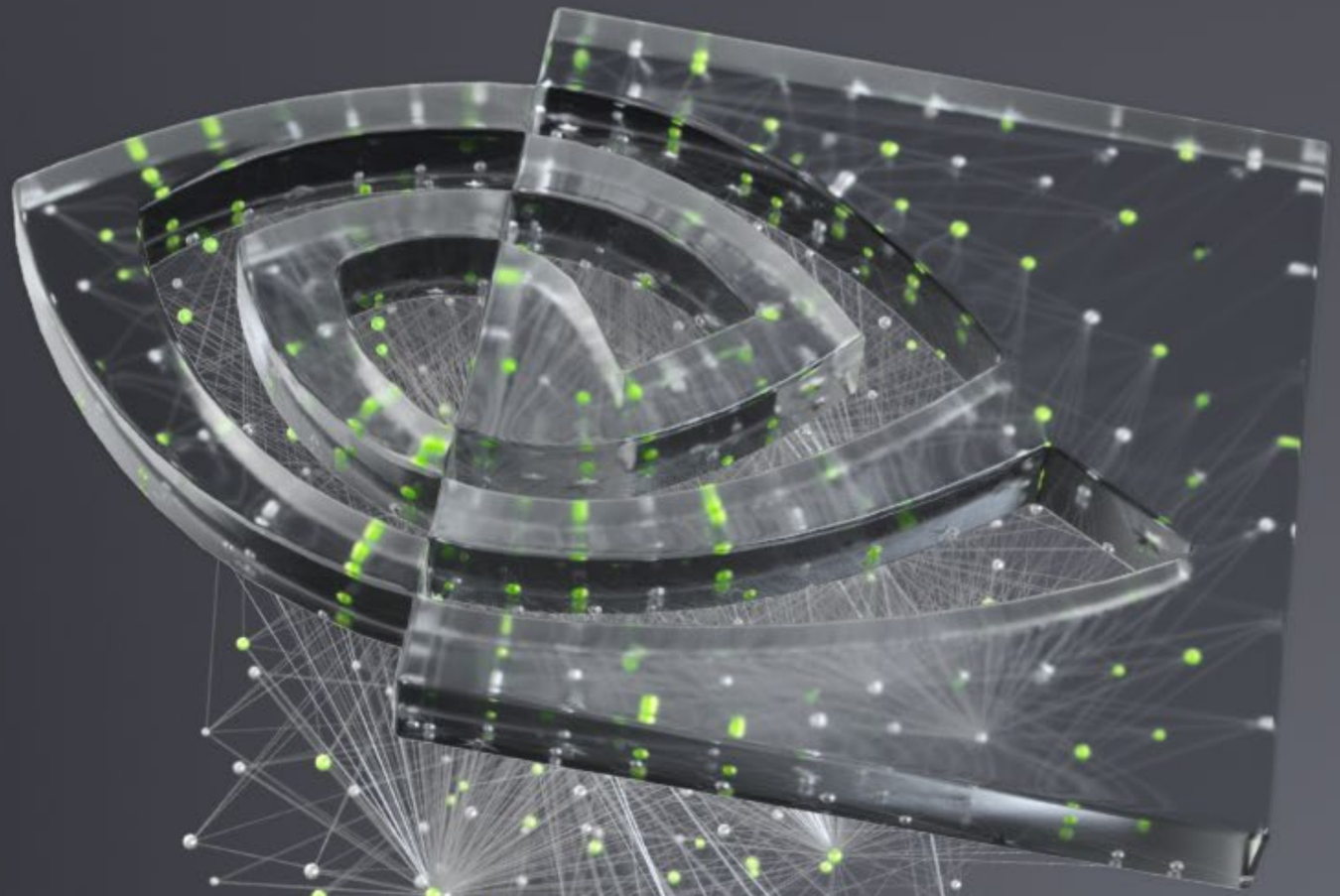
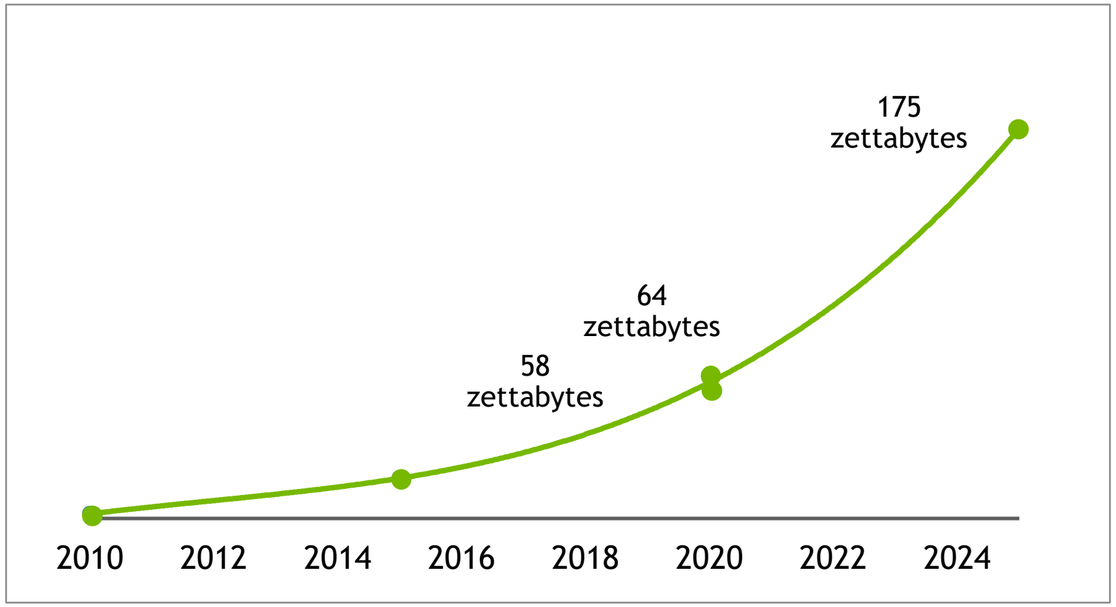


MAGNUM IO GPUDIRECT STORAGE (GDS)

Kushal Datta | STORAGE DEVELOPER CONFERENCE, SEP 28-29, 2021

AI #EXTREMESCALE

Exploding Data in HPC, Cloud and Enterprise



BIG DATA GROWTH

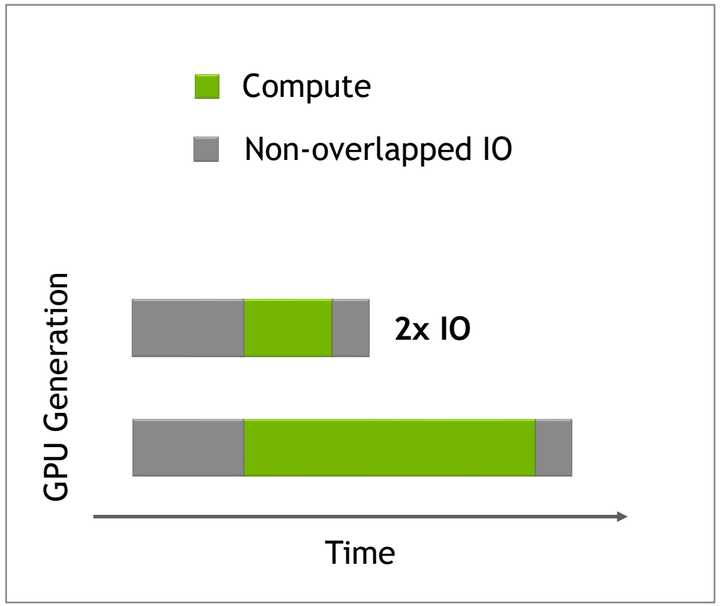
90% of the world's data in the last 2 years

GROWTH IN SCIENTIFIC DATA

Fueled by accurate sensors and simulations

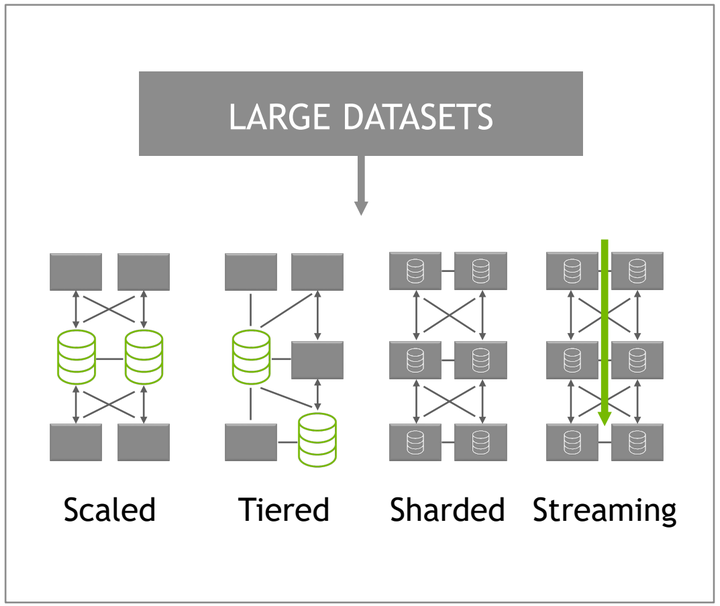
Source for Big Data Growth chart: IDC – The Digitization of the World (May 2020) storagenewsletter.com: 64.2ZB of Data Created or Replicated in 2020

