

Path to GPU-Initiated I/O for Data-Intensive Systems

SAMSUNG

IT UNIVERSITY OF COPENHAGEN

RAD

Karl B. Torp, Samsung

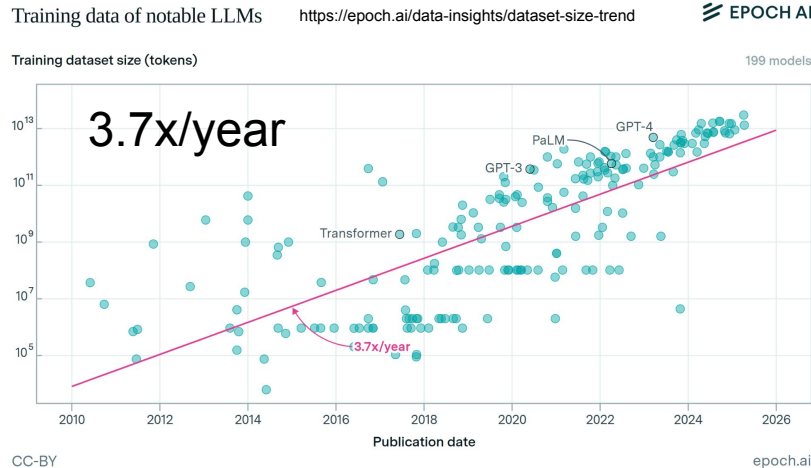
Simon A. F. Lund, Samsung

Pinar Tözün, IT University of Copenhagen

Why GPU-initiated Storage?

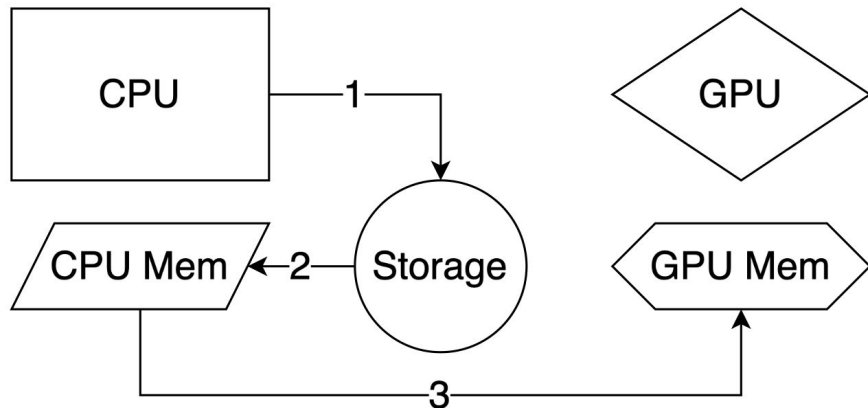
NVIDIA GPU	Memory	Year
V100	16 / 32 GB	2017
A100	40 / 80 GB	2020
H100	80 GB	2022
H200	141 GB	2024

- We need to identify:
 - What are the key technologies in the field of GPU-initiated Storage?
 - How does performance of GPU-initiated Storage compare to CPU-centric Storage?



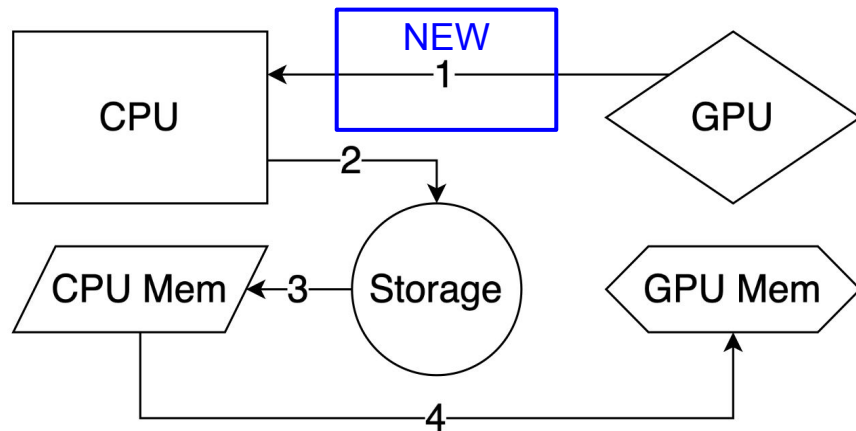
The Conventional Approach

- CPU-initiated
 - CPU loads data to CPU memory
 - Copy data to GPU memory
- Typically POSIX
- Ecosystem support, but CPU-bound and high overhead from memory copy



