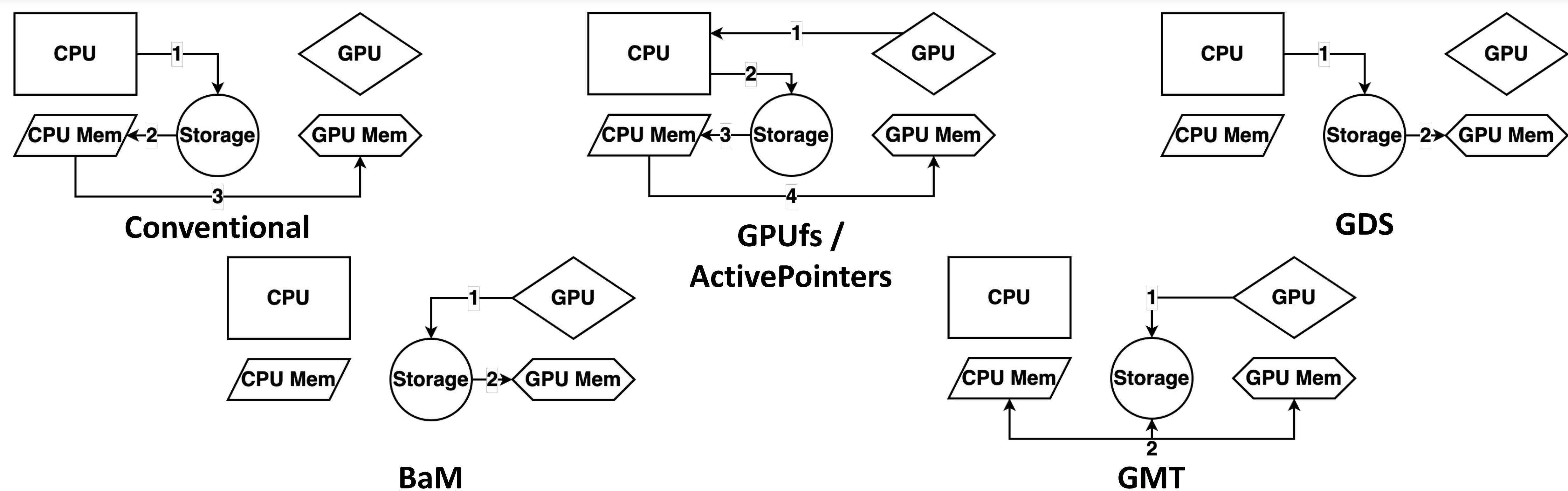


# Path to GPU-Initiated I/O for Data-Intensive Systems

Karl B. Torp (k.torp@samsung.com) - Simon A. F. Lund (simon.lund@samsung.com) - Pinar Tözün (pito@itu.dk)

## GPU-centric Storage



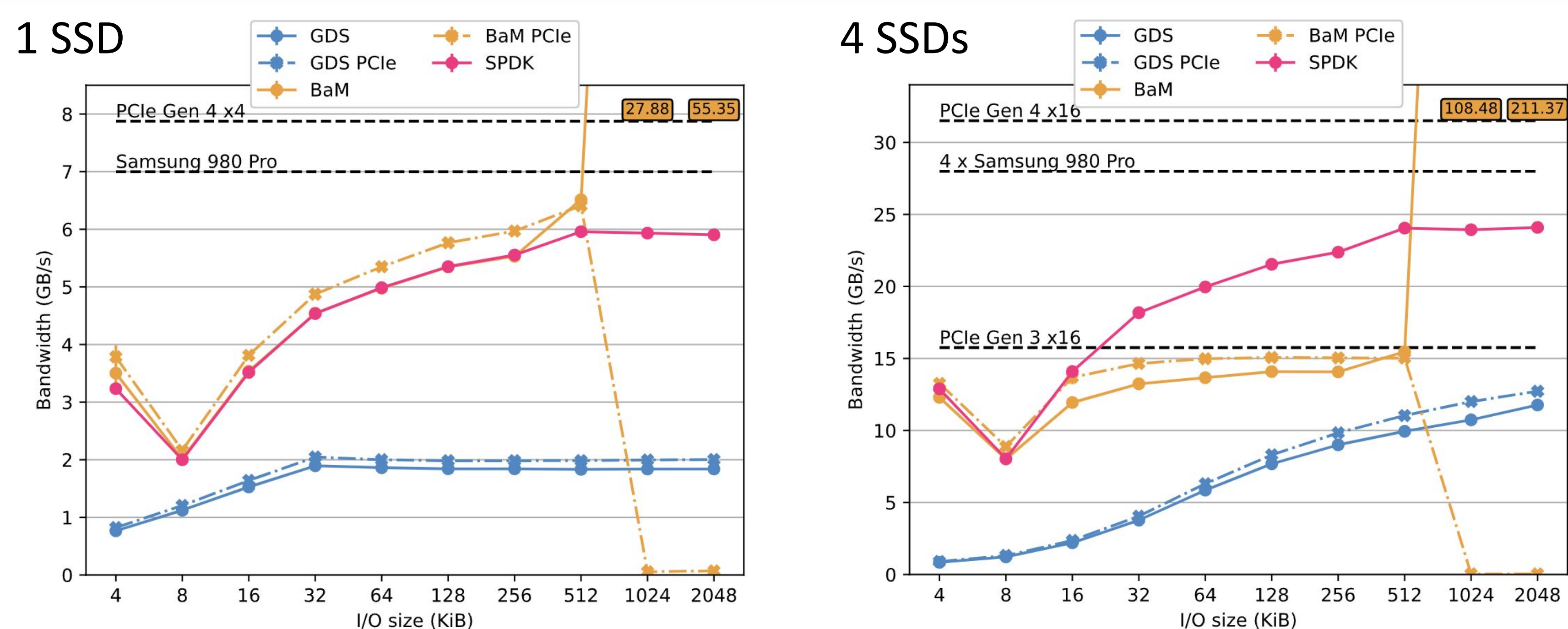
Technology	Focus		Initiation		Uses CPU Memory
	Programmability	Performance	GPU	CPU	
GPUfs	X		X		X
ActivePointers	X		X		X
GDS		X		X	
BaM		X	X		
GMT		X	X		X