EXTREME-SCALE AI REQUIRES EXTREME IO



LATENCY

IO latency dominates as compute shrinks



THROUGHPUT

Extreme throughput required as microservices scale across data center

180
Industry-Leading Organizations

2,500
Active Contributing Members

COMPATIBILITY

Diverse set of standards and solutions in the ecosystem

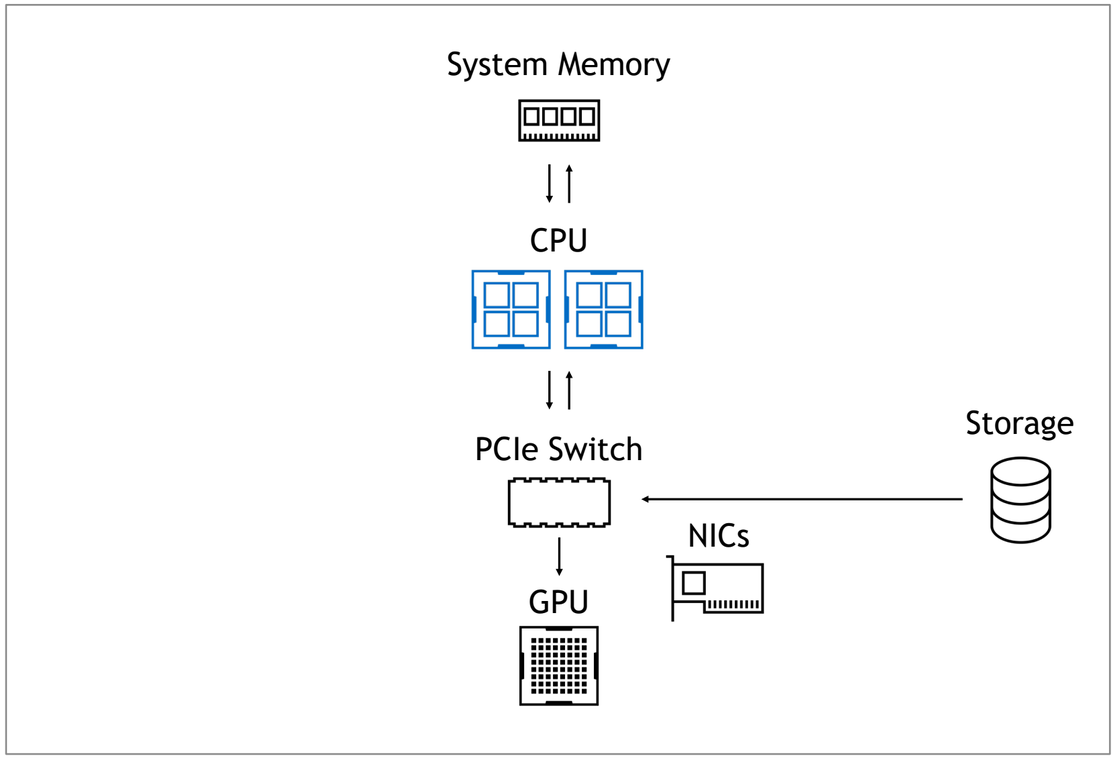
GDS Architecture and cuFILE

Benchmarks and Use Cases

Ecosystem and CTA

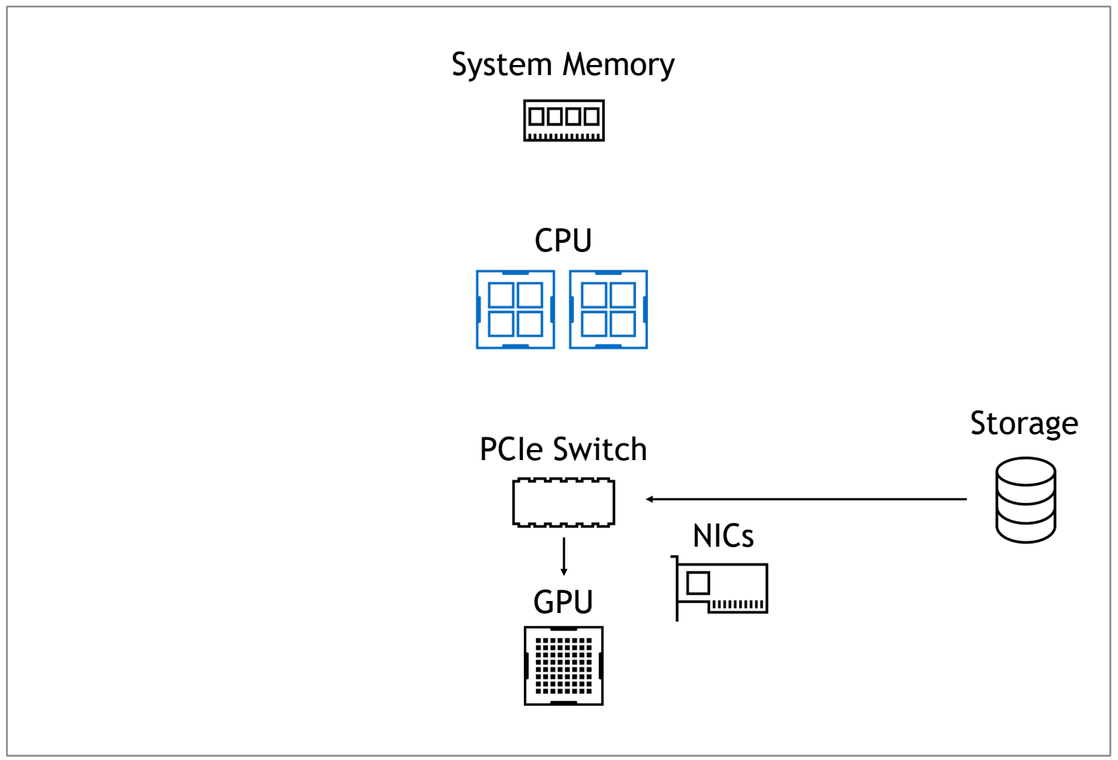
GETTING DATA TO GPUs: IO

WITHOUT GPUDIRECT STORAGE



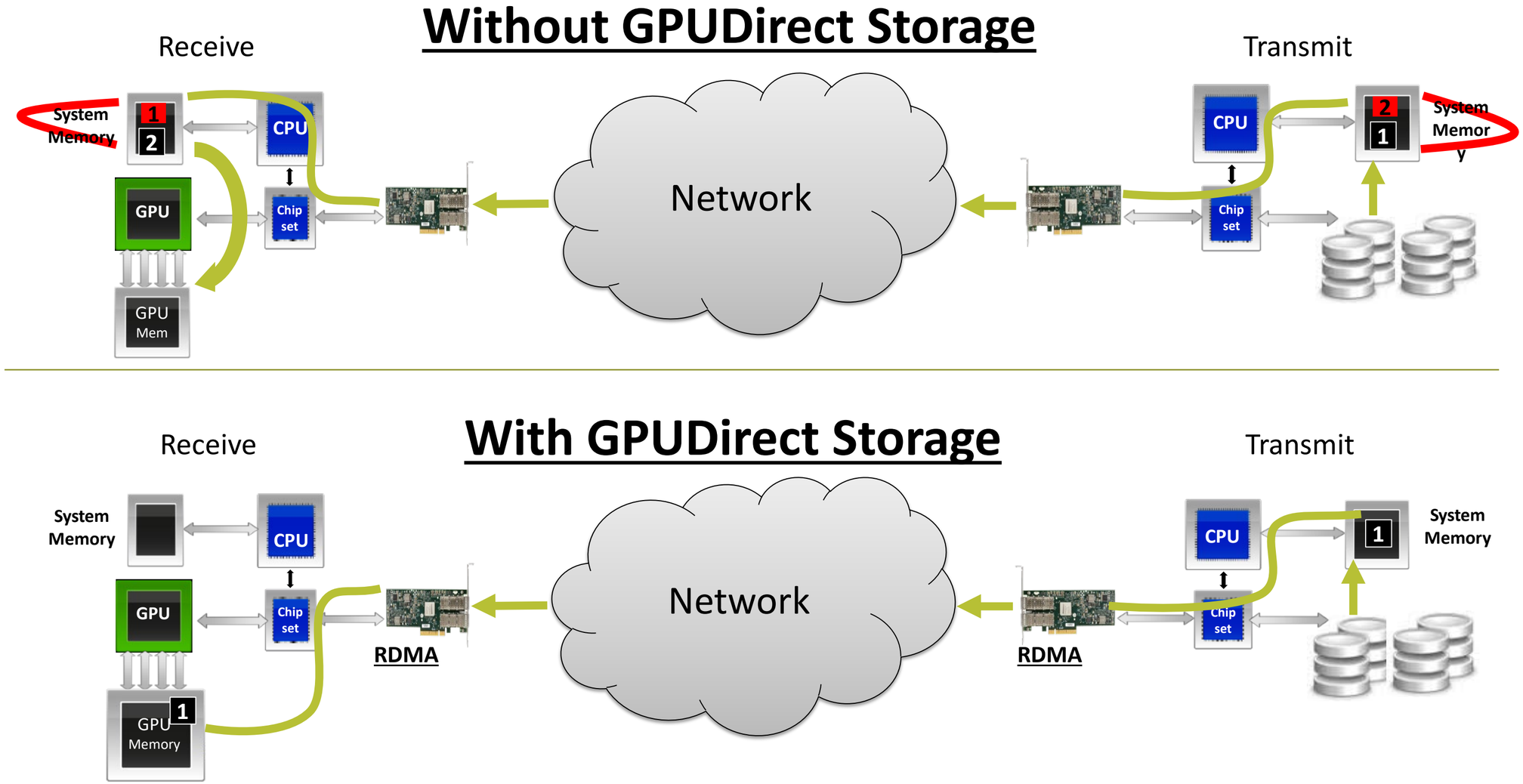
Low Bandwidth | High Latency | Limited Capacity

