I/O acceleration with GPU for I/O-bound applications

Kento Sato[†1, †2], Akira Nukada[†1], Naoya Maruyama[†3, †1, †5] and Satoshi Matsuoka[†1, †4, †5]

†1 Tokyo Institute of Technology, †2 Research Fellow of the Japan Society for the Promotion of Science,
†3 RIKEN Advanced Institute for Computational Science, †4 National Institute of Informatics, †5 JST/CREST

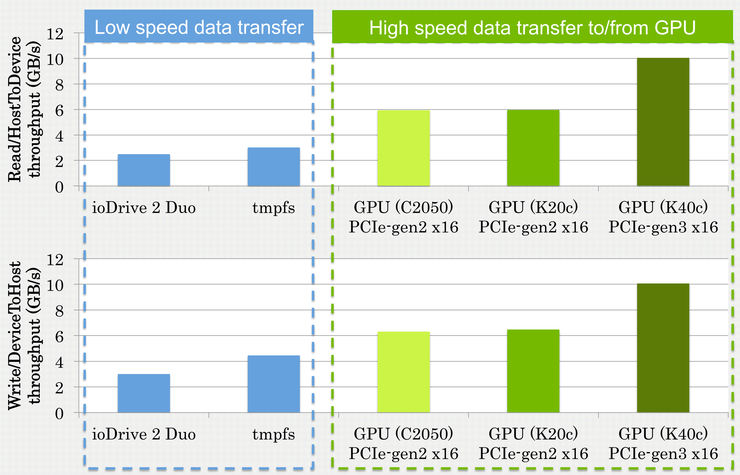
Background

GPU-enabled supercomputers

- Several supercomputers usually have GPUs on each compute node to accelerate computation
- Not all applications can be accelerated by GPUs
- Performance of I/O-bound applications is limited by underlining I/O device performance
- Such I/O-bound applications require more I/O bandwidth rather than computational power
- If we execute such non-GPU applications, the GPUs is not utilized, and we waste the resources

Data transfer throughputs of GPUs

- Data transfer throughputs with POSIX read/write for ioDrive 2 Duo & tmpfs, and with cudaMemcpy for GPUs



ioDrive 2 Duo	Flash devices on ioDrive can become a bottleneck
tmpfs	A file system becomes a bottleneck
GPUs	GPUs can exploit PCIe's bandwidth

- PCIe-attached GPUs can be used as I/O devices for accelerating I/O-bound applications

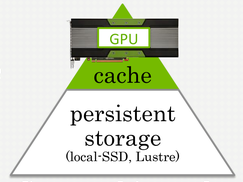
Objective

- Accelerate I/O operations using GPU device memory
- Exploit PCIe's bandwidth even with I/O operations
- Provide well-known I/O interfaces

Methods

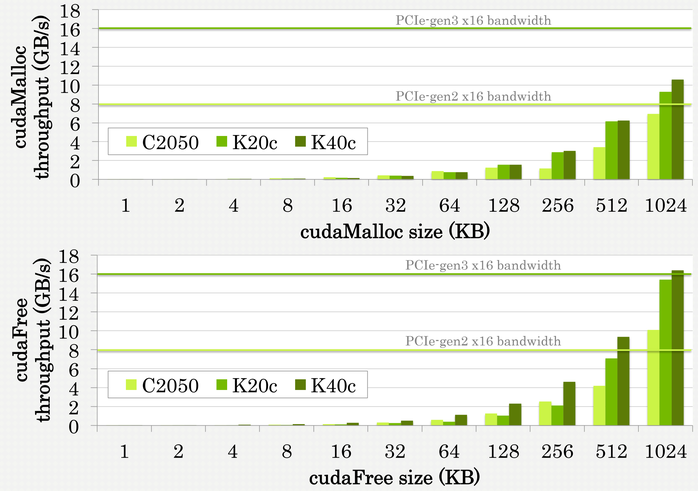
gmfs: GPU-accelerated I/O interface

- gmfs is an I/O interface which uses GPU as buffer cache of persistent storage
- gmfs provides well-known I/O interfaces (open, read, write and close) on top of the hierarchical storage



