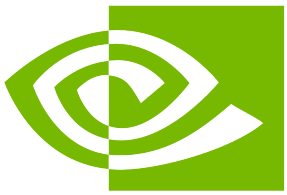

NVIDIA GPUDirect Storage Release Notes

Release r1.13.1

NVIDIA Corporation

Mar 13, 2025

Contents

Chapter 1. NVIDIA GPUDirect Storage Release Notes

Release information for NVIDIA® Magnum IO GPUDirect® Storage.

Chapter 2. Introduction

Release information for NVIDIA® GPUDirect® Storage (GDS) for developers and users.

NVIDIA Magnum IO GPUDirect Storage (GDS) is one of the members of the GPUDirect family of technologies. GDS enables a direct data path for direct memory access (DMA) transfers between GPU memory and storage. This direct path increases IO bandwidth, decreases IO latency and reduces the utilization load on the host CPU.

GDS is generally available on third party storage solutions such as DDN EXAScaler, Dell EMC Isilon, IBM Spectrum Scale, NetApp ONTAP and BeeGFS, WekaFS, VAST NFS, Dell Isilon, and Micron. See the *Support Matrix* for the complete list. GDS documents and online resources provide additional context for the optimal use of and understanding of GPUDirect Storage.

Refer to the following guides for more information about GDS:

- GPUDirect Storage Design Guide
- GPUDirect Storage Overview Guide
- cuFile API Reference Guide
- GPUDirect Storage Best Practices Guide
- GPUDirect Storage Installation and Troubleshooting Guide
- GPUDirect Storage Benchmarking and Configuration Guide Guide
- GPUDirect Storage O_DIRECT Requirements Guide

To learn more about GDS, refer to the following posts:

- GPUDirect Storage: A Direct Path Between Storage and GPU Memory
- The Magnum IO blog series.

Chapter 3. New Features and Changes

v1.13.1

- Assorted bug fixes.

Features introduced in previous releases:

v1.13

- Improved unregistered buffer IO performance for larger IO size using threadpool.
- Improved unregistered buffer IO performance by increasing the internal bounce buffer size. The size can be configured up to a max of 16MiB using the JSON parameter `per_buffer_cache_size_kb`.
- On Grace+Hopper systems, DDN Exascaler GDS p2p mode is supported with 64K kernel PAGE SIZE for transfer sizes that are 4KB aligned but not 64K aligned.
- Updated nvidia-fs.ko to support more PCIe Devices, to support Amazon Fsx for Lustre.
- Stability fixes during driver close and batch close operations.
- Added support for new “NVME P2PDMA” feature for GDS. This mode supports GDS with NVMe linux upstream drivers, minimum kernel version >= 6.2 (Ubuntu 22.04) and 5.14 (RHEL 9.4) on x86_64 based platforms. This feature will eliminate the need for custom MOFED NVMe patches and nvidia-fs.ko to support GDS with Ext4 and XFS with NVMe drives.

v1.11.1

- Assorted bug fixes.

v1.11

- Added support for RHEL 8.10, RHEL 9.4 and UB 24.04.
- Assorted bug fixes.

v1.10.1

- Assorted bug fixes.

v1.10

- Assorted bug fixes.

v1.9.1

- Assorted bug fixes.

v1.9

- Added support for RHEL 9.3 and UB 22.04.3.

v1.8.1

- Added support for RHEL 9.2 on Grace Hopper platform with 64K host OS page size for EXT4 filesystems with Local NVMe.
- Improved the IO throughput performance for applications by adding topology awareness in compatibility mode.

v1.8

- Assorted bug fixes.

v1.7.2

- Added Grace Hopper platform Support with 64K host OS page size for EXT4 filesystems with Local NVMe on Ubuntu 22.04 with HWE kernels.
- Proprietary NVIDIA kernel module is not supported. Only the NVIDIA open kernel module will be supported.
- cuFile APIs can be used in Cloud-service-providers environments in compatibility mode.

v1.7

- Support for APIs `cuFileStreamRegister`, `cuFileStreamDeregister`, `cuFileReadAsync`, and `cuFileWriteAsync` is complete. This enables use of CUDA Streams with cuFile APIs.
- cuFile APIs can be used with system memory.
- cuFile APIs can now be used with non-O_DIRECT file descriptors.
- Threadpool support is enabled by default and is required for cuFile APIs supporting CUDA streams.

v1.6.1

- Improved batch API performance.
- Implemented threadpool in the cuFile library to enable parallelism and improve throughput of a large IO request using a single user thread.

v1.5.1

- Assorted bug fixes.

v1.5:

- Added support for `cuMem*` memory allocations with cuFile APIs.

v1.4:

- Hopper PCIe support.
- RHEL 9.0 and Ubuntu 22.04 support.

v1.3.1:

- GDS can be now installed through CUDA `.run` files.
- Support for Ubuntu 22.04 and RHEL9.
- Improvements to NIC to GPU affinity for userspace RDMA file systems.

v1.3

- Initial support for Linux dma-buf.

