Spectrum Scale Expert Talks

Episode 18:

NVIDIA GPU Direct Storage with IBM Spectrum Scale

Show notes:
www.spectrumscaleug.org/experttalks

Join our conversation:
www.spectrumscaleug.org/join

SSUG::Digital

Welcome to digital events!

Show notes:
www.spectrumscaleug.org/experttalks

Join our conversation:
www.spectrumscaleug.org/join

About the user group

- Independent, work with IBM to develop events
- Not a replacement for PMR!
- Email and Slack community
- https://www.spectrumscaleug.org/join

We are ...

Current User Group Leads

- Paul Tomlinson (UK)
- Kristy Kallback-Rose (USA)
- Bob Oesterlin (USA)

Former User Group Leads

- Simon Thompson (UK)
- Bill Anderson (USA)
- Chris Schlipalius (Australia)

IBM CHAMPION

#SSUG

Check https://www.spectrumscaleug.org/experttalks
for charts, show notes and upcoming talks

- Past talks:
  - 001: What is new in Spectrum Scale 5.0.5?
  - 002: Best practices for building a stretched cluster
  - 003: Strategy update
  - 004: Update on performance enhancements in Spectrum Scale (file create, MMAP, direct IO, ESS 5000)
  - 005: Update on functional enhancements in Spectrum Scale (inode management, vCPU scaling, NUMA considerations)
  - 006: Persistent Storage for Kubernetes and OpenShift environments
  - 007: Manage the lifecycle of your files using the policy engine
  - 008: Multi-node scaling of AI workloads using Nvidia DGX, OpenShift and Spectrum Scale
  - 009: Continental: Deep Thought – An AI Project for Autonomous Driving Development
  - 010: Data Accelerator for Analytics and AI (DAAA)
  - 011: What is new in Spectrum Scale 5.1.0?
  - 012: Lenovo - Spectrum Scale and NVMe Storage
  - 013:Event driven data management and security using Spectrum Scale Clustered Watch Folder and File Audit Logging
  - 014: What is new in Spectrum Scale 5.1.1?
  - 015: IBM Spectrum Scale Container Native Storage Access
  - 016: What is new in Spectrum Scale 5.1.2?
  - 017: Multiple Connections over TCP (MCOT)
- This talk
  - 018: NVIDIA GPU Direct Storage with IBM Spectrum Scale

Speakers

Kiran Modukuri
Principal software engineer
NVIDIA DGX Platform software

Ingo Meents
IT Architect
IBM Spectrum Scale Development

User Group Host: Kristy Kallback-Rose

GPUDirect Storage

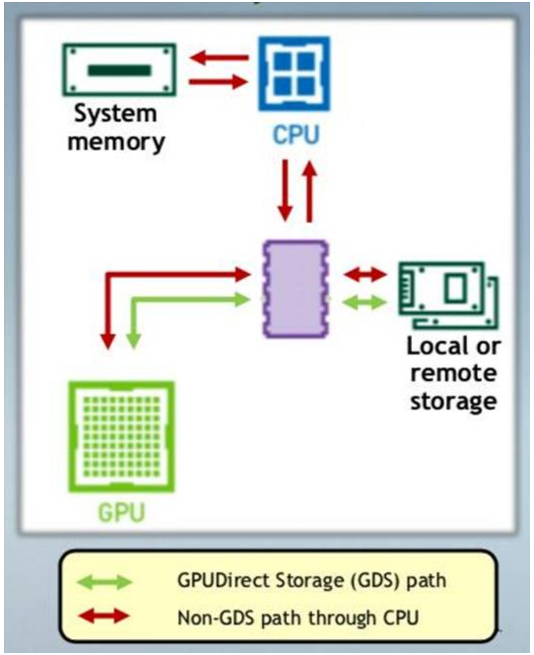
1/19/2022

GPUDIRECT® STORAGE (GDS)

Accelerating data movement between GPUs and storage

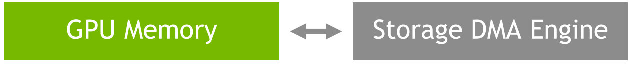
- Skips CPU bounce buffer via DMA
- Works for local or remote storage, with/without PCIe switch
- Accessed via new CUDA cuFile APIs on CPU
- No special HW

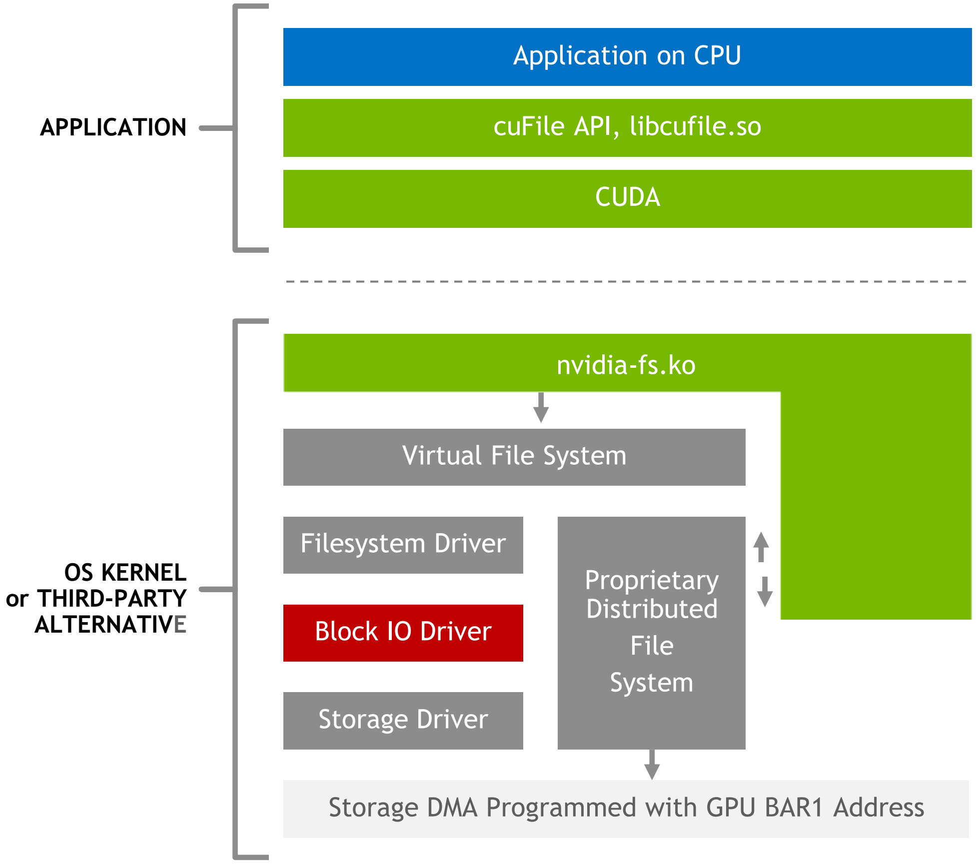
- Advantages
  - Higher peak bandwidth, especially with switch
  - Lower latency by avoiding extra copies and dynamic routing that optimizes path, buffers, mechanisms
    - Most relevant for smaller xfers, even without switch
  - Less jitter than fault-based methods
  - Greater generality, e.g. alignment



