Lustre for Grace Hopper: Current Status Report

Sohei Koyama
Shuichi Ihara,
DataDirect Networks, Japan

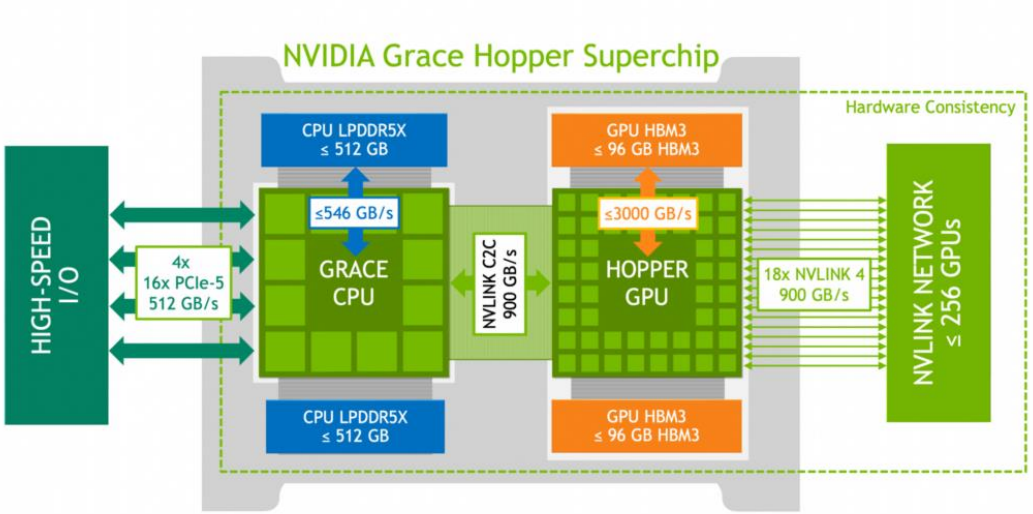
NVIDIA GH200 Grace Hopper Superchip is out

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.



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

#1: Impact of page size

Grace Hopper supports 64k page size.

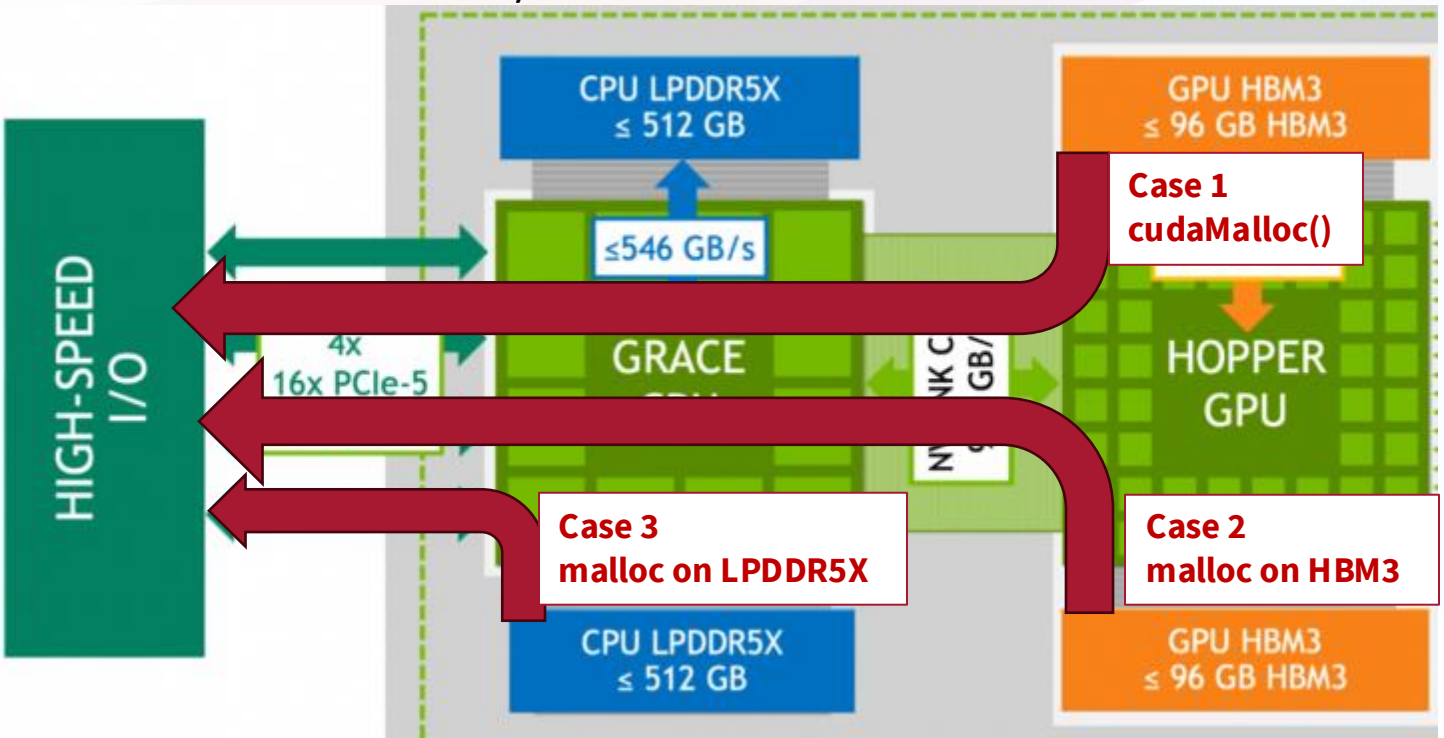
We evaluate the performance of Buffered I/O and Direct I/O with 4k / 64k page sizes by IOR [https://github.com/hpc/ior] benchmark.

