Solid-State Drives (SSDs)

Jin-Soo Kim
(*jinsoo.kim@snu.ac.kr*)

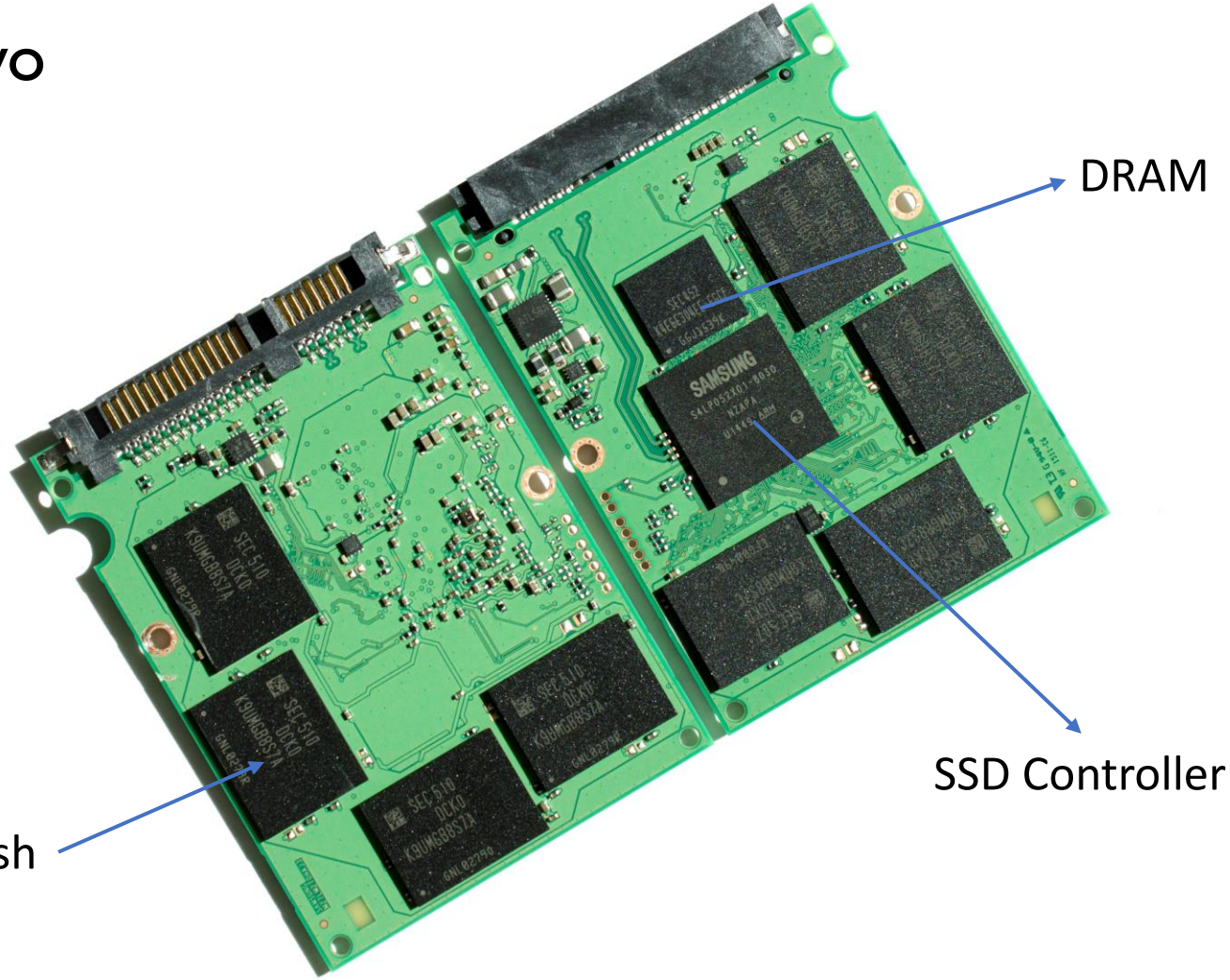
Systems Software &
Architecture Lab.

