From ARIES to MARS: Transaction Support for Next-Generation, Solid-State Drives

Joel Coburn*, Trevor Bunker*, Meir Schwarz, Rajesh Gupta, Steven Swanson

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

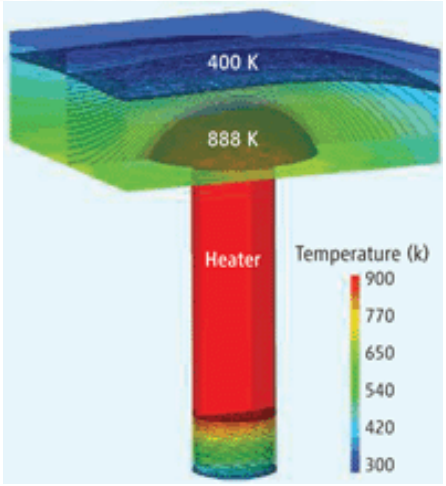
* Now at Google

NVSL
Non-volatile Systems Laboratory

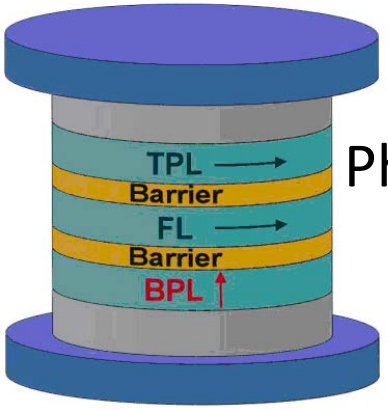
UCSDCSE
Computer Science and Engineering

Faster than Flash Non-volatile Memories

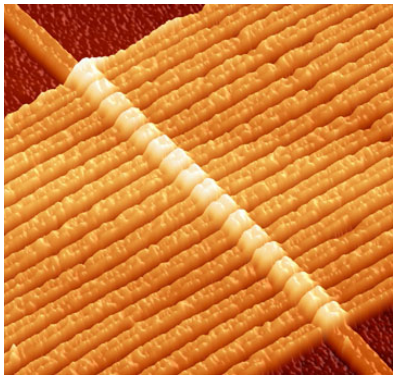
- Flash is everywhere but has its idiosyncrasies
- New device characteristics
  - Nearly as fast as DRAM
  - Nearly as dense as flash
  - Non-volatile
  - Reliable
- Applications
  - DRAM replacements
  - Fast storage