WITH GPUDIRECT STORAGE



Higher Bandwidth | Lower Latency
O(PB) capacity | CUDA programming model

GPUDIRECT STORAGE(GDS) ADVANTAGE



NFSoRDMA
NVMe-oF/RDMA

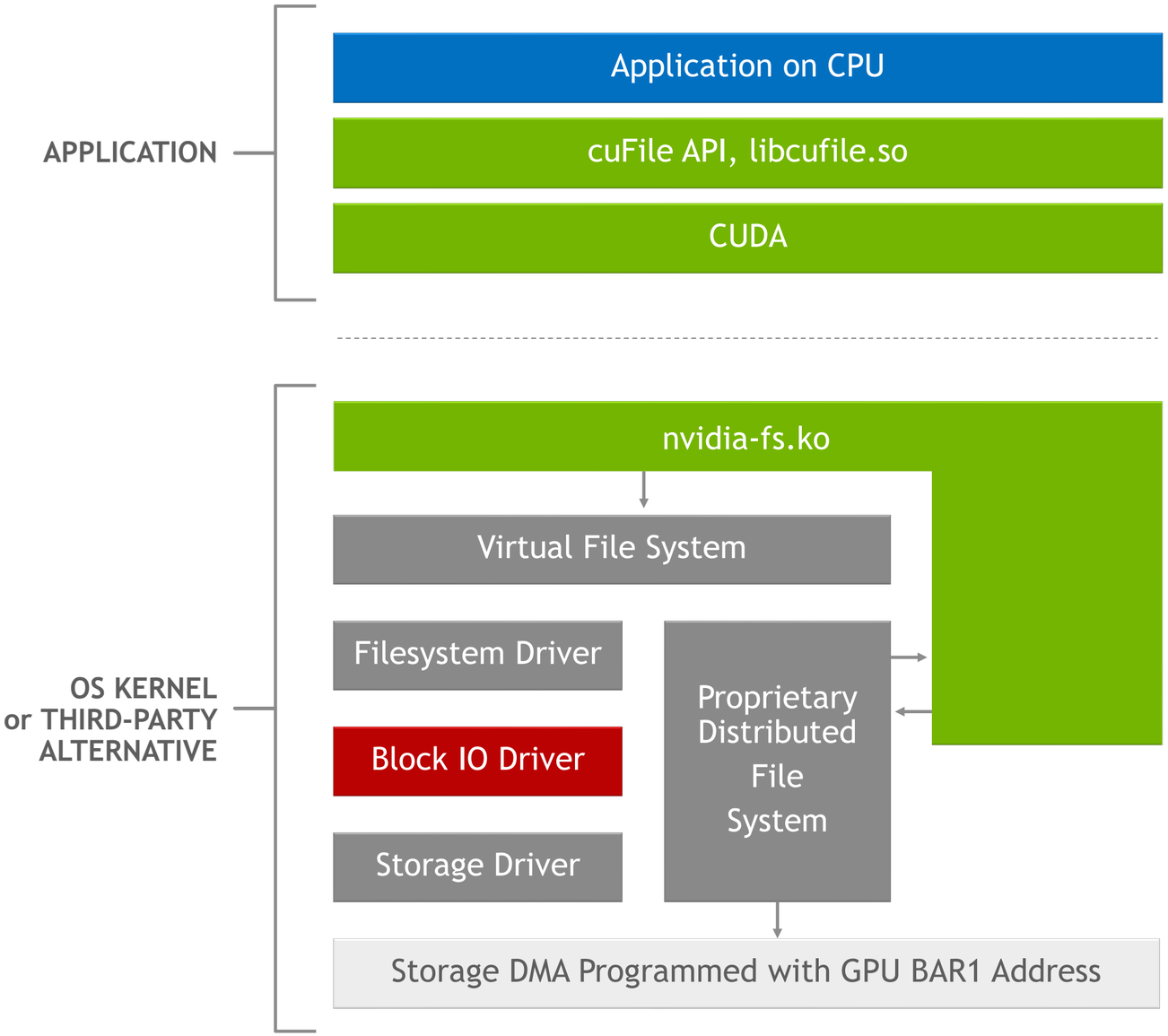
GPUDIRECT STORAGE SW ARCHITECTURE

cuFile user API
Enduring API for applications and frameworks

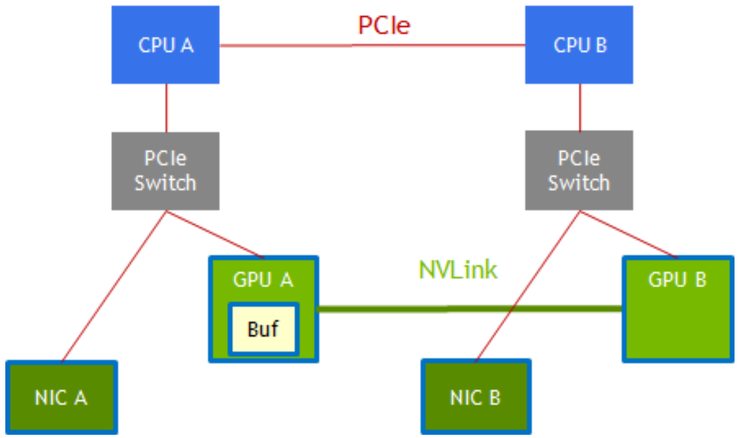
nvidia-fs driver API
For file system and block IO drivers

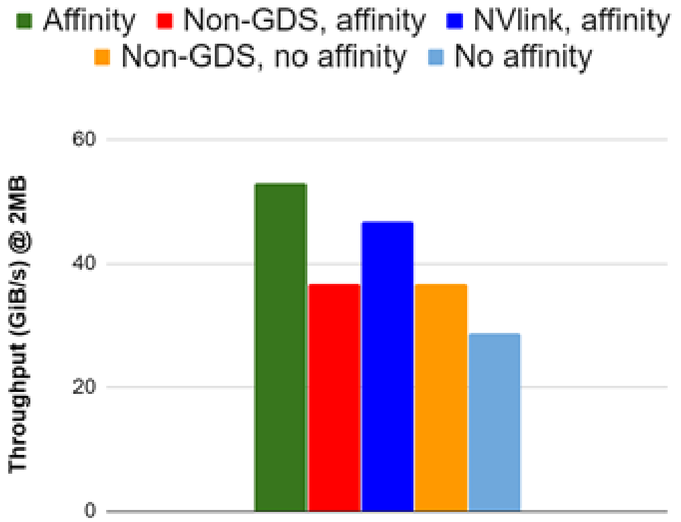
Vendor-proprietary solutions: no patching avoids lack of Linux enabling

NVIDIA is actively working with the community on upstream first to enable Linux to handle GPU VAs for DMA



GPUDIRECT STORAGE: DYNAMIC ROUTING





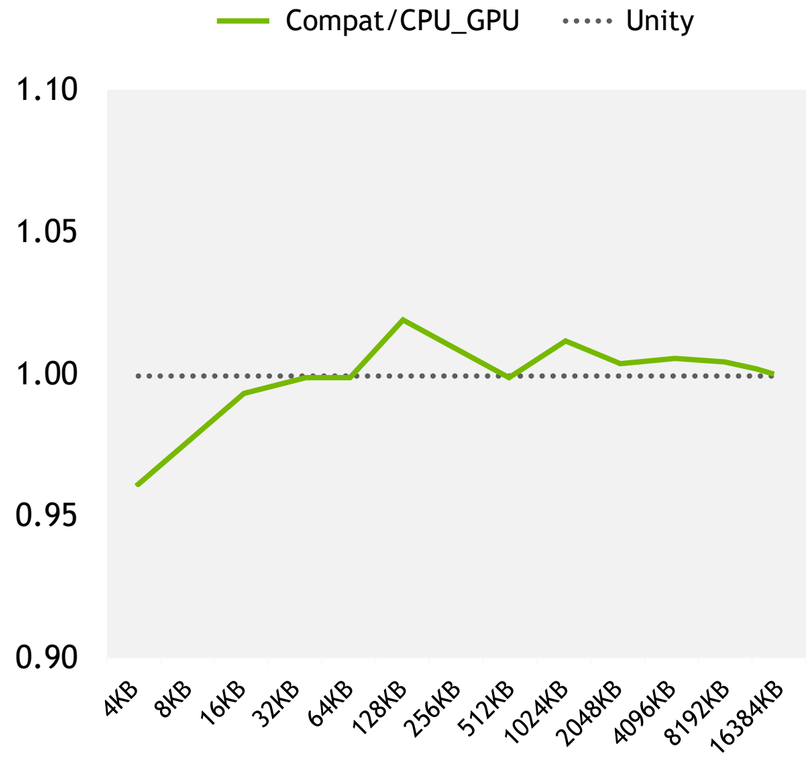
4 Gen4 drives
DGX A100

GDS, affinity	**$NVMe_A \rightarrow GPU_A$ Target**	**Best overall**
Non-GDS, affinity	**$NVMe_A \rightarrow CPU_A$ Buf $\rightarrow GPU_A$ Target**	Slower: bounce buffer
NVLink, no affinity	**$NVMe_A \rightarrow GPU_A$ Buf $\rightarrow GPU_B$ Target**	Best for cross-socket
Non-GDS, no affinity	**$NVMe_A \rightarrow CPU_A$ Buf $\rightarrow GPU_B$ Target**	Agnostic to affinity thru CPU
GDS, no affinity	**$NVMe_A \rightarrow GPU_A$ Buf $\rightarrow$ CPU $\rightarrow GPU_B$**	Some overheads

