

Hot Topics in Academic Flash and NVM Research

Saugata Ghose

Carnegie Mellon University



August 8, 2019
Santa Clara, CA

What Will This Talk Cover?

- Overview of recent trends and major topics in academic research
- Broadly centered around some highlighted works
 - Three papers selected by a committee of industrial experts with strong academic ties
 - Other selections representative of the larger areas of active work
- Where to explore more academic research
- Limiting factors to academic research
- Discussing personal reflections on academic research and the value of industrial collaboration

- Academic Coordinator of Flash Memory Summit
- Publishing actively in the following areas:
 - Processing-in-memory
 - NAND flash memory reliability
 - SSD controllers
 - Low-latency memory
 - Hybrid memory technologies
- More information
 - <http://ece.cmu.edu/~saugatag/>
 - <https://safari.ethz.ch/>



The Impact of Low-Latency Drives on Software

Integrating Non-Volatile Main Memory

Exposing Persistency in Systems

Device-Level Memory Innovations

Reliability and Security Exploits

- **10 works shortlisted**
 - Must be published in 2018 or 2019
 - Cannot include works with a conflict of interest
- **3 highlight works selected by a committee from the shortlist**



Robert Peglar



Jim Ballingall



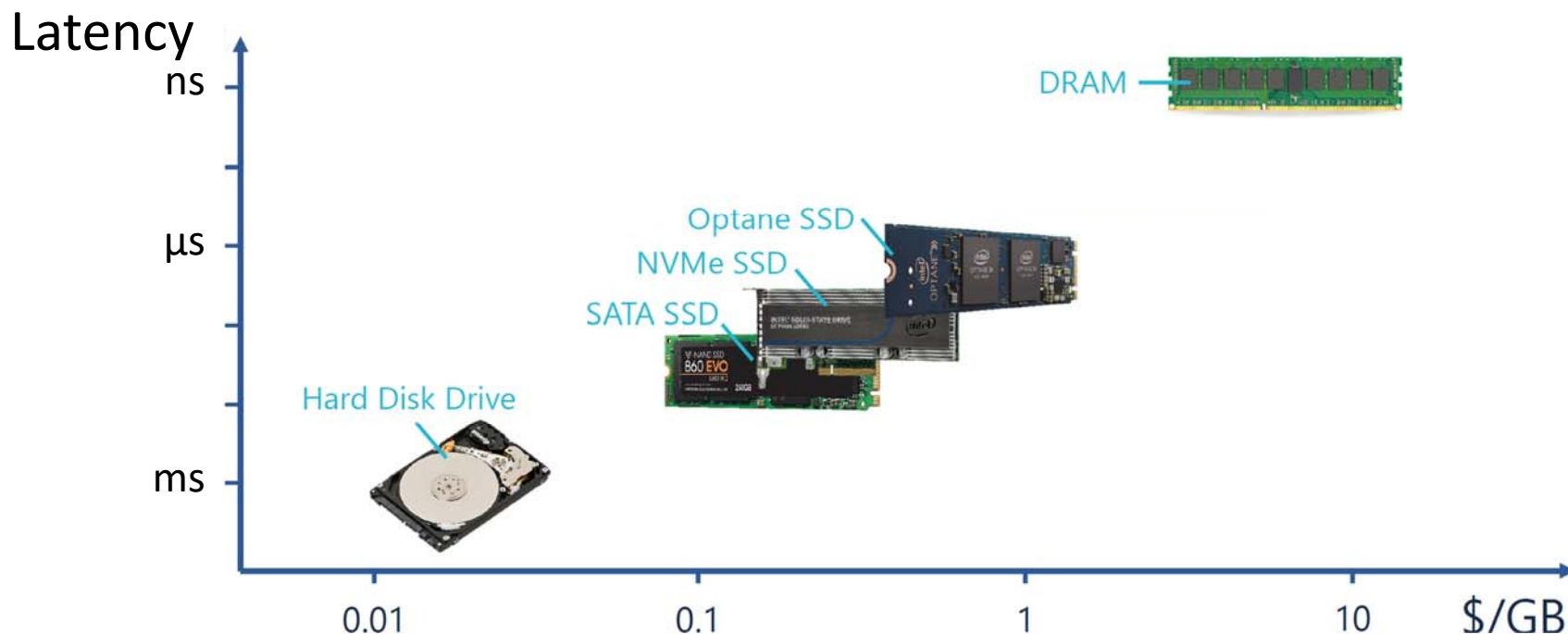
Erich Haratsch



The Impact of Low-Latency Drives on Software

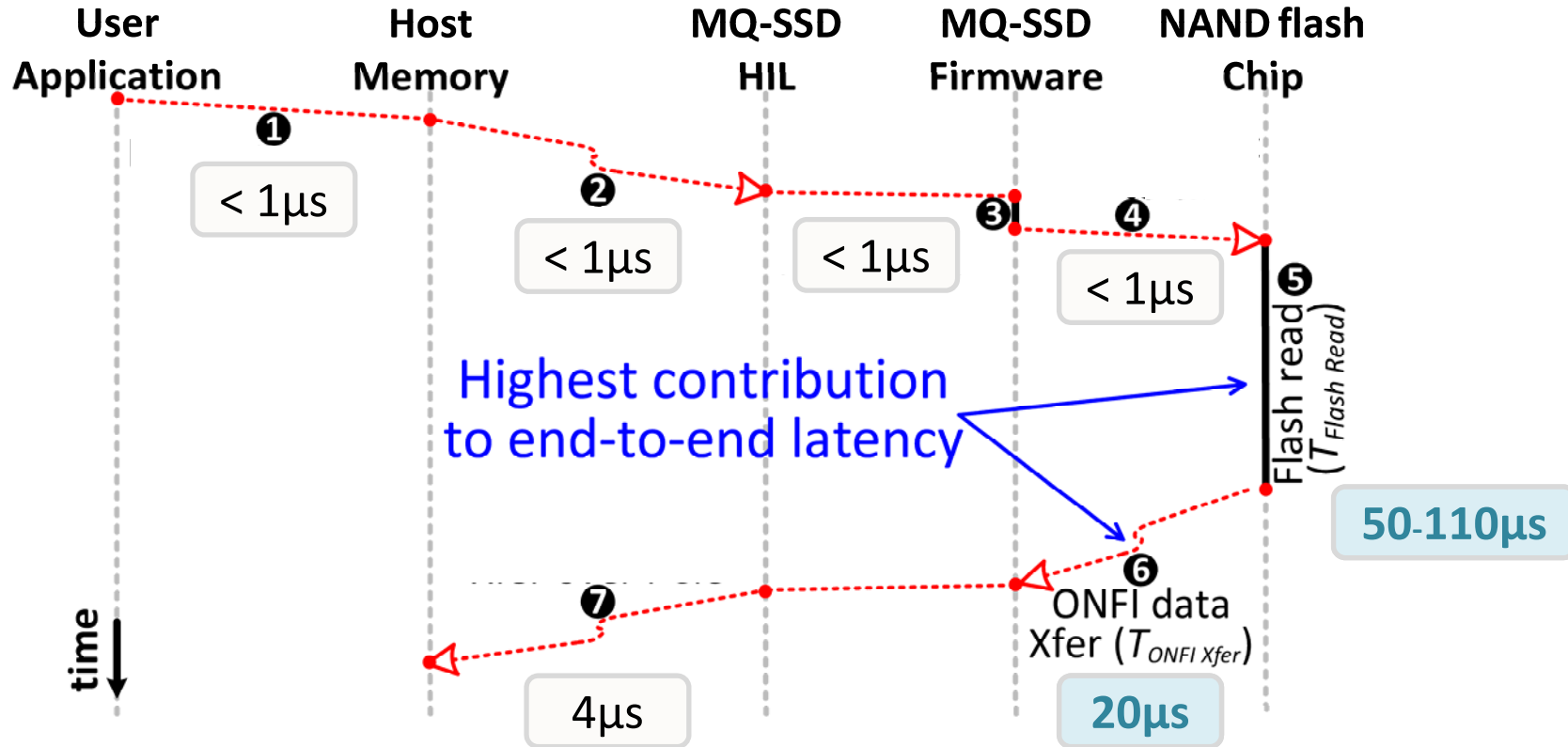
The Impact of Ultra-Fast Storage

- Solid-state storage is filling in the gap between NAND flash-based SSDs and DRAM

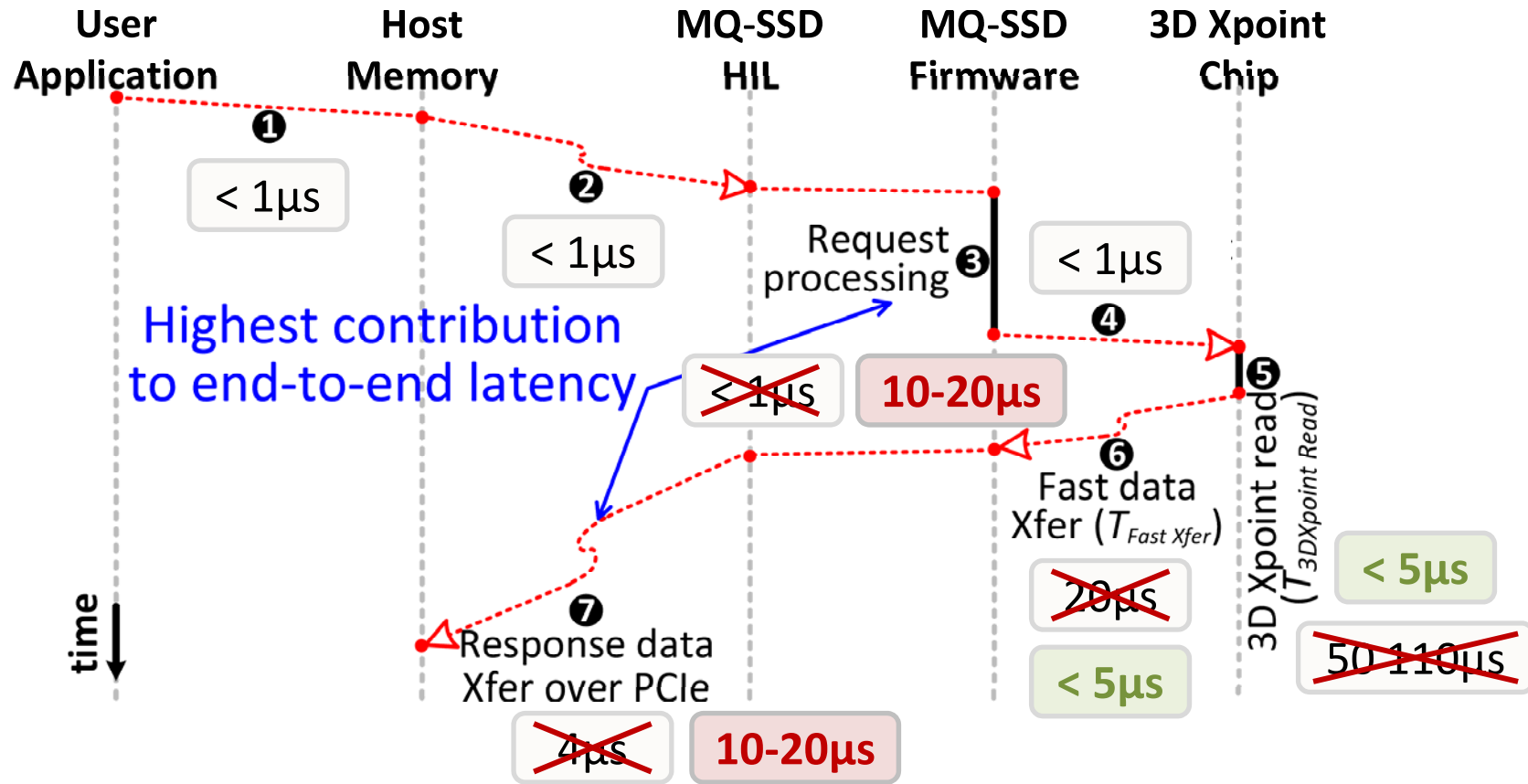


3D NAND	BiCS	V-NAND	Z-NAND
# layer	48	64	48
t_R	$45\mu s$	$60\mu s$	$3\mu s$
t_{PROG}	$660\mu s$	$700\mu s$	$100\mu s$
Capacity	256Gb	512Gb	64Gb
Page Size	16KB/Page	16KB/Page	2KB/Page