GPUfs (2014) & ActivePointers (2016/2018)

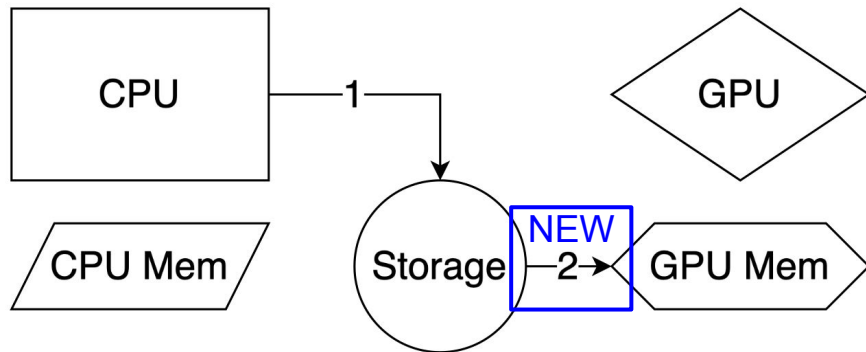
- Provide a POSIX-like API for GPUs
- GPU-initiated
 - CPU does the work



- Improve GPU programmability at the price of slightly worse performance

GDS: NVIDIA GPUDirect Storage (2019)

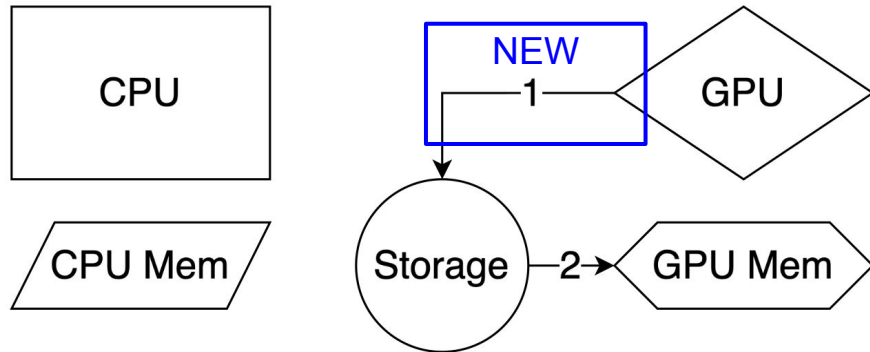
- Allows data transfer directly between storage and GPU memory
- CPU-initiated
 - Doesn't use CPU Mem



- Faster than the conventional approach, performance is CPU-bound

BaM: Big Accelerator Memory (2023)

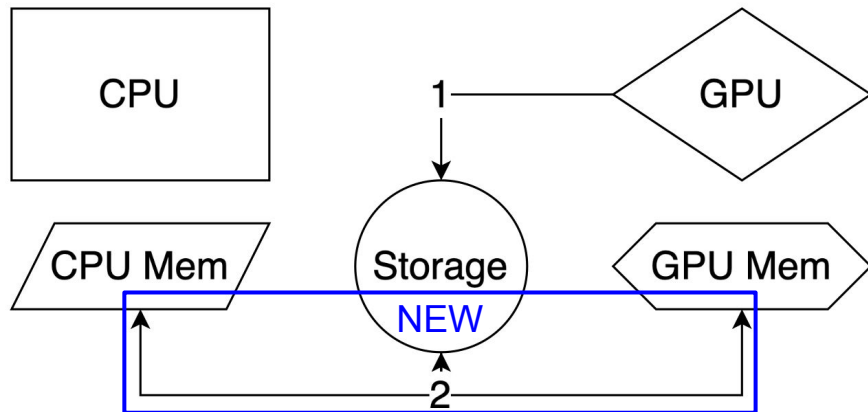
- Completely bypass the CPU when accessing storage
- GPU-initiated
 - CPU is only used for setup