COMPATIBILITY MODE

No Appreciable Degradation Relative to Non-GDS at 128+KB

- cuFile APIs remain functional when:
  - GDStorage-enabled drivers and nvidia-fs.ko are not installed
  - Mounted file system is not GPUDirect Storage compatible
  - File system's specific conditions for O_DIRECT are not met
  - Development is without root privileges
- Configurable in config.json by admin or user
- Advantages:
  - Enable code development on platforms without GPUDirect Storage support
  - Containerized deployment across a range of systems
- cuFile vs. POSIX APIs for moving data to GPU memory:
  - Performance is on par: slight overhead at smallest IO sizes
  - Measured results on DGX-2 and DGX A100 with local storage



CUFILE API

cuFILE LIBRARY

- **Features**
  - Shared user-space file IO library for GDS-ready DMA and RDMA file systems
  - Supports aligned and unaligned file IO offsets and sizes
  - Avoids cudaMemcpy, intermediate copies to page cache, and user-allocated sys memory
  - User-space RDMA connection and buffer registration management
  - Dynamically routes IO using optimal path based on HW topology and GPU resources
  - Supports compatibility mode that works with user-level installation only for development
- **Requirements**
  - Needs files to be opened in O_DIRECT mode to bypass page cache and buffering in system memory for POSIX-based file systems
- **Documentation**
  - https://docs.nvidia.com/gpudirect-storage/overview-guide/index.html

USAGE EXAMPLE

Write from GPU Memory to Local or Remote Storage

```
int main(void) {
        …                                                       // set GPU device, etc.

        CUfileError_t status;
        CUFileDescr_t cf_descr;                                  // general enough to support Windows
        CUFileHandle_t cf_handle;

        status = cuFileDriverOpen();                            // initialize

        fd = open(TESTFILE, O_WRONLY|O_DIRECT, 0644);           // interop with normal file IO
        cf_descr.handle.fd = fd;
        status = cuFileHandleRegister(&cf_handle, &cf_descr);   // Check support for file at this mount

        cuda_result = cuMemAlloc(&devPtr, size);                // User allocates memory

        status = cuFileBufRegister(devPtr, size);               // performance optimization
        assert(cudaMemset((void *) devPtr, 0xab, size) == cudaSuccess);
        ret = cuFileWrite(cf_handle, devPtr, size, 0, 0);       // ~pwrite: file handle, GPU base address, size,

                                                                //  offset, GPU buffer offset

        status = cuFileBufDeregister(devPtr);                   // optional cleanup for good hygiene
        cuFree(devPtr);
        cuFileHandleDeregiser(cf_handle);
        close(fd);
        cuFileDriverClose();
        return 0;
}
```

HOW TO DEVELOP WITH GDS?

What SW Is Needed to Get GDS to Work with DGX A100

USAGE SCENARIO	MECHANISM OF ACTION
C++ CUDA application	Install GDS (UML + KMD) Code change required to use cuFile API*
Python application w/ NumPy reader	Install GDS (UML + KMD) Import DALI reader (open-source)*
Spark/RAPIDS application	Install GDS (UML + KMD) GDS integrated

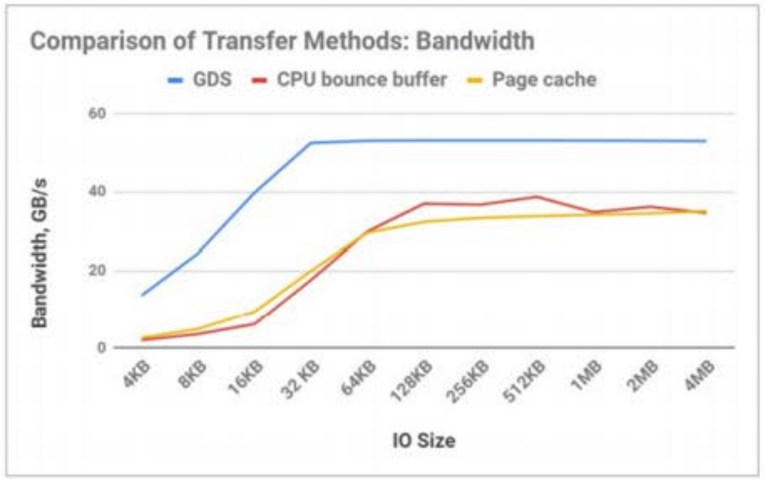
*ONE CHANGE FITS ALL

EXT4 with local (NVMe) or direct attached storage (NVMe-oF)
NFS (VAST, Isilon/PowerScale) | DFS (EXAScaler, WekaFS)

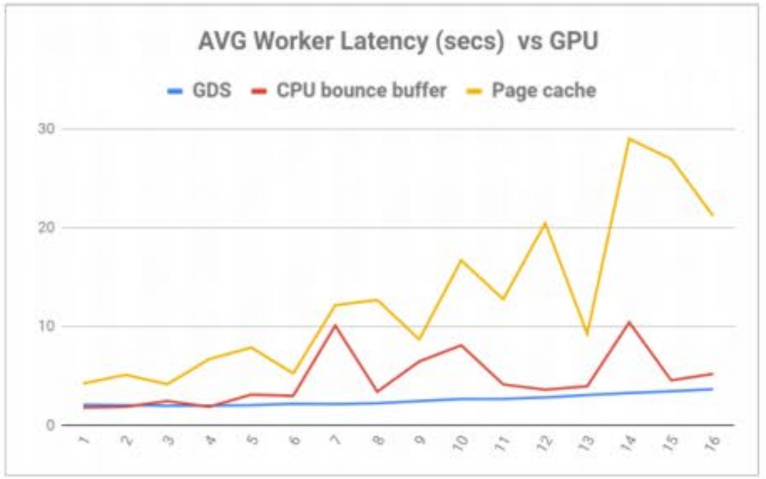
BENCHMARKING GDS

GPUDIRECT STORAGE - BENEFITS

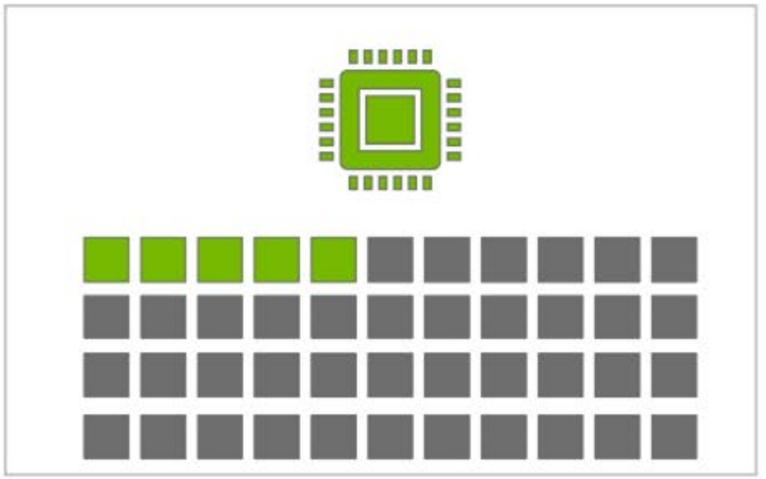
Higher Bandwidth, Lower Latency, Lesser CPU utilization



Higher bandwidth – Direct path leads to more throughput



Low latency with GPUDirect Storage
Fairly flat (Predictable)



