Performance Optimizations for Advanced Non-volatile Storage Arrays

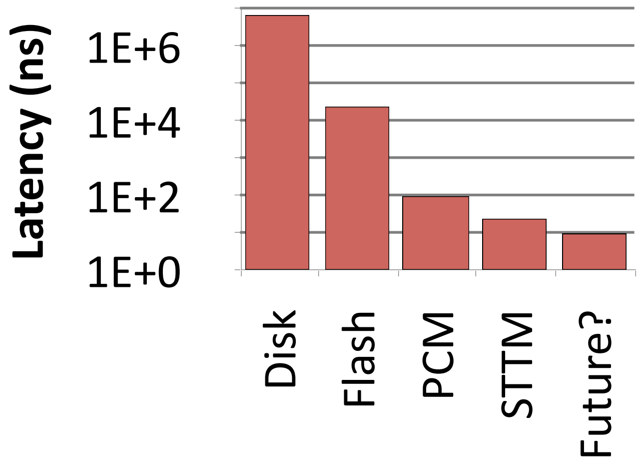
Adrian Caulfield, Joel Coburn, Todor Mollov, Arup De, Ameen Akel, Jiahua He, Arun Jagatheesan, Rajesh Gupta, Allan Snavely, Steven Swanson

Non-Volatile Systems Laboratory
Department of Computer Science and Engineering
University of California, San Diego

Advances in Storage Technology

- New memories will revolutionize the way we treat storage
  - 10s-100s of nanoseconds latencies
  - Interconnect saturating bandwidth (PCIe, SATA)
  - Increased parallelism from many small memory devices

- Flash memory is already replacing disks in many applications because of its low latency
- Emerging NVMs will be even faster and behave more like DRAM
  - Phase Change Memory
  - Spin-Transfer Torque Memory
  - Memristor



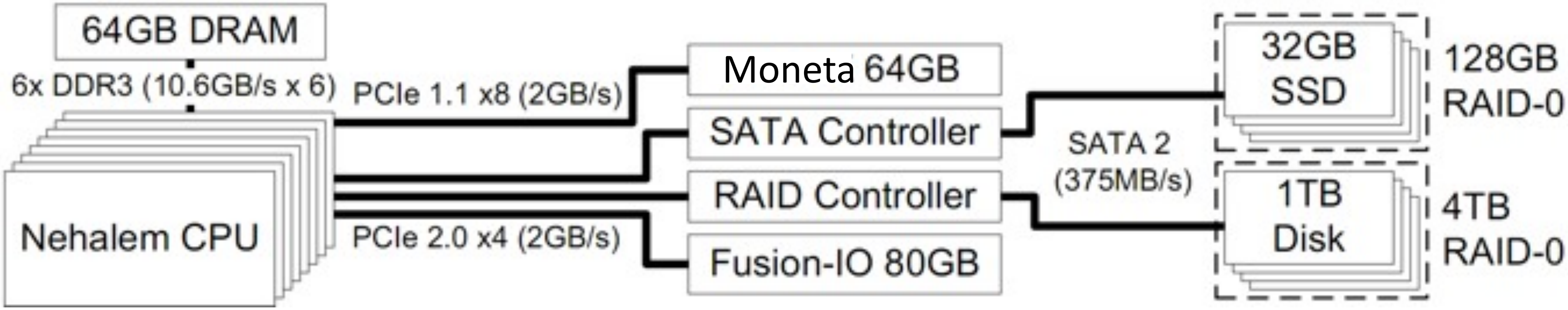
Applications

- Fast storage impacts:
  - Software disk caches
  - Read/Write system calls
  - Log structured file systems
  - IO schedulers
  - Software drivers
  - Interrupt processing
  - CPU requirements for IO
- Who benefits from improved storage?
- IO intensive applications
  - File system accesses
  - Databases
  - Scientific workloads
  - Huge working sets
  - Virtualization

Overview

- Motivation
- System Overview
- Basic IO Performance
- Application Performance
- Conclusion

System Overview

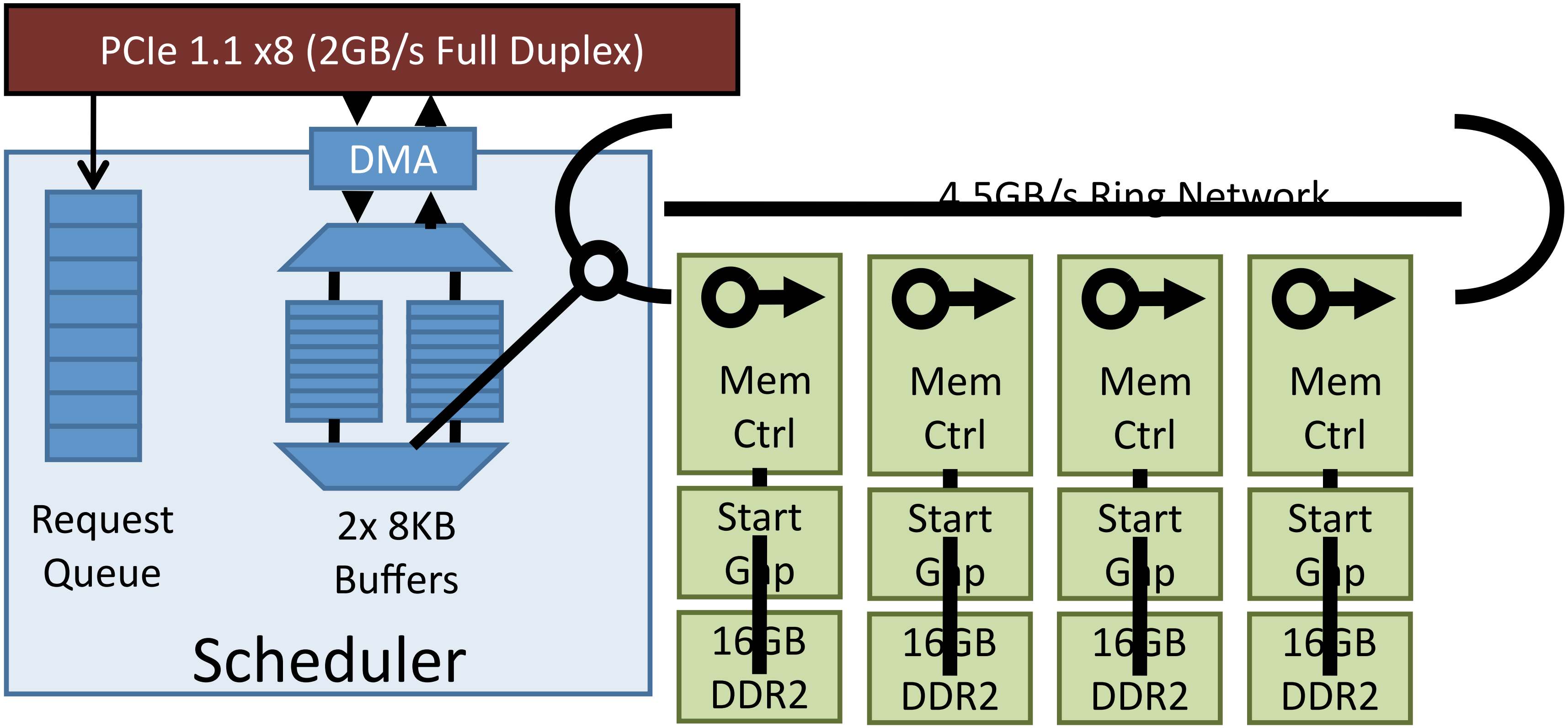


Memory and Device	Interconnect	Capacity
Fusion-IO IODrive	PCIe 2.0 4x	80GB
SLC NAND Flash SW	PCIe 2.0 4x SATA 2	128GB
Disk HW RAID-0	PCIe 2.0 4x RAID	4TB
DDR3-attached PCM	6x DDR3 Channels	64GB
PCIe-attached PCM	PCIe 1.1 8x	64GB