- Fast storage access at the cost of saturating the GPU

GMT: GPU Orchestrated Memory Tiering (2024)

- Access storage through a 3-tier cache: GPU, CPU, Storage
- GPU-initiated
 - CPU is used for transfer to and from CPU memory



- Fast storage access, if reuse percentage is high, at the cost of spending both GPU and CPU resources

Recap

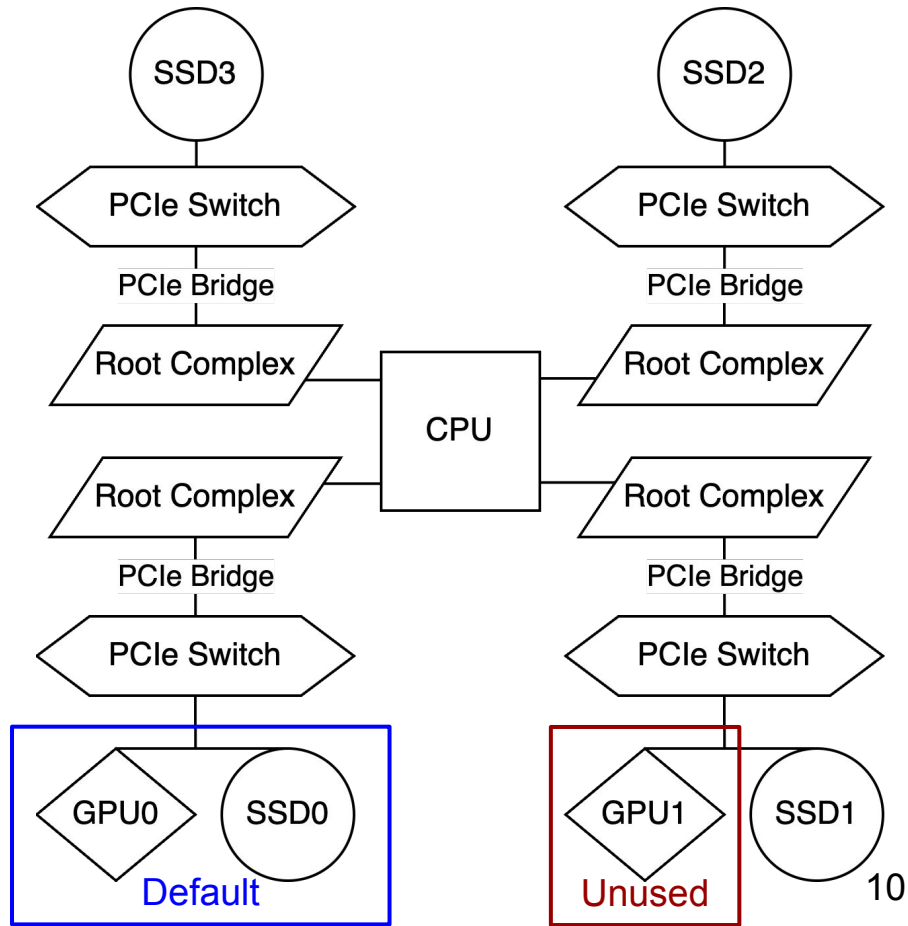
Technology	Focus		Initiation	
	Programmability	Performance	GPU	CPU
GPUfs	X		X	
ActivePointers	X		X	
GDS		X		X
BaM SOTA		X	X	
GMT		X	X	

GPU- vs CPU-centric Storage

- **SPDK represents the state-of-the-art in CPU-centric storage**
- How does the bandwidth of BaM compare to GDS and SPDK?
- How does BaM scale across multiple SSDs compared to SPDK?
- How is the resource consumption of BaM compared to SPDK?