Storage hierarchy

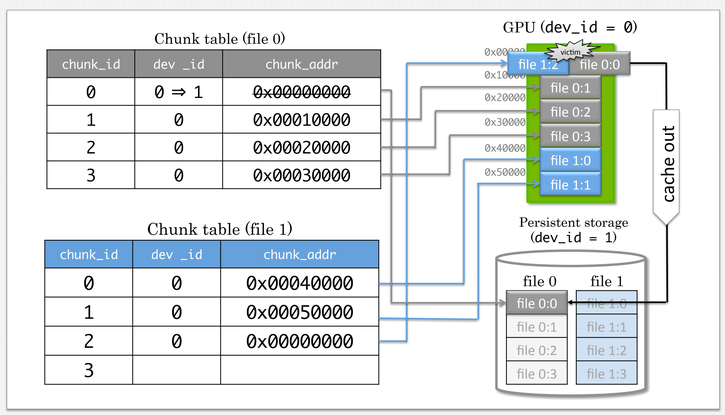
Requirement for self memory management



Challenges	How to exploit PCIe's bandwidth on multiple small writes requiring small cudaMallocs and cudaFrees

gmfs's file management

- First allocate entire GPU memory to gmfs
- gmfs divides the GPU memory into chunks, and chunk tables manage chunk locations (round-robin cache mgmt.)



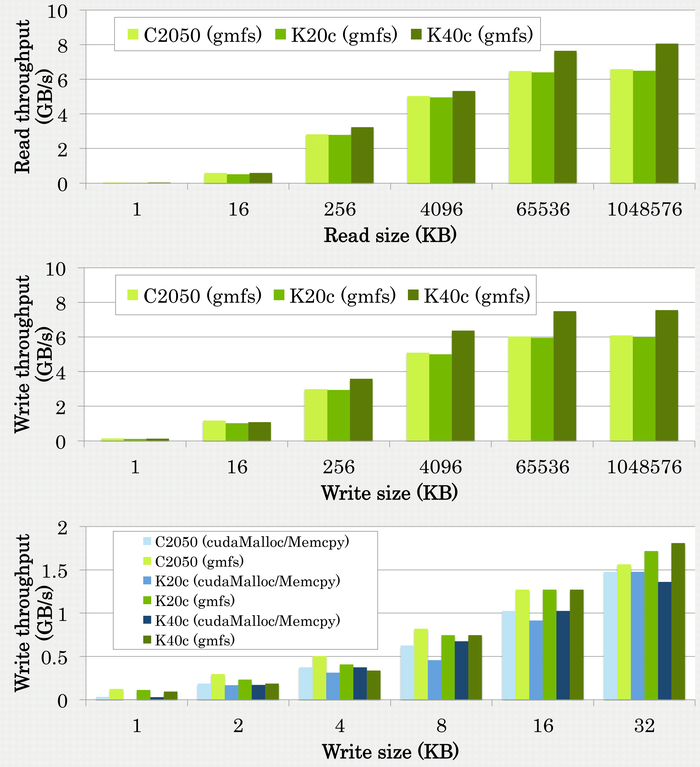
Experiments

Experimental environment

M/B	ASUSTeK P9X79 WS
CPU	Intel(R) Core(TM) i7-3930K CPU @ 3.20GHz
Memory	DDR3 DIMM 1333Hz (16GB)
GPU	NVIDIA GPU (Tesla: C2050, K20c, K40c)
CUDA	5.5
Persistent storage	OCZ VTX3-25SAT3-120G (SSD MLC) (Max Read: up to 550MB/s, Max Write: 500MB/s)

Results

- Sequential Read/Write throughput comparisons



Conclusion

- gmfs can accelerate sequential read/write, and utilize 82% of PCIe-gen2 peak bw, 50% of PCIe-gen3 peak bw
- gmfs can improve small write operations