Harmonia: A Globally Coordinated Garbage Collector for Arrays of Solid-state Drives

Presented by Youngjae Kim

Collaborators: Sarp Oral, Galen M. Shipman, Junghee Lee

David Dillow, and Feiyi Wang

May 26, 2011

A Demanding Computational Environment

Jaguar XT5	18,688 Nodes	224,256 Cores	300+ TB memory	2.3 PFlops
Jaguar XT4	7,832 Nodes	31,328 Cores	63 TB memory	263 TFlops
Frost (SGI Ice)	128 Node institutional cluster			
Smoky	80 Node software development cluster			
Lens	30 Node visualization and analysis cluster			

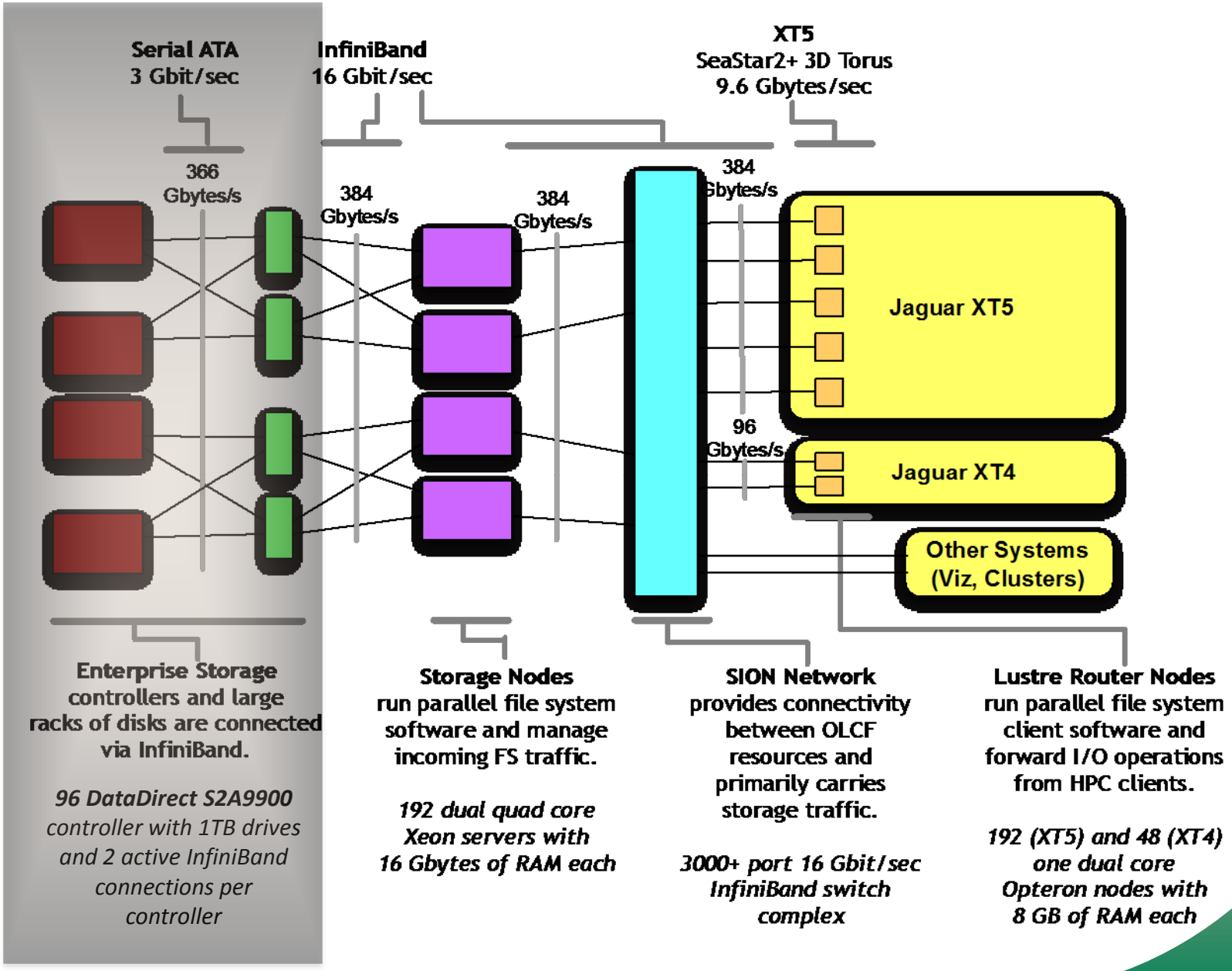
Spider: A Large-scale Storage System

- **Center-wide File System**
  - Based on Lustre file system
- **192 Lustre I/O Servers**
  - Over 3TB of memory (on Lustre I/O servers)

- **Back-end Disk Arrays**
  - Over 10.7 PB of RAID 6 formatted capacity
  - 13,400 x 1 TB HDDs

- **IB Network**
  - Available to many compute systems through high-speed IB network
    - Over 2,000 IB ports
    - Over 3 miles (5 kilometers) cable
    - Over 26,000 client mounts for I/O
    - Peak I/O performance is 240 GB/s

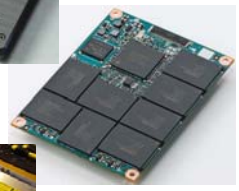
Spider Architecture



Hard Disk Drive

- **Main Storage Media for Object Storage Targets (OSTs)**
- **OST = 10 x 1TB Disks (8+2 RAID 6 Configuration)**
- **Hard Disk Drive**
  - Mechanical device
  - Spindle and voice-coil motors

Emergence of NAND Flash based SSD



MacBook Air

- **NAND Flash vs. Hard Disk Drives**
  - **Pros:**
    - Semi-conductor technology, no mechanical parts
    - Offer lower access latencies
      - *µs* for SSDs vs. *ms* for HDDs
    - Lower power consumption
    - Higher robustness to vibrations and temperature
  - **Cons:**
    - Limited lifetime
      - 10K - 1M erases per block
    - High cost
      - About 8X more expensive than current hard disks
    - Random writes can be sometimes slow
      - *Performance variability*

SSD based Object Storage Target (OST)

- **SSD based OSTs**
  - PCI Express SSDs
    - Fusion IO ioDrive, Virident tachIOn, OCZ RevoDrive, etc
  - SATA SSDs
    - Intel, SuperTalent, Samsung, etc

$13,990/640GB

~1.3GB/s

Fusion io 640GB
MLC PCIe DUO ioDrive

- PCIe SSDs versus RAID of SATA SSDs

SSD Type	Performance	Cost
PCIE SSD (Fusino IO)	High	Expensive
Array of SATA SSDs	High	Relatively Cheap

