## Lustre for Grace Hopper: Current Status Report

Sohei Koyama Shuichi Ihara, DataDirect Networks, Japan





#### **NVIDIA GH200 Grace Hopper Superchip is out**

How to achieve high performance I/O to Lustre from Grace Hopper?

CPU and GPU share a single page table.

The placement of memory is determined by

- 1. Which touches first: CPU or GPU?
- 2. Memory bind (numaif.h's mbind())
- 3. Memory usage, etc.



© DDN 2024

https://developer.nvidia.com/blog/nvidia-grace-hopper-superchip-architecture-in-depth/



#### **#1: Impact of page size**

Grace Hopper supports 64k page size.

We evaluate the performance of Buffered I/O and Direct I/O with 4k / 64k page sizes

by IOR [https://github.com/hpc/ior] benchmark.

Environment DDN AI400X2 NVIDIA Mellanox ConnectX-6 (200Gbps, InfiniBand) Lustre 2.14.0 ddn168 Stripe count is 8; stripe size is 1MiB

© DDN 2024



### #2: The cuFile API and memory allocation methods

We examine whether GPUDirect Storage is truly utilized and whether it delivers optimal performance with each memory allocation method.





#### Single process IOR w/ 4k or 64k page size, Buffered or Direct I/O

#### Read



ior -w -b 256g -t \$XFER\_SIZE -k -o "\$OUTPUT\_FILE.\$XFER\_SIZE" -F



Bandwidth [MiB/s]

#### cuFileRead/cuFileWrite Result cuFileRead **cuFileWrite** GPUDirect Storage enabled ( only with cudaMalloc ! ) Buffered - cudaMalloc 14000 Buffered - malloc on HBM3 Buffered - cudaMalloc 15000 Buffered - malloc on LPDDR5X Buffered - malloc on HBM3 12000 Direct - cudaMalloc Buffered - malloc on LPDDR5X 12500 Direct - malloc on HBM3 Direct - cudaMalloc -----Direct - malloc on HBM3 Direct - malloc on LPDDR5X 10000 Direct - malloc on LPDDR5X 10000 Bandwidth [MiB/s] 8000 7500 6000 5000 4000 2500 2000 0 0 64k 128k 256k 512k 1m 2m 4m 8m 16m 32m 64m 128m 256m 512m 1g 64k 128k 256k 512k 1m 2m 4m 8m 16m 32m 64m 128m 256m 512m 1g Xfer Size Xfer Size



#### Conclusion

- 1. Page size 64k has better I/O performance (especially Buffered I/O)
- 2. posix\_memalign()'ed buffer cannot leverage GPUDirect Storage
- 3. Use Buffered I/O or Direct I/O appropriately

# **THANK YOU**