Seoul National University

Fall 2020

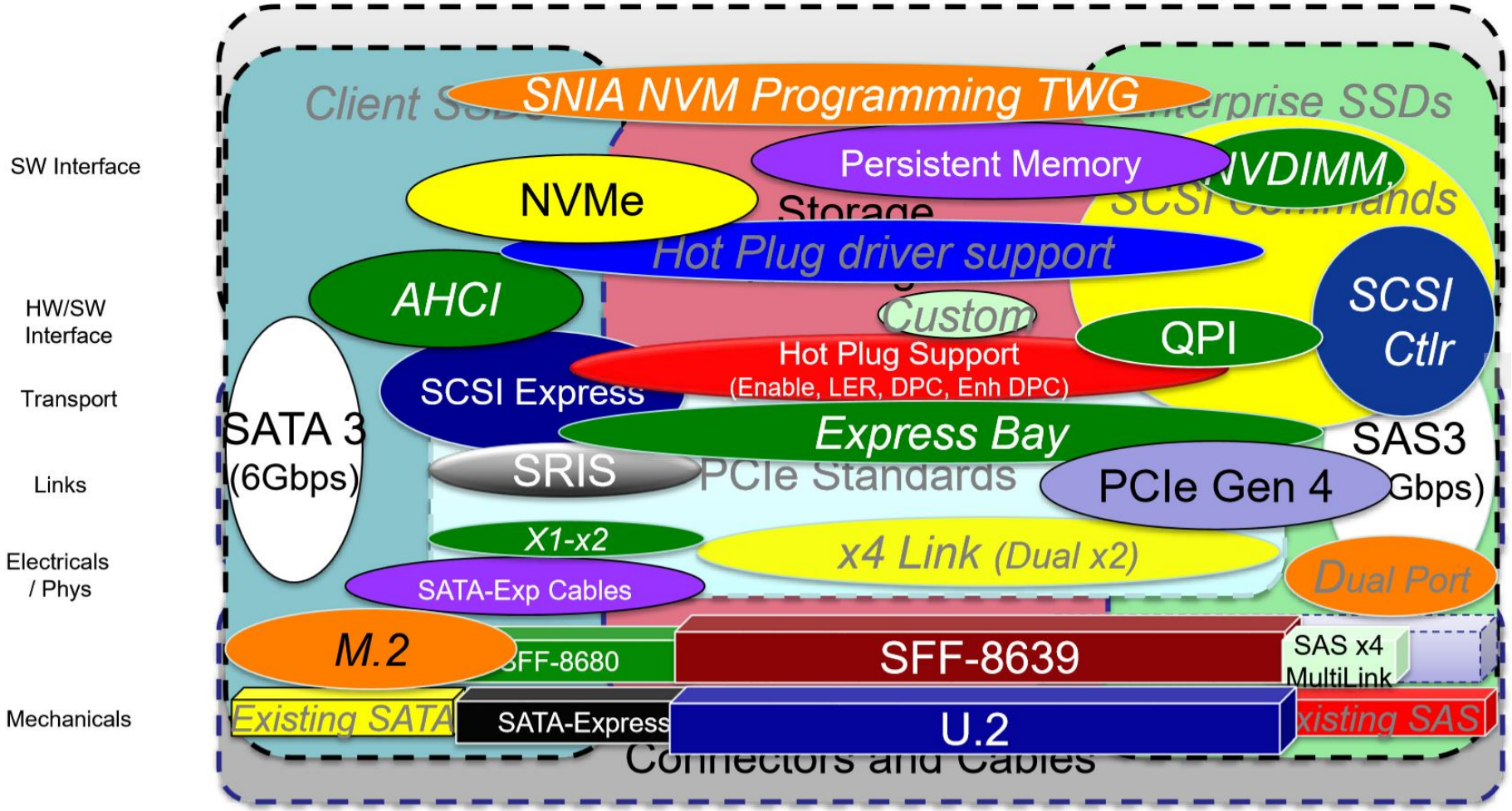
Anatomy of an SSD

- Samsung 850 Evo



