work with newer drivers.

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

1.2. New Features

General

- Added support for the following NVIDIA GPU products:
  - NVIDIA A100X
  - NVIDIA A30X
  - NVIDIA A10M
- OpenCL Vulkan Interop

  New external memory and semaphore sharing extensions provide a generic framework that enables OpenCL to import external memory and semaphore handles to synchronize with the external runtime, coordinating the use of shared memory. Refer to "NVIDIA OpenCL Vulkan Interop" on page 4 for details.
- NVIDIA OpenCL Compiler Upgrade

  The embedded OpenCL Just-In-Time compiler will offer an opt-in version utilizing CLANG 7.0 and NVVM 7.0 components, providing support for 16-bit floating point and 128-bit integer data types. Refer to "NVIDIA OpenCL Compiler Upgrade" on page 5 for details.

OpenCL Vulkan Interop

Starting with R510 driver version 511.65, the NVIDIA OpenCL driver has added support for the following new provisional extension specifications released by Khronos. The specifications are for OpenCL external semaphore and external memory.

- https://www.khronos.org/registry/OpenCL/specs/3.0-unified/html/OpenCL_Ext.html#cl_khr_semaphore
- https://www.khronos.org/registry/OpenCL/specs/3.0-unified/html/OpenCL_Ext.html#cl_khr_external_semaphore
- https://www.khronos.org/registry/OpenCL/specs/3.0-unified/html/OpenCL_Ext.html#cl_khr_external_memory

NVIDIA is seeking developer feedback on this new extension support.

OpenCL External semaphore and memory extensions

The set of new External Memory and Semaphore Sharing extensions provides a generic framework that enables OpenCL to import external memory and semaphore handles exported by external APIs—using a methodology that will be familiar to Vulkan developers—and then use those semaphores to synchronize with the external runtime, coordinating the use of shared memory.

The following key features are supported as part of these extensions:

- Importing memory into buffers using FD, Win32 KMT and NT handles
- Importing memory into images using FD, Win32 KMT and NT handles
- Importing binary semaphores using FD, Win32 KMT and NT handles
- Synchronizing using Wait and Signal on imported semaphores
- Using buffers and images imported in OpenCL kernels and other APIs such as other regular cl_mem.

Limitations of the Current Implementation

- Support for importing external memory and semaphores using FD handles on Linux and Win32 NT and KMT handles on Windows. No other handle types are currently available.
- Support for binary semaphores only.
- No support for exporting semaphore or memory from OpenCL.
- `clEnqueueAcquireExternalMemObjectsKHR` and `clEnqueueReleaseExternalMemObjectsKHR` APIs are currently not required as execution hand-off can be managed through semaphore wait and signal. But, these may be required in the future for correct functionality.

NVIDIA OpenCL Compiler Upgrade

The NVIDIA OpenCL driver uses an embedded OpenCL Just-In-Time (JIT) compiler based on the legacy 3.4 versions of the NVVM optimizer component and CLANG compiler front-end. NVIDIA plans to upgrade its OpenCL JIT compiler to use version 7.0 of the CLANG front-end and NVVM optimizer component.

NVIDIA is releasing this new OpenCL compiler as an opt-in feature in this driver release, with the default OpenCL compiler remaining the same. In a subsequent driver release, the default compiler will be switched to use the new components, replacing the old compiler. This approach provides an opportunity to solicit feedback from OpenCL customers and work to ensure this feature meets functional and performance expectations for key use-cases prior to the final release.

Enabling the OpenCL NVVM 7.0 Compiler

As part of this driver install, a new compiler library should be visible in the system folder as “libnvidia-compiler-next” along with the old “libnvidia-compiler”. The driver by default will pick the old compiler unless the following environment variable is set: “`NVCL_USE_NVVM70_COMPILER=1`”

NVVM 7.0 New Compiler Features

The new NVVM 7.0 based compiler takes advantage of years of development in the Clang +LLVM framework. In addition to several minor bug fixes and diagnostic improvements, this compiler introduces the following noteworthy features:

- **16-bit floating point (half) type**

16-bit floating point types or “half” type is available as a native data type in the new compiler.