Phase change memory



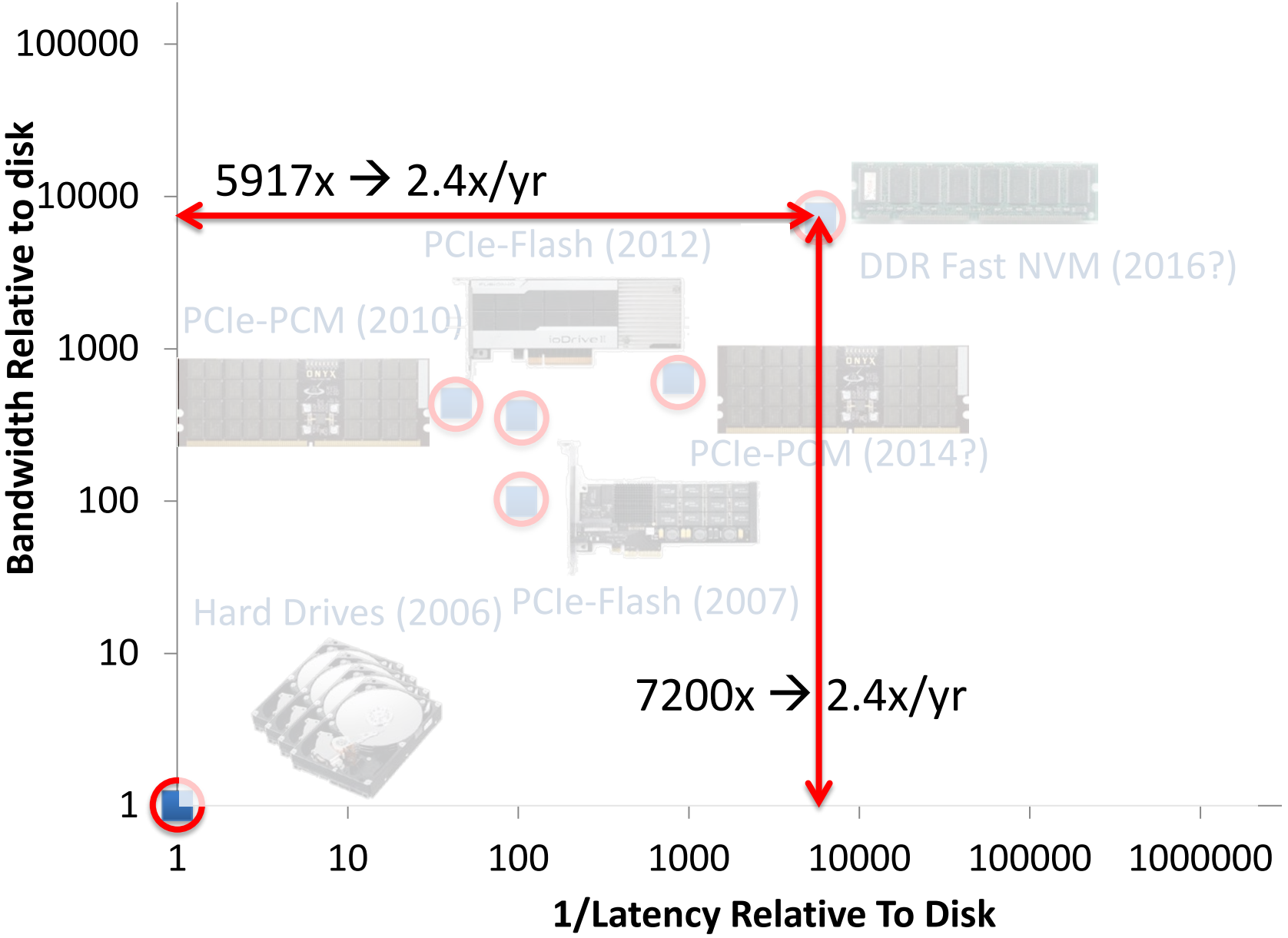
Spin-torque MRAM



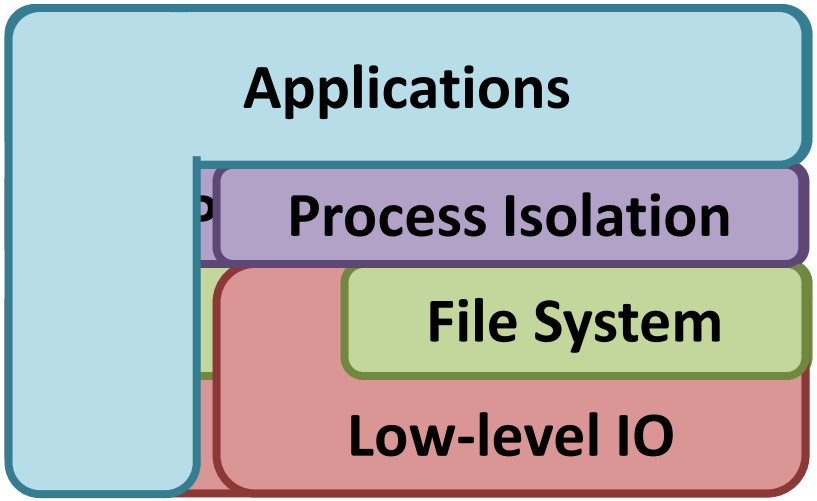
Memristor

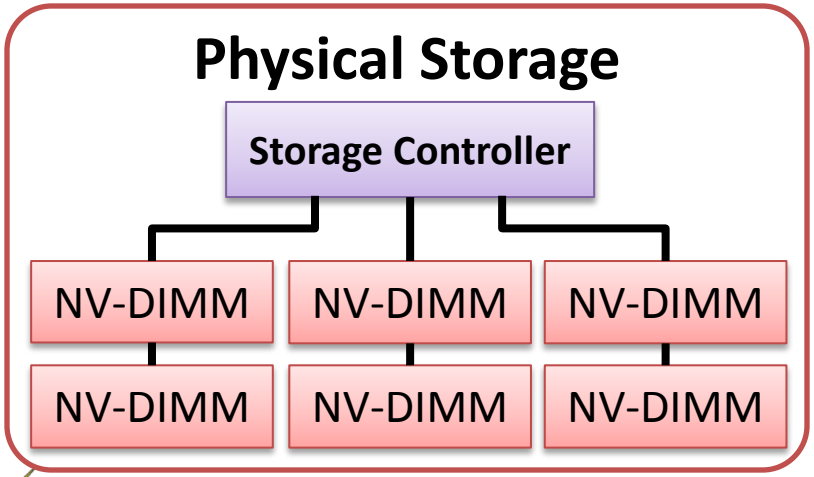
NVSL Non-volatile Systems Laboratory

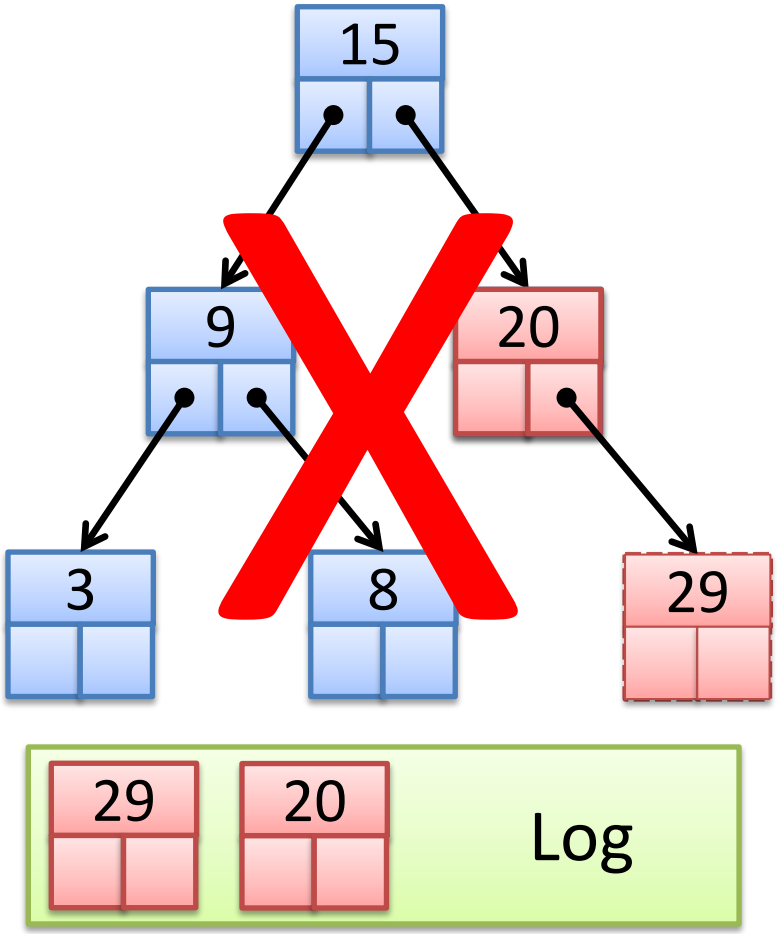
More than Moore's Law Performance



Realizing the Potential of fast NVMs



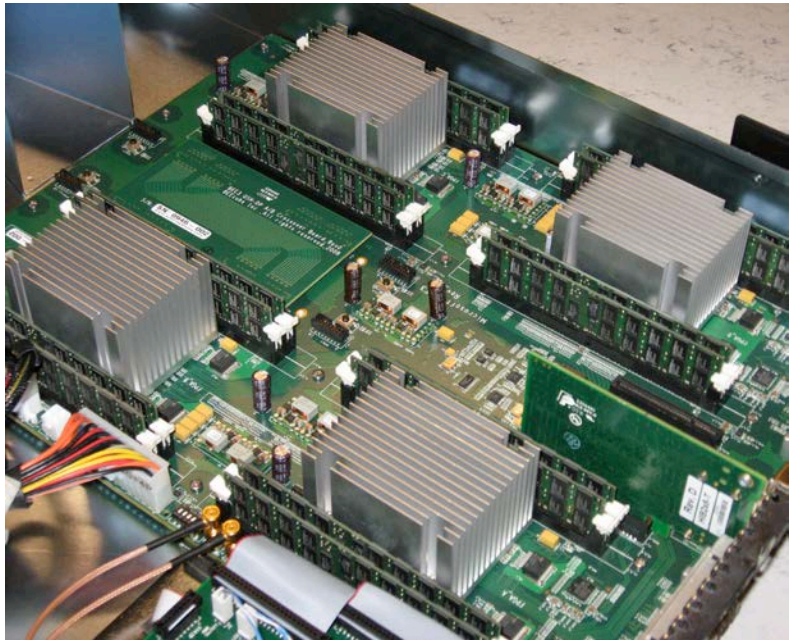




WAL algorithms were designed for disk!