Peak IO bandwidth using fewer CPU cores for IO

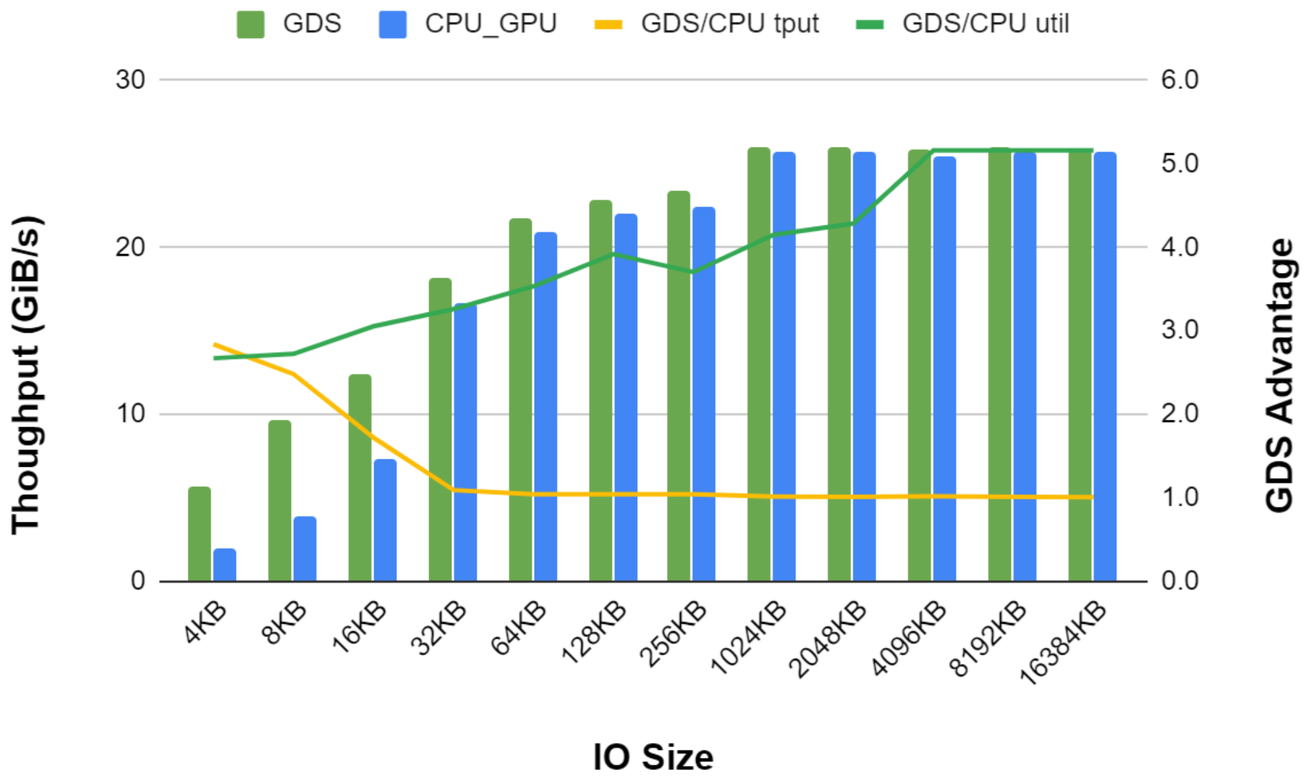
DGX A100 LOCAL STORAGE COMPARISON

GDS advantage is capped by limited throughput of local drives

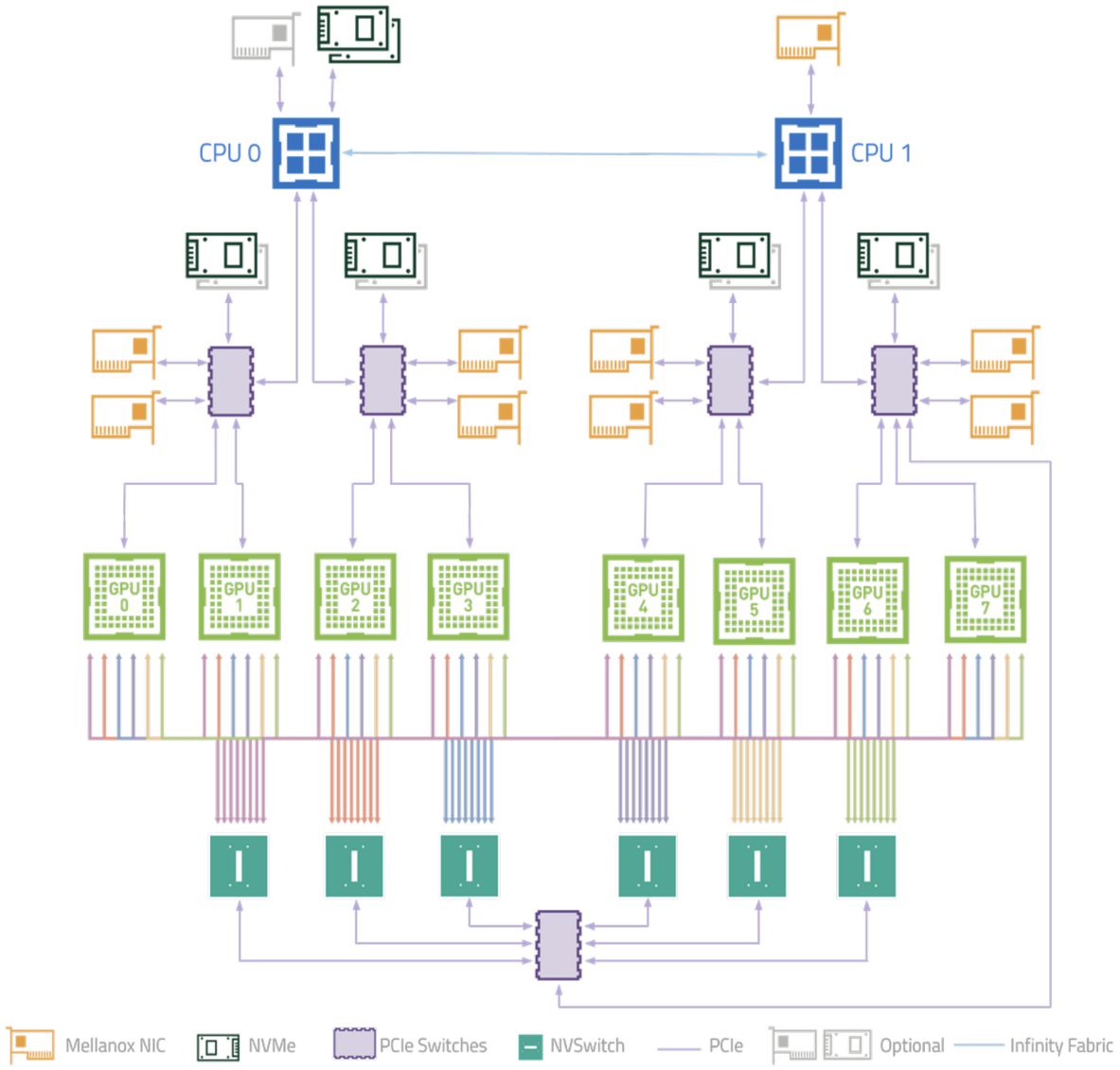
2.8x performance advantage
at 4K for no copy shrinks to 1x when amortized on larger IO sizes