This type is enabled by the “cl_khr_fp16” feature guard pragma

Example:

```
#pragma OPENCL EXTENSION cl_khr_fp16 : enable

half scalar_arith(half n, half k) {
  half w = n + k; 8
  half x = n - k; -1
  half y = w * x; -8
  half z = y / x; 8
  return -z;
}
__kernel void foo(__global int* x) {
  half a = 3.5H, b = 4.5H;
  if (scalar_arith(a, b) == -8.0)
   *x = 1;
  return;
}
```

- 128-bit integer type

  128-bit integer types or “(un)signed long long” is available as a native data type in the new compiler. This type is enabled by default and does not require any macros to be defined

  Example:

```
typedef unsigned long long ULL;
typedef long long LL;

LL scalar_arith(ULL n, ULL k) {
  LL w = n + k;
  LL x = n - k;
  LL y = w * x;
  LL z = y / x;
  return -z;
}

__kernel void foo(__global int* x) {
  ULL a = 0x123456789ABCDEF0ULL;
  ULL b = 0xFEDCBA9876543210ULL;
  if (scalar_arith(a, b) < 0)
   *x = 1;
  return;
}
```

- Upgraded math libraries

  The built-in standard math functions (e.g. sin(), cos()) have been upgraded to be on par with CUDA C++. This ensures that your application benefits from high-performance math routines optimized for the latest GPU architectures

Known Issues with NVVM 7.0-based Compiler

The new Clang/NVVM 7.0 based compiler has stricter error checking compared to the previous compiler. The following use-cases which were allowed with the older compiler may now throw an error.

- Updating const variables after they have been assigned

- Using address spaces other than global for kernel pointer parameters
- Using variadic arguments in functions and blocks

1.3. Fixed Issues

- **Security updates**: See *Security Bulletin: NVIDIA GPU Display Driver - February 2022*, which is available on the release date of this driver and is on the NVIDIA Product Security page.

1.4. Known Issues

General

- `nvidia-release-upgrade` may report that not all updates have been installed and exit.

  When running the `nvidia-release-upgrade` command on DGX systems running DGX OS 4.99.x, it may exit and tell users: "Please install all available updates for your release before upgrading" even though all upgrades have been installed.

  Users who see this can run the following command:

  ```
  sudo apt install -y nvidia-fabricmanager-450/bionic-updates --allow-downgrades
  ```

  After running this, proceed with the regular upgrade steps:

  ```
  sudo apt update
  sudo apt full-upgrade -y
  sudo apt install -y nvidia-release-upgrade
  sudo nvidia-release-upgrade
  ```

- By default, Fabric Manager runs as a systemd service. If using `DAEMONIZE=0` in the Fabric Manager configuration file, then the following steps may be required.

  1. Disable FM service from auto starting. (`systemctl disable nvidia-fabricmanager`)
  2. Once the system is booted, manually start FM process. (`/usr/bin/nv-fabricmanager -c /usr/share/nvidia/nvswitch/fabricmanager.cfg`). Note, since the process is not a daemon, the SSH/Shell prompt will not be returned (use another SSH shell for other activities or run FM as a background task).

- On NVSwitch systems with Windows Server 2019 in shared NVSwitch virtualization mode, the host may hang or crash when a GPU is disabled in the guest VM. This issue is under investigation.

GPU Performance Counters

The use of developer tools from NVIDIA that access various performance counters requires administrator privileges. See this note for more details. For example, reading NVLink utilization metrics from `nvidia-smi (nvidia-smi nvlink -g 0)` would require administrator privileges.

NoScanout Mode

NoScanout mode is no longer supported on NVIDIA Data Center GPU products. If NoScanout mode was previously used, then the following line in the "screen" section of /etc/X11/xorg.conf should be removed to ensure that X server starts on data center products:

```
Option         "UseDisplayDevice" "None"
```

NVIDIA Data Center GPU products now support one display of up to 4K resolution.

Unified Memory Support

Some Unified Memory APIs (for example, CPU page faults) are not supported on Windows in this version of the driver. Review the CUDA Programming Guide on the system requirements for Unified Memory

CUDA and unified memory is not supported when used with Linux power management states S3/S4.

IMPU FRU for Volta GPUs

The driver does not support the IPMI FRU multi-record information structure for NVLink. See the Design Guide for Tesla P100 and Tesla V100-SXM2 for more information.

OpenCL 3.0 Known Issues

- Device-Side-Enqueue related queries may return 0 values, although corresponding built-ins can be safely used by kernel. This is in accordance with conformance requirements described at https://www.khronos.org/registry/OpenCL/specs/3.0-unified/html/OpenCL_API.html#opencl-3.0-backwardscompatibility
- Shared virtual memory - the current implementation of shared virtual memory is limited to 64-bit platforms only.

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.

Supported Hypervisors

The following hypervisors are supported:

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 2016 Hyper-V Generation 2

Data Center GPU products now support one display of up to 4K resolution.