Moneta-Direct SSD for Fast NVMs

- FPGA-based prototype
  - DDR2 DRAM emulates PCM
  - PCIe: 2GB/s, full duplex
- Optimized kernel driver and device interface
  - Eliminate disk-based bottlenecks in IO stack
- User-space driver
  - Eliminates OS and FS costs in the common case



[SC 2010, Micro 2010, ASPLOS 2012]

5µs latency,
1.8M IOPS for 512B requests

Characteristics of Fast SSDs

	Disk	Moneta
Latency (4KB)	7000µs	7µs
Bandwidth (4KB)	2.6MB/s	1700MB/s
Sequential/random performance	~100:1	1:1
Minimum request size/alignment	Block	Byte
Parallelism	1	64
Internal/external bandwidth	1:1	8:1

Existing Support for Transactions

- Disk-based systems
  - Write-ahead logging approaches: ARIES [TODS 92], Stasis [OSDI 06], Segment-based recovery [VLDB 09], Aether [VLDB 10]
  - Device/HW support: Logical Disk [SOSP 93], Atomic Recovery Units [ICDCS 96], Mime [HPL-TR 92]
  - Shadow paging in file systems: ZFS, WAFL
- Non-volatile main memory
  - Persistent regions: RVM [TOCS 94], Rio Vista [SOSP 97]
  - Programming support: Mnemosyne, NV-heaps [ASPLOS 11]
- Flash-based SSDs
  - Transactional Flash [OSDI 08]
  - FusionIO's AtomicWrite [HPCA 11]

ARIES: Write-Ahead Logging Recovery Algorithm for Databases

Fast, flexible, and scalable ACID transactions

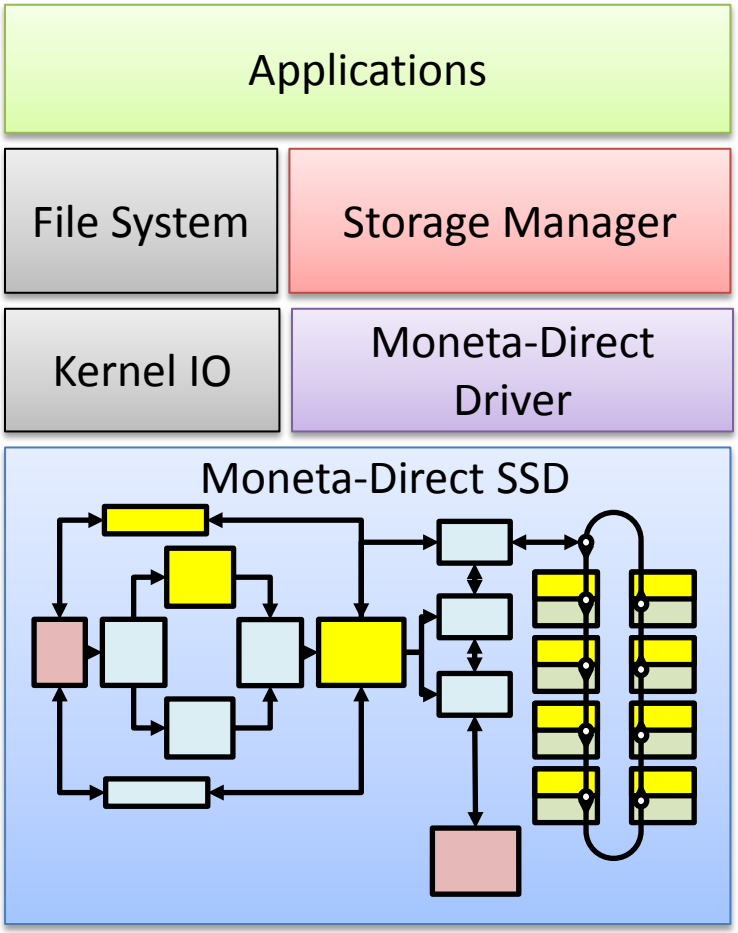
Feature	Benefit(s)
Flexible storage management	Supports varying length data and high concurrency
Fine-grained locking	High concurrency
Partial rollbacks via savepoints	Robust and efficient transactions
Recovery independence	Simple and robust recovery
Operation logging	High concurrency lock modes

ARIES Disk-Centric Design

Design Decision	Advantages	How?
No-force	Eliminate synchronous random writes	Flush redo log entries to storage on commit
Steal	Reclaim buffer space (scalability) Eliminate random writes Avoid false conflicts on pages	Write undo log entries before writing back dirty pages
Pages	Simplify recovery and buffer management Match the semantics of disk	All updates are to pages Page writes are atomic
Log Sequence Numbers (LSNs)	Simplify recovery Enable features like operation logging	LSNs provide an ordering on updates

Good for disk, not good for fast SSDs

MARS:
Modified ARIES Redesigned for SSDs



Simplified ARIES Replacement

+

Editable Atomic Writes

+

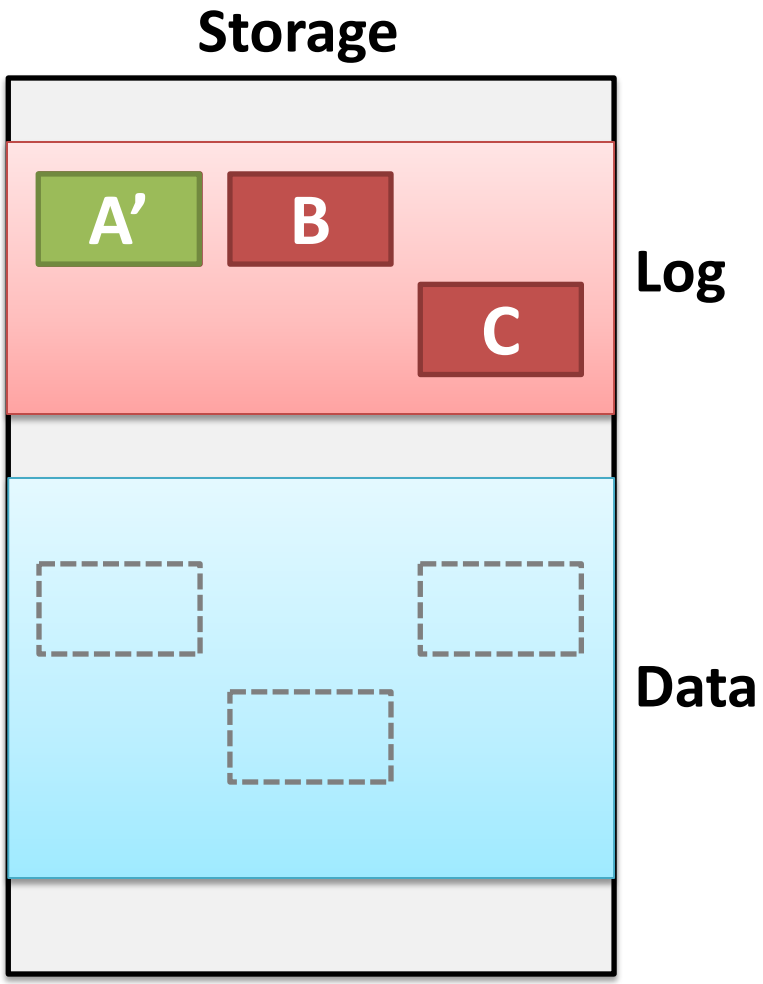
Hardware support

Editable Atomic Writes (EAWs)

```
Atomic {
  Write A
  Write B
  Write C
  …
  If(x)
    Write A'
  …
}
```