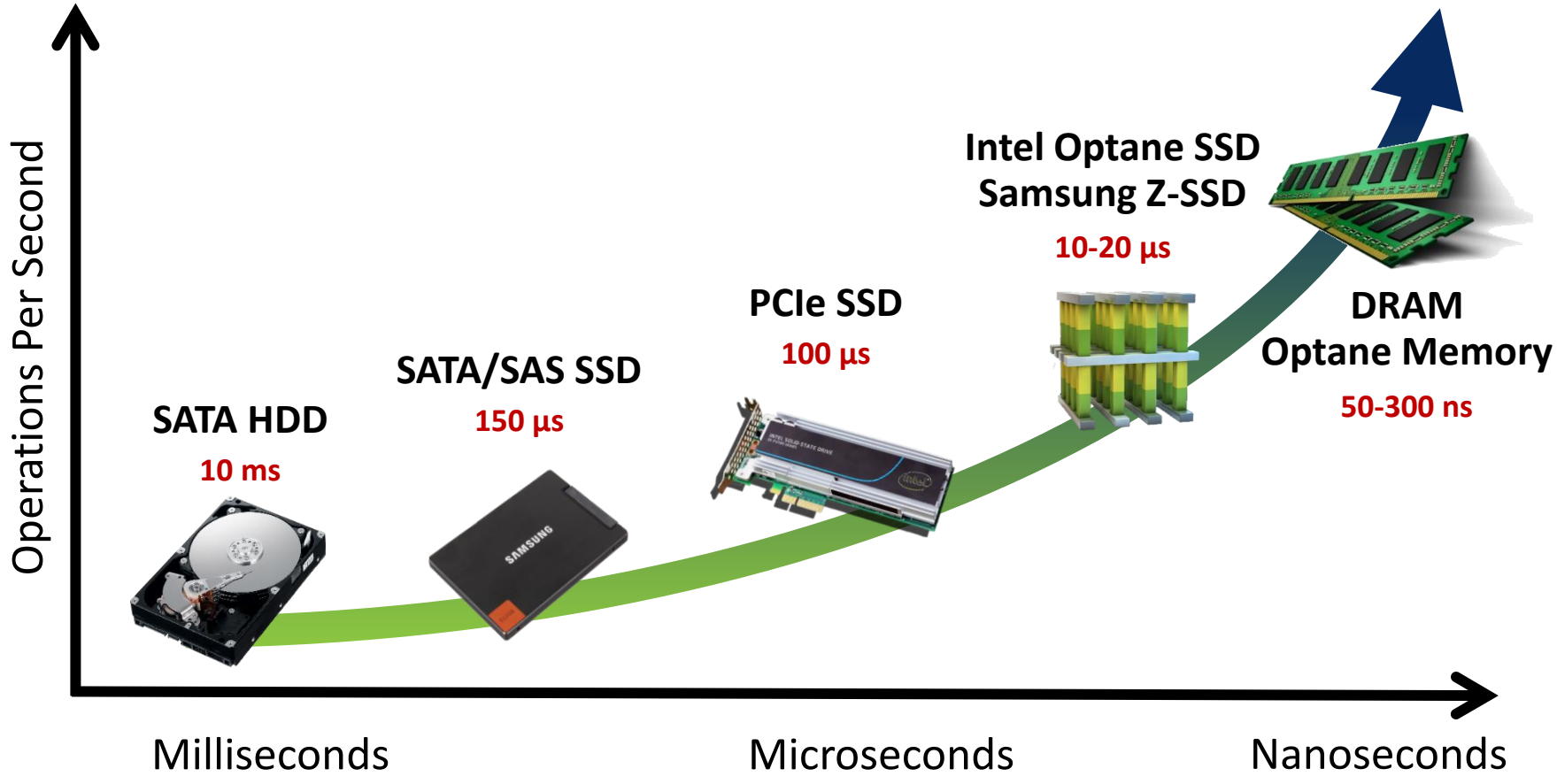
http://www.anandtech.com/show/9451/the-2tb-samsung-850-pro-evo-ssd-review