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





USAGE EXAMPLE

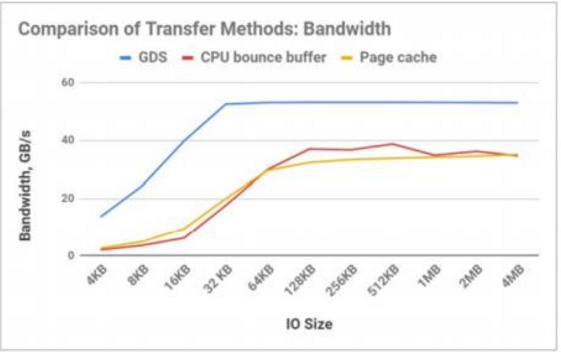
Read from local or remote storage directly into the GPU memory

```
status = cuFileDriverOpen();                              // initialize

fd = open(TESTFILE, O_RDONLY|O_DIRECT, 0, true);          // 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 = cudaMalloc(&devPtr, size);                  // user allocates memory
status = cuFileBufRegister(devPtr, size);                 // initialize and register the buffer

ret = cuFileRead(cf_handle, devPtr, size, 0, 0);          // ~pread: file handle, GPU base address, size,
                                                          //  offset in file, offset in GPU buffer
status = cuFileBufDeregister(devPtr);                     // Cleanup
cuFileHandleDeregister(cf_handle);
close(fd);
cuFileDriverClose();
/* Launch Cuda Kernels */
```

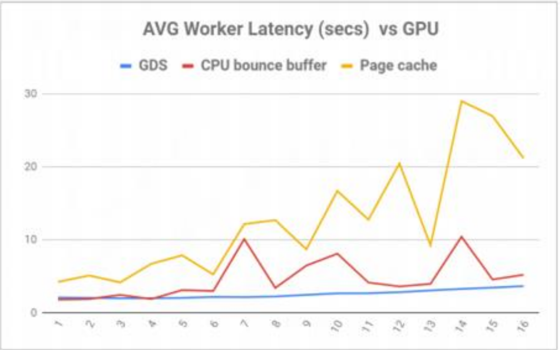
Note: please follow this link for complete example cuFile Sample

GPUDIRECT STORAGE - BENEFITS

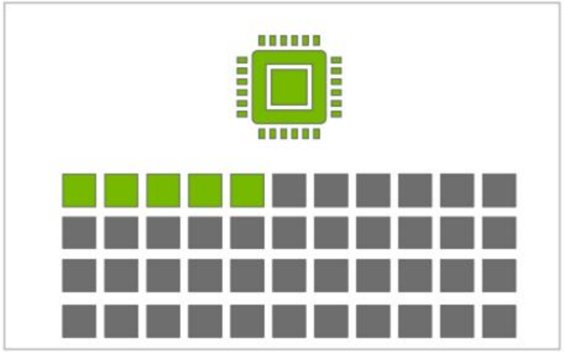
Higher Bandwidth, Lower Latency, Lesser CPU utilization



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

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
  - Need ib_verbs support for Dynamically connected transport.

nvidia-fs.ko driver

- Provides kernel interface for GPU buffer BAR1 lifecycle management and IO operations for cuFile library.
- Interfaces with Linux VFS layer for IO operations to all supported POSIX filesystems.
- Registers with DMA /RDMA ready network filesystems, block devices and storage drivers.
- Registers as IB memory peer client to support memory registration from userspace using ibv_register_mr (using nvidia_peermem.ko starting from 11.5U1)
- Provides DMA callbacks for GPU virtual address to BAR1 addresses.

https://github.com/NVIDIA/gds-nvidia-fs

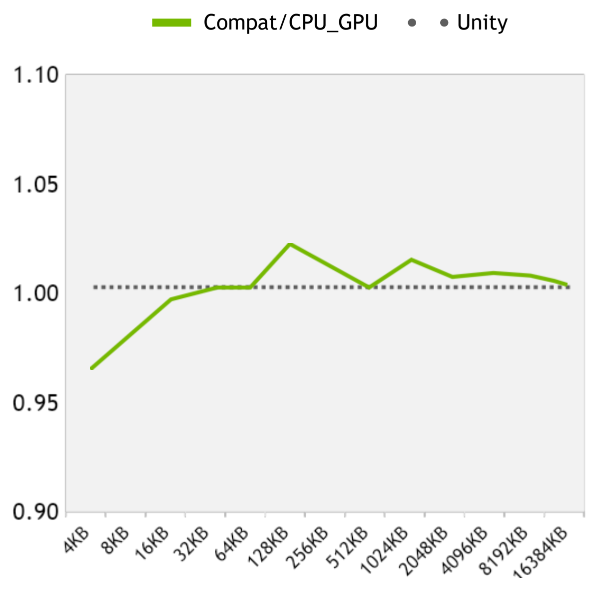
Dynamic Routing

- Choose the optimal path between the storage and the GPU.
- Discover additional hardware support between the storage and the GPU. Eg. NVlinks.
- Understand the PCIe hierarchy and path latencies between GPU and the storage.
- Route the IO requests by segmenting them based on available GPU BAR1 memory
- Route the IO requests based on the type of memory allocated by use cudaMalloc vs cudaMallocManaged memory
- Routing based on IO offset and buffer offset alignments.
- Route the IO requests based on the IO size thresholds for reduced latencies.