Moneta: Modeling Advanced NVMs

- FPGAs connected via PCIe
- DDR2 memory to emulate NV memories
- Add latency to the existing DDR commands
  - t_rcd: RAS-CAS Delay –delay to read a row into a buffer
  - t_wrp: Write/Read – delay to write a row into memory

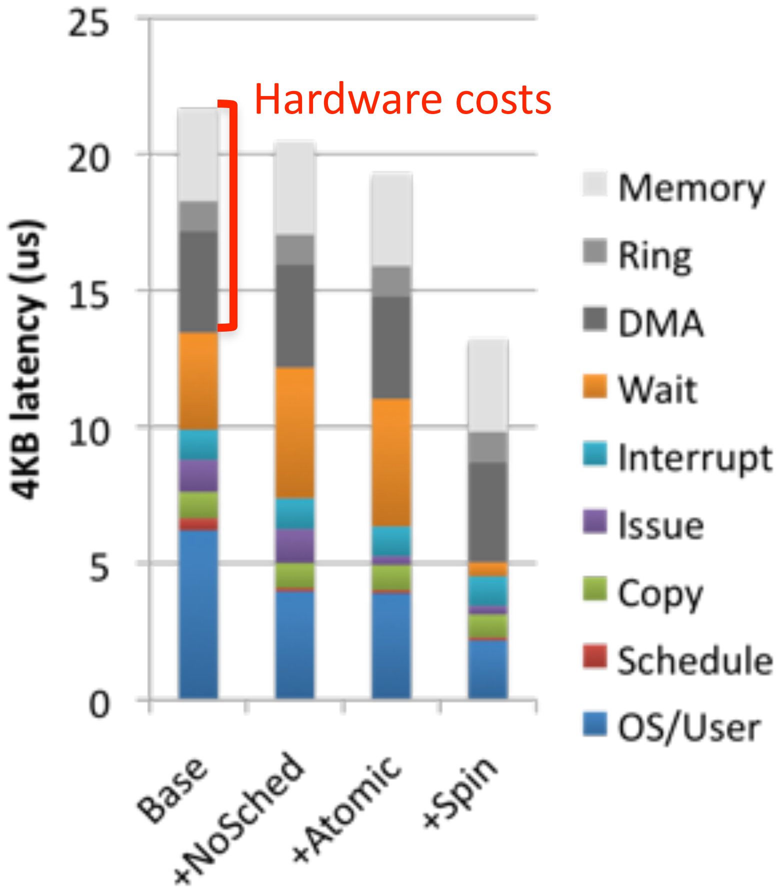
Moneta Architecture



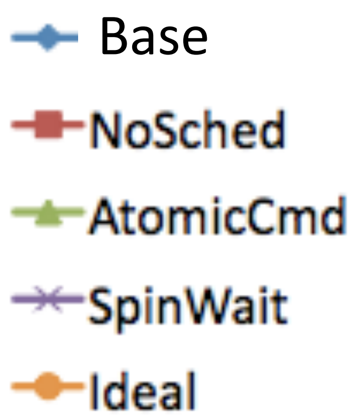
Built on the BEE3 FPGA prototyping board

A Good Driver is Critical

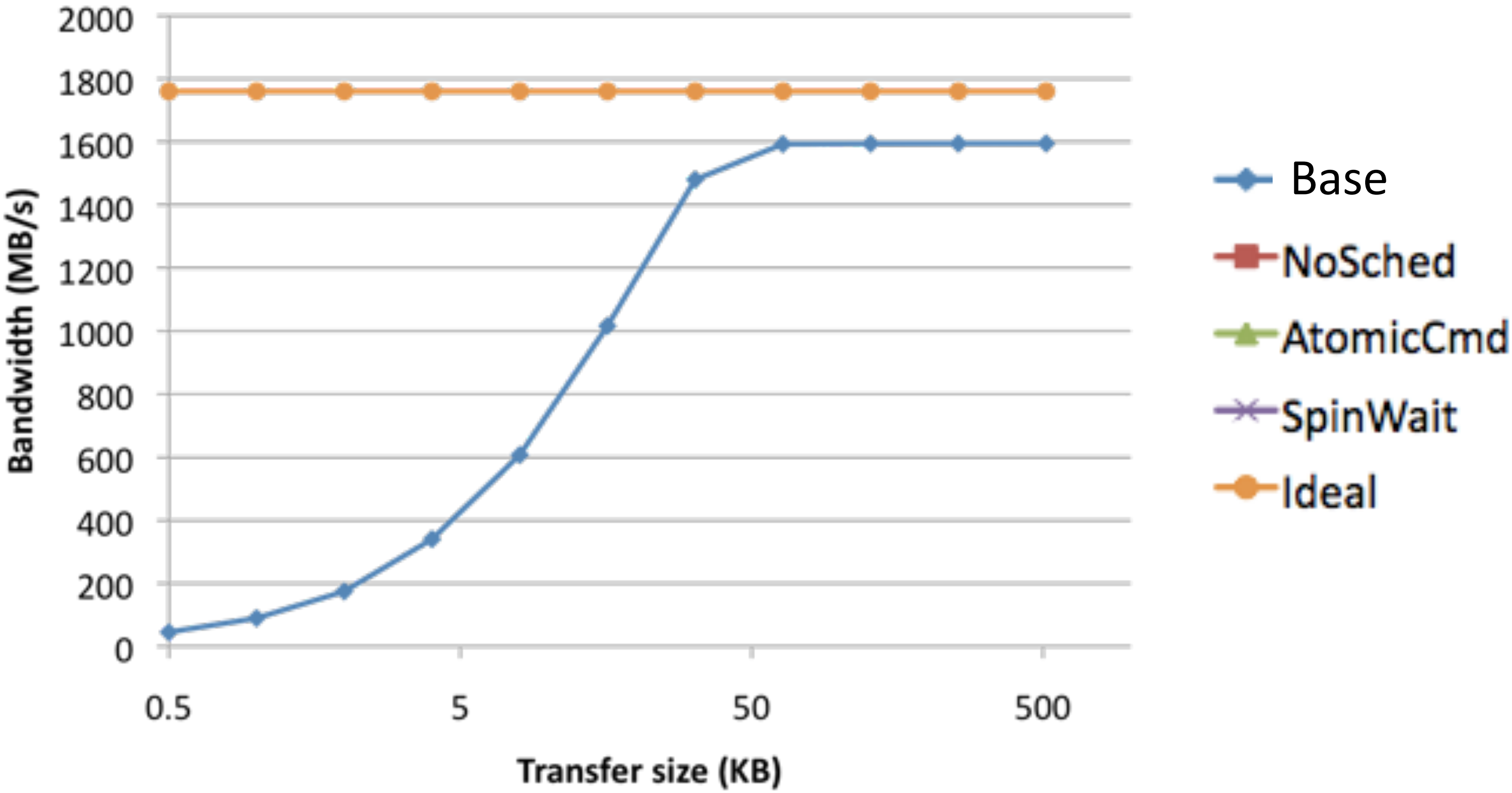
- Optimizations
  - Baseline
  - No scheduler
  - Atomic command issue
  - Spin wait for completion
- Removed 2/3 of SW latency
- Removed all locks
- What remains?
  - Interrupt processing
  - Entering/leaving the kernel