v1.2.1

- GDS now supports vGPU in VMware context. Refer to https://docs.nvidia.com/grid/latest/grid-vgpu-release-notes-generic-linux-kvm/index.html#gpudirect-technology-support and https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html#cuda-open-cl-support-vgpu for more information.

v1.2

- Support for BeeGFS.
- Support for XFS.
- Batch APIs available for use (Alpha level support).

v1.1.1:

- Use `nvidia_peermem` default for userspace RDMA filesystems (GPFS, Weka). In order to use `nvidia_peermem`, load it using:

```
# modprobe nvidia_peermem
```

- Added support for BeeGFS (preview).

v1.1:

- The XFS file system has been added to the list of supported file systems at a a beta support level.
- Improved support for unregistered buffers.
- Added options `start_offset` and `io_size` to gdsio config file per job options.
- Improved performance of 4K and 8K IO sizes for local file systems.
- Added user-configurable priority for internal cuFile CUDA streams.

v1.0:

- New configuration and environment variables for the cuFile library.
- Fixed error handling behavior for Weka retriable and unsupported errors.
- Removed hard dependency on `librcu-bp`.
- Added read support for IBM Spectrum Scale.

v0.95:

- Compatibility with POSIX IO is enabled by default.
- Alpha level support for RHEL 8.3.
- GDS is available as Technical preview for DGX OS.
- Support for MLNX_OFED 5.3 for NVMe and NVMeOF.
- Support for Excelero NVMesh devices.
- Support for ScaleFlux computational storage.
- Integration with DALI® and PyTorch.
- Experimental RAPIDS integration for cuDF, unoptimized, reads only.

Chapter 4. MLNX_OFED and File System Requirements

The following are the MLNX_OFED and file system requirements for GDS:

- MLNX_OFED must be installed **before** installing GDS. Refer to Installing GPUDirect Storage for more information about installing MLNX_OFED.
- `nvidia-fs.ko` requires Linux kernels 4.15.x and above.

Note

Ubuntu 22.04.3 is not supported with any publicly available MLNX_OFED versions at this time.

MLNX_OFED version	Distros supported	Notes
5.4-x (LTS)	Ubuntu 18.04, 20.04,22.04, RHEL 8.x (>8.4), RHEL 9	Long-term support version
5.5-x	Ubuntu 18.04, 20.04, RHEL 8.4, RHEL 8.6	
5.6-x	Ubuntu 18.04, 20.04, RHEL 8.4, RHEL 8.6	
5.7-x	Ubuntu 18.04, 20.04, RHEL 8.4, RHEL 8.6	Does not support RHEL9 and UB22.04
5.8-x (LTS)	Ubuntu 18.04, 20.04,22.04, RHEL 8.x (>8.4), RHEL 9, Rocky Linux 9.x, RockyLinux 8.x	
5.9-x	UB22.04 and RHEL 9.1, 8.7	NVMeOF is not functional.
23.04-x	UB22.04 and RHEL 9.2, 8.8	NVMeOF is not functional.
23.07-x	UB22.04 and RHEL 9.2, 8.8	NVMeOF is not functional
23.10	UB22.04 and RHEL 9.2	
24.04-x	UB24.04 and RHEL 9.4	

Chapter 5. Support Matrix

Supported GPUs: Data Center and Quadro (desktop) cards with compute capability > 6 listed here are supported in GDS mode. All other cards are supported only in compatibility mode.

Partner/Distributed File Systems

Partner Comany	Partner Product Version	Compatible GDS Version	Date
DDN	EXAScaler 5.2 and newer EXAScaler 6.0 and newer	1.1 and higher	November 2021
DellEMC	PowerScale 9.2.0.0	1.0	October 2021
Hitachi Vantara	HCSF	1.0	October 2021
HPE Ezmeral	5.5	1.3.1 and higher	February 2023
HPE Cray ClusterStor	Neo 4.2 and newer	1.0 and higher	September 2021
HPE GreenLake File Storage	3.0	1.10	June 2024
IBM	Spectrum Scale 5.1.2 and newer	1.1 and higher	November 2021
NetApp	ONTAP 9.10.1	1.0 and higher	January 2022
NetApp ThinkParQ System Fabrics Works	7.3.0	1.1.1 and higher	March 2022
Pure Storage	FlashBlade	1.7 and higher	December 2023
VAST	Universal Storage 4.1	1.1 and higher	November 2021
WekaIO	WekaFS 3.13	1.0	June 2021

Note

Distributed file systems are not supported on Grace CPU (NVIDIA's Arm-based CPU) based platforms.

Chapter 6. GDS Enabled Libraries/Frameworks

GDS has been enabled in the following libraries and frameworks:

- RAPIDS cuDF: More details
- CLARA cuCIM: More details
- DALI: Python frameworks such as PyTorch are enabled to use DALI, which is in turn enabled with GDS: More details
- MONAI: Python Framework for Medical Imaging and Deep learning: More details
- Clara Parabricks: More details

Chapter 7. Included Packages

The GDS package contains the following Debian packages:

- `gds-tools-12-8_*.deb`
- `libcufile-12-8*.deb`
- `libcufile-dev-12-8_*.deb`
- `nvidia-fs_2.24.*.deb`
- `nvidia-fs-dkms_2.24.*.deb`
- `nvidia-gds-12-8_*.deb`
- `nvidia-gds_12-8.*.deb`

Note

Each component has a README file. For example, for `gds-tools`, the README file is in the `/usr/local/CUDA-12-8/gds/tools/` directory.