$799/64GB

Intel X25-E
64GB SSD

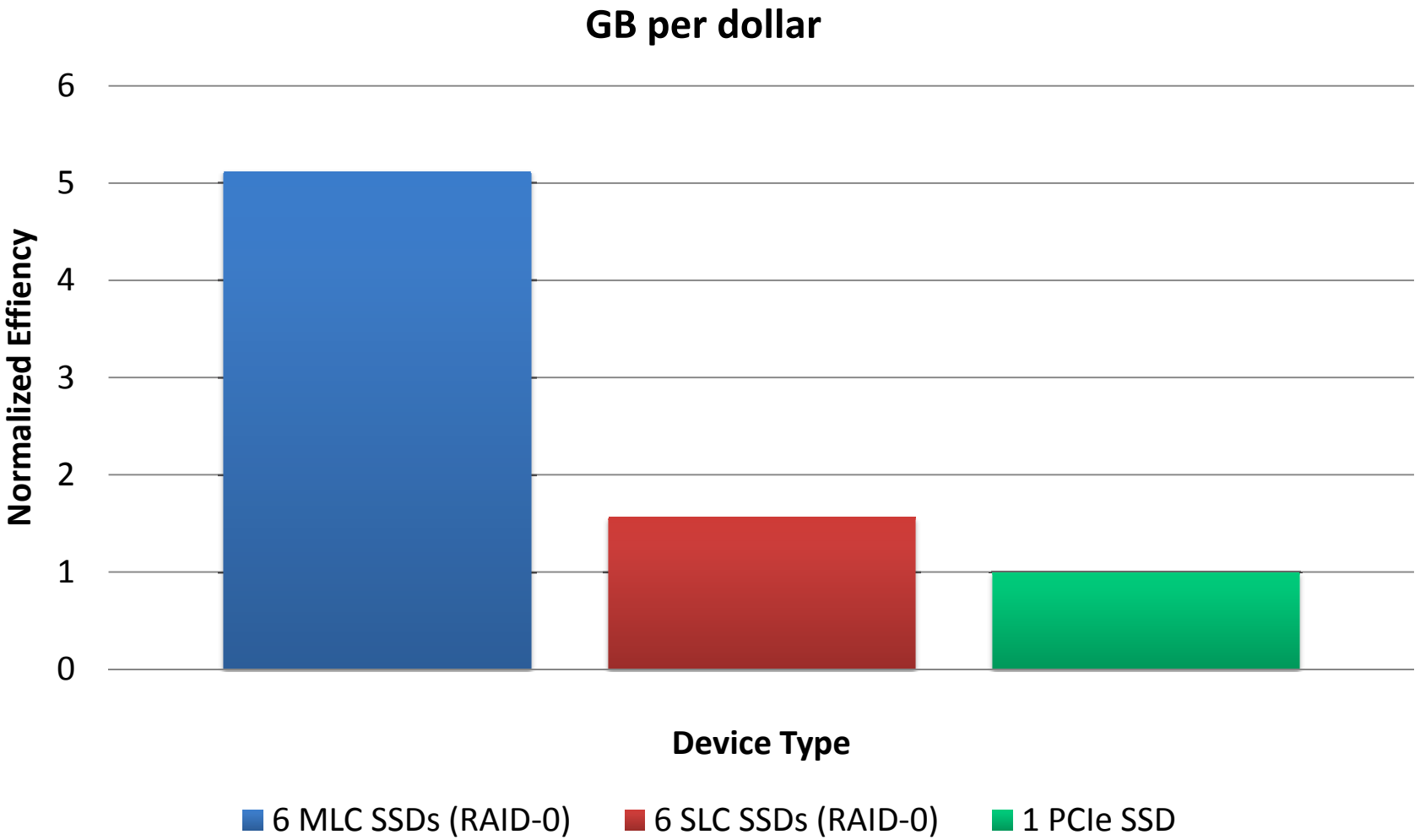
~280MB/s

Efficiency Analysis of SSD RAID

- **RAID of SSDs**
  - Configured 6 SSDs in RAID-0 using Mega RAID controller
  - Mega RAID controller is only able to connect up to 6 SSDs.
- **Cost efficiency analysis**
  - Metric (GB per $ and MB/s per $)
  - Compared RAID-0 of 6 x SATA SSDs versus 1 x PCIE SSD
- **SSDs used**

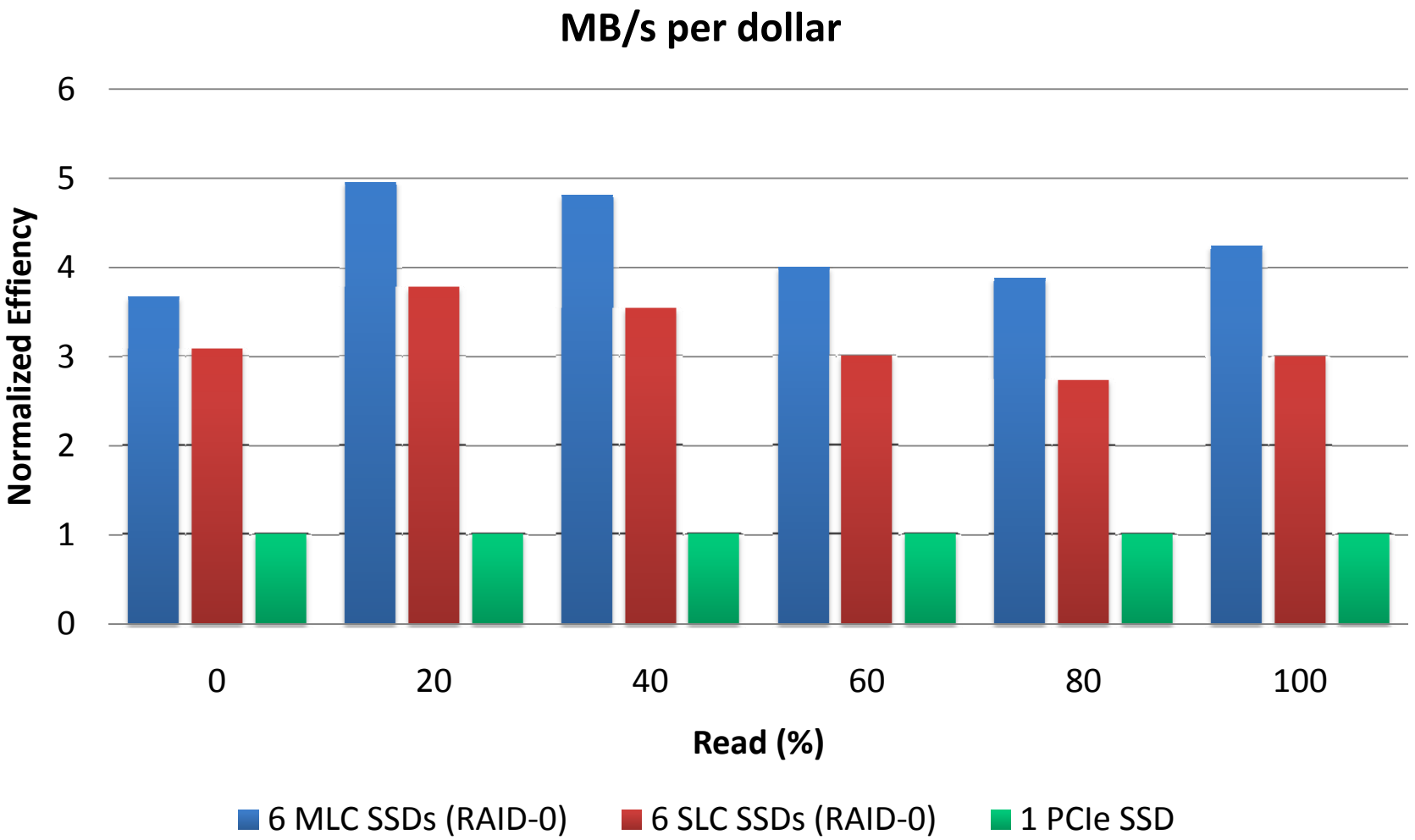
SSD Type	Specification	Size (GB)	Price ($)	MB/$
MLC SSD	Super-Talent MLC SATA II SSD	120	415	296
SLC SSD	Intel SLC SATA II SSD	64	799	82
PCIe SSD	Fusion-io ioDrive Duo MLC PCIe x8 SSD	640	13,990	46

Capacity Efficiency Analysis



- **Total cost**
  - N (RAID controller) x $ (RAID controller) + N (SSD) x $ (SSD)
  - We used $579 for PCIE LSI Mega RAID controller card.

Performance Efficiency Analysis



- **Total cost**
  - N (RAID controller) x $ (RAID controller) + N (SSD) x $ (SSD)
  - We used $579 for PCIE LSI Mega RAID controller card.

Lessons Learned

- **From the cost-efficiency analysis, we learned:**
  - RAID of SSDs is more cost-efficient than PCIE SSD in terms of capacity per dollar and bandwidth per dollar.
  - In particular, MLC based SSDs in RAID is more cost-efficient than SLC based SSDs.
- **Then what are problems and challenges in SSD RAID?**
  - Does SSD RAID offer sustainable bandwidth?
  - If not, why not? Any solution?

RAID of SSDs?