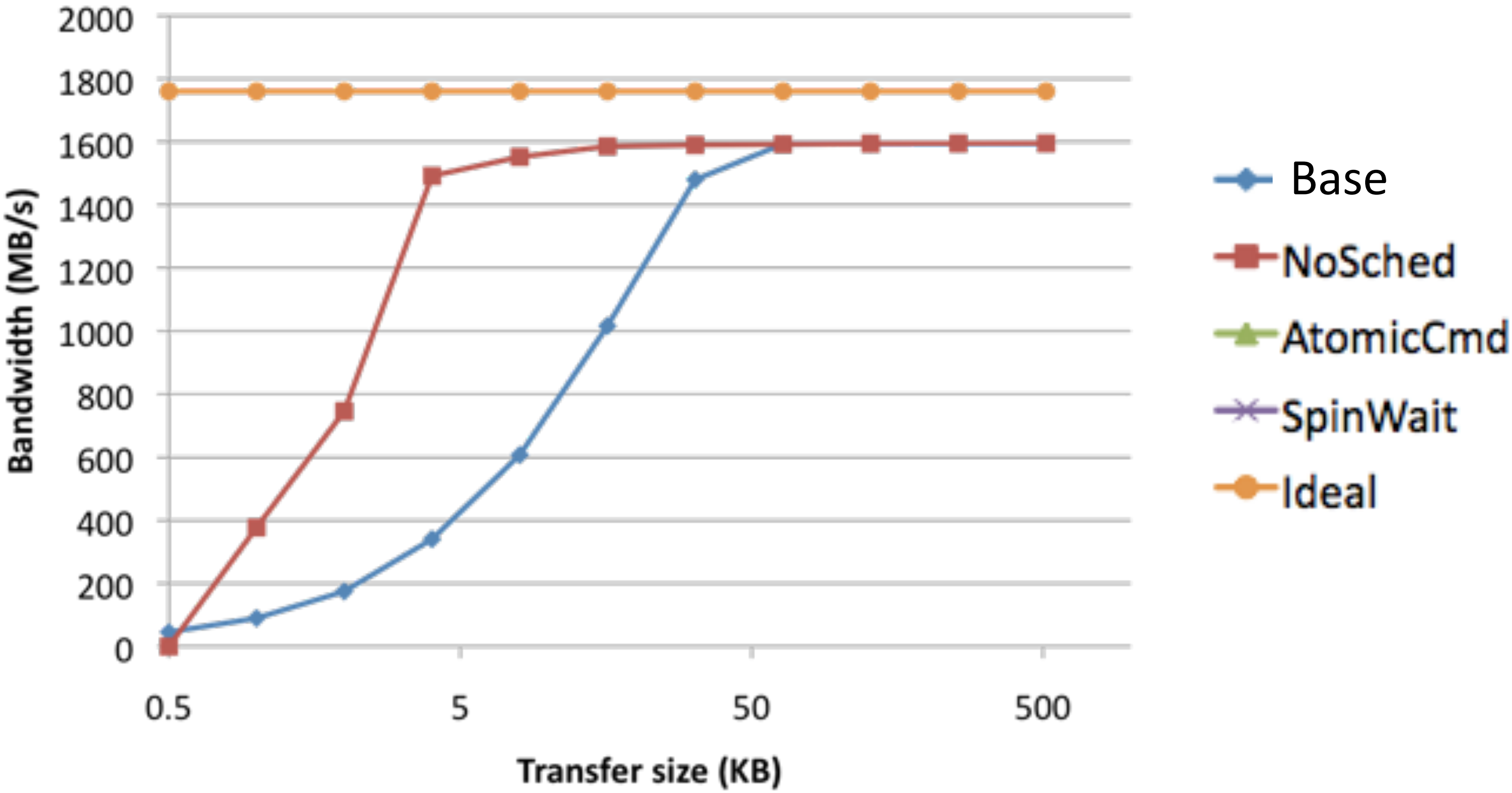
Moneta IO Performance (Writes)



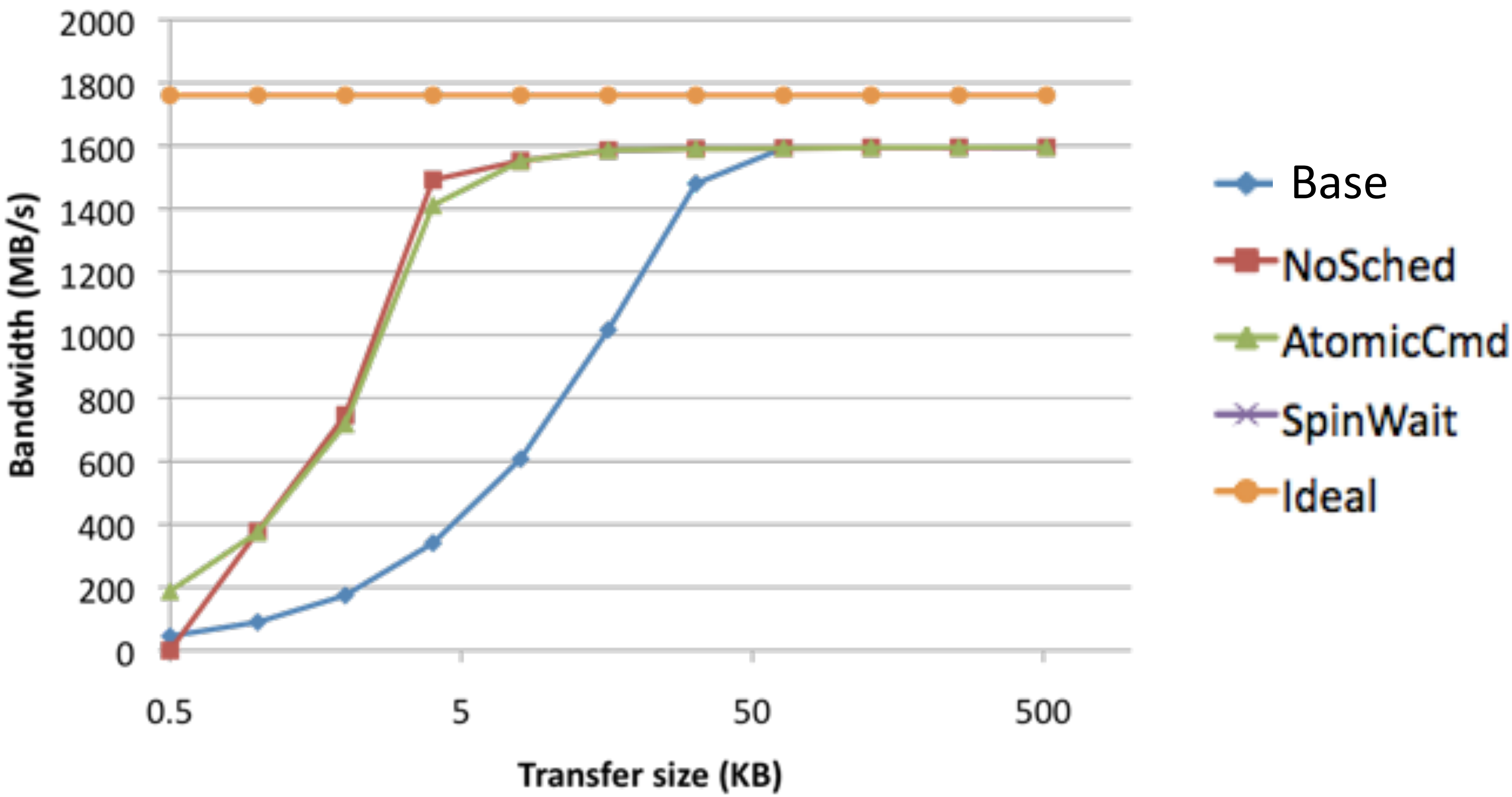
Moneta IO Performance (Writes)