Write the log

Commit



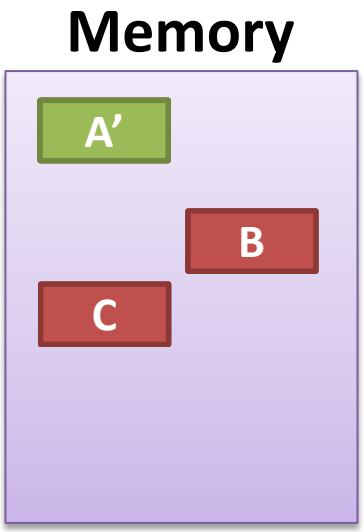
Applications can access and edit the log prior to commit.

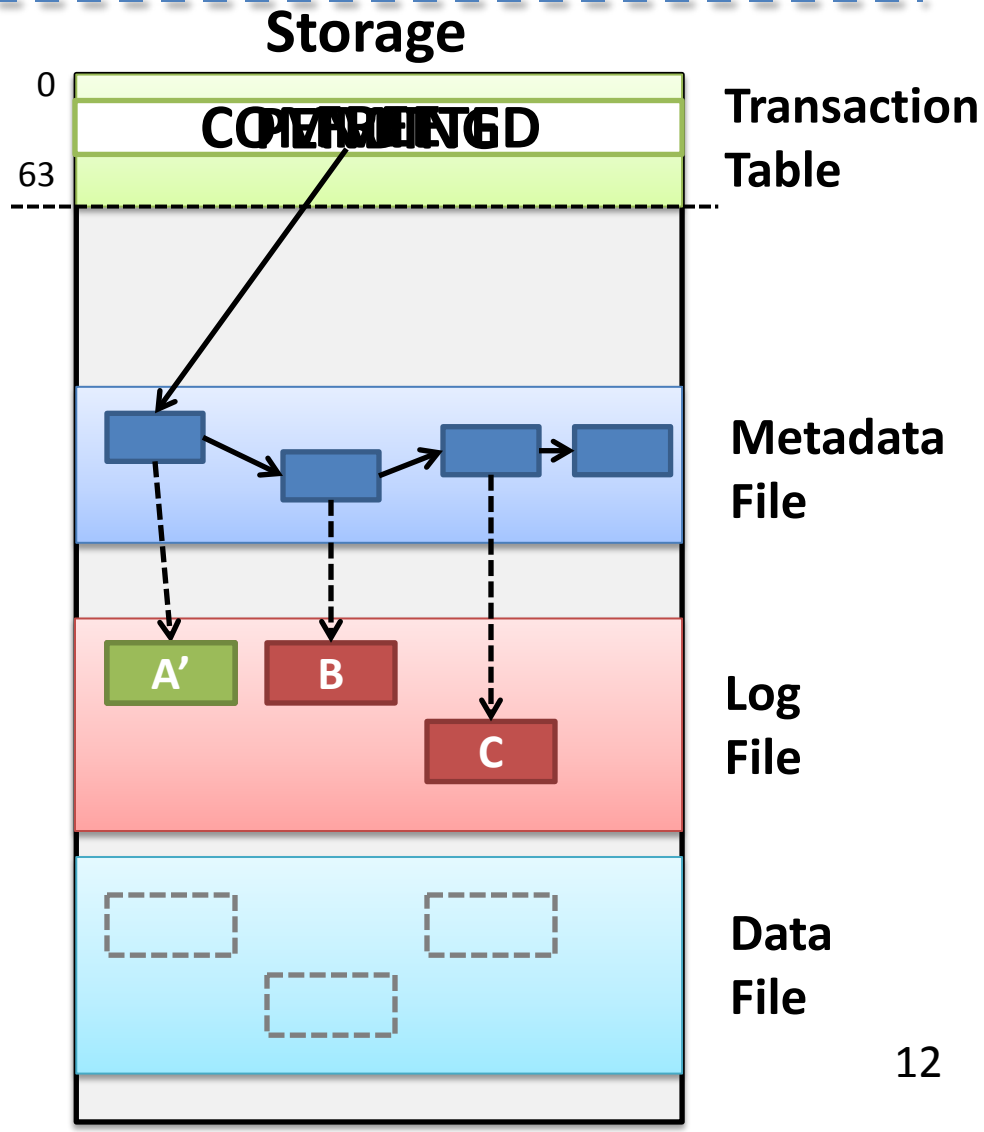
Hardware copies data in-place.