Request Latency of NAND Flash Based SSD



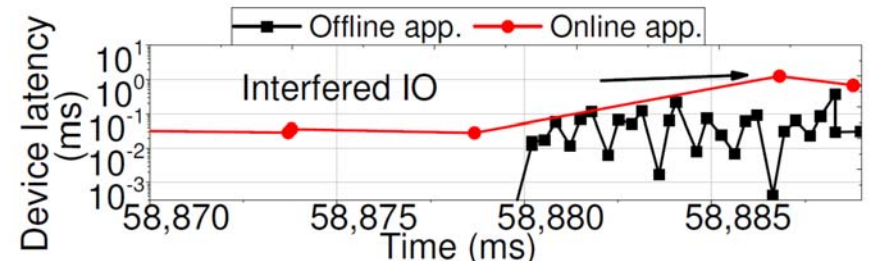
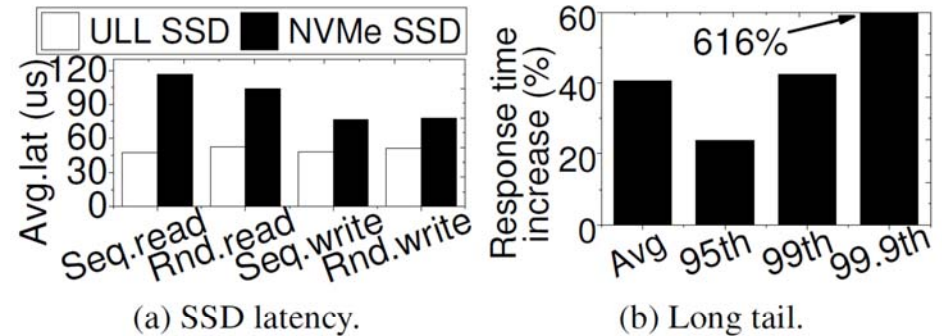
Request Latency of Optane Based SSD

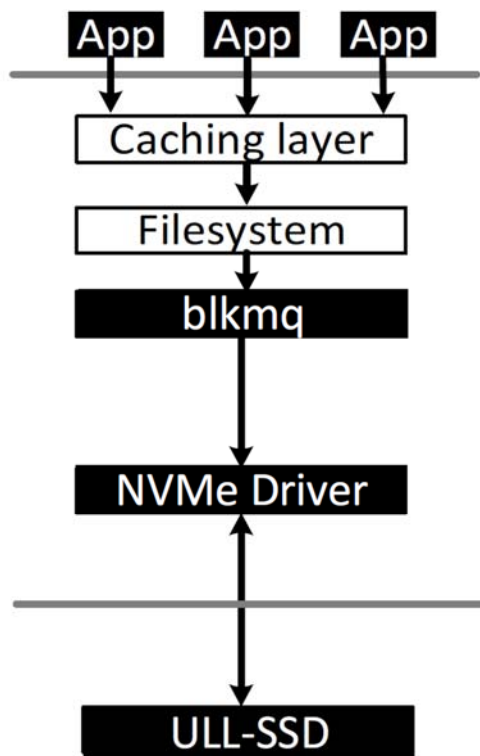


Rethinking the Software Stack

- The traditional OS I/O stack was designed with **long-latency drives** (e.g., magnetic hard disks) in mind
- NVMe and other modern protocols **eliminate parts of the stack**
 - No more I/O scheduling in the software
 - Side effect: Lack of fairness control**

- Even with new protocols, the remaining parts of the OS software stack can **hide the benefits of ultra-low-latency (ULL) SSDs**





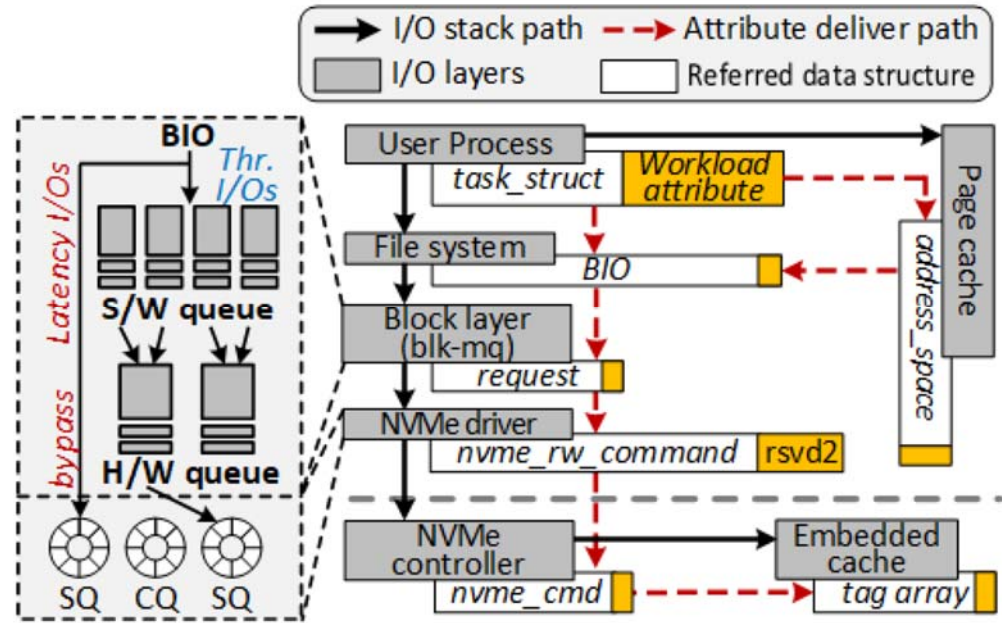
The storage stack is unaware of the characteristics of both latency-critical workload and ULL-SSD

The current design of blkmq layer, NVMe driver, and SSD firmware can hurt the performance of latency-critical applications.