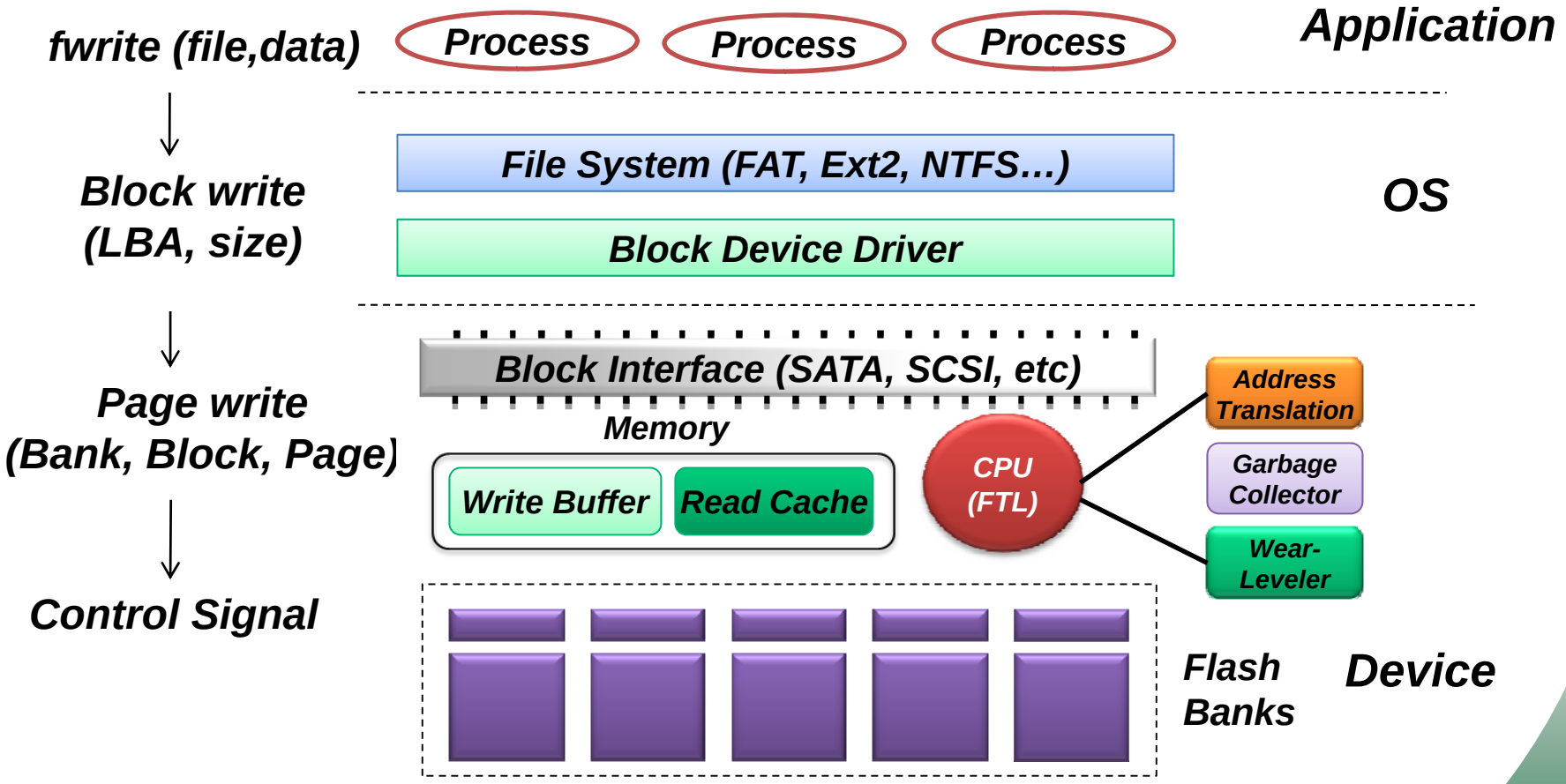
- **Problems**
  - Overall bandwidth of RAID of SSD is dependent on the slowest SSD.
  - GC process of each SSD in RAID of SSDs is is not globally coordinated.

- **Challenges**
  - There is no functional support for coordinating individual GC processes at the conventional RAID controller.
  - We need to develop a mechanism for RAID controller to be able to coordinate individual GC processes in RAID of SSDs.

- **Idea and Solution**
  - Harmonia!
  - A Coordinated Garbage Collector for RIAD of SSDs

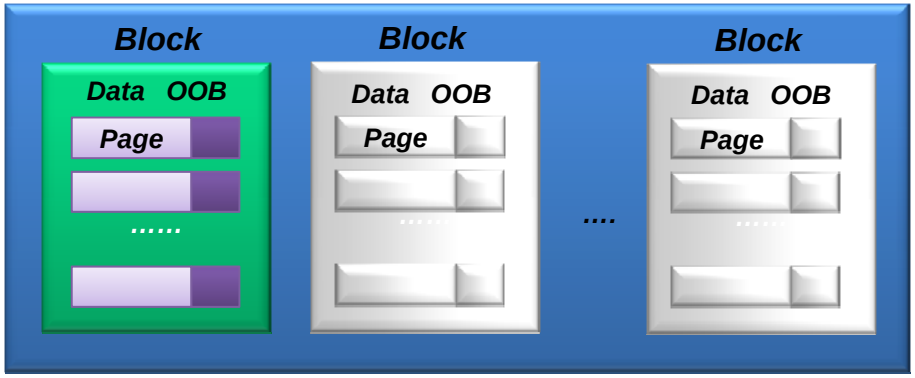
NAND Flash based SSD

System Architecture



Basics of NAND Flash Memory

NAND Flash Chip



Flash Chip Block Diagram

- Block, Page (Data + OOB)
- OOB (Out-Of-Band)
  - ECC, Logical Page Address, State of page (valid, invalid, free)

Three Operations – Read, Write, Erase

- Reads and writes are done at the granularity of a ***PAGE***
- Erases are done at the granularity of a ***BLOCK***

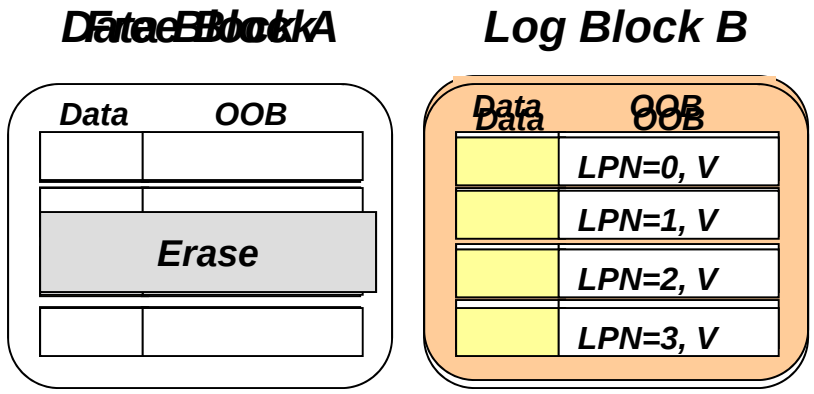
Flash	Size			Access Time		
	Page (Data)	Page (OOB)	Block	Page Read	Page Write	Block Erase
Large Block	2KB	64B	(128 + 4)KB	130.9 us	405.9 us	2 ms

Out-of-place update operation (vs. In-place update)

- is more efficient than in-place update operation, however needs to collect garbage (invalid pages)