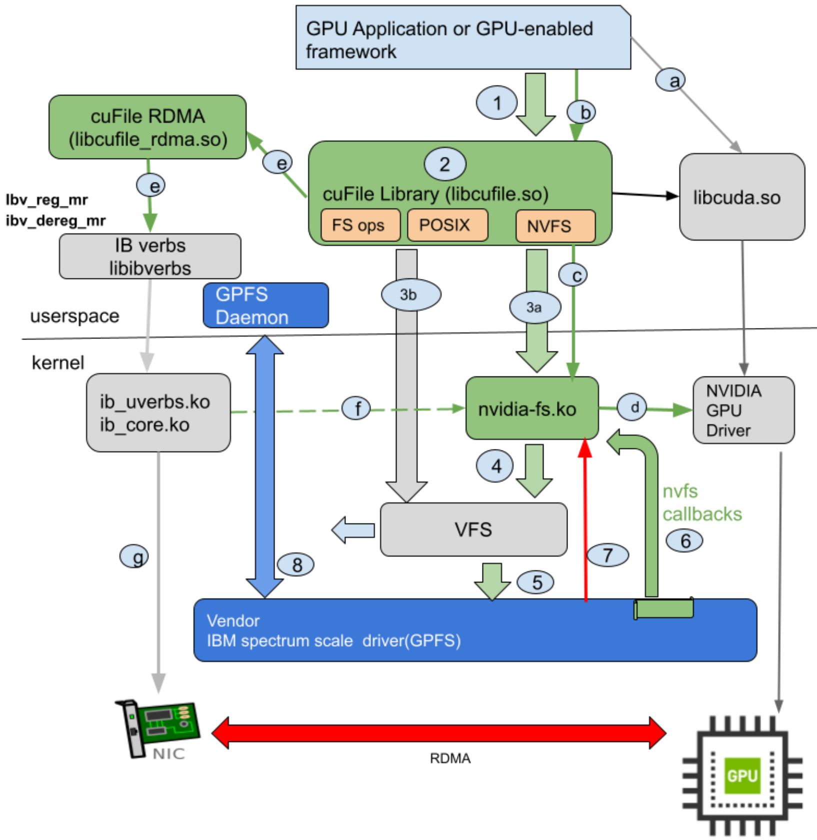
COMPATIBILITY MODE

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



Detailed Architecture



Nvidia-fs callback interface

NVIDIA ref implementation

GPFS implementation

No change

Alloc and Map path

a) cuMemAlloc
b) cuFileBufRegister
c) Call ioctl to map the buffer
d) Get VA to PA mapping for GPU memory and setup metadata to track mapping.
e) Register device memory with ib_verbs using Peer Direct support
f) Get DMA mapping from nvidia-fs.ko
g) Setup VA-PA mappings in NIC IOMMU and get RDMA info for memory region (Remote Key).

IO Path

1) App calls `cuFileRead or cuFileWrite` operation.
2) Validate and prepare buffers.
3) determine the IO path
    a) If GPFS filesystem and valid driver, call nvidia-fs driver IOCTLs.
    b) Otherwise route through POSIX API.
4) nvidia-fs validates, prepares IO and issue request to VFS
5) VFS calls vendor drivers
6) GPFS drivers call appropriate nvidia-fs.ko DMA callbacks for getting RDMA information
7) If IO is not O_DIRECT ready return -EOPNOTSUPP and retry step 3b.
8) GPFS driver communicates to GPFS daemon. GPFS Driver issues IO completion callback for async IO submissions.

HW & SW support

Supported HW

- GPU
  - Data Center and Quadro (desktop) cards with compute capability > 6
- NIC
  - ConnectX-5 and ConnectX-6
- CPU
  - x86-64
- NVMe
  - Version 1.1 or greater
- Systems
  - DGX, EGX
- Computational Storage
  - ScaleFlux CSD

SW Requirement

- GPU driver 418.x onwards
- CUDA version 11.4 and above
- MOFED
  - Preferred 5.4 or 5.3 (DKMS support)
- DGX BaseOS 5.0.2 or later
- Linux Kernel
  - 4.15 or later
  - NO kernel > 5.4
- Linux Distros
  - GDS mode
    - UB 18.04 (4.15.x),20.04 (5.4.x)
    - RHEL (8.4)
  - Compat mode only
    - Rhel 7.0, Debian 10, CentOS 8.3, SLES 15.2
    - OpenSUSE 15.2

Current GDS Restrictions

- Design supported for Linux only.
- Not supported on virtual machines yet.
- Not supported on power9 or ARM based architectures
- Not supported on file systems which cannot perform IO using strict O_DIRECT semantics like compression, checksum, inline IO, buffering to page cache.
- Not supported on non Mellanox adapters.
- IOMMU support is limited to DGX platforms.

IBM Spectrum Scale Restrictions

- The following cases are handled in compatibility mode:
  - Spectrum Scale does not support GDS write (cuFileWrite).
  - Spectrum Scale does not support GDS on files less than 4096 bytes in length.
  - Spectrum Scale does not support GDS on sparse files or files with pre-allocated storage (for example, fallocate(), gpfs_prealloc(), etc)
  - Spectrum Scale does not support GDS on files that are encrypted.
  - Spectrum Scale does not support GDS on memory-mapped files.
  - Spectrum Scale does not support GDS on files that are compressed or marked for deferred compression
  - Spectrum Scale does not support GDS if the mmchconfig option "disableDIO" is set to "true" (the default value of "disableDIO" is "false")

ECOSYSTEM

Frameworks, Readers

Partner Ecosystem

FRAMEWORK AND APPLICATION DEVELOPMENT

Broadening the ecosystem

Frameworks, apps

- Visualization, e.g. IndeX
- Health, e.g. cuCIM in CLARA
- Data analytics, e.g. RAPIDS,cuCIM, SPARK, nvTABULAR
- HPC, e.g. molecular modeling
- DL, e.g. PyTorch,TensorFlow, MxNet (Via DALI)
- Databases, e.g. HeteroDB
- Benchmarking Tools: gdsio, fio, ElBencho

Readers

- RAPIDS cuDF, DALI
- HDF5 — Serial, LBNL (repo); passed HDF5 regression suite, IOR layered on HDF5
- OMPIO — U Houston; early PHDF5 functionality
- MPI-IO — engaging with ANL, others
- ADIOS ORNL
- Zarr — prototyped with Pangeo community

Key
NVIDIA functional
NVIDIA WIP/planned
Functional
WIP/planned

USE CASES

Real Time Video Streaming
Data visualization

Inference

Training

Seismic

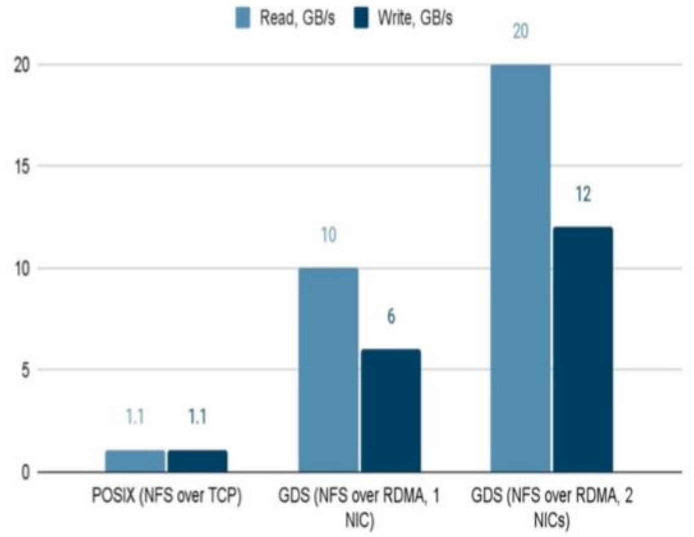
GDS Benchmarking

VOLUMETRIC VIDEO PROCESSING

Verizon

GPU Direct Storage

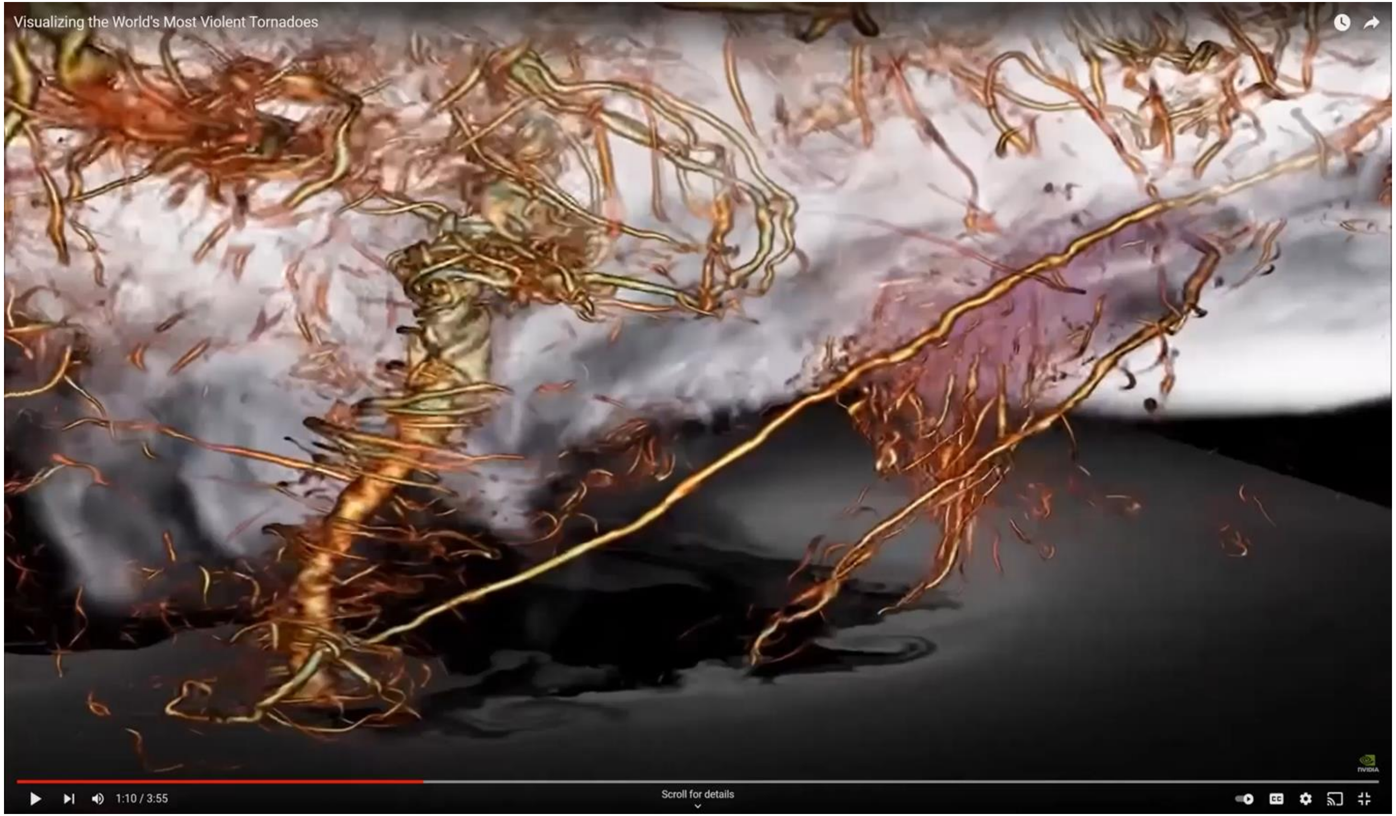
- Enables shortest path from GPU to storage device
- Accelerated Read/Write
    - 10x speed compared to NFS over TCP
- Read and Write as GPU RDMA memory accelerates I/O speed
- Enables both write to high capacity storage and process of data at the same time
- Higher read speeds enable offline workflows to be near realtime



GTC Keynote Demo

TORNADO VISUALIZATION

Fast Data Streaming with GPUDirect Storage



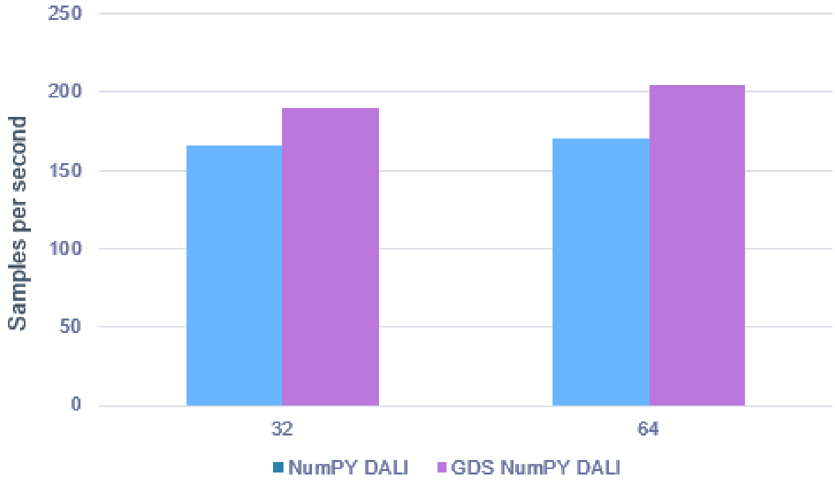
10 fps achieved on DGX A100 with 8 Samsung P1733 NVMe drives

47 GB/s IO bandwidth

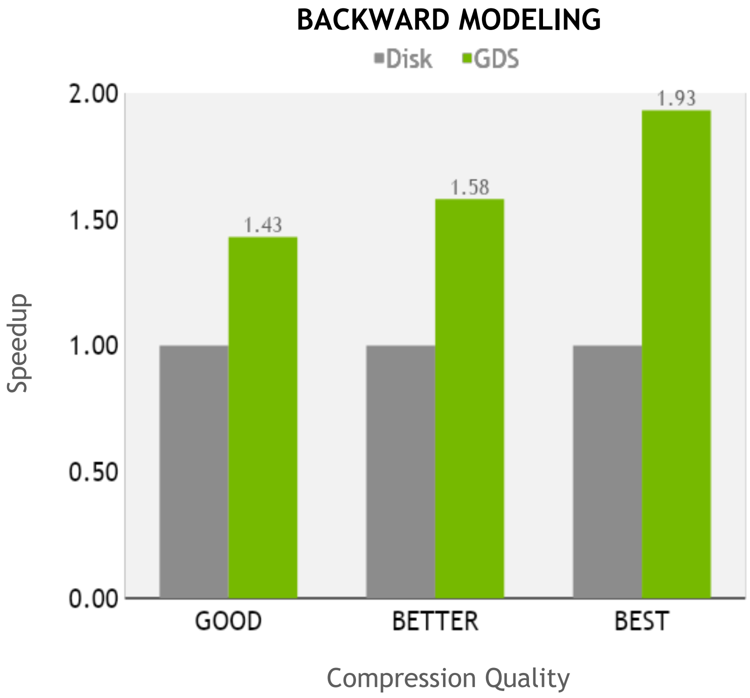
See also: SC demo on Tornado visualization

TRAINING ON PYTORCH+DALI+GDS

~1.17x gain for a single node



HPC Benchmark - Reverse Time Migration



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

Accelerating IO for GPUs with IBM Spectrum Scale and GPUDirect Storage

Jan 19, 2022

Ingo Meents
IT Architect
Spectrum Scale Development
IBM Systems Group

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

IBM reserves the right to change product specifications and offerings at any time without notice. This publication could include technical inaccuracies or typographical errors. References herein to IBM products and services do not imply that IBM intends to make them available in all countries.

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Agenda

- GDS in Spectrum Scale
  - GDS READ data path
  - How to use (HW & SW Prerequisites)
  - Use of cuFileRead
  - Performance numbers
  - Writes
  - Outlook
  - References

Why GPUDirect Storage?

Short latencies & ***High throughput***

for GPU accelerated HPC and AI applications.