- blk-mq is a software request queue for I/O blocks
- Latency-critical requests get held up by the queue for a long time
- Hardware queues don't know the latency-criticality of requests

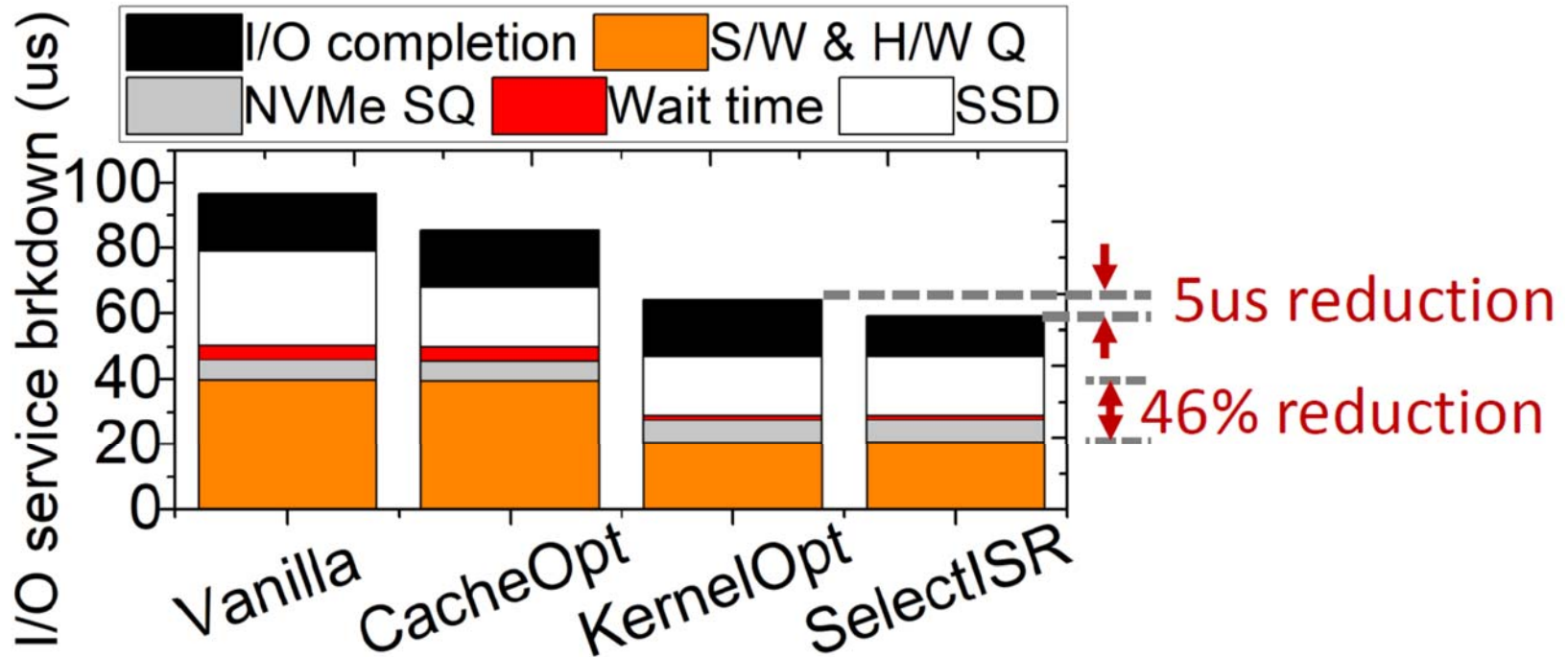
FlashShare: A Server I/O Stack for ULL SDDs

- Redesign several parts of the stack to let latency-critical requests have fast access
- Modified Linux process control block to include workload attributes
- blk-mq: identifies, bypasses latency-critical requests



- Use a dedicated NVMe hardware queue for bypassed requests
- Modify the NVMe scheduler to prioritize bypassed requests
- Partition the in-SSD DRAM cache to dedicate space for latency-critical data

FlashShare Evaluation



- Average turnaround time reduced by 22%
- 99th percentile turnaround reduced by 31%

**Significant benefits from
holistically eliminating queuing**

“New media such as Z-NAND and 3D XPoint have significantly lower latencies than 3D NAND, and the NVMe protocol also allows for lower latencies than previous storage interfaces. This paper shows that in order to achieve low latencies on the application level, further optimizations are required.”

FLASHSHARE: Punching Through Server Storage Stack from Kernel to Firmware for Ultra-Low Latency SSDs

Jie Zhang¹, Miryeong Kwon¹, Donghyun Gouk¹, Sungjoon Koh¹, Changlim Lee¹,
Mohammad Alian², Myoungjun Chun³, Mahmut Taylan Kandemir⁴,
Nam Sung Kim², Jihong Kim³, and Myoungsoo Jung¹

Yonsei University¹,

Computer Architecture and Memory Systems Laboratory,

University of Illinois Urbana-Champaign², Seoul National University³, Pennsylvania State University⁴

<http://camelab.org>

Abstract

A modern datacenter server aims to achieve high energy efficiency by co-running multiple applications. Some of such applications (e.g., web search) are latency sensitive. Therefore, they require low-latency I/O services to fast respond to requests from clients. However, we ob-

ices [8]. As such applications are often required to satisfy a given Service Level Agreement (SLA), the servers should process requests received from clients and send the responses back to the clients within a certain amount of time. This requirement makes the online applications latency-sensitive, and the servers are typically

- OSDI 2018 Paper:

<https://www.usenix.org/system/files/osdi18-zhang.pdf>

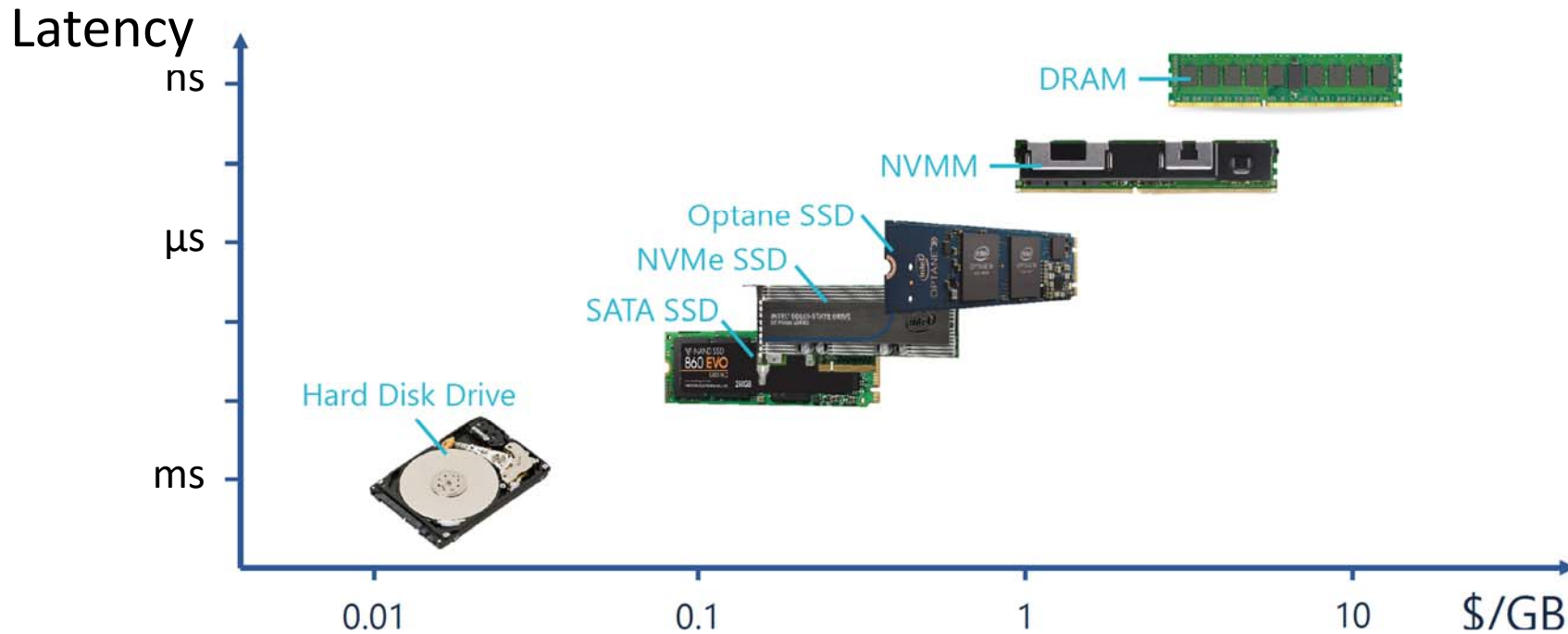
- Presentation:

<https://www.usenix.org/conference/osdi18/presentation/zhang>



Integrating Non-Volatile Main Memory

Non-Volatile Main Memory (NVMM)



- Byte-addressable, fast, persistent memory
- Two common management approaches
 - Software-transparent modules that are compatible with DRAM standards
 - NVMM-aware file systems





Ziggurat: A Tiered File System with NVMM

- NVMM for speed
- Disks for capacity



- Goal: design a latency-first file system
- Not all I/O requests should go to the NVMM

Placing Data in Ziggurat

Data Placement		Synchronicity predictor	
		Synchronously-updated	Asynchronously-updated
Write size predictor	Large writes	NVMM 	Disk 
	Small writes	NVMM 	NVMM 

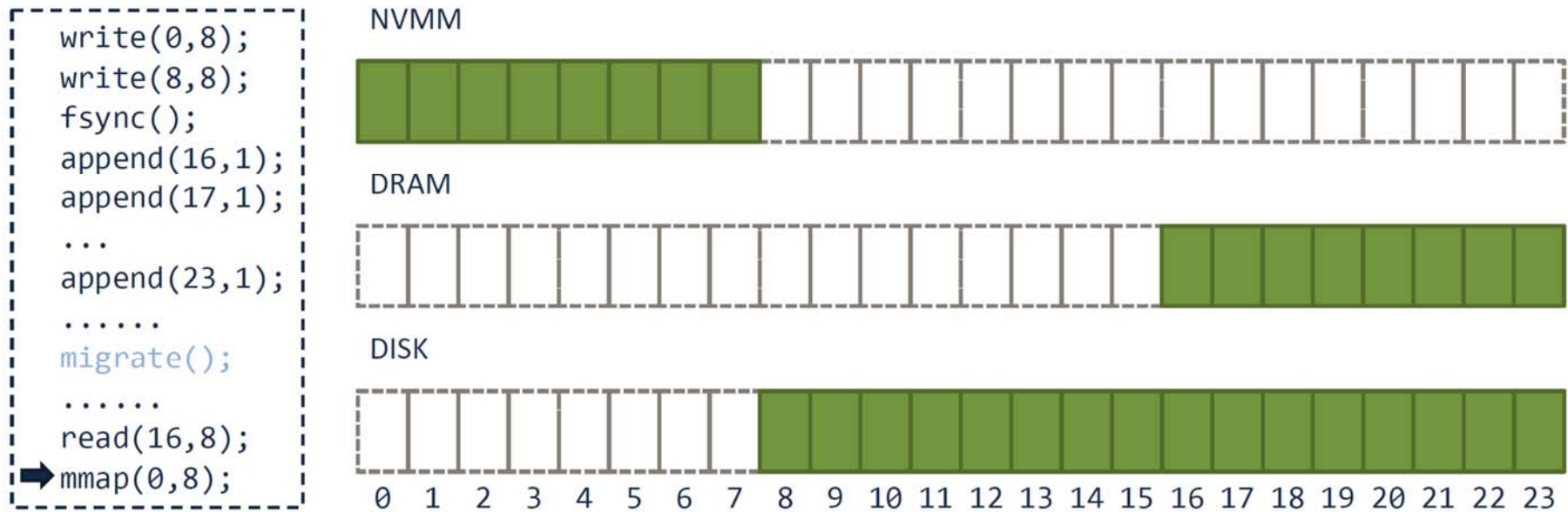
■ Synchronicity predictor

- Counts the number of blocks between each `fsync()` call for each file
- If counter exceeds a predetermined threshold, the file is classified as asynchronous; otherwise, file is classified as synchronous

■ Write size predictor

- Counter inside each write entry
- Checks if size is large, and if the write size remains stable (i.e., if write is at least half as big as original write)

Example File Operations with Ziggurat File System



- With 2GB of NVMM and a 400GB SSD, Ziggurat achieves up to 38.9x/46.5x throughput over EXT4/XFS vs. SSD alone
- As NVMM size grows, Ziggurat's performance improves until it nearly matches the performance of an NVMM-only file system

“Caching and tiering are well known methods for file operations and managing data.