GPU- vs CPU-centric Storage: System

System	Gigabyte G292-Z20
CPU	AMD EPYC 7402P 24-Core Processor
DRAM	8 X 32GB SK Hynix DDR4 2400MHz
GPUs	2 X NVIDIA Tesla V100-16GB PCIe Gen 3
SSDs	4 X 1TB Samsung 980 PRO w/ Heatsink
OS	Ubuntu 20.04 LTS (Linux 5.8)
NVIDIA	Driver 550, CUDA 12.6
BaM	GitHub 'master' branch
GDS	Matching CUDA (12.6)
SPDK	v24.09

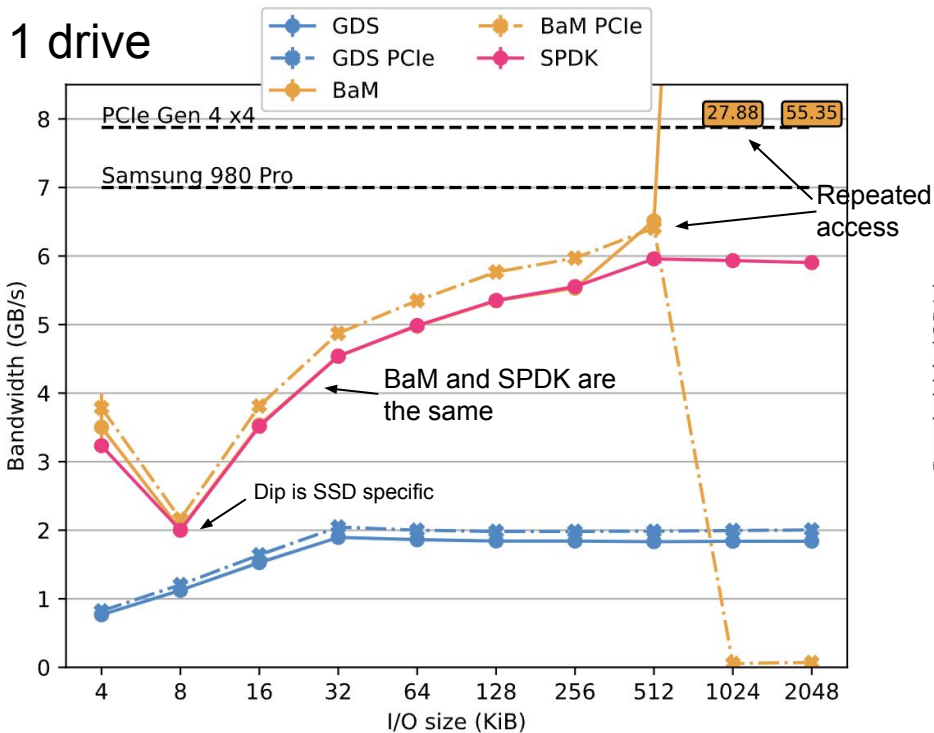


GPU- vs CPU-centric Storage: Workload

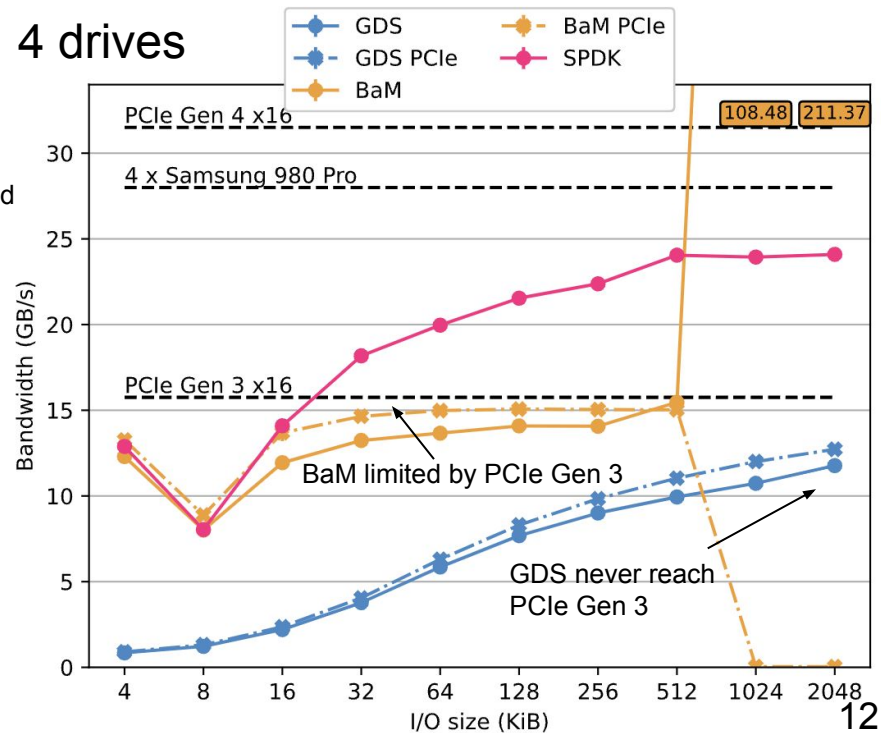