(Messy) Storage Interfaces



J. Pappas, Annual Update on Interfaces, FMS, 2015.

Moving Closer to the Processor

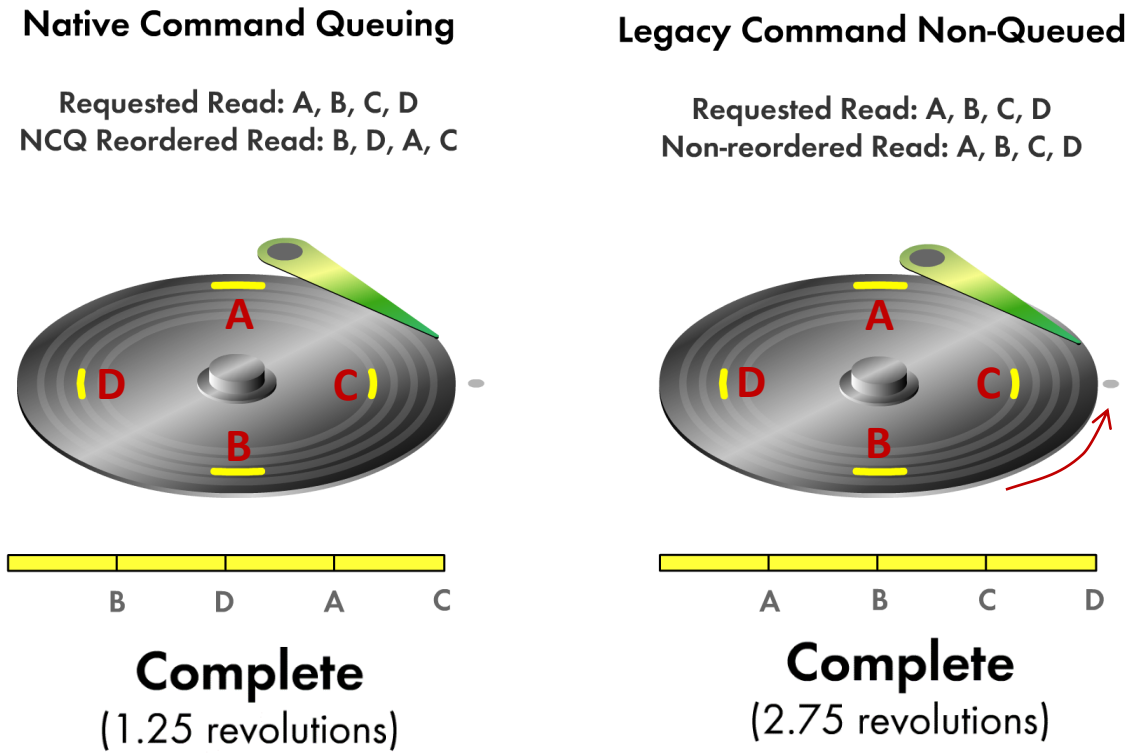


Serial ATA (SATA)

- Primary internal storage interconnect for desktop and mobile PCs
  - Evolved from (Parallel) ATA
  - More than 1.1 billion SATA drives shipped during 2001-2008
  - Market share (as of 2008): Desktop (99%), Mobile PC (97.7%), Enterprise (27.6%)
- Serial, point-to-point, half duplex
- Why SATA?
  - Lower pin count (cost, space), Lower voltage support (5V → 0.7V)
  - Higher performance: SATA 3 – 600MB/s @ 6Gbps
  - Simple drive configuration (no slave)
  - Greater reliability (CRC/packet)
  - Migration to servers (hot plug, NCQ, …)