Ziggurat is an advance to more intelligently move data among disk-based storage and NVM-based storage with high performance.

Ziggurat profiles the application's access stream online to predict the behavior of individual writes.”

Ziggurat: A Tiered File System for Non-Volatile Main Memories and Disks

Shengan Zheng^{†*} Morteza Hoseinzadeh[§] Steven Swanson[§]
[†]*Shanghai Jiao Tong University* [§]*University of California, San Diego*

Abstract

Emerging fast, byte-addressable Non-Volatile Main Memory (NVMM) provides huge increases in storage performance compared to traditional disks. We present Ziggurat, a tiered file system that combines NVMM and slow disks to create a storage system with near-NVMM performance and large capacity. Ziggurat steers incoming writes to NVMM, DRAM, or disk depending on application access patterns, write size, and the likelihood that the application will stall until the

ory is higher than SSD, and SSDs and hard drives scale to much larger capacities than NVMM. So, workloads that are cost-sensitive or require larger capacities than NVMM can provide would benefit from a storage system that can leverage the strengths of both technologies: NVMM for speed and disks for capacity.

Tiering is a solution to this dilemma. Tiered file systems manage a hierarchy of heterogeneous storage devices and place data in the storage device that is a good match for the

- FAST 2019 Paper:

<https://www.usenix.org/system/files/fast19-zheng.pdf>

- Presentation:

<https://www.usenix.org/conference/fast19/presentation/zheng>



Exposing Persistency in Systems

The Benefits and Troubles of Persistency

- Several benefits to integrating NVMs into systems as main memory
 - A way to overcome limitations of DRAM scaling
 - Can be integrated as a hybrid memory alongside DRAM
- Many applications can make use of the non-volatility (e.g., databases)

Performance



Density



Non-volatility



What happens if the system crashes?

- In the past, DRAM lost all state – restart program, but **no inconsistent state**, as commits to disk were synchronized
- **NVMM keeps partial (maybe inconsistent) records** due to its persistency, regardless of synchronization boundaries

- We need to do something to ensure that commits are consistent
- Several solutions from academia and industry

- **Solution Type 1: ISA-level enhancements**
 - Significant programmer burden to make use of low-level ISA instructions
 - Programs are not portable to other architectures

- **Solution Type 2: programming language level constructs**
 - e.g., Acquire–Release Persistency for C++11
 - Do not support sequential consistency for data-race-free (SC-DRF) model

- Can we provide persistency guarantees without having to rely on costly synchronization mechanisms?

Persistency for Synchronization-Free Regions

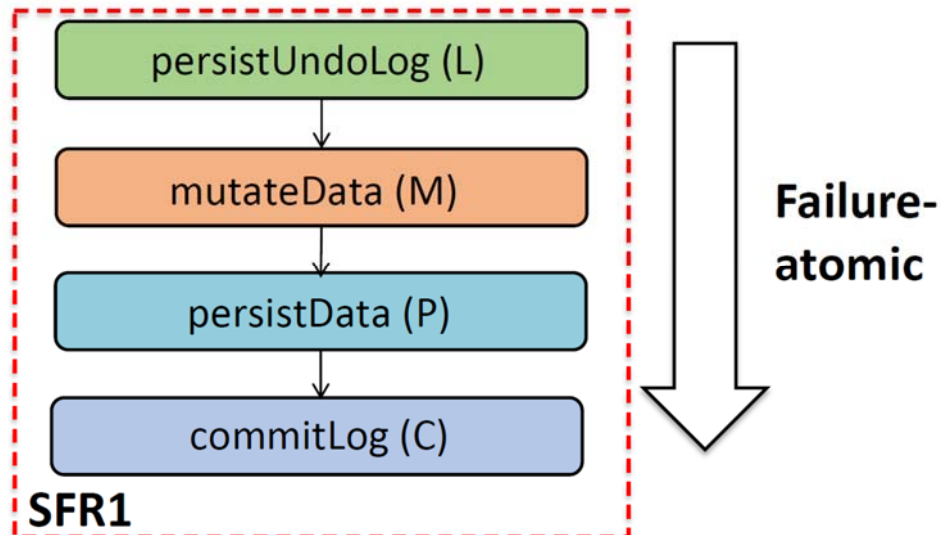
- Synchronization free regions (SFRs):**
 thread regions delimited by synchronization operations or system calls
 - Operations within an SFR logically appear atomic to other SFRs
 - Useful for both fault recovery code, fault-free code
- Goal: provide *failure-atomicity* for an SFR**
- Provide per-SFR undo-logging**
 - Log both undos and commits
 - Coupled-SFR: only latest SFR may be lost, but waits for persist/commit
 - Decoupled-SFR: delay P/C
- 65.5% average performance improvement**

```

l1.acq();
  x -= 100;
  y += 100;
l2.acq();
  a -= 100;
  b += 100;
l2.rel();
l1.rel();
    
```

SFR1

SFR2



“First, the topic is extremely important and timely, given the advent of new persistent memory technologies, in combination with how CPUs are typically organized with multiple SRAM cache regions.

Second, the further importance of recognizing (and programming for) the concepts of atomicity and synchronization with regard to data.

The applicability of synchronization-free regions of code is front and center today, and will be for the foreseeable future.”