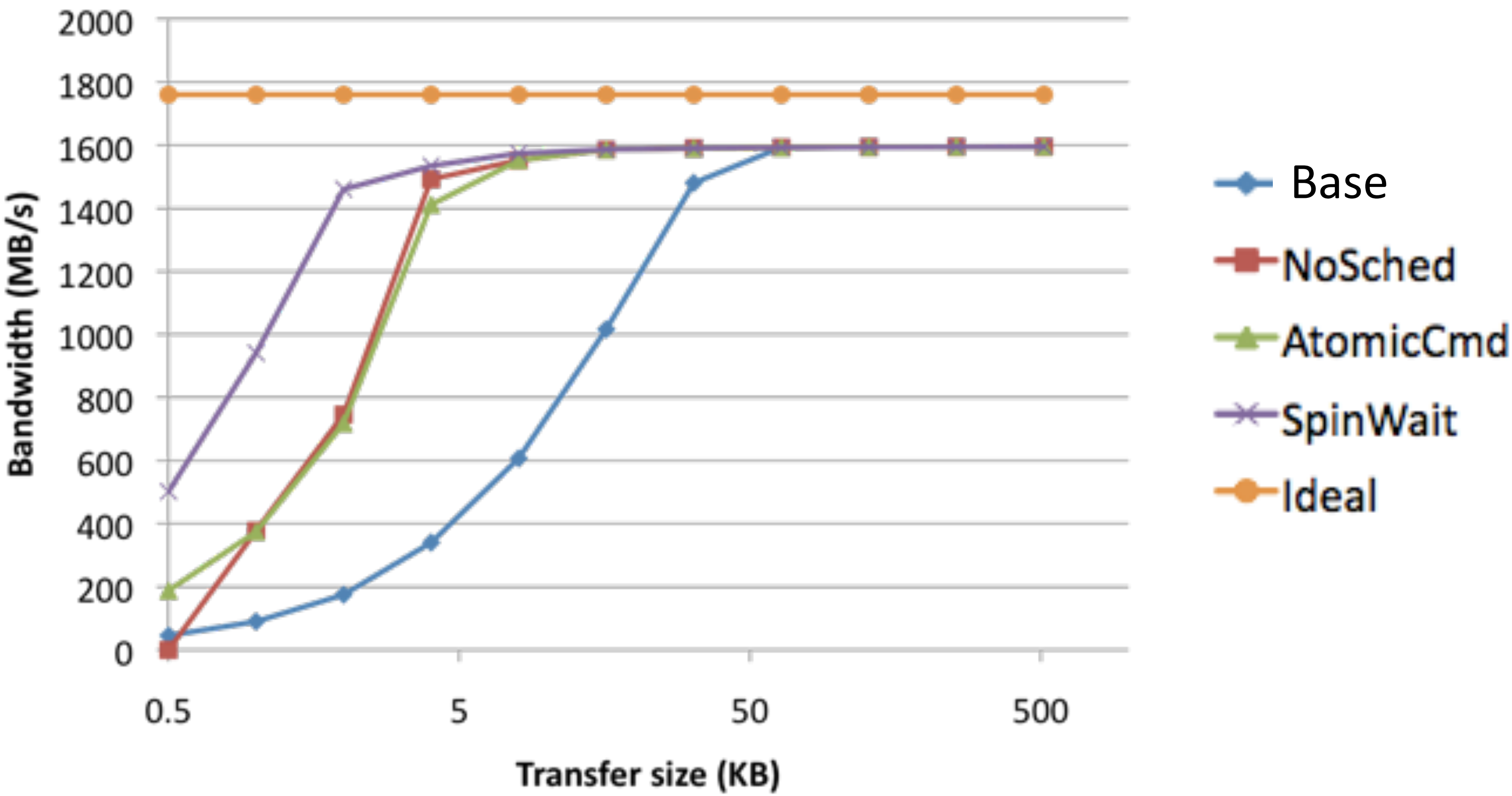
Moneta IO Performance (Writes)



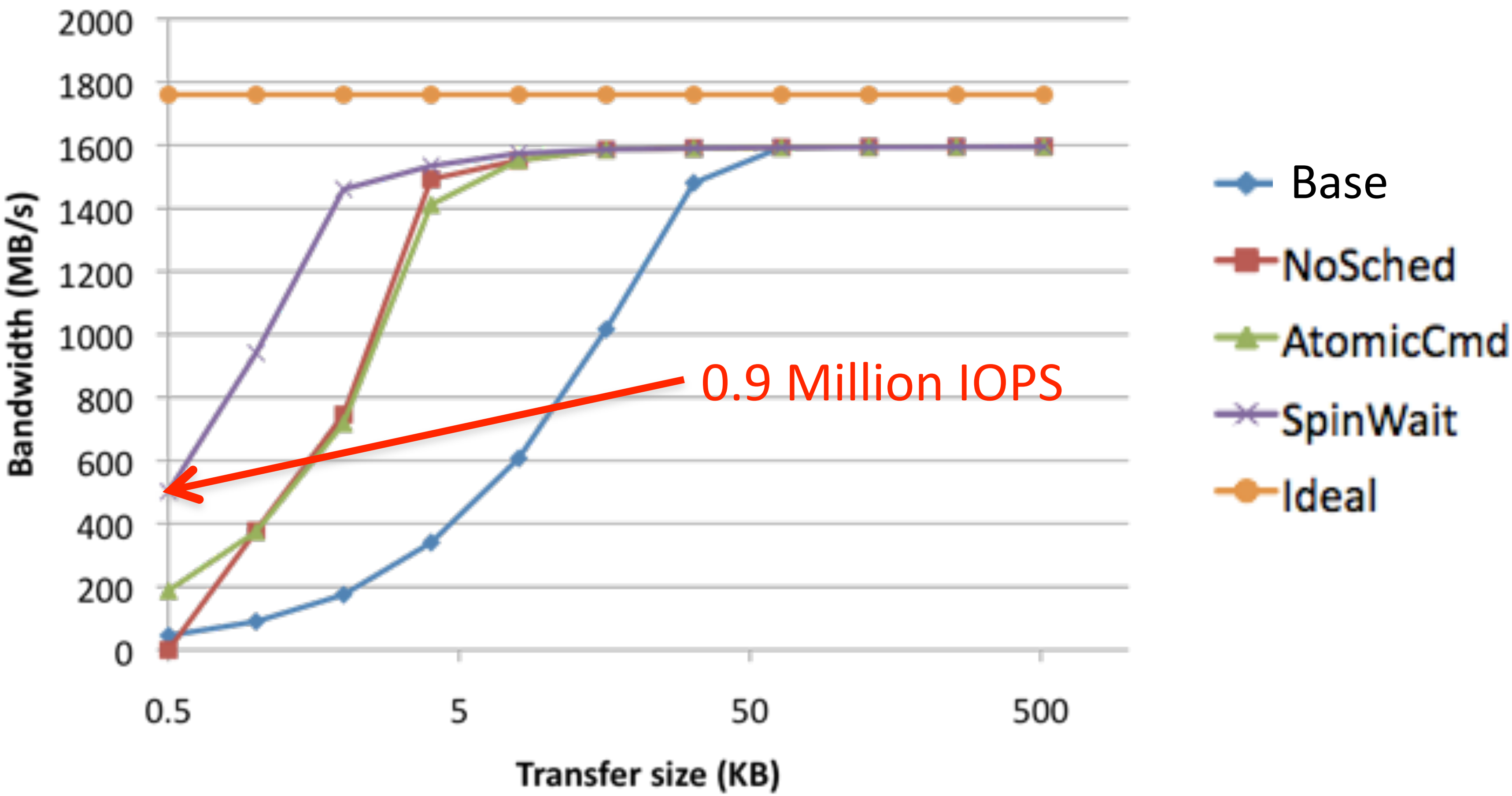
Moneta IO Performance (Writes)