NVSL
Non-volatile Systems Laboratory

Editable Atomic Write Execution

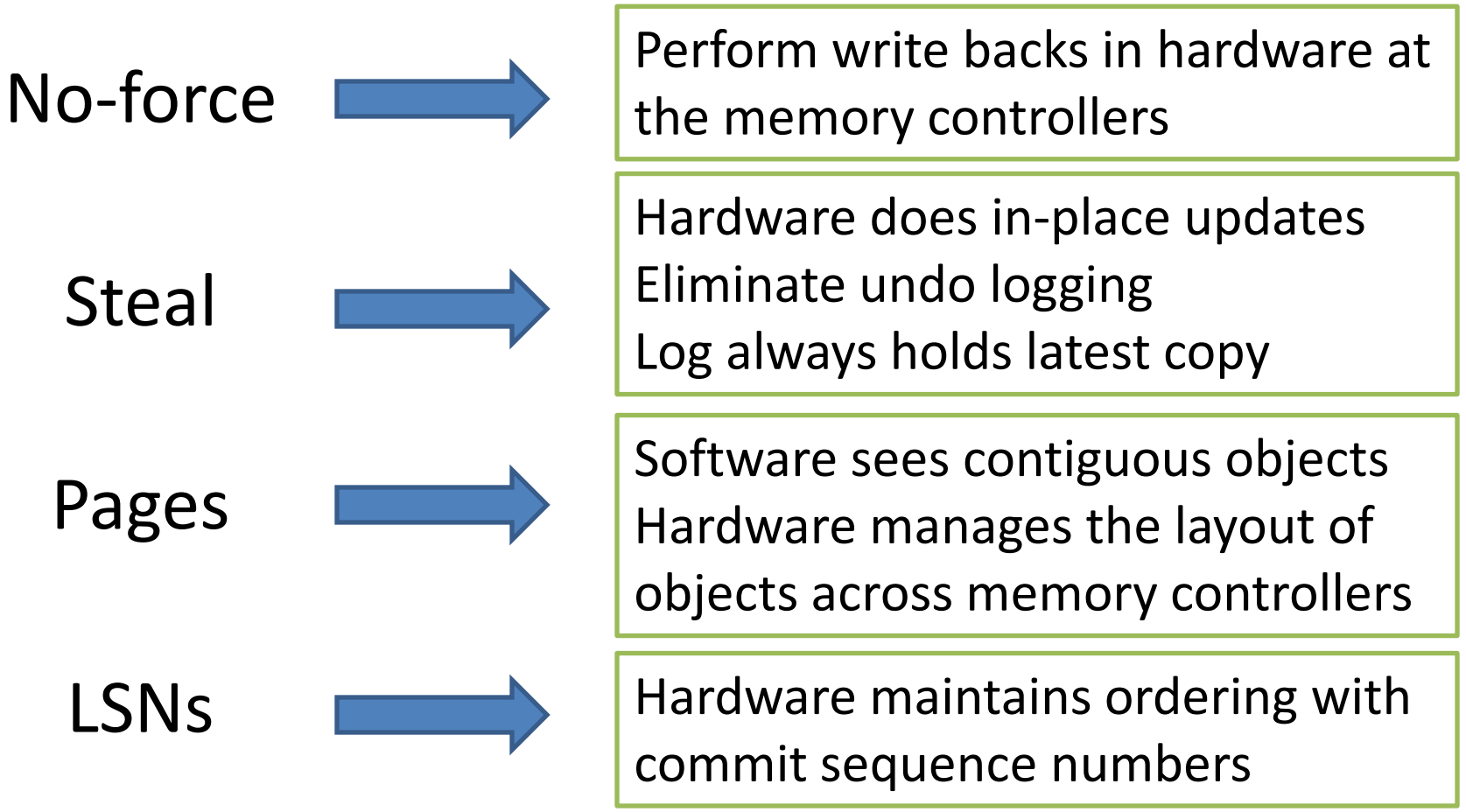
```
LogWrite(t1,memA,dataA,logA);
LogWrite(t1,memB,dataB,logB);
LogWrite(t1,memC,dataC,logC);
If(x) Write(memA,logA);
Commit(t1);
// WriteBack(t1);
```





NVSL
Non-volatile Systems Laboratory

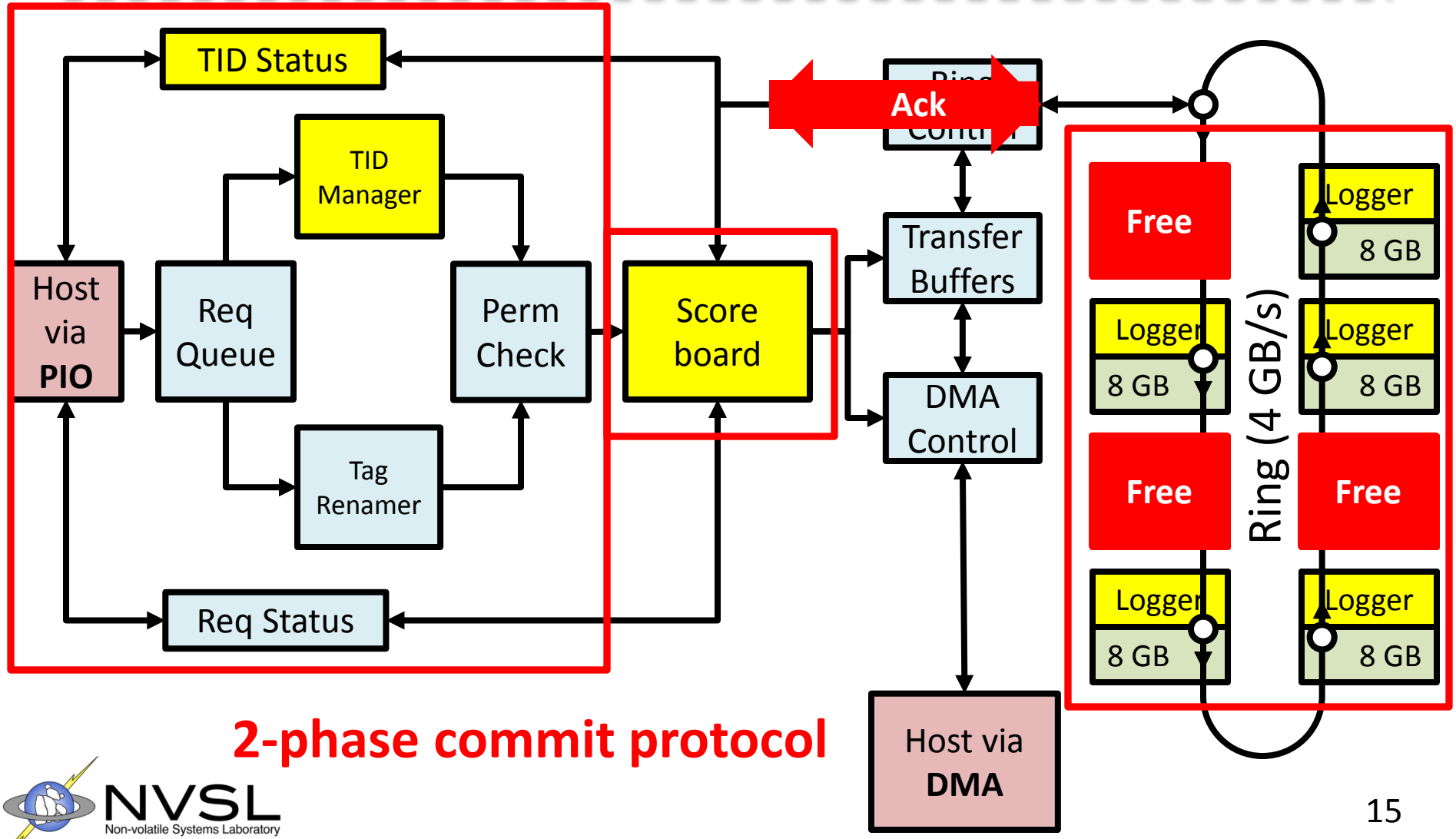
Designing MARS for Fast NVMs



MARS Features using EAWs

Feature	Provided by MARS?
Flexible storage management	✓
Fine-grained locking	✓
Partial rollbacks via savepoints	✓
Recovery independence	✓
Operation logging	N/A