Environment

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

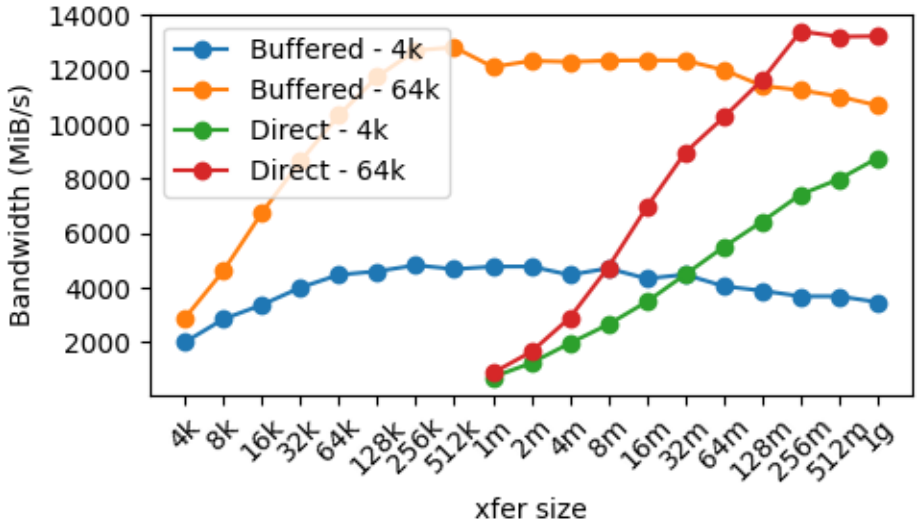
#2: The cuFile API and memory allocation methods