Moneta IO Performance (Writes)



Moneta IO Performance (Writes)



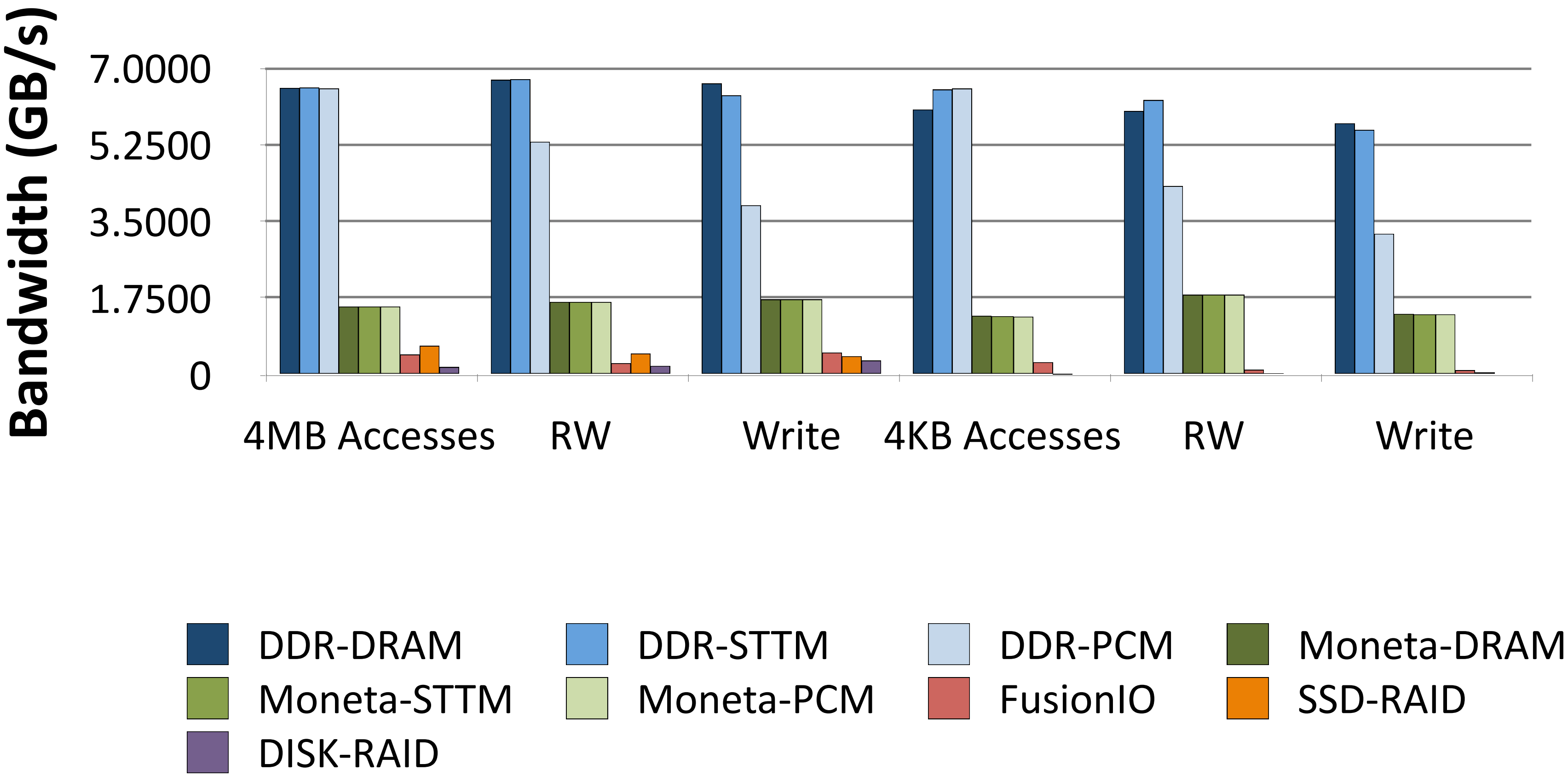
Overview

- Motivation
- System Overview
- Basic IO Performance
- Application Performance
- Conclusion

XDD Bandwidth and Latency

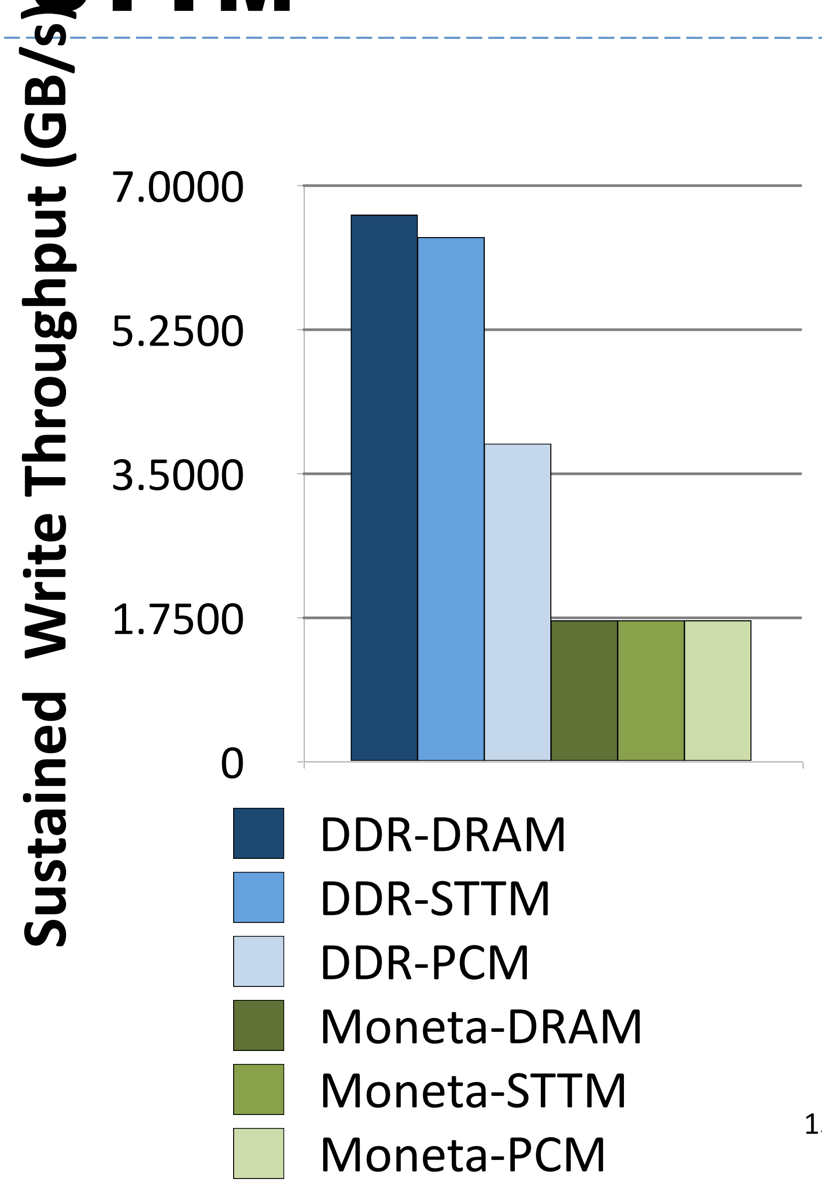
- XDD is a low-level IO benchmarking tool
- Request size: 4KB or 4MB
- Request operation: Read, Write, 50/50 R/W
- XFS and Raw device access