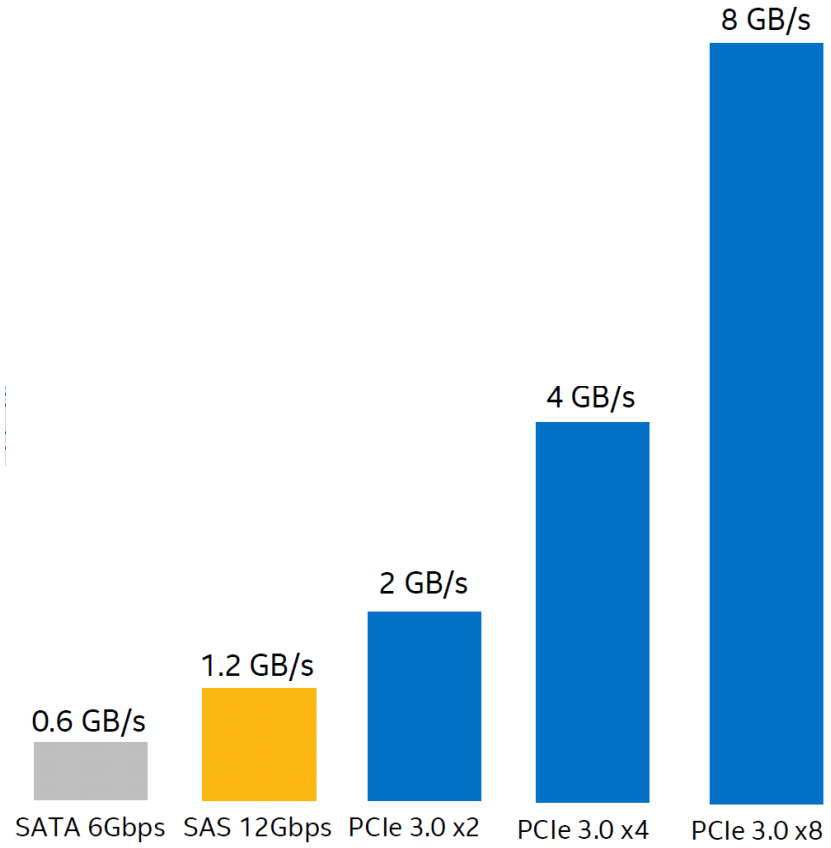
SATA NCQ

- Enqueue up to 32 commands in the drive
- Process them in an out-of-order fashion



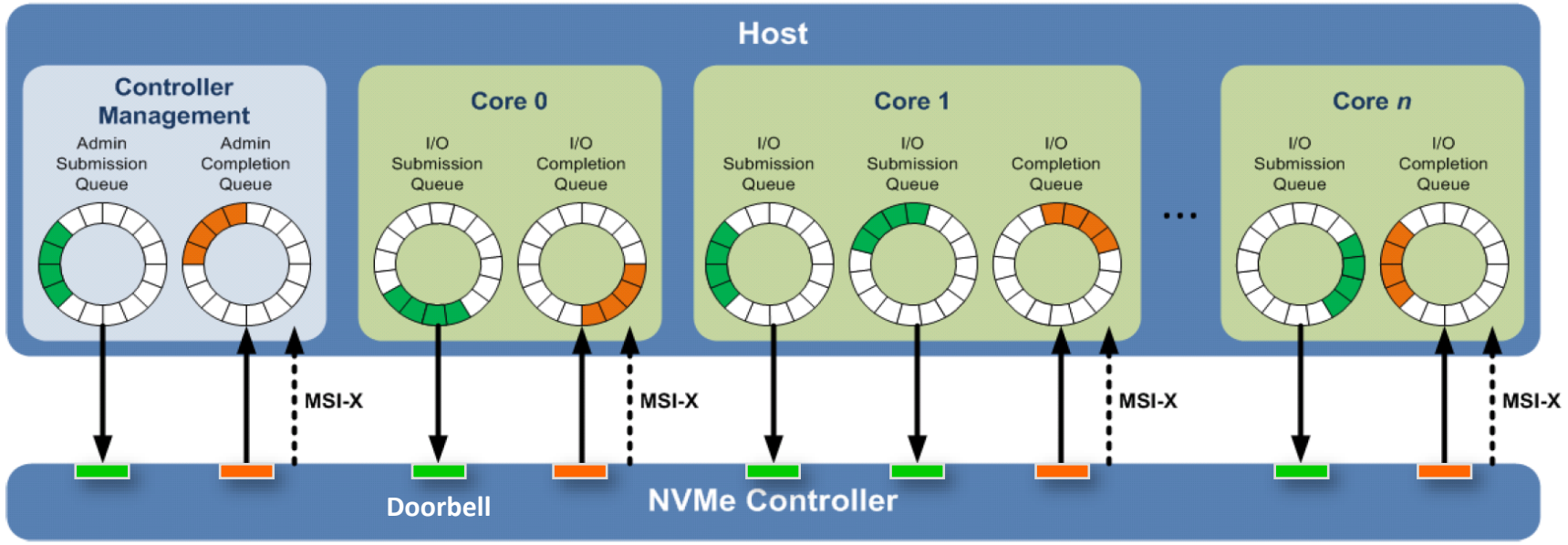
NVMe (NVM Express)