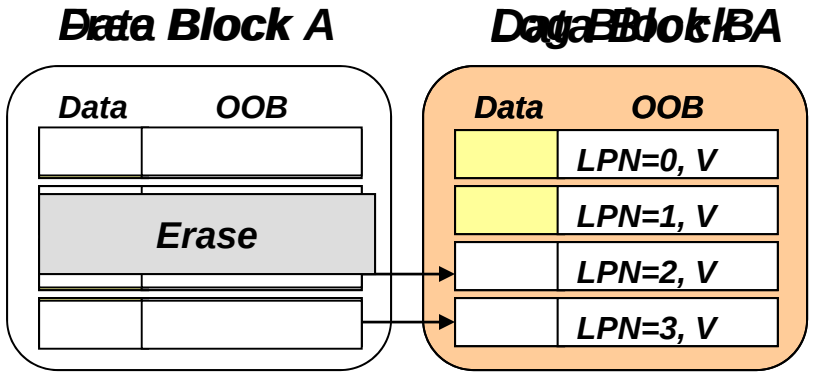
Merge Operations in Garbage Collection

Switch and Partial Merge Operations



Switch Merge

Overhead = 1 Block Erase



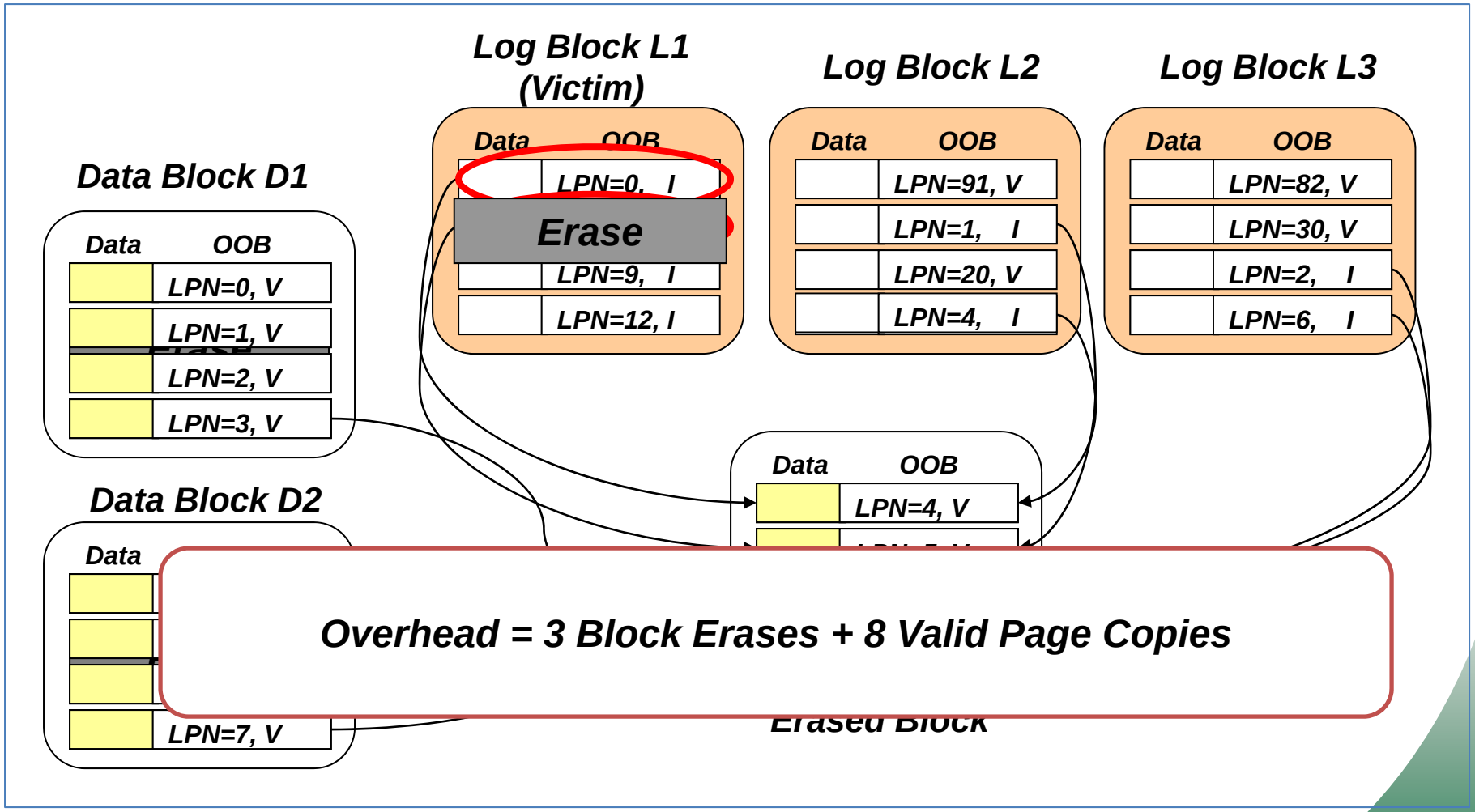
Partial Merge

Overhead = 1 Block Erase + 2 Valid Page Copies

15

Problem: Expensive Full Merge Operation

Full Merge Operation



16

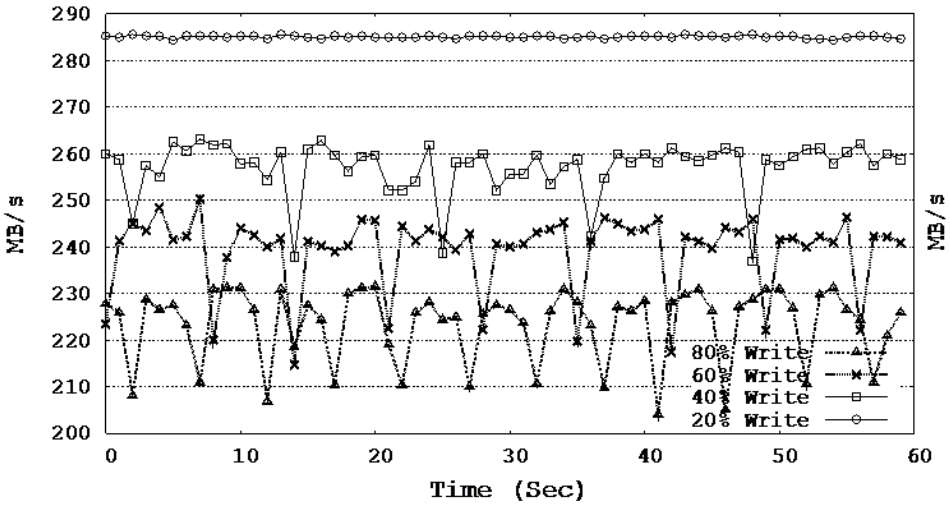
Pathological Behavior of SSDs

- **Does GC have an impact on the foreground operations?**
  - If so, we can observe sudden bandwidth drop
    - More drop with more write requests
    - More drop with more bursty workloads

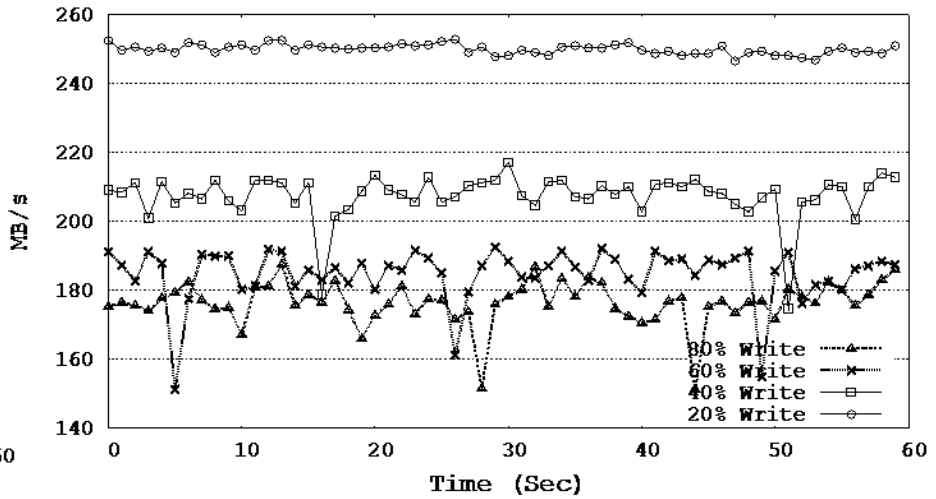
- **Experimental Setup**
  - SSD devices
    - Intel (SLC) 64GB SSD
    - SuperTalent (MLC) 120GB SSD
  - I/O generator
    - Used *libaio* asynchronous I/O library for block-level testing

Bandwidth Drop for Write-Dominant Workloads

- **Experiments**
  - Measured bandwidth for 1MB by varying read-write ratio



Intel SLC (SSD)