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

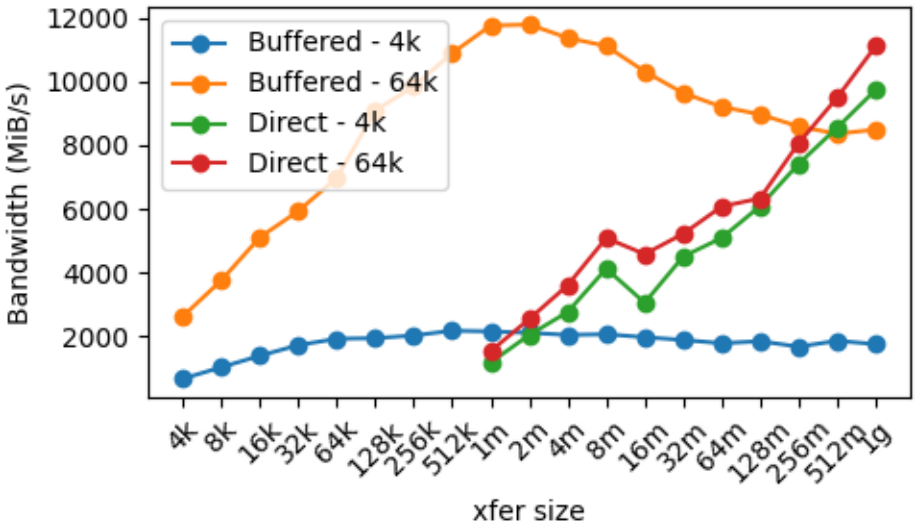


Single process IOR w/ 4k or 64k page size, Buffered or Direct I/O

Read



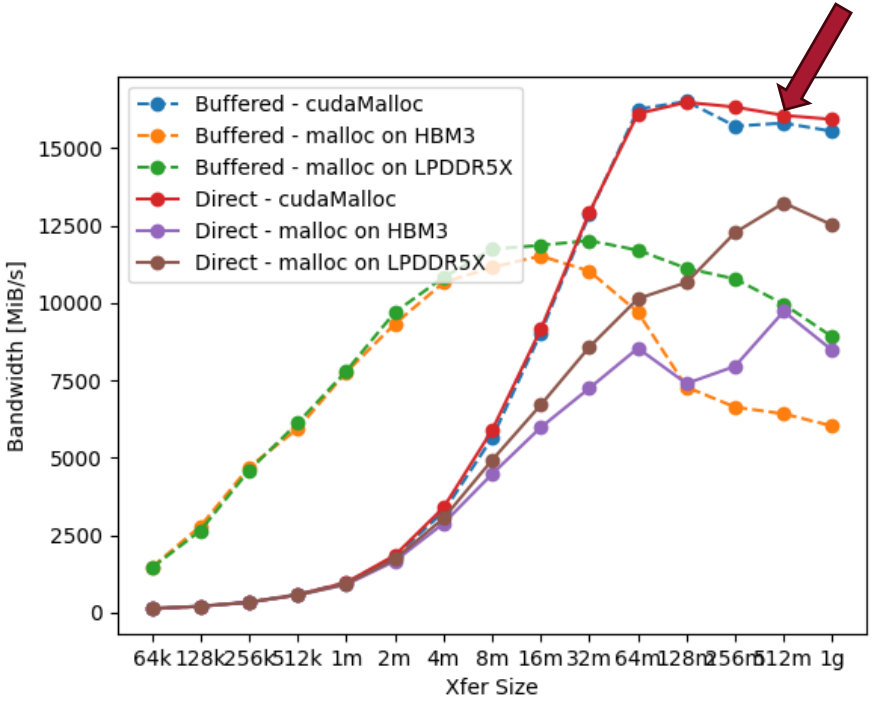
Write




```
ior -w -b 256g -t $XFER_SIZE -k -o "$OUTPUT_FILE.$XFER_SIZE" -F
```

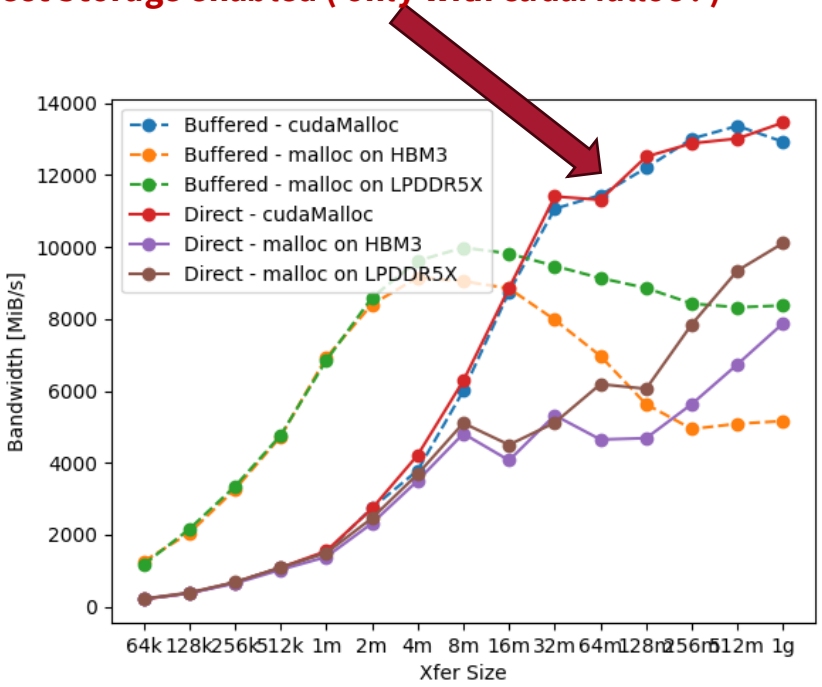
cuFileRead/cuFileWrite Result

cuFileRead



cuFileWrite

GPUDirect Storage enabled (only with cudaMalloc !)



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