Persistency for Synchronization-Free Regions

Vaibhav Gogte
University of Michigan, USA
vgogte@umich.edu

Stephan Diestelhorst
Arm Research, UK
stephan.diestelhorst@arm.com

William Wang
Arm Research, UK
william.wang@arm.com

Satish Narayanasamy
University of Michigan, USA
nsatish@umich.edu

Peter M. Chen
University of Michigan, USA
pmchen@umich.edu

Thomas F. Wenisch
University of Michigan, USA
twenisch@umich.edu

Abstract

Nascent persistent memory (PM) technologies promise the performance of DRAM with the durability of disk, but how best to integrate them into programming systems remains an open question. Recent work extends language memory models with a persistency model prescribing semantics for updates to PM. These semantics enable programmers to design data structures in PM that are accessed like memory and

Keywords Persistent memories, persistency models, language memory models, failure-atomicity, synchronization-free regions

ACM Reference Format:

Vaibhav Gogte, Stephan Diestelhorst, William Wang, Satish Narayanasamy, Peter M. Chen, and Thomas F. Wenisch. 2018. Persistency for Synchronization-Free Regions. In *Proceedings of 39th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'18)*. ACM, New York, NY, USA, 16 pages.

- PLDI 2018 Paper:

https://web.eecs.umich.edu/~pmchen/papers/gogte18_1.pdf

- Presentation:

<https://www.youtube.com/watch?v=6ztCb1nTzo4>

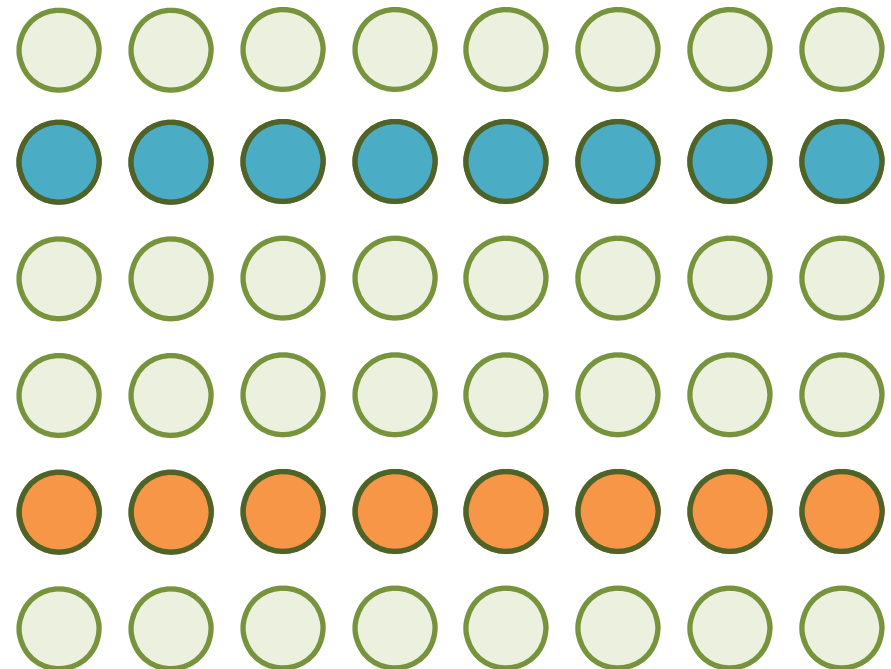


Other Noteworthy Research Areas

Device-Level Memory Innovations

- Significant reliability improvements for RRAM/memristive devices (e.g., advanced selectors)
- Memory technologies that support processing-in-memory
 - Combine charge/resistance/etc. in cells to perform bitwise operations
 - Requires simple changes to the memory arrays
 - New research challenges: how do we support efficient programming and control flow models?
- Machine learning accelerators
 - Perform efficient analog math operations
 - Typically have RRAM embedded with compute logic

X AND Y



- **Storage reliability remains a major area of work**
 - Characterization of new storage technologies (e.g., 3D flash, Optane)
 - Mechanisms to improve lifetime (e.g., adapting MLC/TLC cells to SLC mode)
 - File system reliability

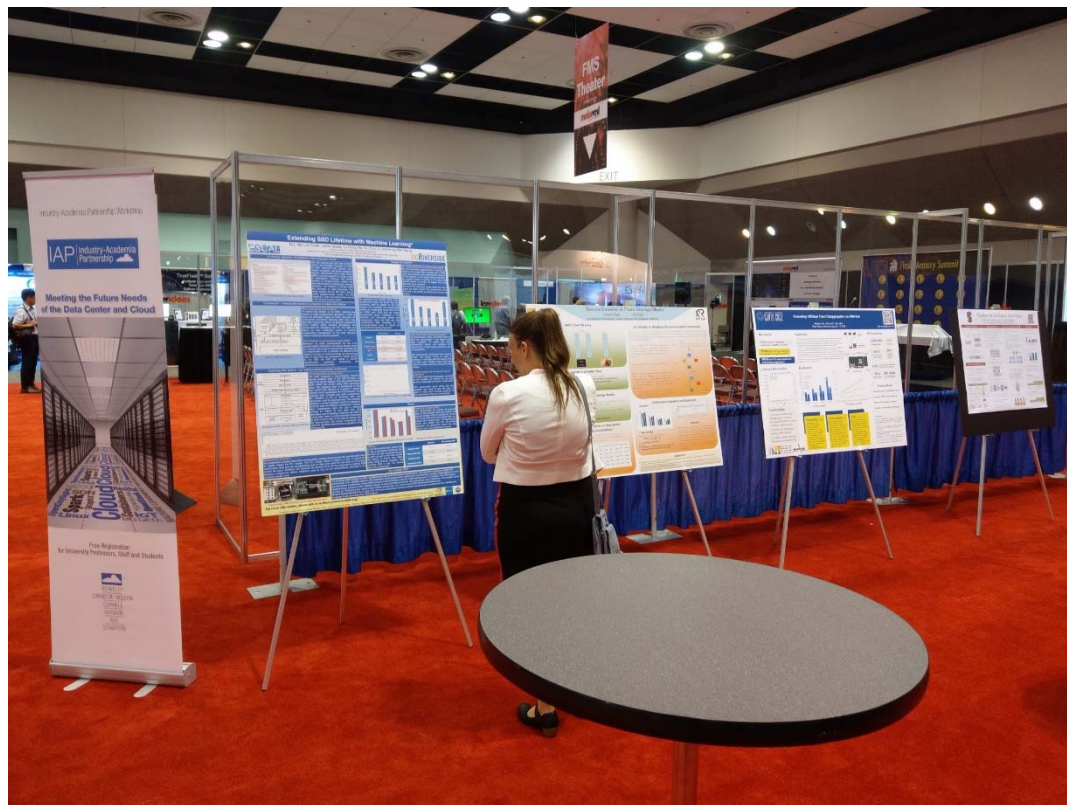
- **Security is a big concern in computer architecture**
 - Meltdown, Spectre exploit speculation to gain root privileges
 - RowHammer in DRAM (i.e., cell-to-cell interference) can be exploited in real systems for many security exploits
 - MLC NAND flash memory susceptible to RowHammer-like attacks