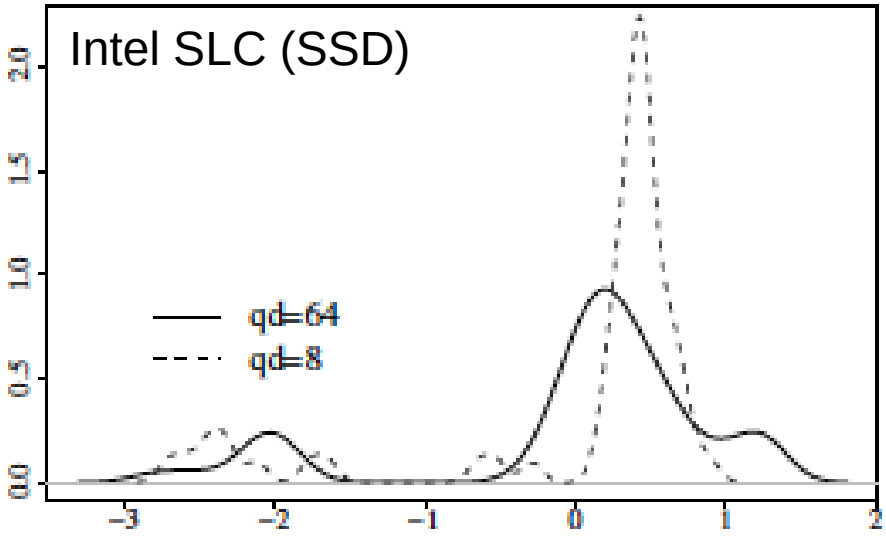


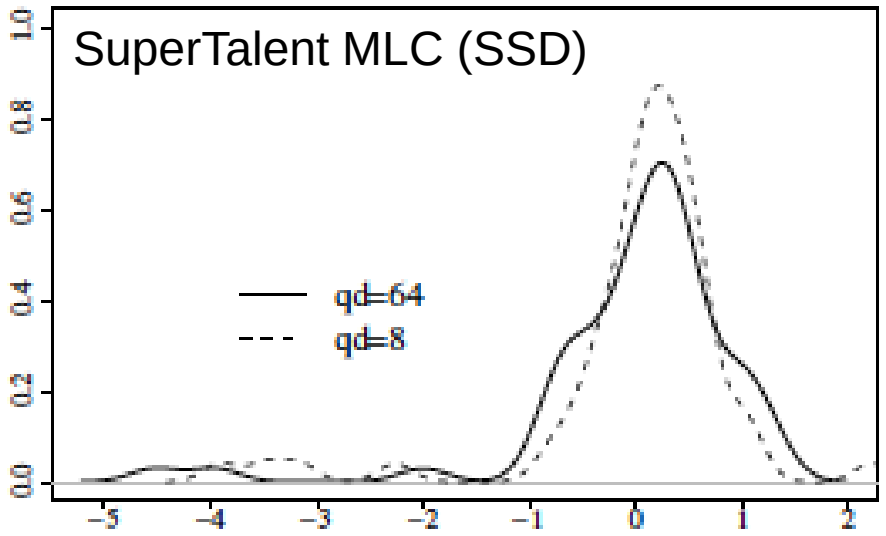
SuperTalent MLC (SSD)

Performance variability increases as we increase write-percentage of workloads.

Performance Variability for Bursty Workloads

- **Experiments**
  - Measured SSD write bandwidth for queue depth (qd) is 8 and 64
  - Normalized I/O bandwidth with a Z distribution





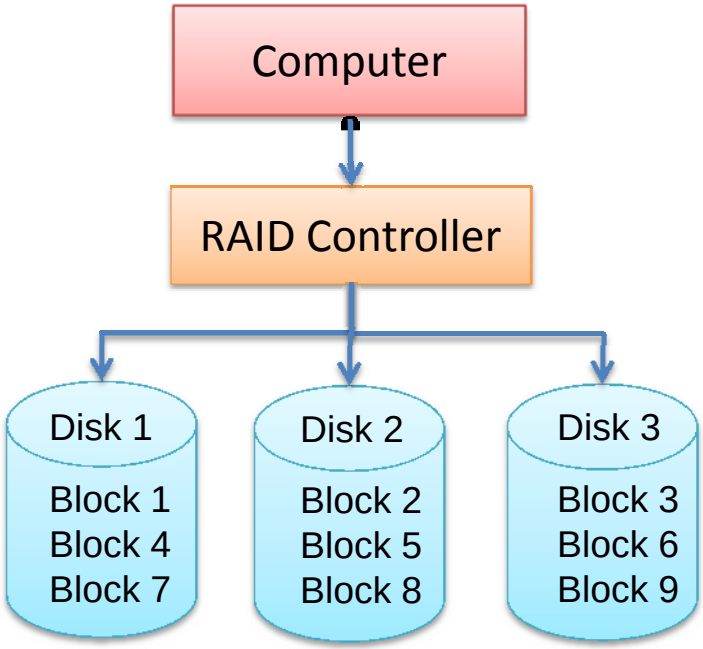
Performance variability increases as we increase the arrival-rate of requests (bursty workloads).

Lessons Learned

- **From the empirical study, we learned:**
  - Performance variability increases as the percentage of writes in workloads increases.
  - Performance variability increases with respect to the arrival rate of write requests.
- **What about the performance variability of RAID of SSDs?**
  - Does it become worse for arrays of SSDs than for individual SSDs?
  - If so, what is the main cause?

Pathological Behavior of RAID of SSDs

- **Does uncoordinated GCs prevent bandwidth improvement?**
  - If so, should we be able to observe higher variability for RAID of SSDs than that for single SSDs?

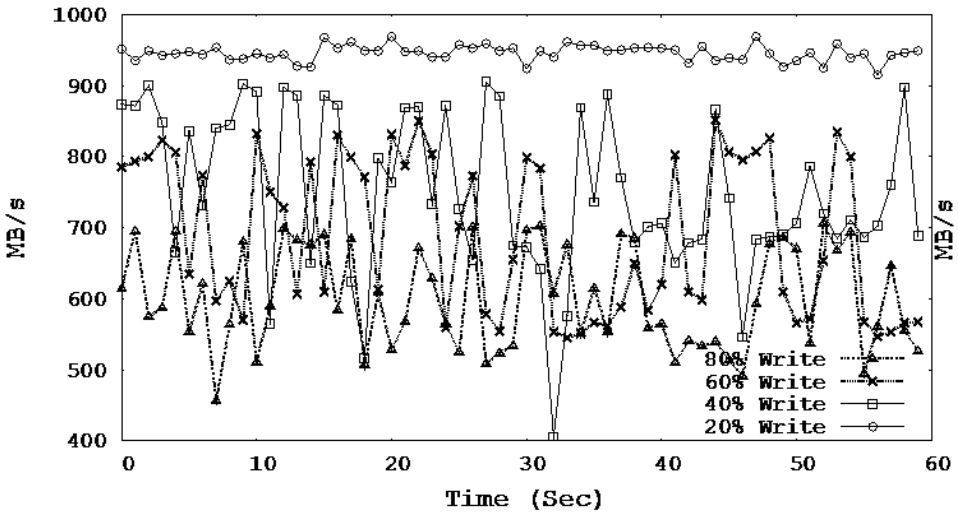


Block Striping

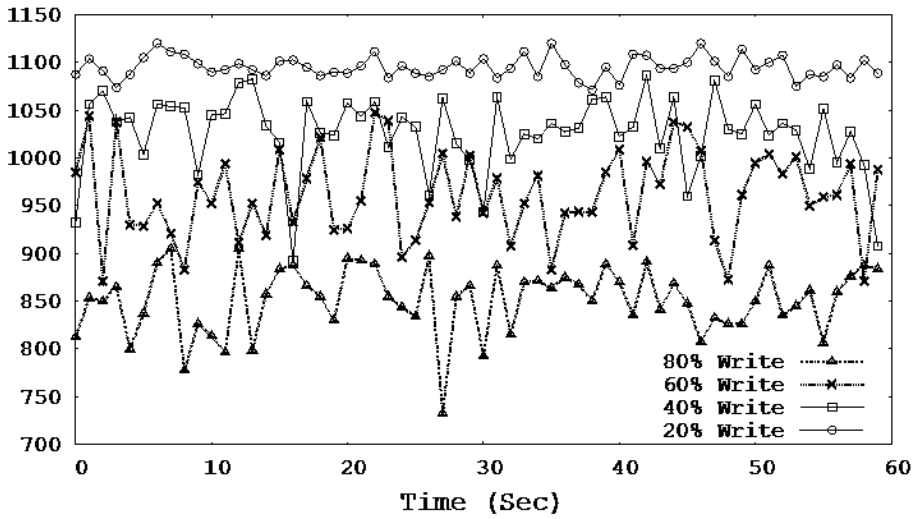
- **Experimental Setup**
  - RAID configuration
    - RAID-0 using 6 SSDs (striping)
  - SSD devices
    - Intel (SLC) 64GB SSD
    - SuperTalent (MLC) 120GB SSD
  - I/O generator
    - Used *libaio* asynchronous I/O library for block-level testing

Bandwidth Drop for Write-Dominant Workloads

- **Experiments**
  - Measured bandwidth for 1.87 MB by varying read-write ratio (qd=64)



RAID-0 of MLC SSDs



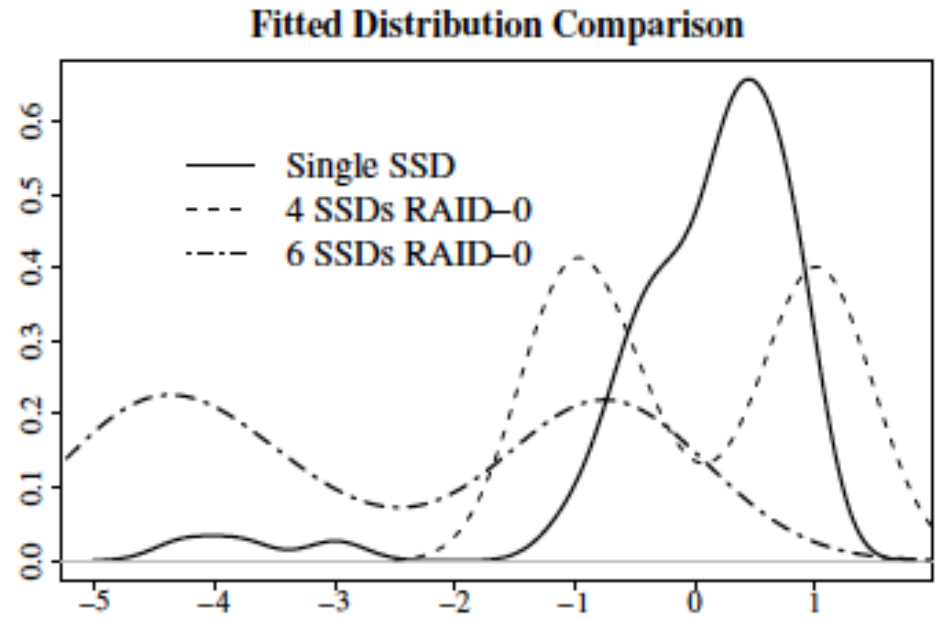
RAID-0 of SLC SSDs

Performance variability increases as we increase write-percentage of workloads.