Sample use case:
Weather Forecasting
deepCAM inference

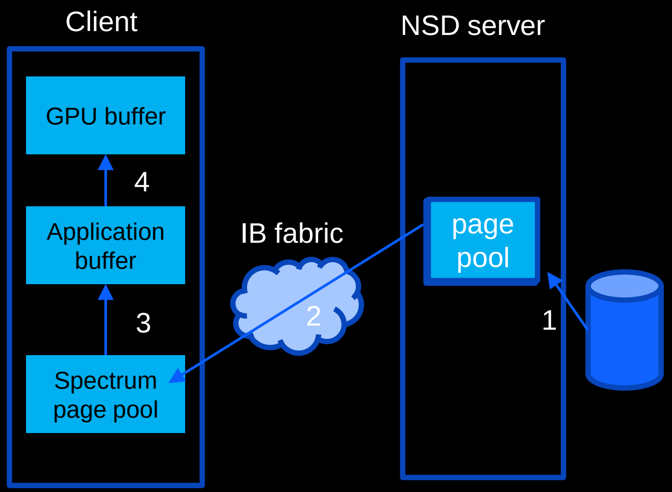
Predicting extreme weather faster

Sample use case:
Oil and gas exploration

4D Seismic imaging for reservoir mapping

GPUDirect Storage (GDS) for Spectrum Scale

Data path for a **READ** into a GPU buffer



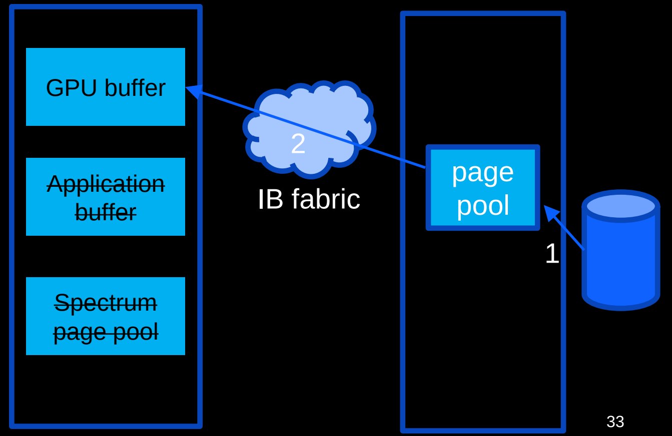
Storage to GPU buffer **without** GDS:

4 data transfers on path from storage media to application GPU buffer

Storage to GPU buffer **with** GDS:

Two data transfers in path are eliminated.
→ ***increased throughput, reduced latency***

Client CPU copy overhead reduced.
→ ***more CPU cycles for client application***



What do I need to use GPUDirect Storage with Spectrum Scale?

https://www.ibm.com/docs/en/STXKQY/gpfsclustersfaq.html#gds

Hardware

- X86 client with a GPU supporting GDS
- Storage Server (NSD server, ESS)
- IB Fabric (Mellanox CX-{4,5,6} and switch)

Spectrum Scale

- 5.1.2 or later

MOFED

- Mellanox OFED stack

CUDA

- CUDA 11.4 or later
- Please look at FAQ for issues and recommendations
- CUDA C/C++ program
- Nvidia DALI (data loading library)

How to exploit – cuFileRead – CUDA Application

```
// open driver
status = cuFileDriverOpen();


// register filehandle with CUDA
cf_descr.handle.fd = fd;   POSIX file handle
cf_descr.type = CU_FILE_HANDLE_TYPE_OPAQUE_FD;
status = cuFileHandleRegister(&cf_handle, &cf_descr);


// reading data from file into device memory
ret = cuFileRead(cf_handle, devPtr, size, 0, 0);


// deregister the handle from cuFile
(void) cuFileHandleDeregister(cf_handle);
```

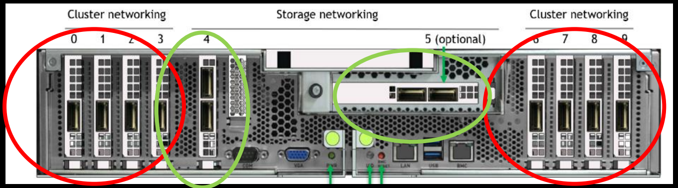
Triggers registration with GPFS

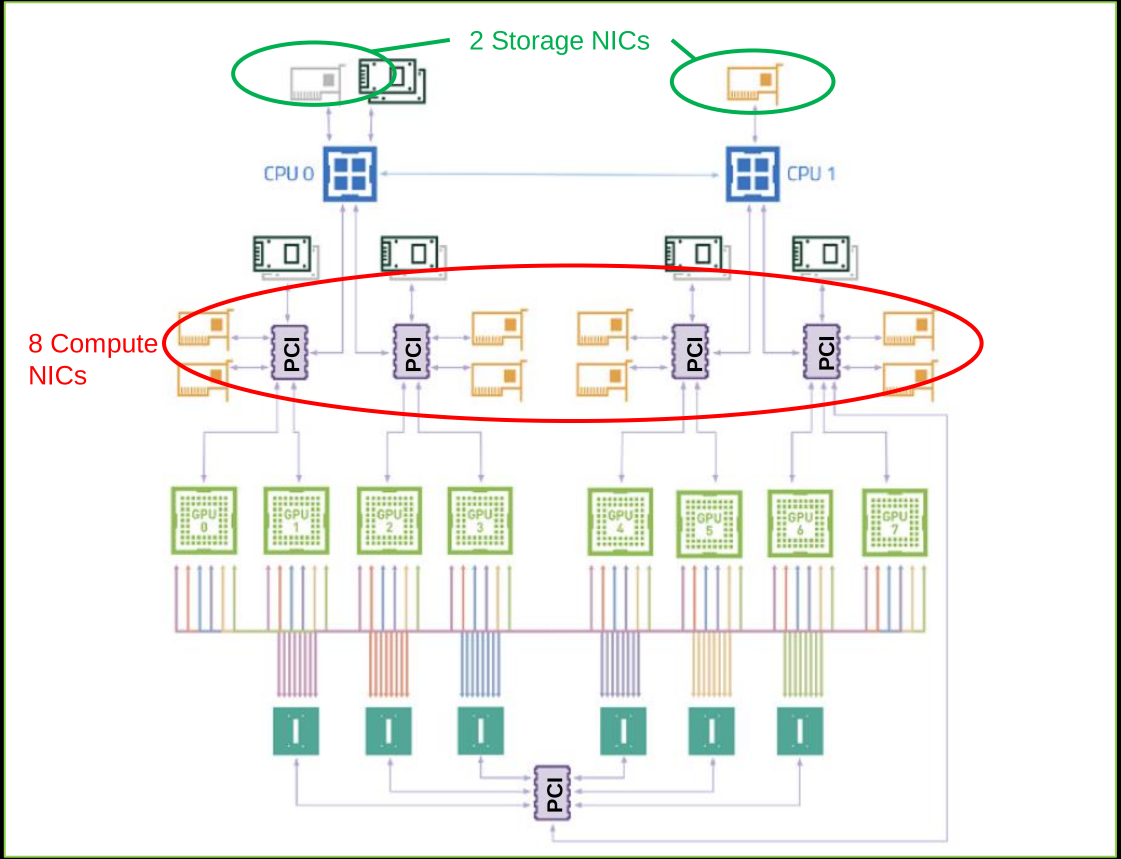
Registers file handle with CUDA for use in cuFileRead