- Random Read
 - BaM: 'nvm-block-bench'
 - 1 thread and 1 I/O per page in cache = 8.59 GB / page size
 - GDS: 'gdsio'
 - 16 CPU threads
 - SPDK: 'bdevperf'
 - 1 Pair of thread siblings
 - 5 repetitions, mean and stddev
 - GPU PCIe traffic measured by NVIDIA 'dcgmi'

Bandwidth of BaM vs GDS vs SPDK

- BaM is comparable to SPDK, but capped by GPU PCIe Gen 3



- GDS can't keep up



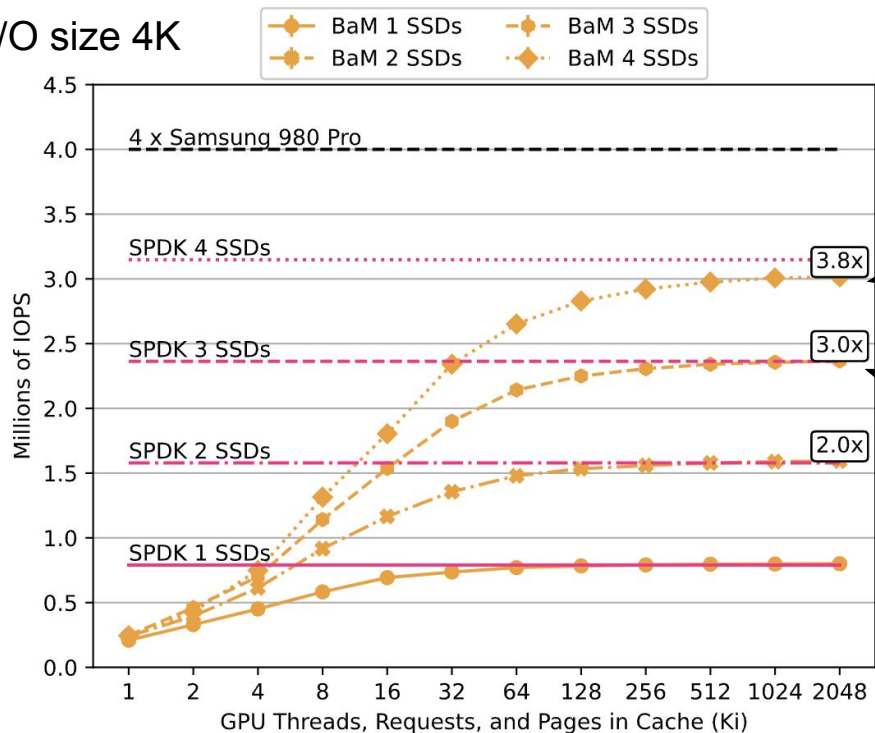
Is BaM scaling linearly?