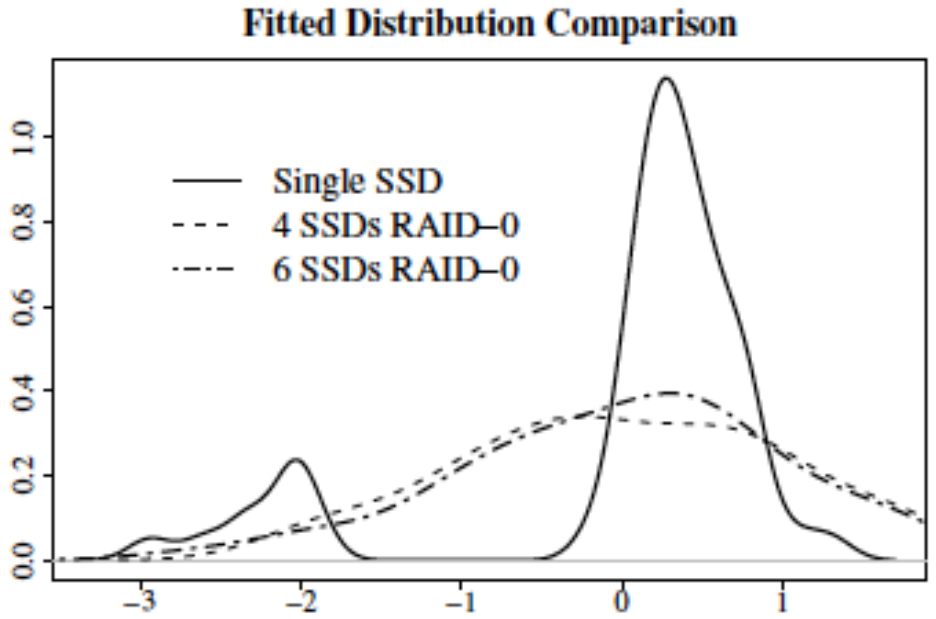
SSD RAIDs go crazy!!!

Performance Variability for Bursty Workloads

- **Experiments**
  - Measured bandwidth for queue depth=64, 60% writes of workloads
  - Normalized I/O bandwidth with a Z distribution