Do GDS IO

Triggers de-registration with GPFS

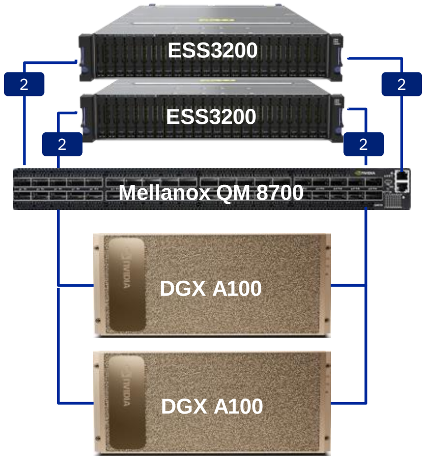
Nvidia DGX A100





Pictures from DGX A100 User guide, https://docs.nvidia.com/dgx/

GDS Read Throughput – Linear scaling



1 ESS 3200: 4 x HDR links
→ ~100 GB/sec max

2 DGX A100: 4 x HDR links
→ ~100 GB/sec max

Use of storage links

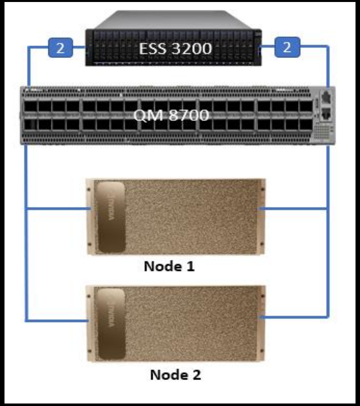
		Scenario 1	Scenario 2
		2 x ESS 3200 1 x DGX A100	2 x ESS 3200 2 x DGX A100
Throughput	Direct IO + cudaMemCopy	22 GB/s	45 GB/s
	GDS	49 GB/s	86 GB/s

Streaming Benchmark:

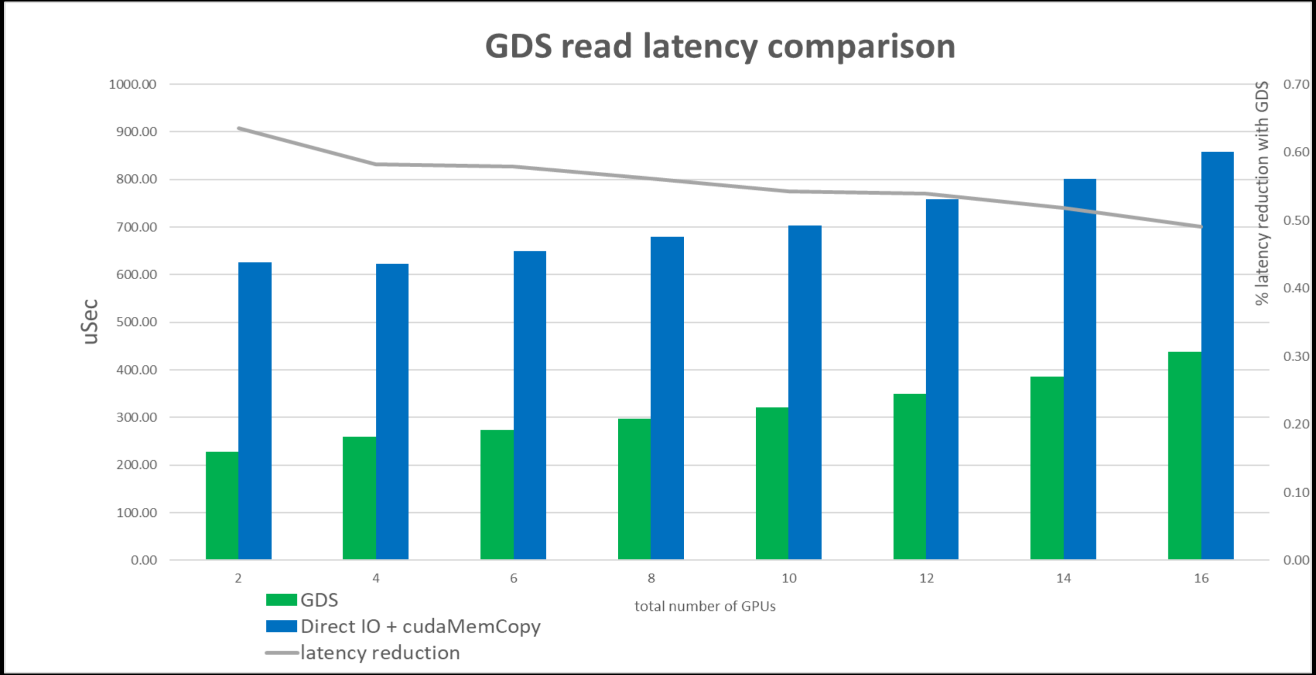
- NVIDIA "gdsio" utility
- 8 GPUs per DGX A100
- 2 or 4 threads per GPU
- 1M I/O size
- Data in GNR cache on ESS server

Typical throughput improvement for DGX A100 with GDS is approx. 2x when the storage and network support the throughput.

GDS Read Latency



1 ESS 3200: 4 x HDR links
2 DGX A100: 4 x HDR links
(storage links)



Benchmark: NVIDIA 'gdsio' benchmark with 1M I/O size and 2 threads per GPU

Typical latency reduction with GPU Direct Storage is 50%.