- GPU memory is limiting cache size

Resource consumption

- BaM fully saturates the GPU
- SPDK requires a single physical core

I/O size 4K



4 Drives: BaM falls off

1-3 Drives: Linear scaling

Discussion

- **What is needed to bring GPU-initiated IO to real-world applications?**
 - Integration into AI frameworks (e.g., Pytorch)
 - CPU/GPU resource management
- **What is the optimal abstraction?**
 - GPUfs, ActivePointers, GDS: File abstraction (CSV, JSON, JPEG etc.)
 - BaM, GMT: Array abstraction (Blocks)

Karl B. Torp, k.torp@samsung.com

Simon A. F. Lund, simon.lund@samsung.com

Pinar Tözün, pito@itu.dk

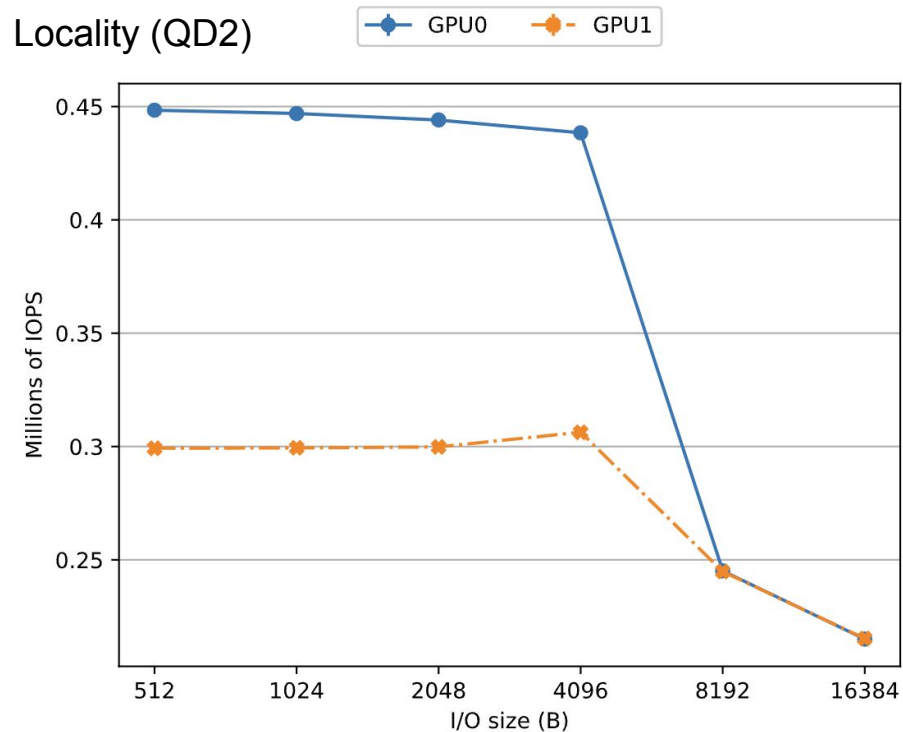
SAMSUNG

IT UNIVERSITY OF COPENHAGEN

RAD

Is BaM affected by locality?

- Locality impact amortized at 8KiB



GPU- vs CPU-centric Storage: Workload

- Random Read
 - BaM: 'nvm-block-bench'
 - 1 I/O per page in cache = 8.59 GB / page size, e.g 2,097,152 pages of size 4K
 - 128 qpairs of depth 1024 for each SSD
 - GDS: 'gdsio'
 - Running for 10 seconds
 - 16 CPU threads and 10GB filesize
 - SPDK: 'bdevperf'
 - Running for 10 seconds
 - 1 Pair of thread siblings, queue depth 256
- Metrics: IOPS and Bandwidth (GB/s)
 - 5 repetitions, mean and stddev reported
 - GPU PCIe traffic measured by 'dcgmi'