Supported Graphics Cards

The following GPUs are supported for device passthrough:

GPU Family	Boards Supported
NVIDIA Ampere GPU Architecture	NVIDIA A100, A40, A30, A16, A10
Turing	NVIDIA T4
Volta	NVIDIA V100
Pascal	Tesla: P100, P40, P4

GPU Family	Boards Supported
Maxwell	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 510 driver is supported on the following operating systems:

- Windows x86_64 operating systems:
  - Microsoft Windows® Server 2022
  - Microsoft Windows® Server 2019
  - Microsoft Windows® Server 2016
  - Microsoft Windows® 11
  - Microsoft Windows® 10
- The table below 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	POWER	Arm64 Server
OpenSUSE Leap 15.y (where y <= 3)	Yes	No	No
Red Hat Enterprise Linux / CentOS 8.y (where y <= 5)	Yes	Yes	Yes
Red Hat Enterprise Linux / CentOS 7.y (where y <= 9)	Yes	No	No
SUSE Linux Enterprise Server 15.y (where y <= 3)	Yes	No	Yes (see note)

Distribution	x86_64	POWER	Arm64 Server
Ubuntu 20.04.z LTS (where z<=3)	Yes	No	Yes
Ubuntu 18.04.z LTS (where z <= 6)	Yes	No	No

Note that SUSE Linux Enterprise Server (SLES) 15.3 is provided as a preview for Arm64 server since there are known issues when running some CUDA applications related to dependencies on `glibc 2.27`.

Supported Operating Systems and CPU Configurations for HGX A100

The Release 510 driver is validated with HGX A100 on the following operating systems and CPU configurations:

- Linux 64-bit distributions:
  - Red Hat Enterprise Linux 8.5 (in 4/8/16-GPU configurations)
  - Red Hat Enterprise Linux 7.9 (in 4/8/16-GPU configurations)
  - CentOS Linux 8.5 (in 4/8/16-GPU configurations)
  - CentOS Linux 7.9 (in 4/8/16-GPU configurations)
  - Ubuntu 18.04.6 LTS (in 4/8/16-GPU configurations)
  - SUSE SLES 15.3 (in 4/8/16-GPU configurations)
- CPU Configurations:
  - AMD Rome in PCIe Gen4 mode
  - Intel Skylake/Cascade Lake (4-socket) in PCIe Gen3 mode

Supported Virtualization Configurations

The Release 510 driver is validated with HGX A100 on the following configurations:

- Passthrough (full visibility of GPUs and NVSwitches to guest VMs):
  - 8-GPU configurations with Ubuntu 18.04.4 LTS
- Shared NVSwitch (guest VMs only have visibility of GPUs and full NVLink bandwidth between GPUs in the same guest VM):
  - 16-GPU configurations with Ubuntu 18.04.4 LTS
  - 1/2/4/8-GPU configurations with Windows x86_64 operating systems:
    - Microsoft Windows® Server 2022
    - Microsoft Windows® Server 2019
    - Microsoft Windows® Server 2016
    - Microsoft Windows® 10

API Support

This release supports the following APIs:

- NVIDIA® CUDA® 11.6 for NVIDIA® Maxwell™, Pascal™, Volta™, Turing™ and NVIDIA Ampere 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 (i.e. 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 Kepler architecture. This includes discontinued support for the following compute capabilities:

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

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

NVIDIA Server Platforms	
Product	**Architecture**
NVIDIA HGX A100	A100 and NVSwitch
NVIDIA HGX-2	V100 and NVSwitch

RTX-Series / T-Series Products	
Product	**GPU Architecture**
NVIDIA RTX A6000	NVIDIA Ampere
NVIDIA RTX A5000	NVIDIA Ampere

RTX-Series / T-Series Products	
Product	**GPU Architecture**
NVIDIA RTX A4000	NVIDIA Ampere
Quadro RTX 8000	Turing
Quadro RTX 6000	Turing
NVIDIA T1000	Turing
NVIDIA T600	Turing
NVIDIA T400	Turing

Data Center A-Series Products	
Product	**GPU Architecture**
NVIDIA A100X	NVIDIA Ampere
NVIDIA A100 NVIDIA A100 80 GB PCIe	NVIDIA Ampere
NVIDIA A40	NVIDIA Ampere
NVIDIA A30, A30X	NVIDIA Ampere
NVIDIA A16	NVIDIA Ampere
NVIDIA A10, A10M	NVIDIA Ampere

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

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

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

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