GDS Read Performance

Experimental config using DGX-A100 compute NICs (*)

Maximum theoretical thruput:
ESS 3200: 4 x HDR = 100 GB/s max
DGX-A100 compute NICs: 8 x HDR = 200 GB/s max

	Scenario 3 (Compute NICs)
	2 x ESS 3200 2 x DGX A100
Aggregate GDS throughput	196 GB/s

> 95% of max fabric bandwidth for 2 x ESS 3200

Benchmark details:
- NVIDIA "gdsio" utility
- 8 GPUs per DGX A100
- 4 threads per GPU
- 1M I/O size
- Data in GNR cache on ESS servers

(*) *Performance numbers shown here with NVIDIA GPUDirect Storage on NVIDIA DGX A100 slots 0-3 and 6-9 ("compute NICs") are not the officially supported NVIDIA DGX A100 network configuration and are for experimental use only. Sharing the same network adapters for both compute and storage may impact the performance of any benchmarks previously published by NVIDIA on DGX A100 systems.*

GDS Write

Spectrum Scale 5.1.2 PTF 1
supports cuFileWrite() in compatibility mode

- user application calls cuFileWrite() API
- NVIDIA GDS library transparently

1. calls cudaMemCopy() to stage the data from the application GPU buffer
   to a temporary system memory buffer internal to the NVIDIA GDS lib
2. issues a Direct IO write to GPFS, sourcing the data from the temporary system memory buffer

Planned Future Enhancements

- Support for GDS over RoCE
- GDS Write acceleration
- Support for NDR
- Performance Improvements

Documentation

Spectrum Scale Knowledge Center:

www.ibm.com/docs/en/spectrum-scale/5.1.2?topic=summary-changes

www.ibm.com/docs/en/spectrum-scale/5.1.2?topic=architecture-gpudirect-storage-support-spectrum-scale

NVIDIA GDS Documentation:

docs.nvidia.com/gpudirect-storage/index.html

developer.nvidia.com/gpudirect-storage

Thank you

Ingo Meents
IT Architect
IBM Research & Development, Germany
—
meents@de.ibm.com

Special thanks to: Felipe Knop, John Divirgilio, Ralph Würthner, Swen Schillig, Willard Davis, Mary Hailu

Trademarks

CUDA, DALI, DGX A100, GPUDirect Storage are trademarks
and/or registered trademarks of NVIDIA Corporation in the U.S. and/or other countries.

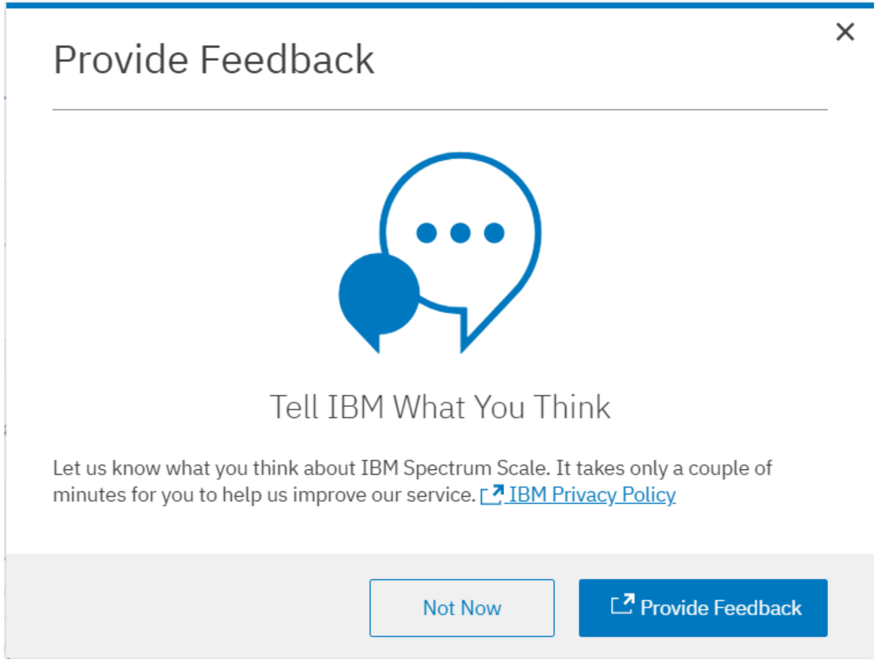
Check https://www.spectrumscaleug.org/experttalks
for charts, show notes and upcoming talks

- Past talks:
  - 001: What is new in Spectrum Scale 5.0.5?
  - 002: Best practices for building a stretched cluster
  - 003: Strategy update
  - 004: Update on performance enhancements in Spectrum Scale (file create, MMAP, direct IO, ESS 5000)
  - 005: Update on functional enhancements in Spectrum Scale (inode management, vCPU scaling, NUMA considerations)
  - 006: Persistent Storage for Kubernetes and OpenShift environments
  - 007: Manage the lifecycle of your files using the policy engine
  - 008: Multi-node scaling of AI workloads using Nvidia DGX, OpenShift and Spectrum Scale
  - 009: Continental: Deep Thought – An AI Project for Autonomous Driving Development
  - 010: Data Accelerator for Analytics and AI (DAAA)
  - 011: What is new in Spectrum Scale 5.1.0?
  - 012: Lenovo - Spectrum Scale and NVMe Storage
  - 013:Event driven data management and security using Spectrum Scale Clustered Watch Folder and File Audit Logging
  - 014: What is new in Spectrum Scale 5.1.1?
  - 015: IBM Spectrum Scale Container Native Storage Access
  - 016: What is new in Spectrum Scale 5.1.2?
  - 017: Multiple Connections over TCP (MCOT)
- This talk
  - 018: NVIDIA GPU Direct Storage with IBM Spectrum Scale

Thank you!

Please help us to improve Spectrum Scale with your feedback

- If you get a survey in email or a popup from the GUI, please respond
- We read every single reply



Spectrum Scale User Group

The Spectrum Scale (GPFS) User Group is free to join and open to all using, interested in using or integrating IBM Spectrum Scale.

The format of the group is as a web community with events held during the year, hosted by our members or by IBM.

See our web page for upcoming events and presentations of past events. Join our conversation via mail and Slack.

www.spectrumscaleug.org