RAID-0 of MLC SSDs

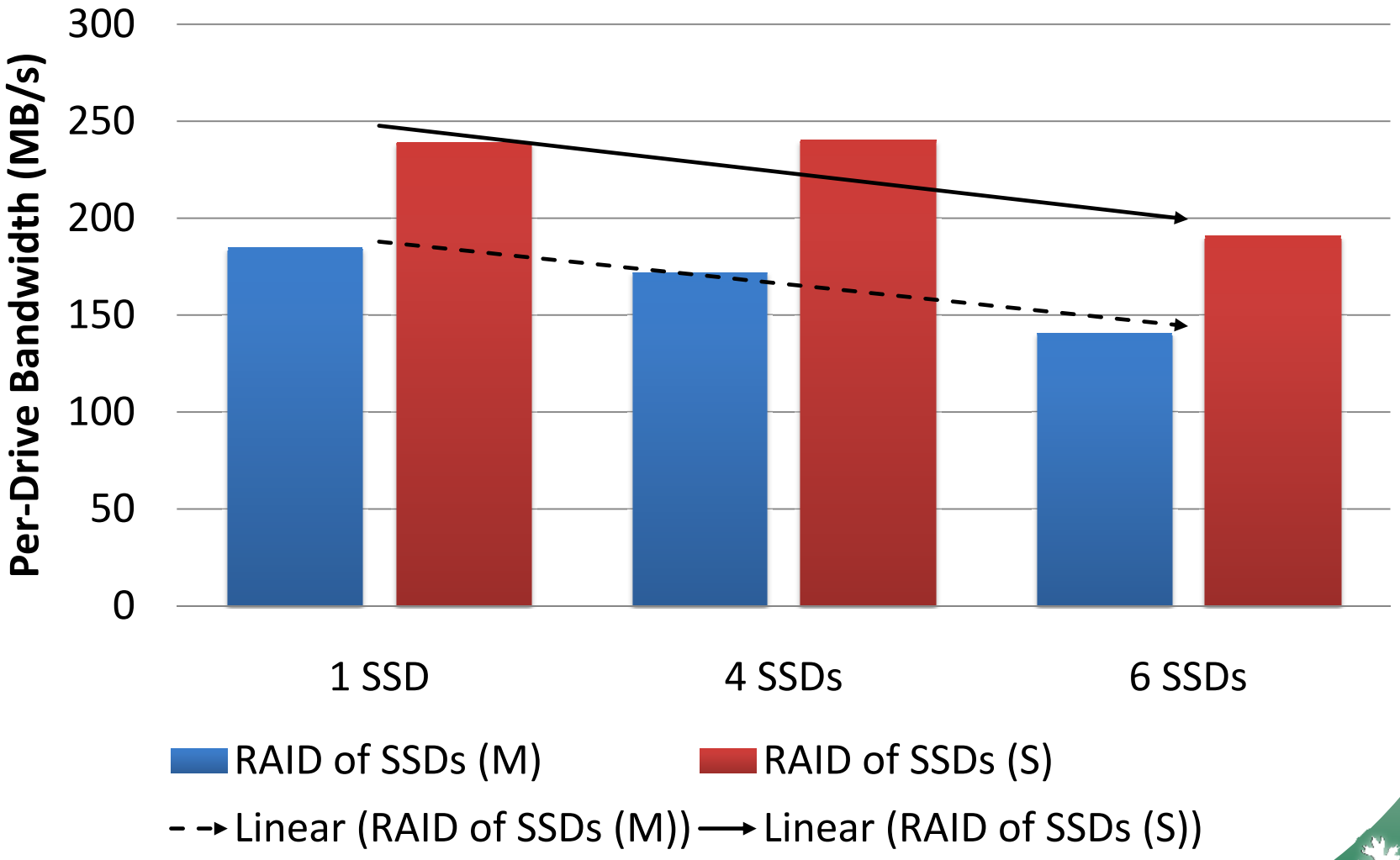


RAID-0 of SLC SSDs

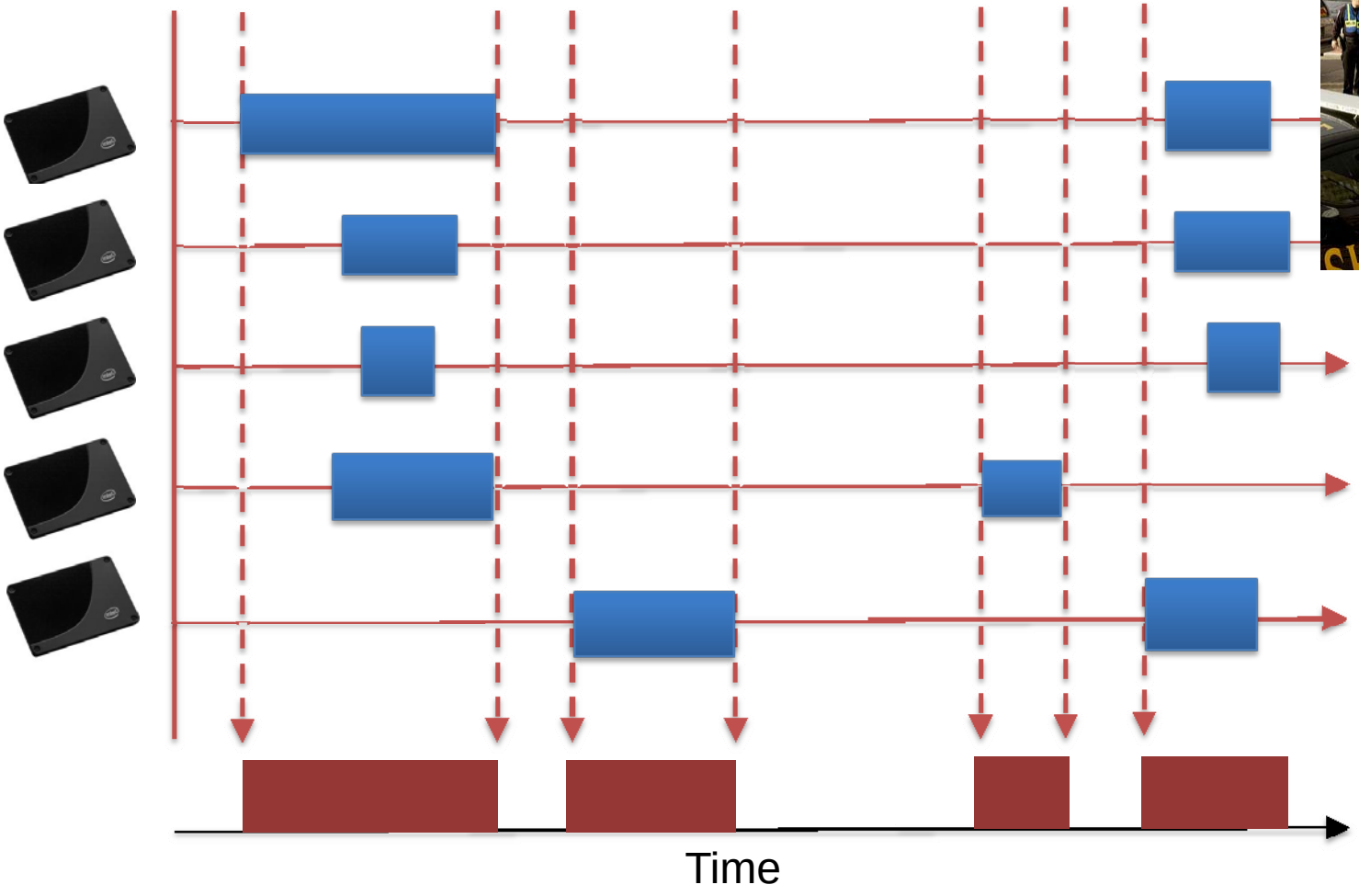
Performance variability increases as we increases the number of participant SSDs in RAID array.

Performance Variability (Cont')

- **Per-Drive Bandwidth (MB/s per drive)**



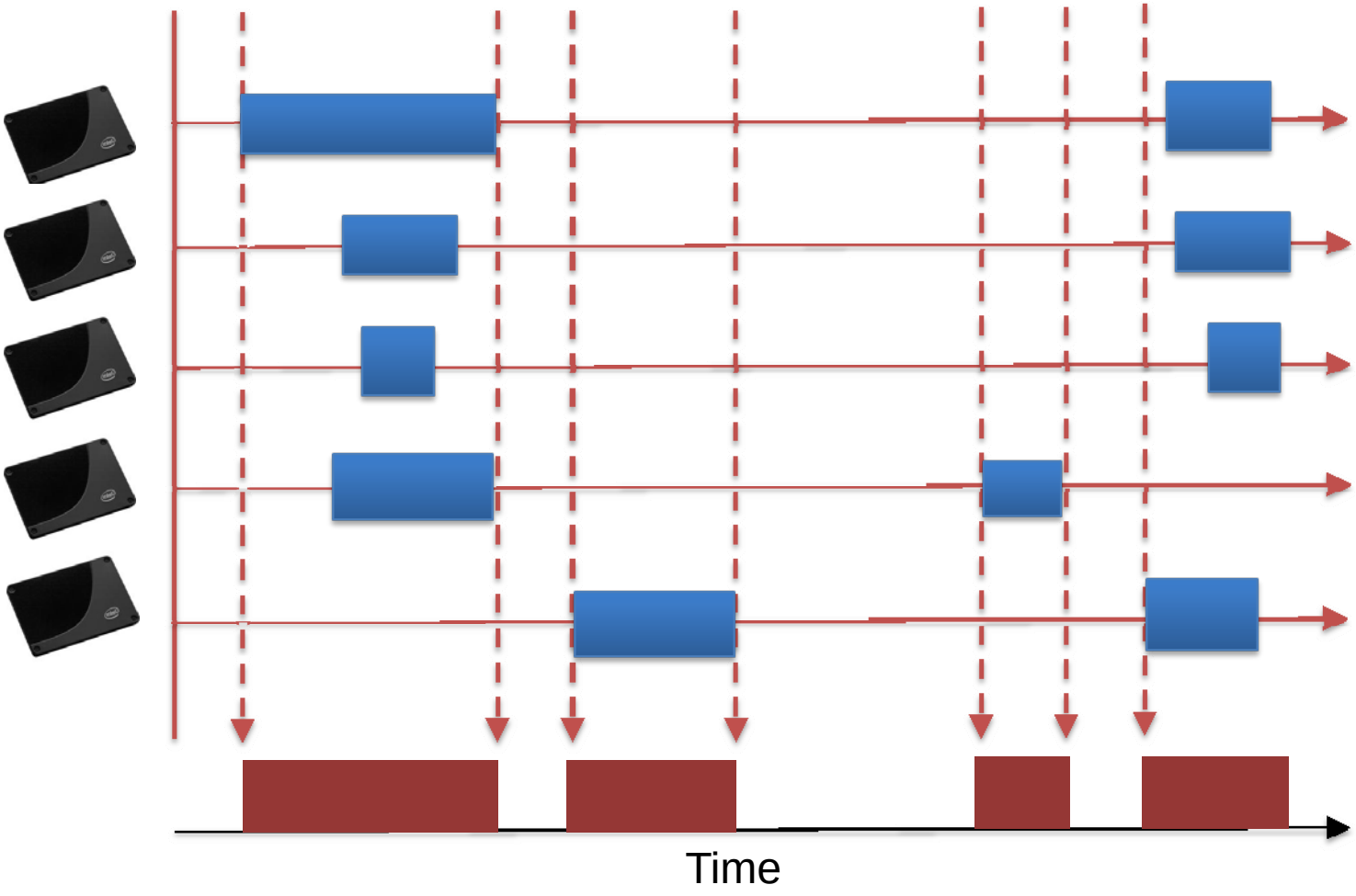
Uncoordinated Garbage Collectors



Local GC process

Aggregate degraded performance

A Globally Coordinated Garbage Collector



Design

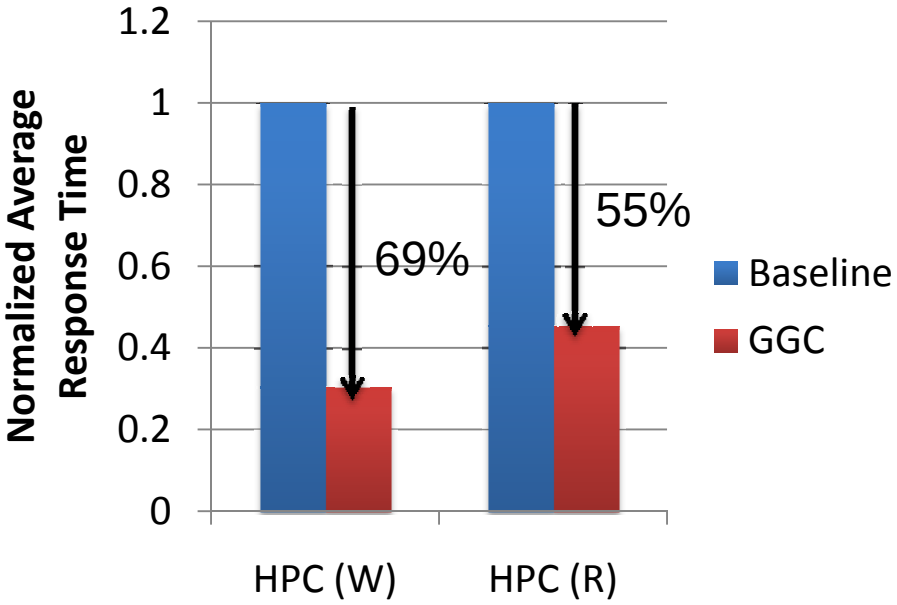
- **SSD optimized RAID controller (O-RAID)**
  - A RAID controller designed to enable global coordination of garbage collection when used with SSDs supporting that capability.
- **Global GC optimized SSD**
  - An SSD designed for participating in a globally coordinated garbage collection process in an O-RAID.
- **GC coordination algorithms**
  - A set of algorithms to perform a globally coordinated GC process on a given SSD-based RAID set comprised of an O-RAID and multiple O-SSD devices.
    - Reactive method vs. Proactive method
- **Extension of storage protocols**
  - Extension of storage protocols such as SATA and SCSI for controlling the additional capabilities of O-SSD device.