Chapter 8. Minor Updates and Bug Fixes

The following minor updates and bug fixes were made in version **1.13.1**:

- Fixed a bug in the gdscheck `-f` option to use the system page size instead of a fixed 4K size for I/O.
- Fixed a race condition between the `cuFileBatchIODestroy` API and the `cuFileDriverClose` API.

Chapter 9. Known Issues

- NVME P2PDMA mode is currently not supported on RHEL 9.5 for x86.
- NVME P2PDMA can fail on x86_64 platforms when KASLR is enabled in the linux kernel. The feature can be disabled by specifying the `nokaslr` kernel parameter.
- Without nvidia-persistenced enabled, NVME P2PDMA can increase the time taken to start the driver initialization for B200 platforms.
- Sparse file IO performance on Ext4 with NVME P2PDMA is slower on x86_64 platforms compared to GDS P2PDMA with `nvidia-fs.ko`.
- When installing via .run file, the installation of GPUDirect Storage components fails if MLN_OFED is not installed.
- The cuFile library may fail to load GPU topology information properly when udev information is not accessible to the application. Ensure the correct udev library is installed.
- `nvidia-fs` can deadlock in `nv_p2p_dma_map_pages` and `nv_p2p_mem_info_free_callback` functions when the user frees the CUDA memory without calling `cuFileBufDeregister` on registered buffers.
- On DDN EXAScaler file systems:
  - With stripe count > 1, `cuFileRead` and `cuFileWrite` do not work with poll mode enabled for versions older than 2.12.5_ddn10.
  - With 2.12.5_ddn10, any reads beyond EOF causes a BUG_ON inside `nvidia-fs`.
- The `cuFileRead` and `cuFileWrite` APIs fail when working on `cuMemMap` allocations with multiple GPUs, when the IO request to a GPU buffer is not 4K aligned and spans across multiple GPUs.

Chapter 10. Known Limitations

This section provides information about the known limitations in this release of GDS.

- GPUDirect Storage with `NVME P2PDMA` feature is not supported with RAID0 and multipath NVMe support.
- vGPU supports the latest GPUs with OpenRM; that is, GPUs with Ada Lovelace, Hopper and later are supported. A100 and earlier GPUs are not supported with OpenRM.
- All OS (BaseOS/RHEL/SLES) for Grace-Hopper need persistence enabled.
- CUDA streams based APIs:
  - CUDA graphs are not supported with cuFile Stream APIs.
  - cuFile Stream APIs for GPFS and WekaFS are supported in compatibility mode only.
  - cuFile Stream APIs are not supported when cuFile configuration parameter `execution.parallel_io` is false or `execution.max_io_threads` is set to 0.
  - `CU_FILE_STREAMS_SUPPORTED` bit is not set in `Props.fflags` when queried with `cuFileDriverGetProperties`.
- Available BAR1 memory reported by the`nvidia-smi` utility is not accurate due to some internal overhead of the CUDA toolkit. Therefore, a huge allocation of BAR1 memory by any GDS application can run into ENOMEM errors, even when `nvidia-smi` utility shows there is available BAR1 memory.
- GPUDirect storage in P2P mode does not support NVMe end-to-end data protection features. To support GDS in p2p mode, the NVMe must be formatted with Protection Information where `Metadata Size` is set to zero bytes.
- CentOS 7.x is no longer supported.
- Checksums on the client-side of file systems must be disabled for GDS.
- cuFile APIs are not supported with applications that use the `fork( )` system call.
- GDS Compatibility mode is only tested on GDS qualified file systems: ext4, EXAScaler, XFS, WekaFS, IBM Spectrum Scale, VAST, and BeeGFS.
- On x86-64 platforms, GDS with “IOMMU=on” or ACS enabled are not guaranteed to work functionally or in a performant way.
- Refer to the following documentation for IBM Spectrum Scale Limitations with GDS: https://www.ibm.com/docs/en/spectrum-scale/5.1.5?topic=architecture-gpudirect-storage-support-spectrum-scale
- Upgrading of Linux Kernel version and `nv_peer_mem`:

WekaFS does not support newer MLNX_OFED versions 5.3.x and above with GDS. `nvidia-peer-memory-dkms=1.1-0-nvidia2` is required for GDS support with WekaFS. Please follow the instructions in section 2.2 of the *GDS Troubleshooting and Installation Guide*.

- RHEL 8.3 or later does not have default udev rules for detecting RAID members, which disables GDS on RAID volumes. Refer to the section “Adding udev Rules for RAID Volumes” in *GDS Installation and Troubleshooting Guide*.
- Sparse file read performance can be slow with the “NVME P2PDMA” feature.

Chapter 11. 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.

Chapter 12. OpenCL

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

Chapter 13. Trademarks

NVIDIA, the NVIDIA logo, CUDA, DGX, DGX-1, DGX-2, DGX-A100, Tesla, and Quadro are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright