GDS

GPUDirect Storage

v11.4 | September 2021

Table of Contents

GPUDirect Storage Overview

GDS enables a direct data path for direct memory access (DMA) transfers between GPU memory and storage. This direct path increases system bandwidth and decreases the latency and utilization load on the CPU.

Benefits of GDS include:

- Reduces the performance impact and dependence on CPUs to process storage data transfer.
- Performance force multiplier on top of the compute advantage for computational pipelines that are fully migrated to the GPU so that the GPU, rather than the CPU, has the first and last touch of data that moves between storage and the GPU.
- Supports interoperability with other OS-based file access, which enables data to be transferred to and from the device by using traditional file IO, which is then accessed by a program that uses the cuFile APIs.

Benefits that are provided by the cuFile APIs and their implementations:

- A family of APIs that provide CUDA applications with the best-performing access to local or distributed file and block storage.

  Block storage validation might be added in the future.
- These APIs are consistent with the long-term direction of the Linux community, for example, with respect to peer to peer RDMA.
- When transferring to and from the GPU, increased performance relative to existing standard Linux file IO.
- Greater ease of use by removing the need for the careful expert management of memory allocation and data movement.
- A simpler API sequence that is relative to existing implicit file-GPU data movement methods, which require a more complex management of memory and data movement on and between the CPU and GPU.
- Broader support for unaligned transfers than POSIX `pread` and `pwrite` APIs with `O_DIRECT`.

  In the application code, the POSIX APIs require a buffered IO or unaligned handling.

- Generality across a variety of storage types that span various local and distributed filesystems, block interfaces, and namespace systems, including standard Linux and third-party solutions.

Here are the benefits that are provided by the Stream subset of the cuFile APIs:

- Asynchronous offloaded operations are ordered with respect to a CUDA stream.
  - 0 after compute: The GPU kernel produces data before it is transferred to IO.
  - Compute after IO: After the data transfer is complete, the GPU kernel can proceed.
- Available concurrency across streams.
  - Using different CUDA streams allows the possibility of concurrent execution and the concurrent use of multiple DMA engines.

The full GDS documentation is available at GPUDirect Storage.

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.

OpenCL

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

Trademarks

NVIDIA and the NVIDIA logo 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