3-5x CPU utilization advantage

4 drives @ 26 GiB/s saturate 1 GPU



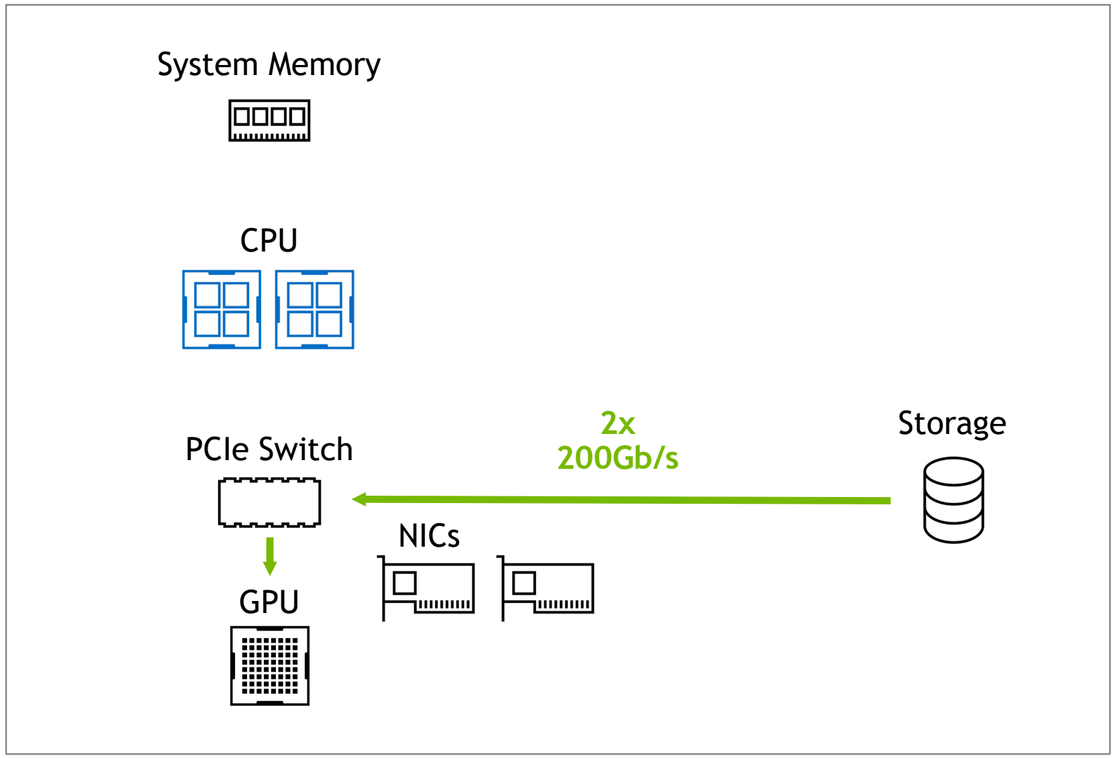
NVIDIA DGX A100 SYSTEM DIAGRAM



GDS CONFIGURATIONS ON DGX

Lower Latencies and Increased CPU Efficiency

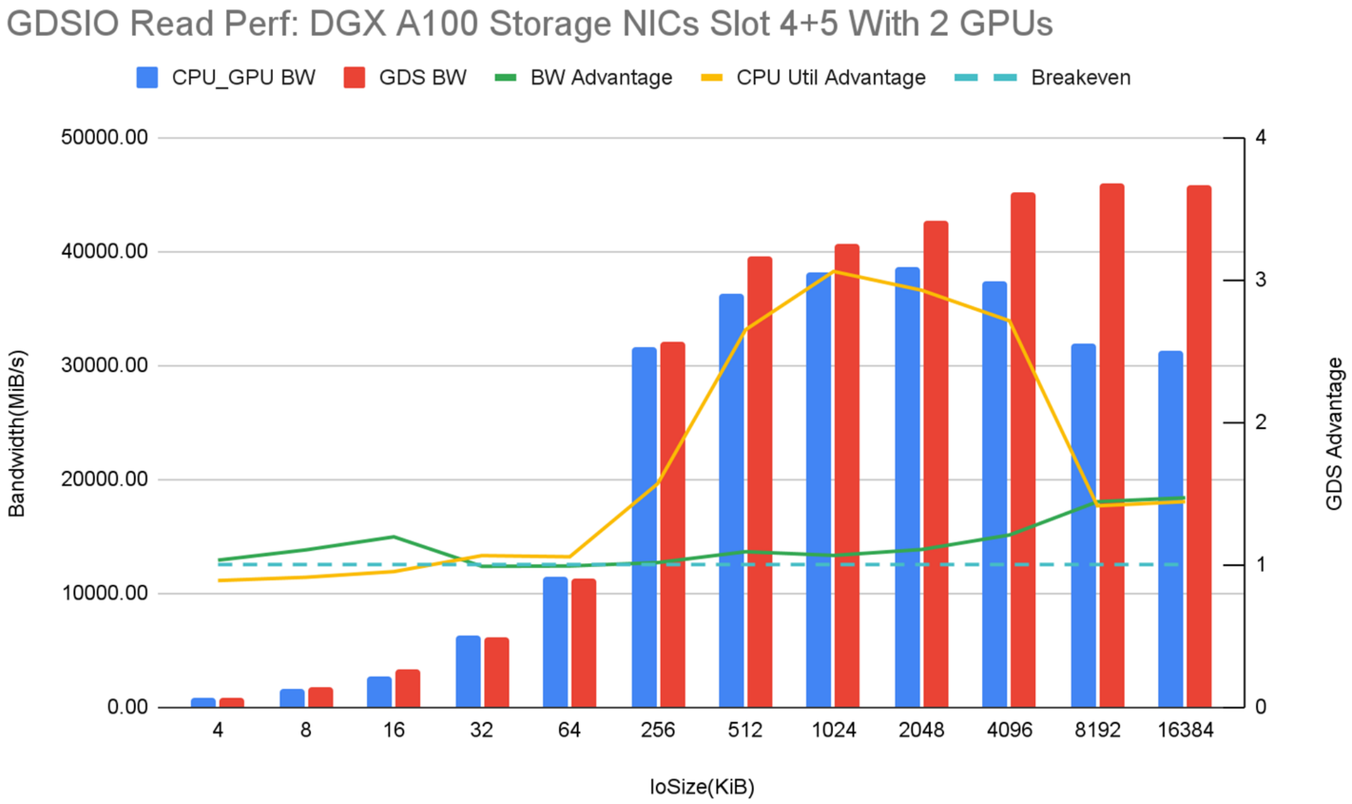
STANDARD GDS CONFIGURATION IN DGX A100



North-south storage configuration adapters 4 and 5 used for GDS

2 N-S NICS - GDS VS CPU-GPU (READ)

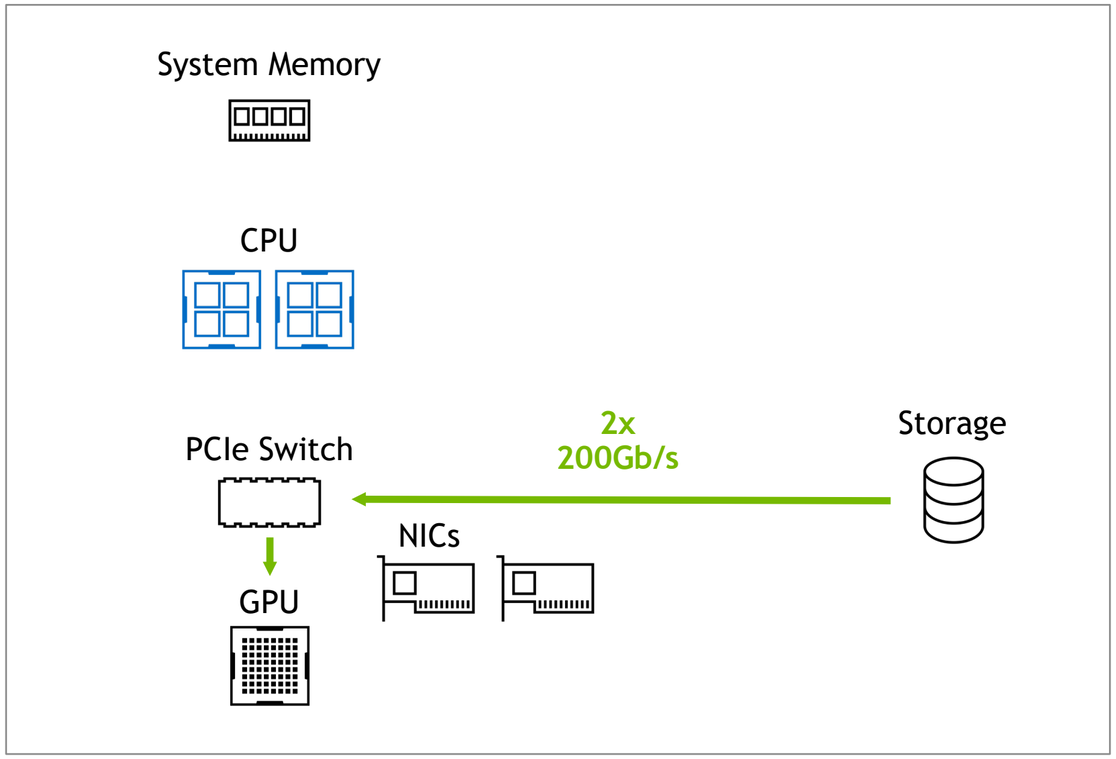
- Use both E-W NIC on slot 4 (socket 1) and slot 5 (socket 0).
- Use GPU 7 (socket 1) and GPU 3 (socket 0) for best perf.
- GDS has up to 1.47x in BW advantage over CPU-GPU.
- GDS has up to 3x in CPU utilization advantage.
- BW achieved with 2 E-W NICs is nearly 2x of 1 E-W NIC.



GDS CONFIGURATIONS ON DGX

Lower Latencies and Increased CPU Efficiency

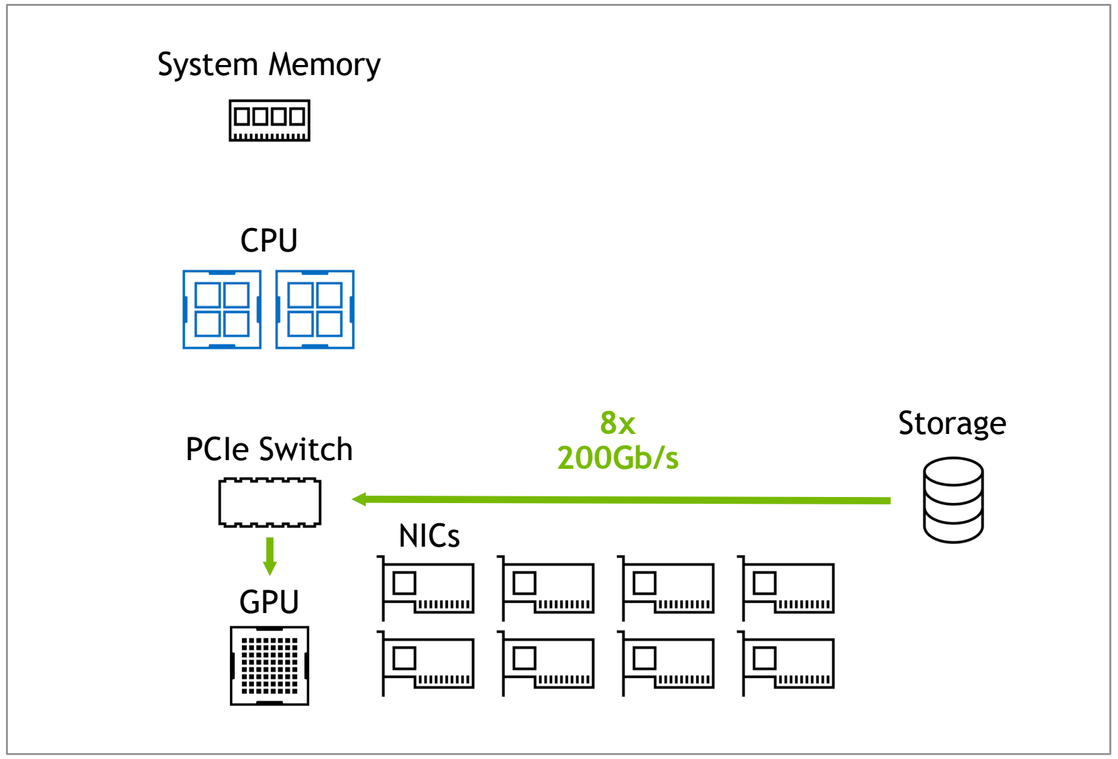
STANDARD GDS CONFIGURATION IN DGX A100



North-south storage configuration adapters 4 and 5 used for GDS

HIGH-THROUGHPUT GDS CONFIGURATION

only recommended for approved LHAs



Converged east-west configuration adapters 0-3, 6-9 used for GDS

IO BENCHMARK ON DGX A100

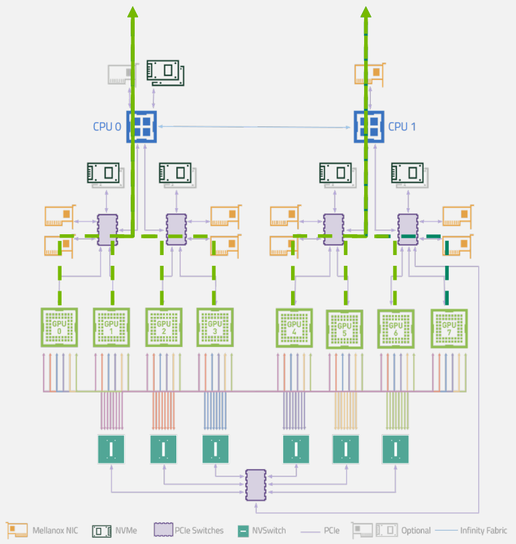
WITH GPUDIRECT STORAGE

1.5x read BW

1/3 CPU utilization

Near line-rate read BW

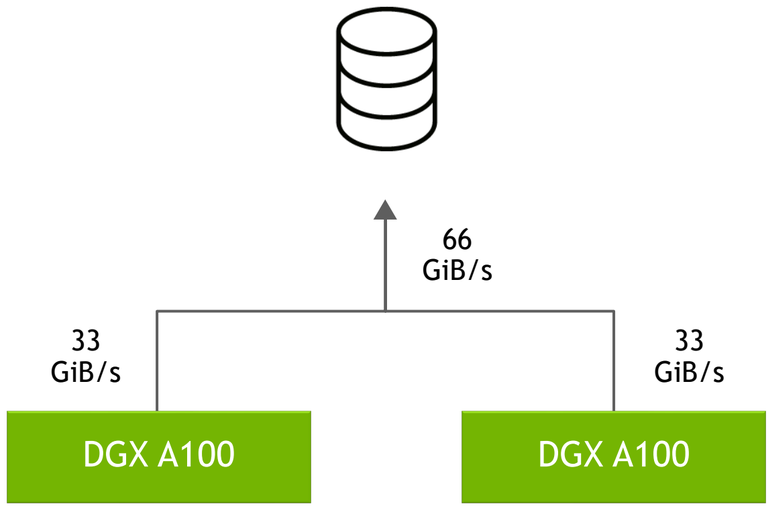
NVIDIA DGX A100 SYSTEM



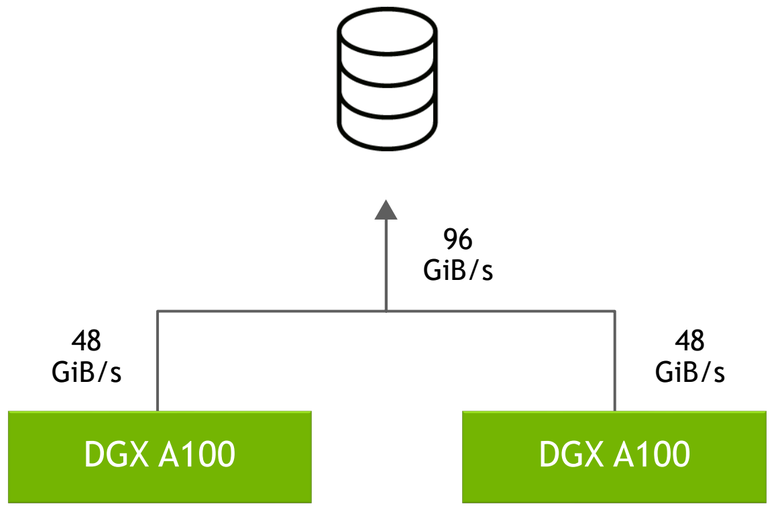
Performance benchmarking done with the gdsio tool using standard GDS configuration
in DGX A100 on N-S network adapters, UB 20.04 MLNX_OFED 5.3 GDS 1.0

GDS ENABLES PEAK IO BANDWITH ON DGX A100

WITHOUT GPUDIRECT STORAGE



WITH GPUDIRECT STORAGE

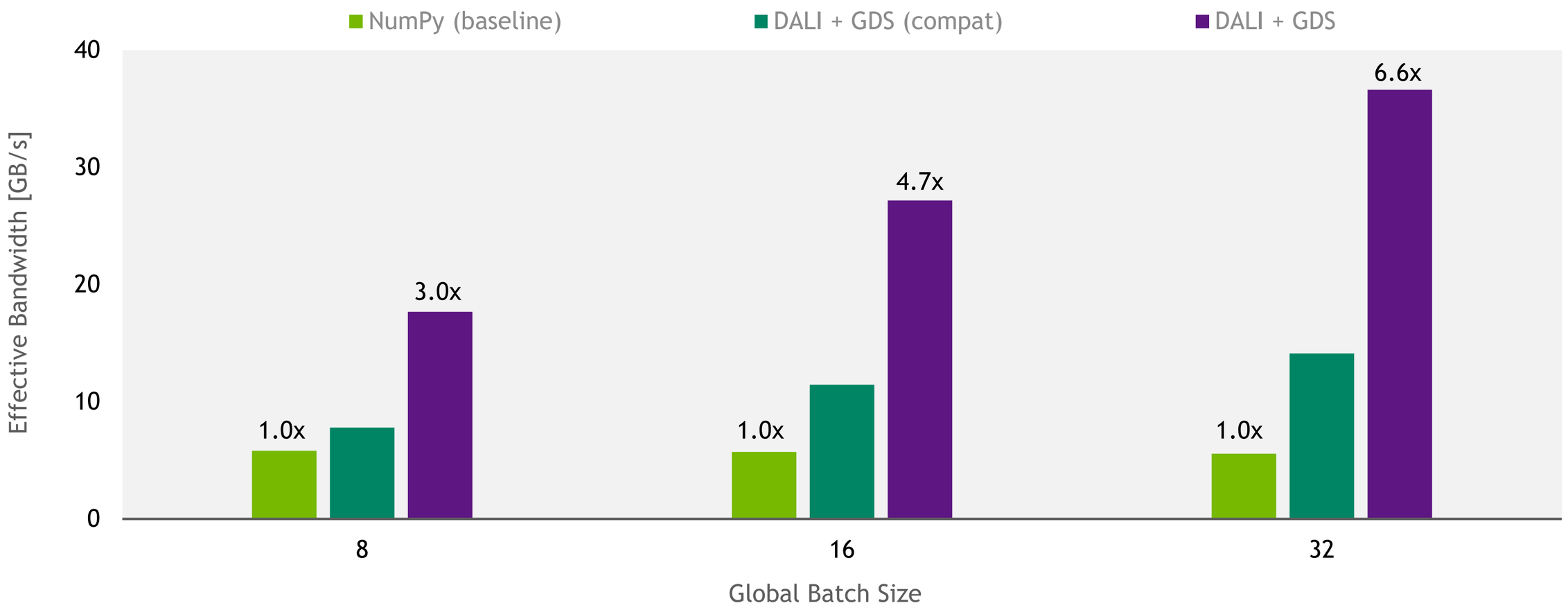


Performance benchmarking done with the gdsio tool using standard GDS configuration
in DGX A100 on N-S network adapters, UB 20.04 MLNX_OFED 5.3 GDS 1.0