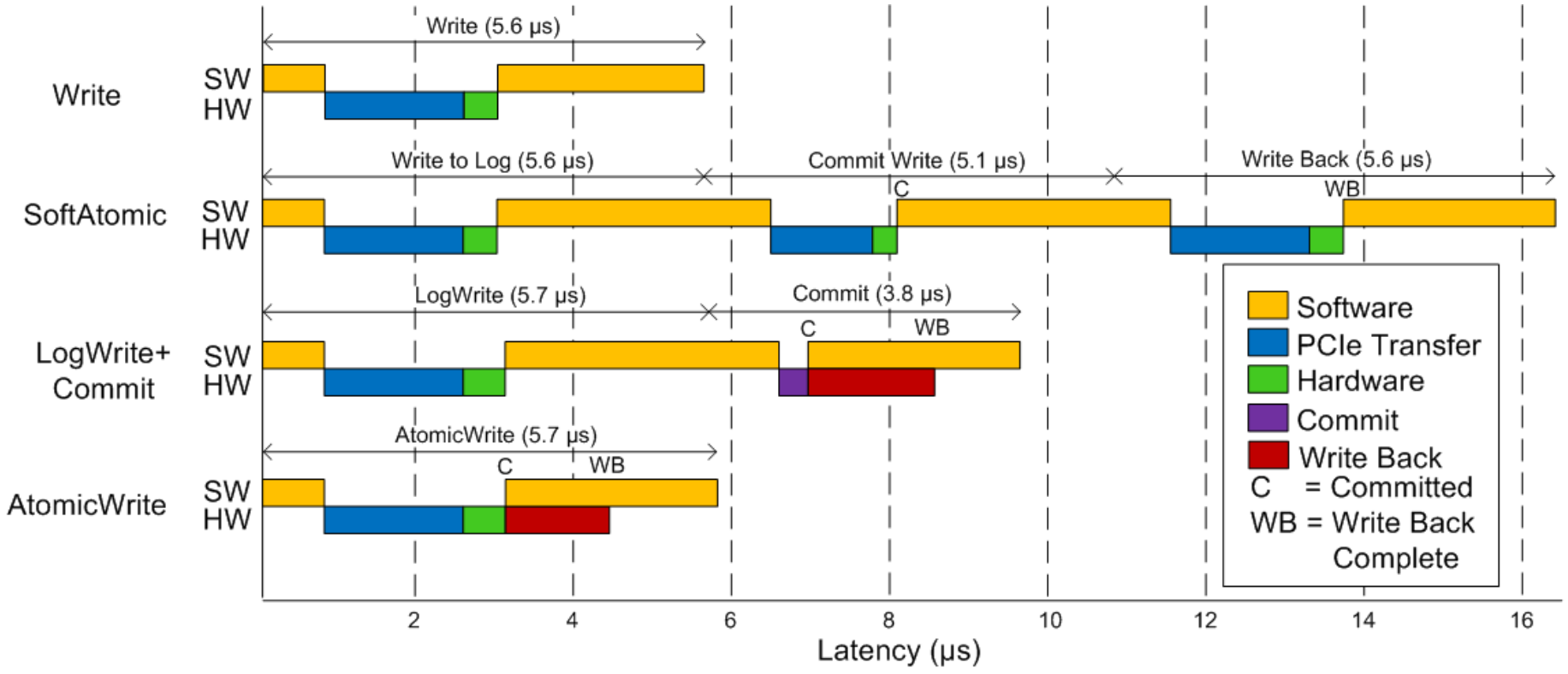
EAW Hardware Architecture



2-phase commit protocol

NVSL
Non-volatile Systems Laboratory

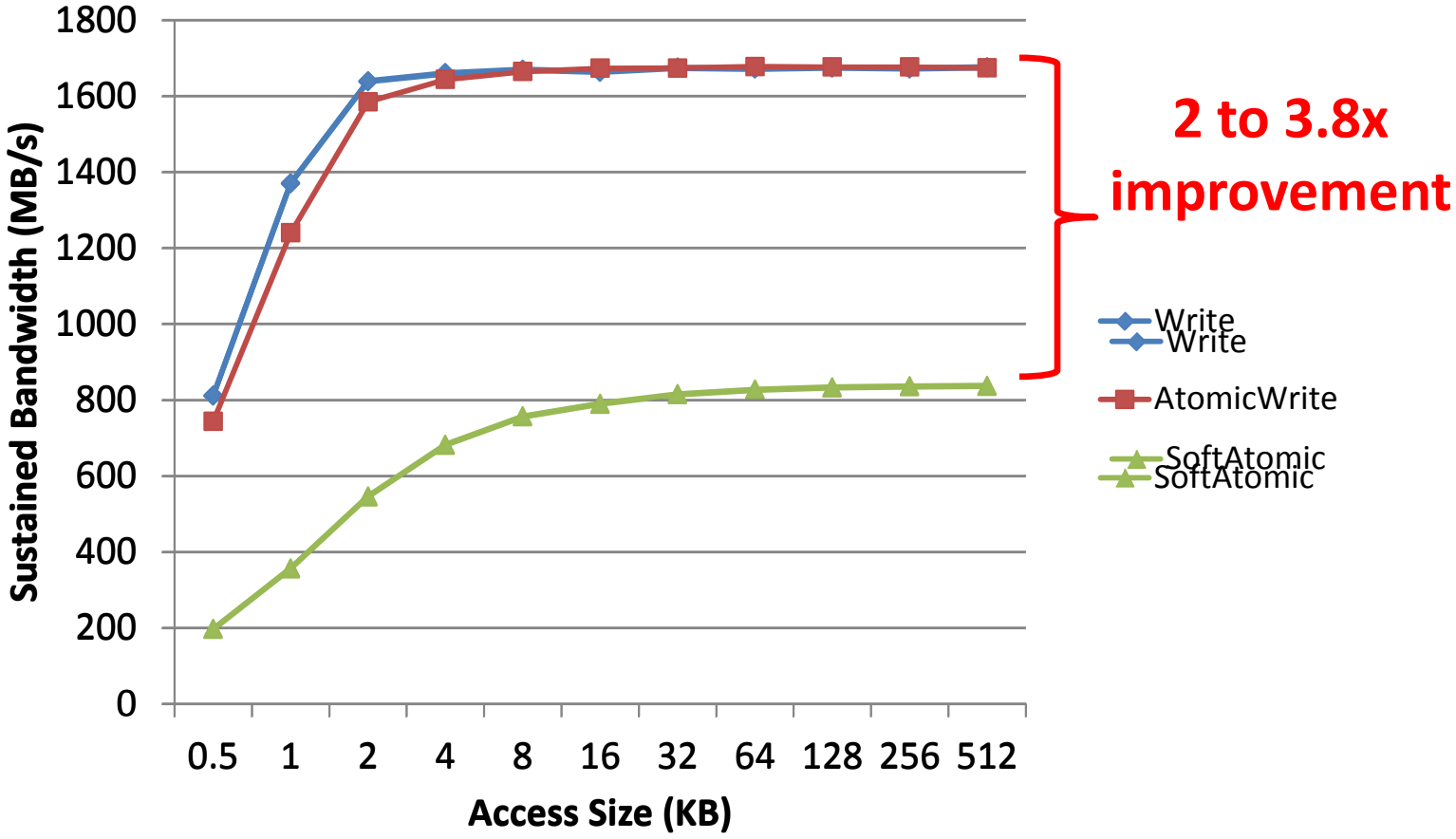
Latency Breakdown



Up to 3x faster than software only

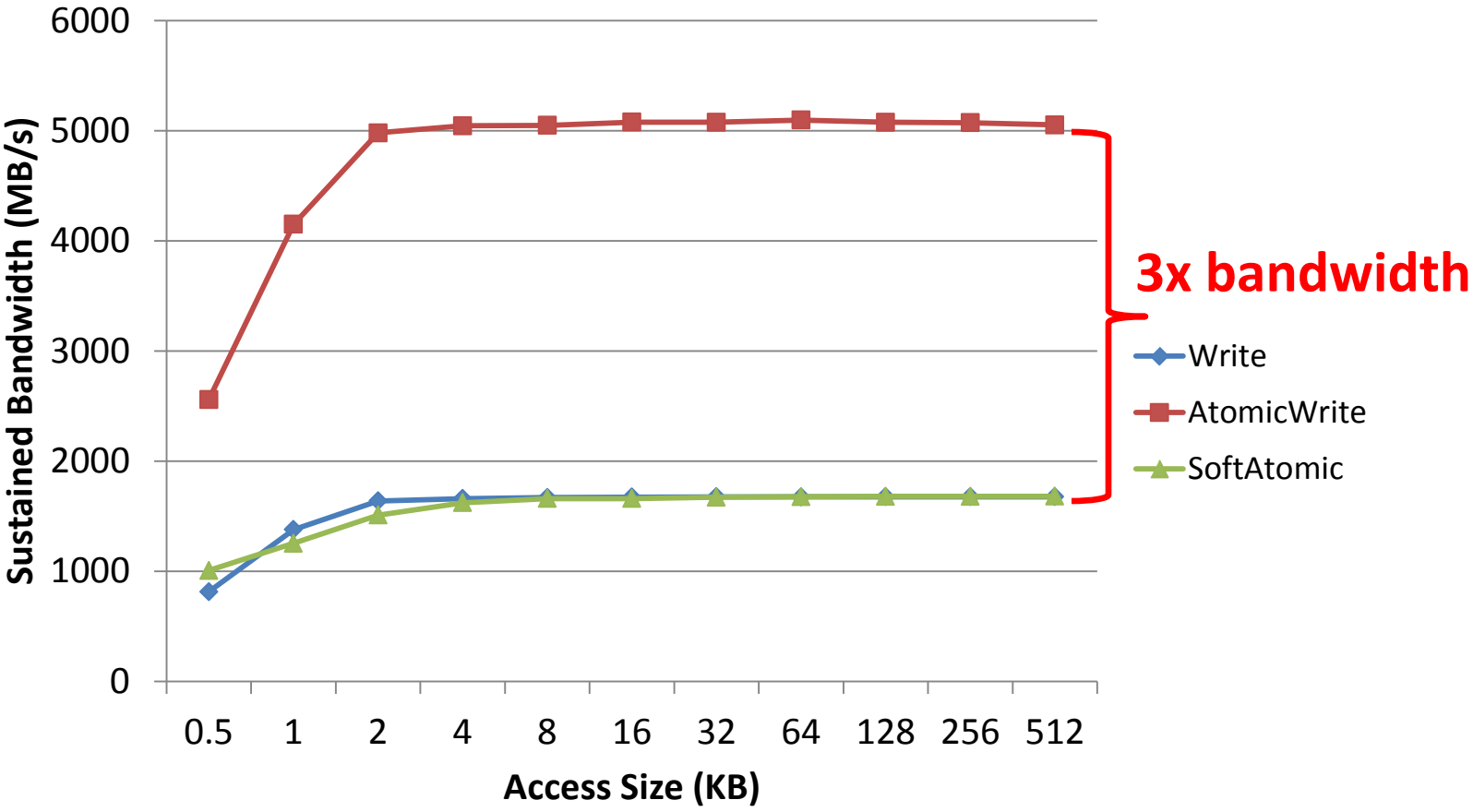
NVSL
Non-volatile Systems Laboratory

Bandwidth Comparison



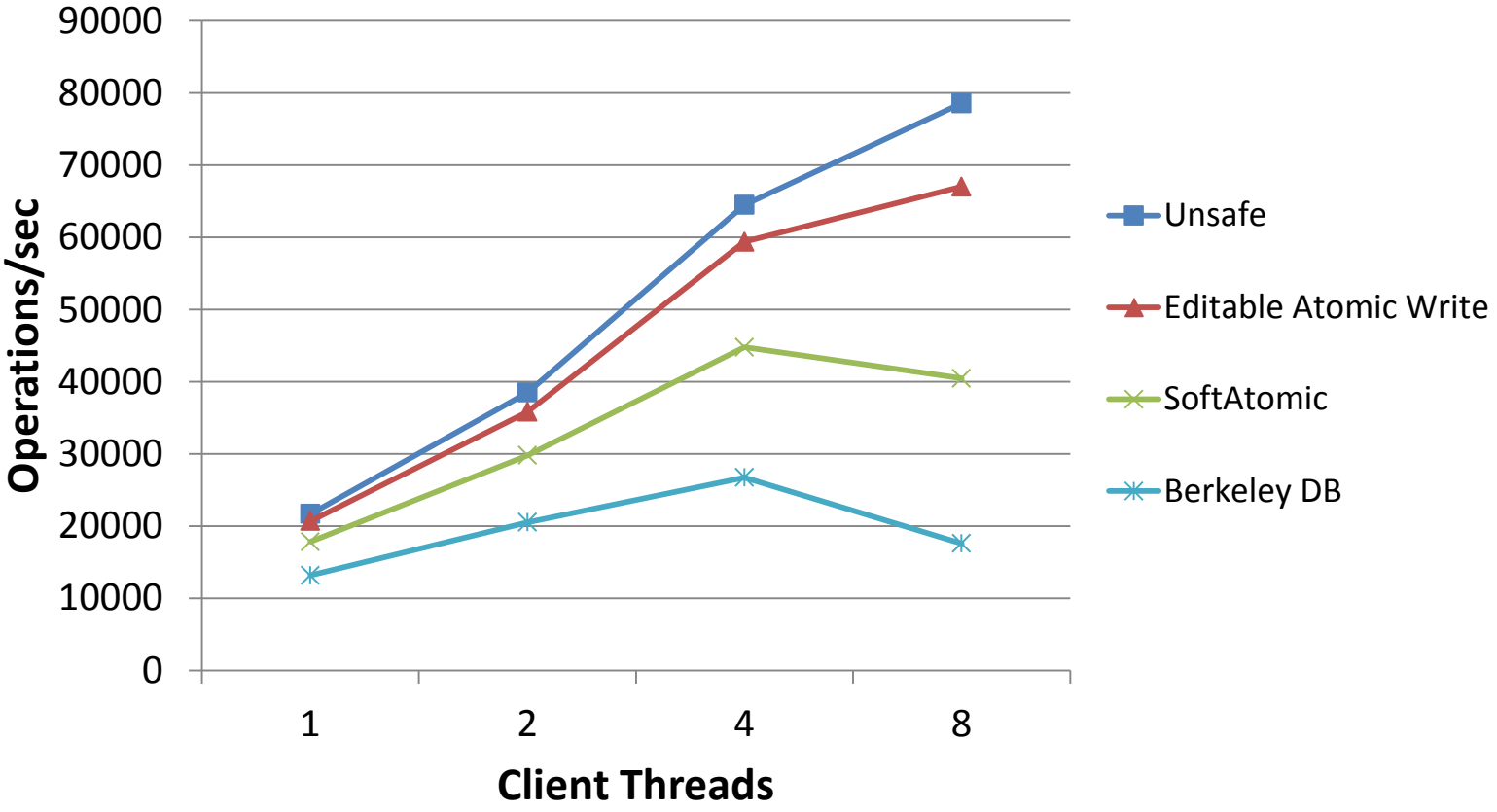
NVSL
Non-volatile Systems Laboratory

Internal Memory Bandwidth



NVSL