Raw Bandwidth



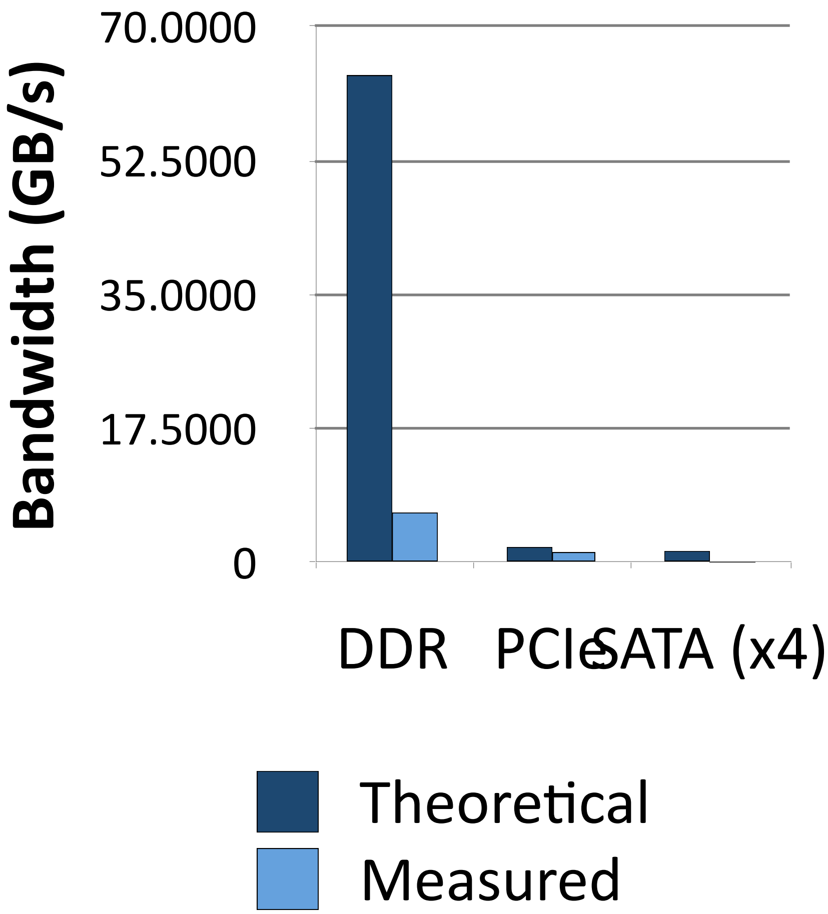
Modeling PCM and STTM

- DDR bus exposes latency
- Requests split into pieces
- DDR
  - 64B accesses (cache-line)
  - 128 row access latencies/8KB
- Moneta hides latency well
  - 8KB accesses (row buffer)
  - 1 row access latency/8KB



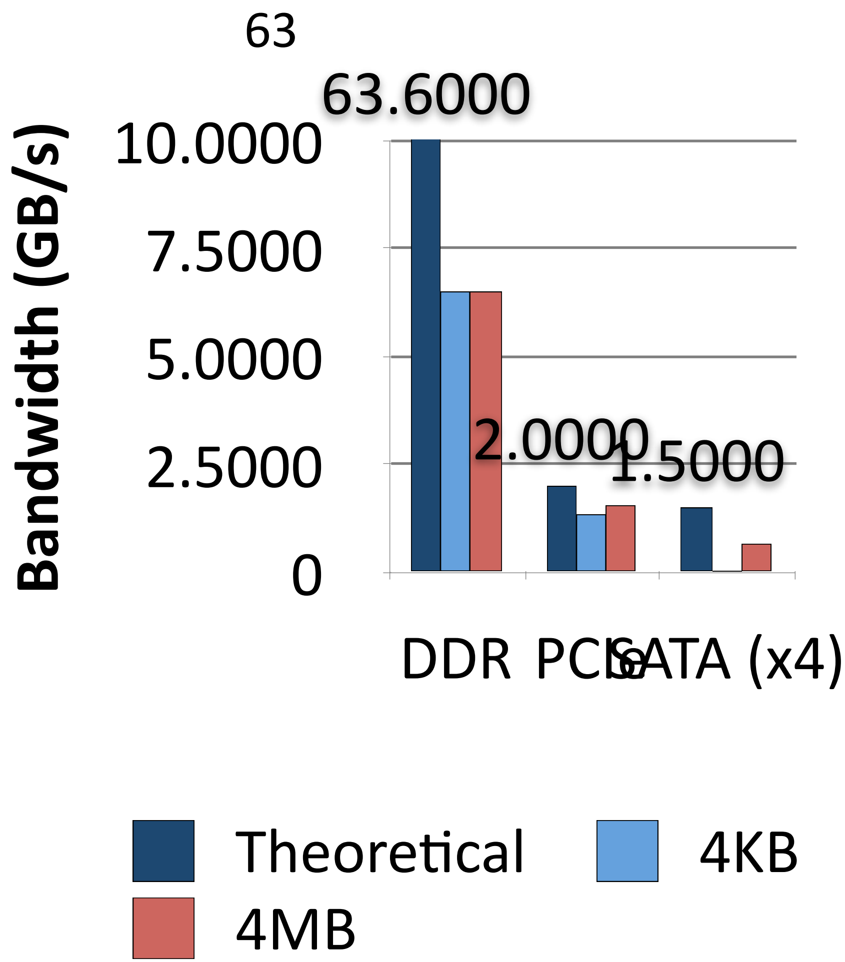
Interconnect Efficiency: 4KB Reads

- Unused bandwidth:
  - 89% DDR
  - 34% PCIe
  - 98% SATA
- Possible limitations:
  - CPU throughput
  - Request overhead