Learning More About Academic Research

Where to Explore Academic Research

- Storage, computer architecture, systems research conferences
 - FAST
 - ISCA, MICRO, HPCA, ASPLOS
 - SOSP, OSDI, USENIX ATC
 - Many more conferences...
- Specialized venues (e.g., NVMW, PIRL)
- Here at the Flash Memory Summit!
 - Several talks
 - **Student posters in booth 745 (by the theater)**



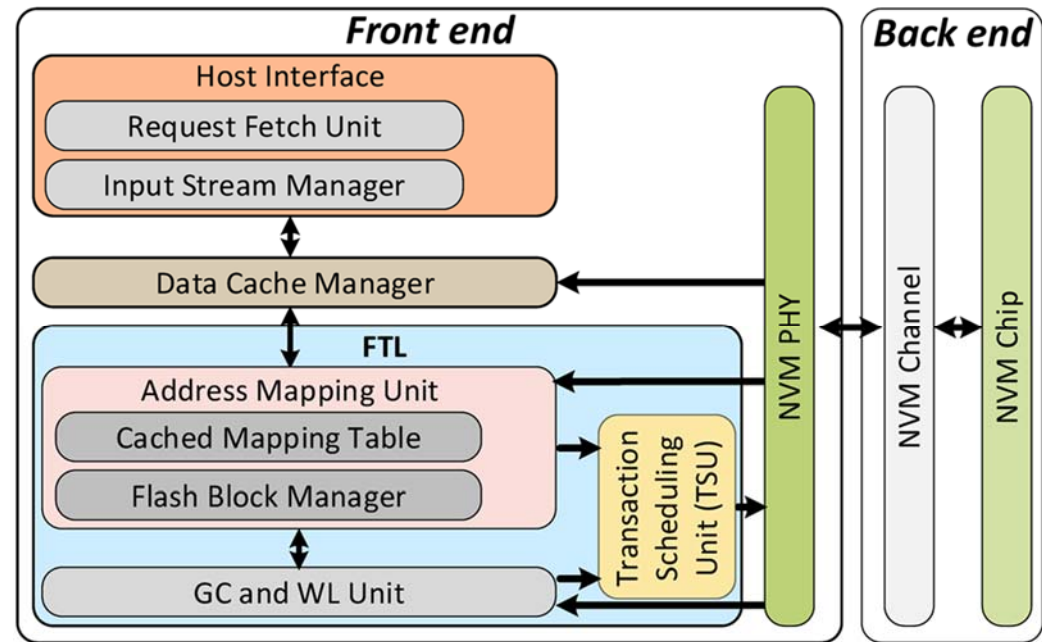
Platforms for Academic Research

■ Emulation

- System-level delays for NVM
- Dummy PCI boards to emulate NVM using DRAM

■ Simulation

- Many simulators miss out on many details of devices, system interactions
- Three new simulators work around this
 - » WiscSee
 - » MQSim
 - » SimpleSSD



■ Real device testing

- Requires significant effort to build testing infrastructure
- Often involves a lot of reverse engineering
- **Difficult to do without some form of industry collaboration**

- An example: our group's collaboration with Seagate
- We **experimentally characterize real**, state-of-the-art chips
- We find that errors are introduced into **unread data** when we read from the SSD: *read disturb errors*
- We find that errors can be introduced into **existing data** when we write to an SSD: *errors in partially-programmed data*
 - MLC NAND flash uses **two-step programming** for each cell
 - *Partially-programmed cells* **much more vulnerable to errors** than fully-programmed cells
- NAND flash errors can be **exploited for security attacks**
- We find **new sources of errors in 3D NAND** flash memory
- Several solutions to **completely eliminate or mitigate vulnerabilities**, increase flash memory lifetime

Our Proceedings of the IEEE paper



Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

By YU CAI, SAUGATA GHOSE, ERICH F. HARATSCH, YIXIN LUO, AND ONUR MUTLU

ABSTRACT | NAND flash memory is ubiquitous in everyday life today because its capacity has continuously increased and cost has continuously decreased over decades. This positive growth is a result of two key trends: 1) effective process technology scaling; and 2) multilevel (e.g., MLC, TLC) cell data coding. Unfortunately, the reliability of raw data stored in flash memory has also continued to become more difficult to ensure, because these two trends lead to 1) fewer electrons in the flash

KEYWORDS | Data storage systems; error recovery; fault tolerance; flash memory; reliability; solid-state drives

I. INTRODUCTION

Solid-state drives (SSDs) are widely used in computer systems today as a primary method of data storage. In comparison with magnetic hard drives, the previously



<https://arxiv.org/pdf/1706.08642>

- Visit our group website: <https://www.ece.cmu.edu/~safari/>

Concluding Remarks

- Academics are working to solve many challenges, and are often **complementing the efforts of industry**
 - The Impact of Low-Latency Drives on Software
 - Integrating Non-Volatile Main Memory
 - Exposing Persistency in Systems
 - NVM Device-Level Innovations
 - Reliability and Security Exploits
- A number of limitations can hinder purely academic research
- Industrial collaborations with academia can provide us with great insights and fruitful results for both!
- Go check out these and other works
 - Several talks here at FMS
 - **Student posters at booth 745**

- <https://www.usenix.org/system/files/conference/hotstorage18/hotstorage18-paper-koh.pdf>
- <https://www.usenix.org/conference/osdi18/presentation/zhang>
- <https://www.usenix.org/conference/fast19/presentation/zheng>
- https://web.eecs.umich.edu/~pmchen/papers/gogte18_1.pdf

Hot Topics in Academic Flash and NVM Research

Saugata Ghose

Carnegie Mellon University



August 8, 2019
Santa Clara, CA