## Experimental Setup

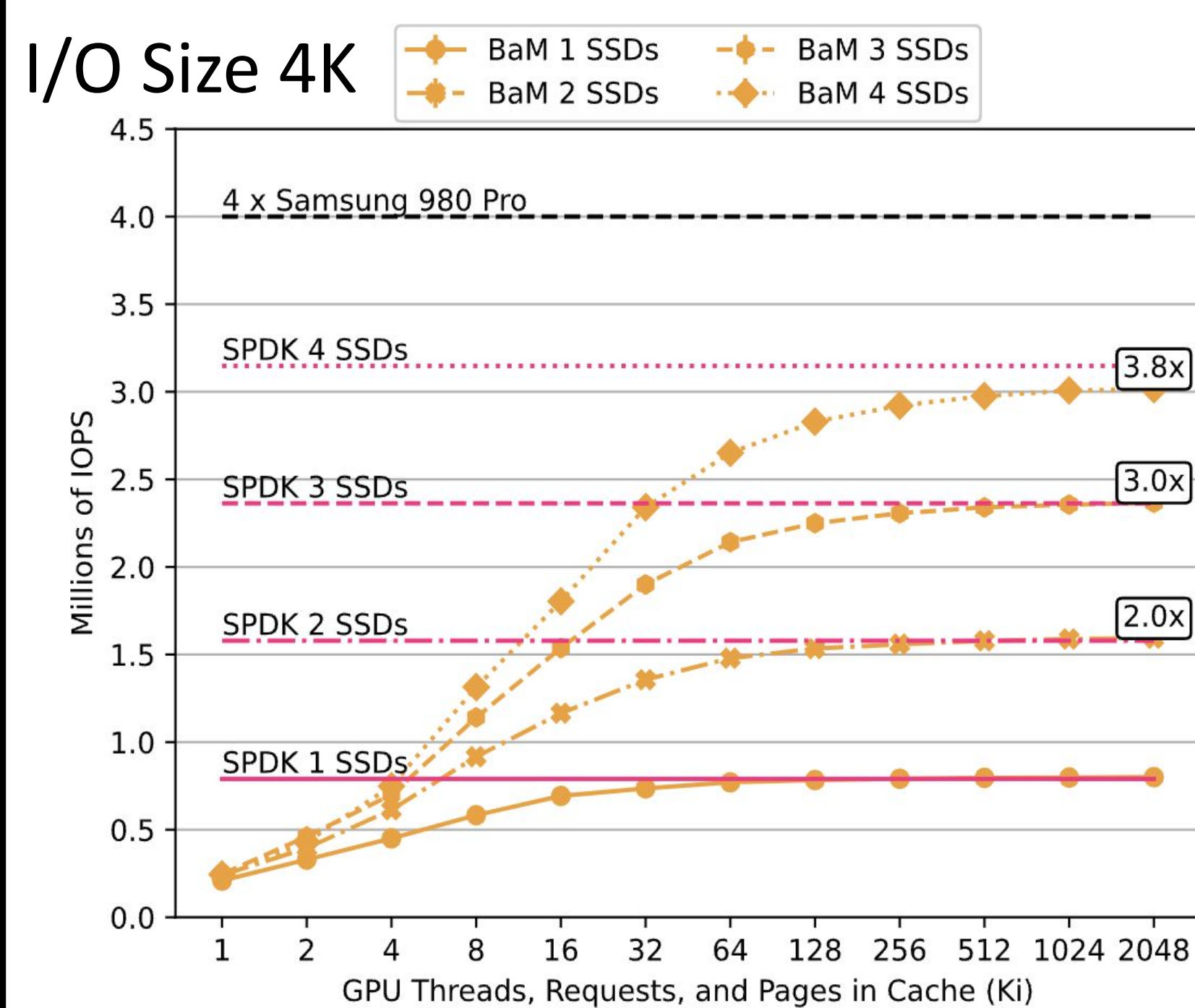
Workload: Random Reads

<b>CPU</b>	AMD EPYC 7402P 24-Core Processor
<b>DRAM</b>	8 X 32GB SK Hynix DDR4 2400MHz
<b>GPUs</b>	2 X NVIDIA Tesla V100-16GB PCIe Gen 3
<b>SSDs</b>	4 X 1TB Samsung 980 PRO w/ Heatsink

## How does BaM compare to GDS and SPDK?



## How does BaM scale?



## Results

### Performance

- **BaM's** bandwidth is comparable to **SPDK's**, but capped by PCIe Gen 3
- **GDS** can't keep up
- **BaM** doesn't scale linearly after 3 SSDs due to limited GPU memory

### Resource consumption

- **BaM** fully saturates the GPU
- **SPDK** can saturate multiple disks with a single physical core
- With **GDS** 16 CPU threads is not enough to saturate PCIe bandwidth

## Discussion

### What is the optimal abstraction?

- **GPUfs, ActivePointers, GDS:** File abstraction
- **BaM, GMT:** Array abstraction

Would a resource aware solution be preferred?

What is the performance in the context of "real" workloads?