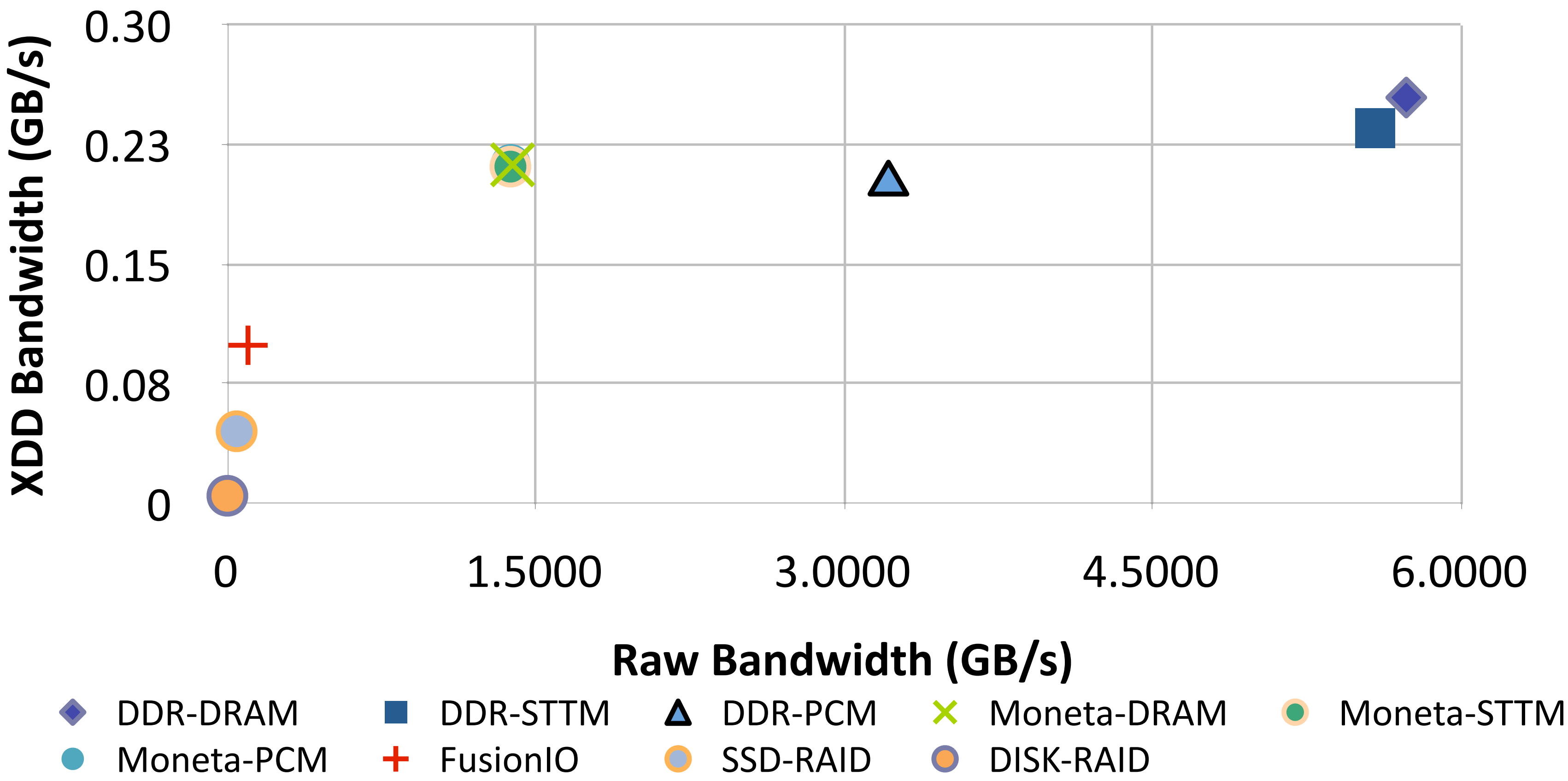


Interconnect Efficiency: 4MB Reads

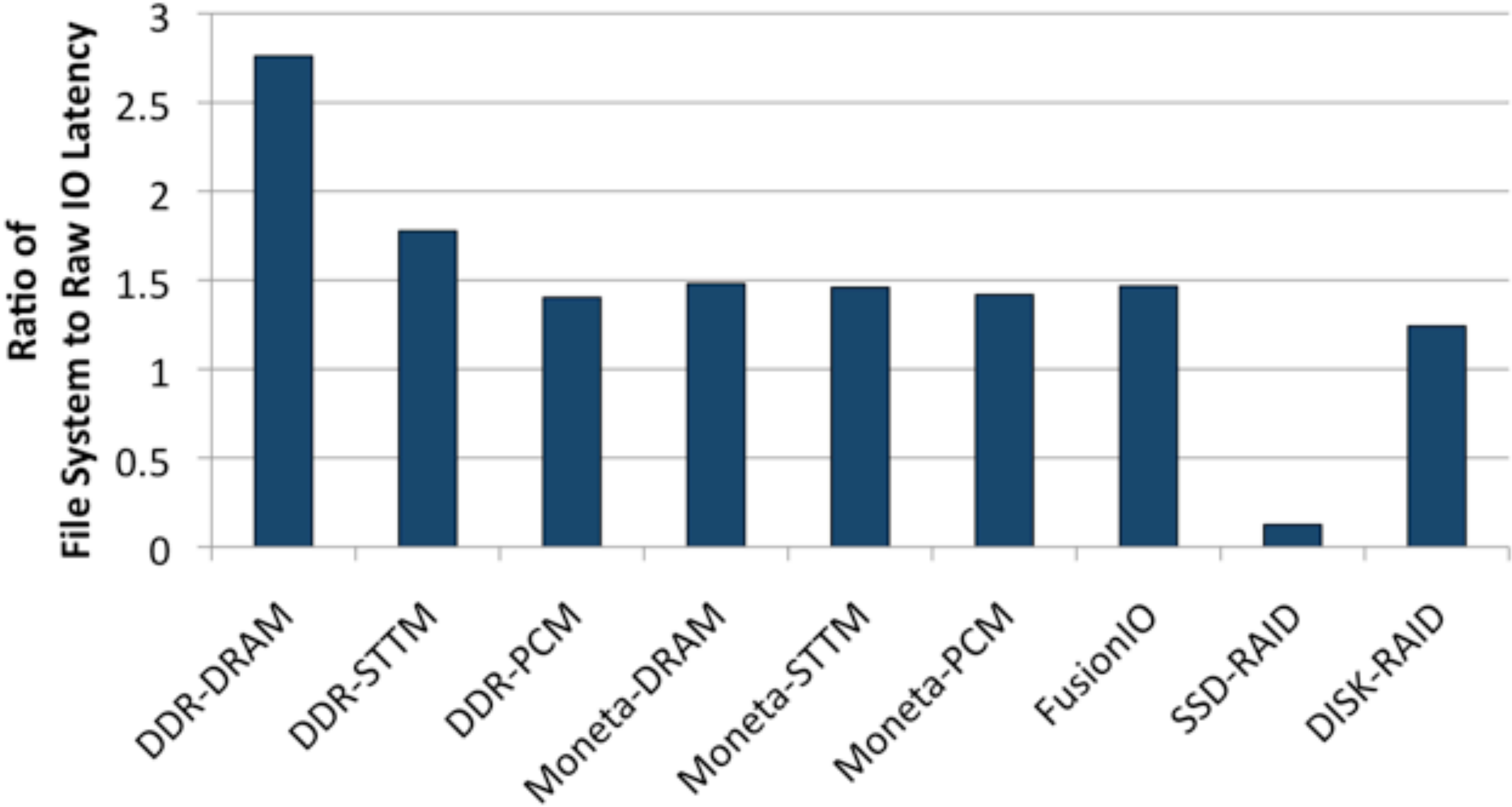
- No DDR improvement
  - Requests broken up
  - Performance limited by 64B accesses
- PCIe and SATA benefit
  - Reduced request overhead
  - Overlap requests
  - Bulk DMA transfer