CASE STUDIES

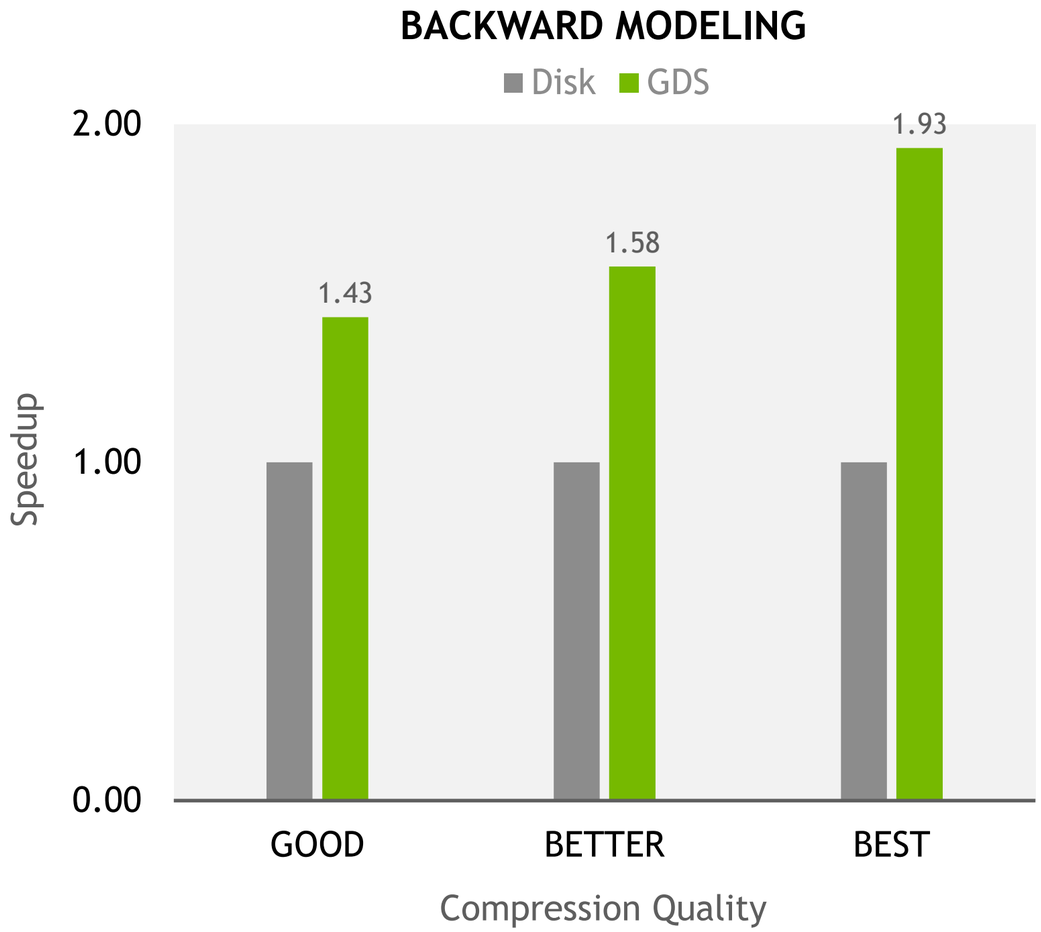
CASE STUDY 1: ACCELERATING DL INFERENCE WITH GDS

GDS-Specific Gains Are up to 2.6X with DALI, 1.9X Without



Performance benchmarking done with the gdsio tool using standard GDS configuration in DGX A100, UB 20.04 MLNX_OFED 5.3 GDS 1.0. Application for batch size >= 32 limited by GPU compute throughput

CASE STUDY 2: ACCELERATING SEISMIC WITH GDS



Testing with 3 different levels of compression quality. Good: snapshots = 1.06TB, Better: snapshots = 1.28TB, Best: snapshots = 1.68TB; Testing RTM using 8 GPUs (V100S) per shot, 512GB host memory, 8 NVMe SSDs

LEARNINGS/BEST PRACTICES

- **Relaxed ordering** on the NICs must be to enabled.
- When running gdsio, the proper **NUMA node**(s) for the GPUs being used need to specified in order to get best performance.
- When using 2 E-W NICs, bind the GPU to NICs' NUMA node respectively for best performance. This way, both NICs can be utilized.
- Need **~96 threads per GPU** to max out a 200 Gbps NIC but the latency might be large.
- **Disable GDS IO path stats** for best performance.
  - echo 0 | sudo tee /sys/module/nvidia_fs/parameters/rw_stats_enabled
- Both client and server need to have the same "peer_credits", "peer_credits_hiw", "concurrent_sends" settings. Both need to increase "lnet_transaction_timeout" to 100. See JIRA.
- Contain accesses to the **same socket** or even the **same PCIe tree**.
- **Avoid writes across CPU Root Ports**.

GPUDIRECT STORAGE PARTNERS

GA

IN DEVELOPMENT

GPUDirect FAMILY WITHIN MAGNUM IO

A natural progression

GPUDirect peer to peer – enables access between GPU peers on PCIe, NVLink

- Used by MPI, UCL, NCCL, NVSHMEM

GPUDirect RDMA – enables direct DMA data path from NIC to GPU memory – no CPU

- Used by MPI, UCL, NCCL, NVSHMEM, GPUDirect Storage

GPUDirect Storage – enables direct DMA data path from storage to GPU memory

- Used by applications, frameworks, and readers

MAGNUM IO TECHNOLOGIES

Historical and New Offerings in Data Movement, Access, and Management

NVIDIA Innovations | Peak Performance | RDMA Everywhere | Flexible Abstraction

NCCL

NVSHMEM

ASAP^2

GDR, GDRCopy

HPC-X for MPI, UCX

MOFED

NETWORK IO

GPUDirect Storage

SNAP

STORAGE IO

SHARP

HW Tag Matching

IN-LINE COMPUTE

Cluster Debug/Profile

Ethernet NetQ, WJH

UFM

NMS

IO MANAGEMENT

CALL TO ACTION

EXPLORE:
https://developer.nvidia.com/gpudirect-storage

GPUDIRECT DEVELOPER BLOG:
https://developer.nvidia.com/blog/gpudirect-storage/

NVIDIA MAGNUM IO ON GITHUB:
https://github.com/NVIDIA/MagnumIO

GDS Engineering

Sandeep Joshi, Aniket Borkar, Rebanta Mitra, Sourab Gupta, Zhen Zeng, Vahid Noormofidi

GDS Product

Maitree Kanungo

GDS Architecture

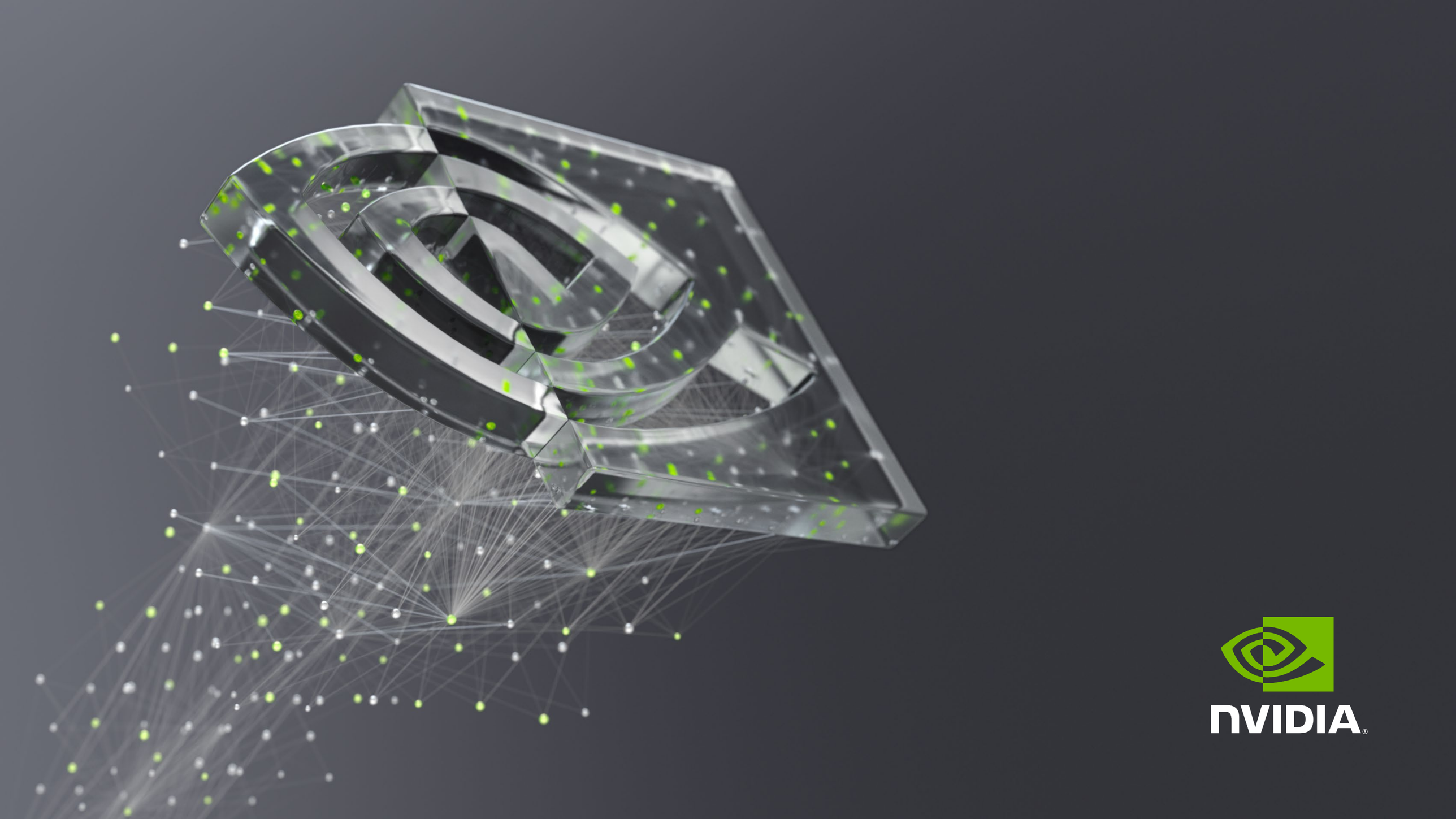
CJ Newburn, Kiran Kumar Modukuri

Ecosystem

Barton Fiske, Michal Shoham, Rob Davis

Management

Ira Nanda


NVIDIA