Non-volatile Systems Laboratory

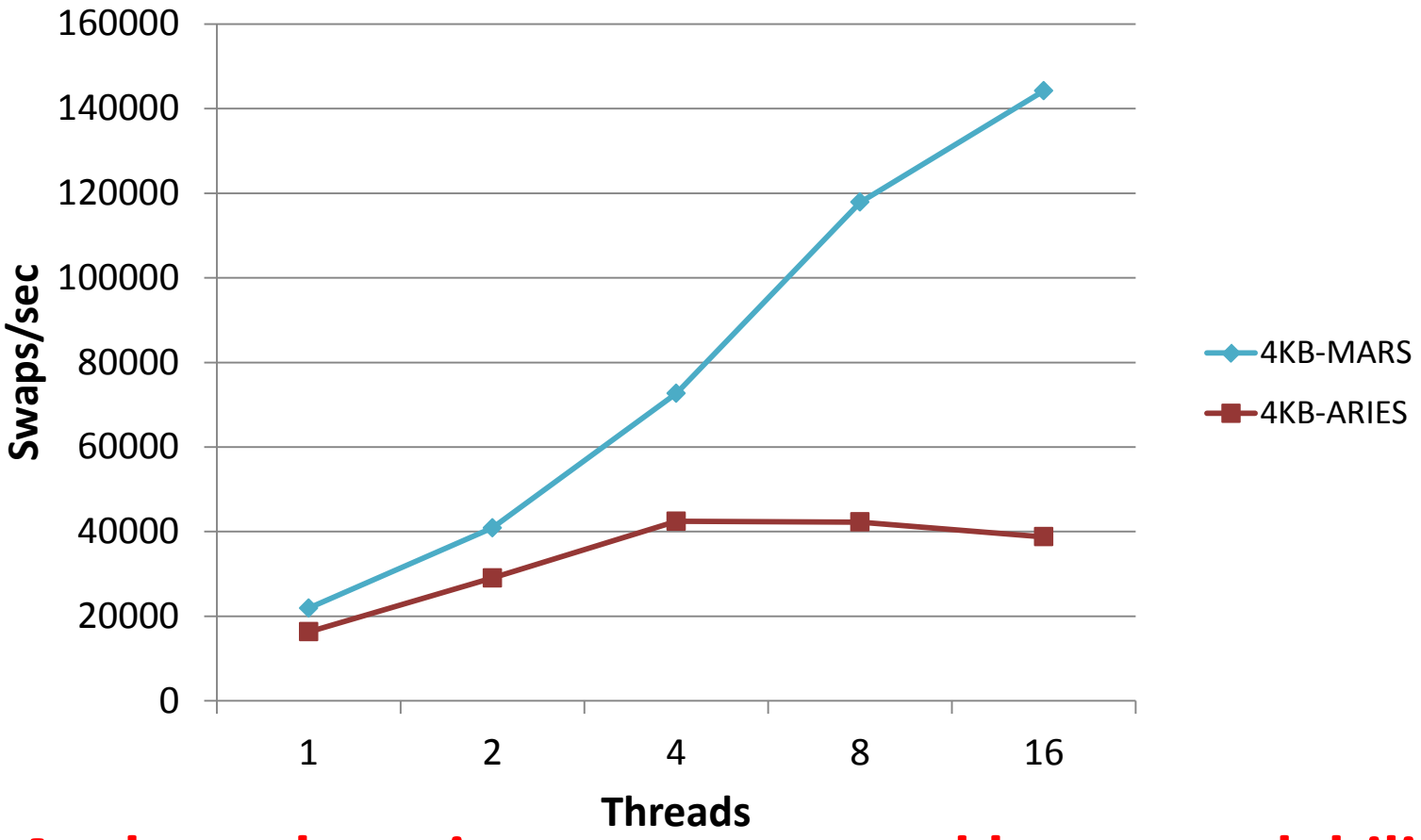
MemcacheDB: Persistent Key Value Store



1.7x faster than SoftAtomic, 3.8x faster than BDB

NVSL
Non-volatile Systems Laboratory

Comparison of MARS and ARIES



4x throughput improvement and better scalability

NVSL
Non-volatile Systems Laboratory

Conclusions from MARS

- MARS: Redesign of write-ahead logging for NVMs
  - Provides the features of ARIES but none of the disk-related overheads in a database storage manager
- Editable Atomic Writes (EAWs)
  - Makes the log accessible and editable prior to commit
  - Minimizes the cost of atomicity and durability
  - Offloads logging, commit, and write back to hardware
- MARS achieves 4x the performance of ARIES
  - Reduces latency and required host/device bandwidth

Thank you!

Any questions?

NVSL
Non-volatile Systems Laboratory

UCSDCSE
Computer Science and Engineering