File System Performance: 4KB Writes



XFS Latency vs Raw IO Latency



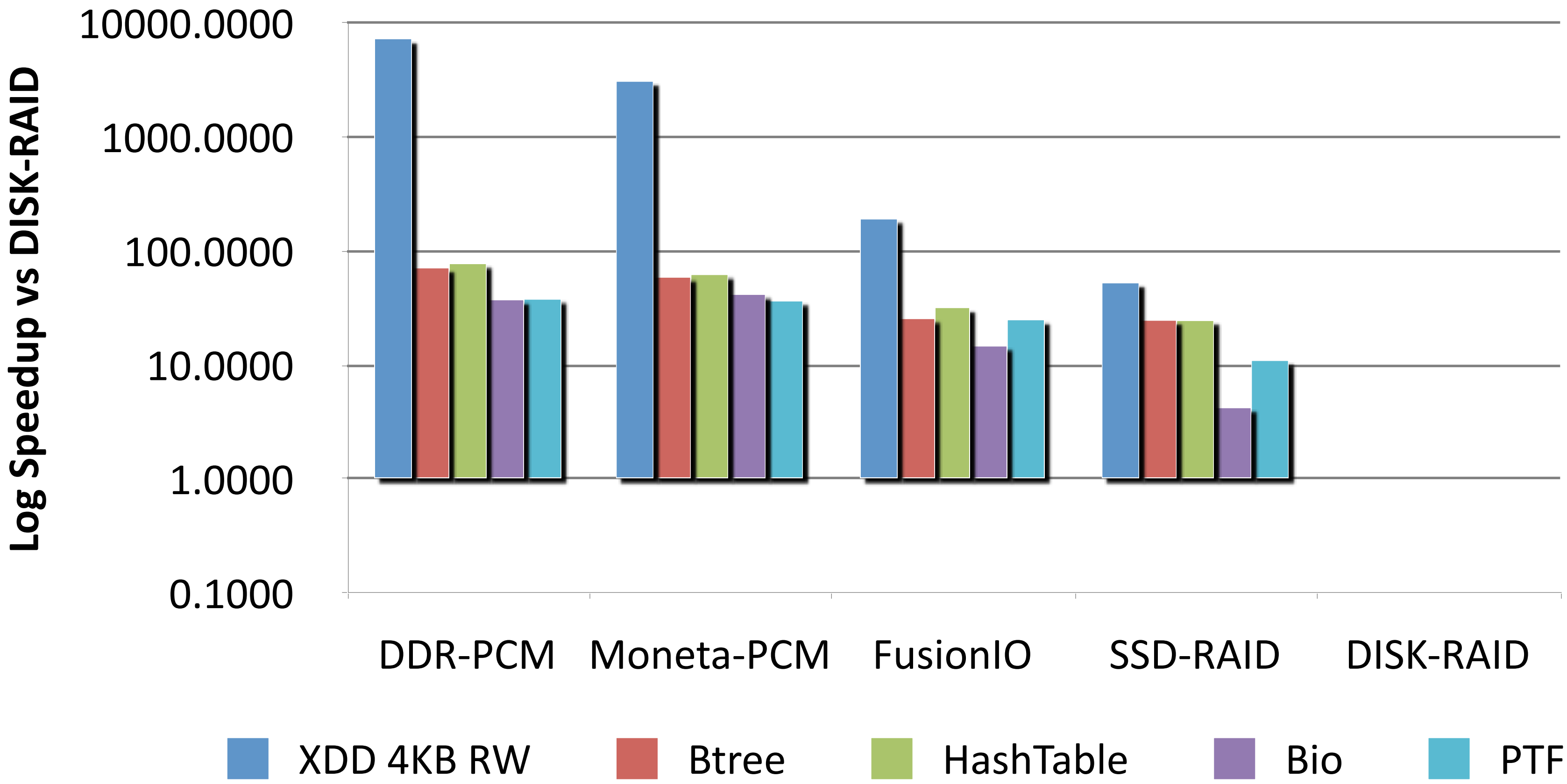
Overview

- Motivation
- System Overview
- Basic IO Performance
- Application Performance
- Conclusion

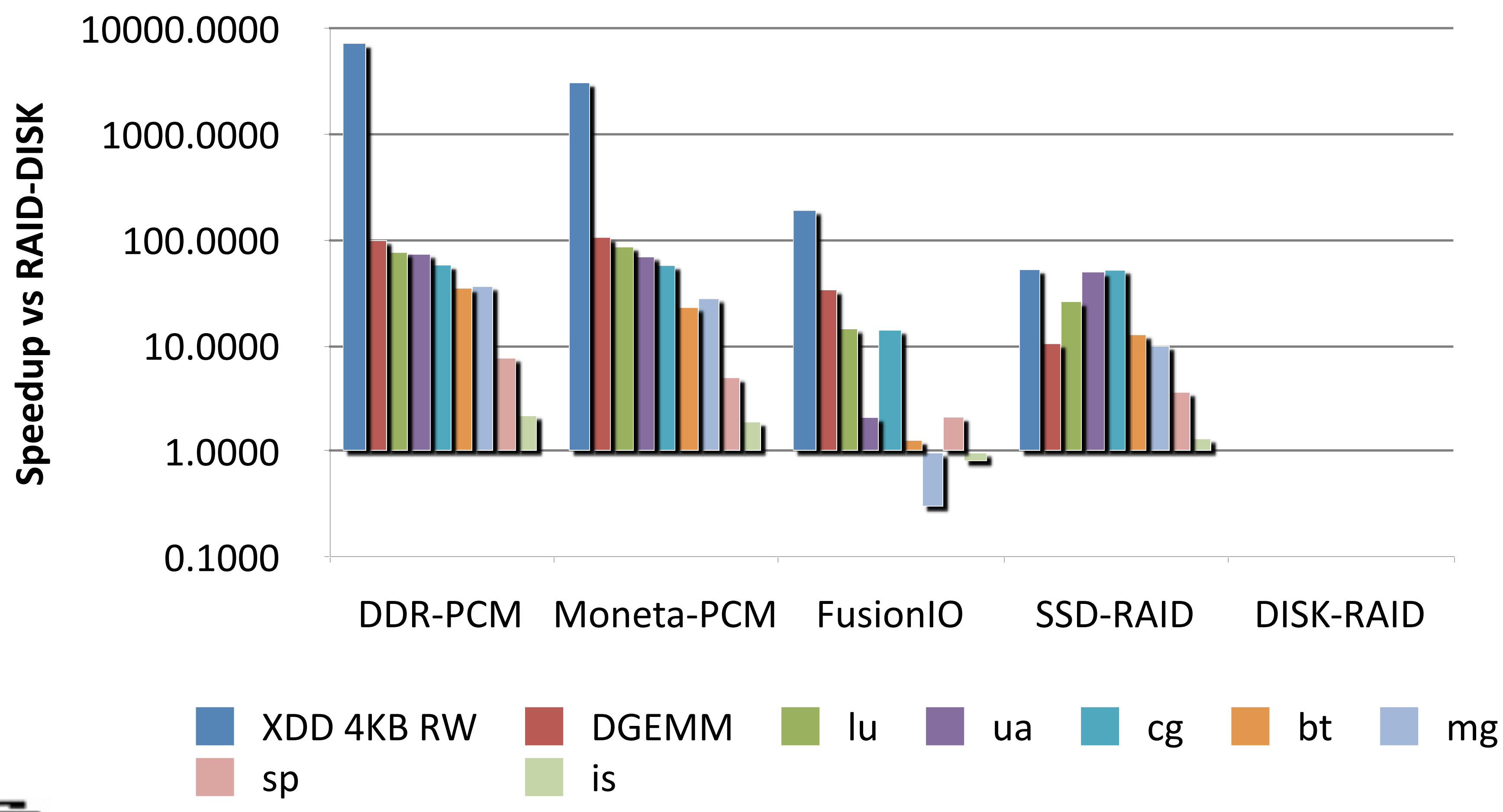
Workloads

Name	Footprint	Description
		Database Applications
Berkeley-DB Btree	16 GB	Transactional updates to btree key/value store
Berkeley-DB HashTable	16 GB	Transactional updates to hash table key/value store
BiologicalNetworks	35 GB	Biological database queried for properties of genes and biological-networks
PTF	50 GB	Palomar Transient Factory sky survey queries
		Memory-hungry Applications
DGEMM	21 GB	Matrix multiply with 30,000 x 30,000 matrices
NAS Parallel Benchmarks	8-35 GB	7 apps from NPB suite modeling scientific workloads

Database Performance



Memory-Hungry App Performance



Conclusion

- Software is not ready to take advantage of fast NVMs
- Flash is starting to break designs based on disk
  - IO schedulers, system calls, file systems, interconnects
  - Applications
- PCM, STTM, others will cause even larger changes
  - Applications will see ~100x speedup
  - There's another 100x on top of that

Thank You!

Any Questions?