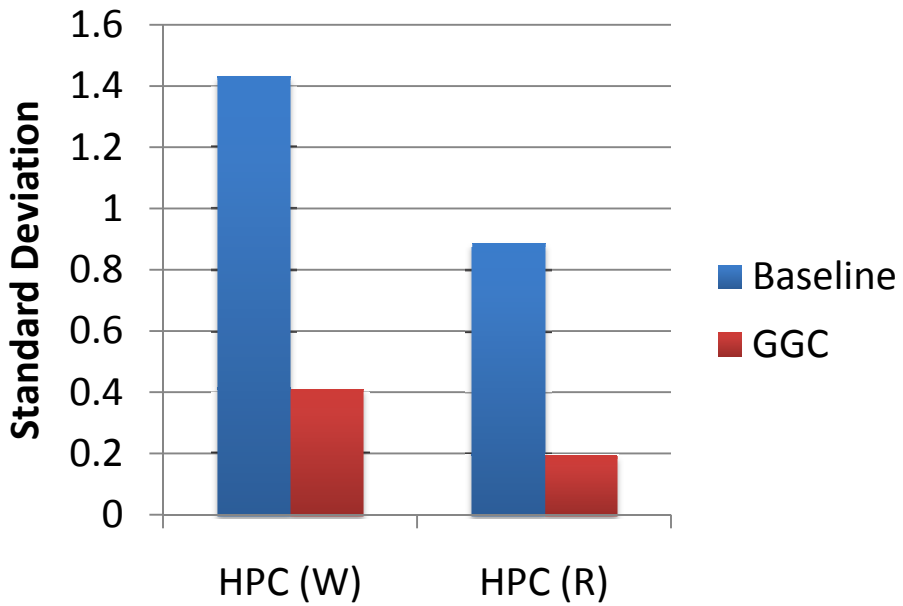
Experimental Setups

- **Simulator**
  - Microsoft Research' SSD simulator based on DiskSim
  - Configured RAID-0 of 8 32GB SSDs using 4KB Stripe unit size
- **Workloads**
  - HPC-like Synthetic workloads
    - Used the synthetic workload generator in DiskSim
    - HPC (W): 80% Writes, HPC (R): 80% Reads
  - Enterprise-scale Realistic workloads

	Workloads	Average request size (KB)	Read ratio (%)	Arrival rate (IOP/s)
Write dominant	Financial	7.09	18.92	47.19
	Cello	7.06	19.63	74.24
Read dominant	TPC-H	31.62	91.80	172.73
	OpenMail	9.49	63.30	846.62

Results for HPC-like Synthetic Workloads

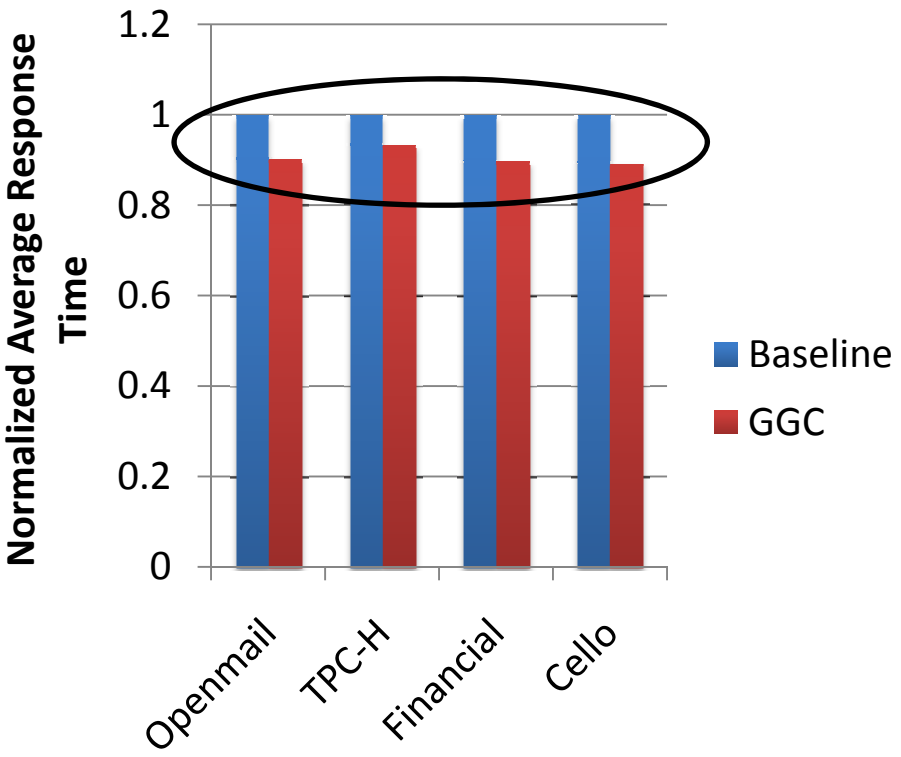


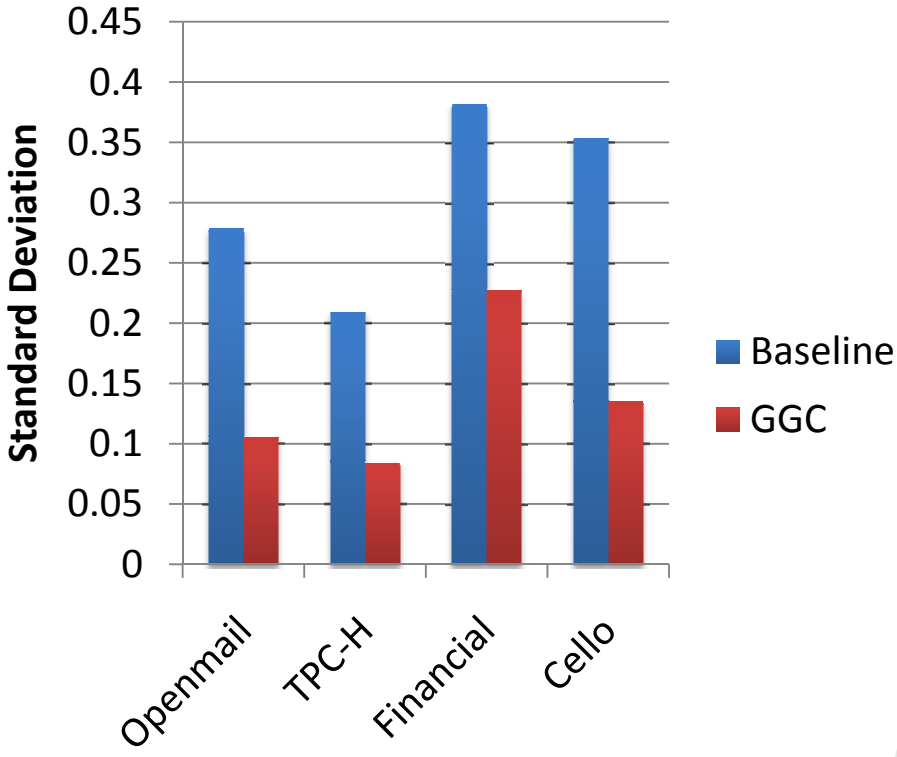


Response time improvements are 69% and 55% for HPC(W) and HPC(R) workloads respectively.

Significant improvement on standard deviations by GGC

Results for Realistic Workloads





Performance improvement is about 10%.

Standard deviation significantly improves by GGC.

Conclusions

- **Empirical experiments using real SSDs**
  - We showed that RAIDs of SSDs exhibit high performance variability due to uncoordinated GC processes.
- **Harmonia: A coordinated garbage collector**
  - We proposed *Harmonia, a global garbage collector,* that coordinates the local GC process of the individual SSDs.
- **Results**
  - We showed that for bursty workloads dominated by large writes, a 69% improvement in response time and a 71% reduction in performance variability when compared to uncoordinated garbage collection.

Questions?

Contact info

Youngjae Kim (PhD)

kimy1@ornl.gov

Oak Ridge National Laboratory

National Center for Computational Sciences