- The industry standard interface for high-performance NVM storage
  - NVMe 1.0 in 2011 by NVM Express Workgroup
  - NVMe 1.2 in 2014
- PCIe-based
- Lower latency
  - Direct connection to CPU
  - No HBA (Host Bus Adapter) required: reduced power and cost
- Scalable bandwidth
  - 1GB/s per lane (PCIe Gen3)
  - Up to 32 lanes



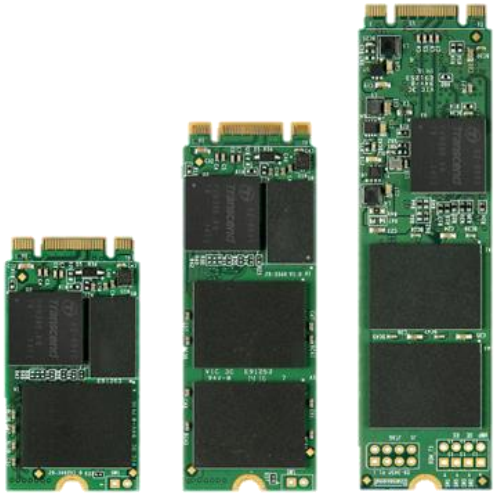
NVMe Overview

- Deep queue: 64K commands per queue, up to 64K queues
- Streamlined command set: only 13 required commands
- One register write to issue a command ("doorbell")
- Support for MSI-X and interrupt aggregation



NVMe SSD Form Factors

Add-in-card (AIC)

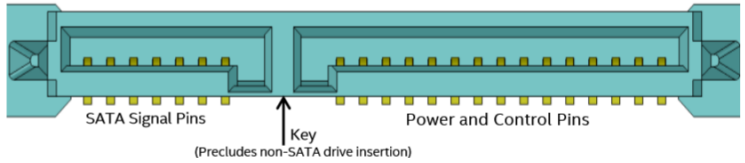


2242 2260 2280

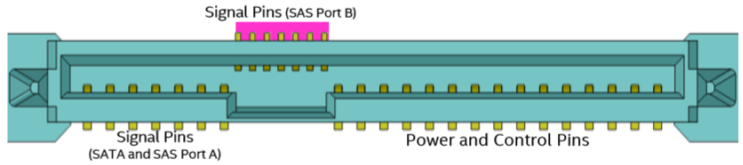
M.2 (PCIe: Up to x4)

U.2
(SFF-8639:
Up to x4)

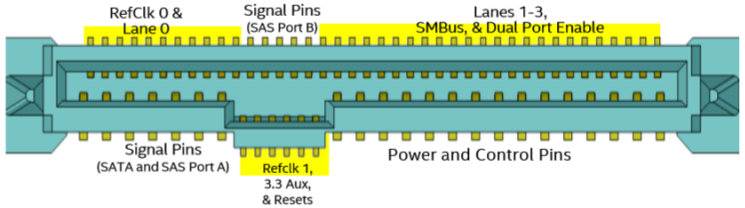
SATA



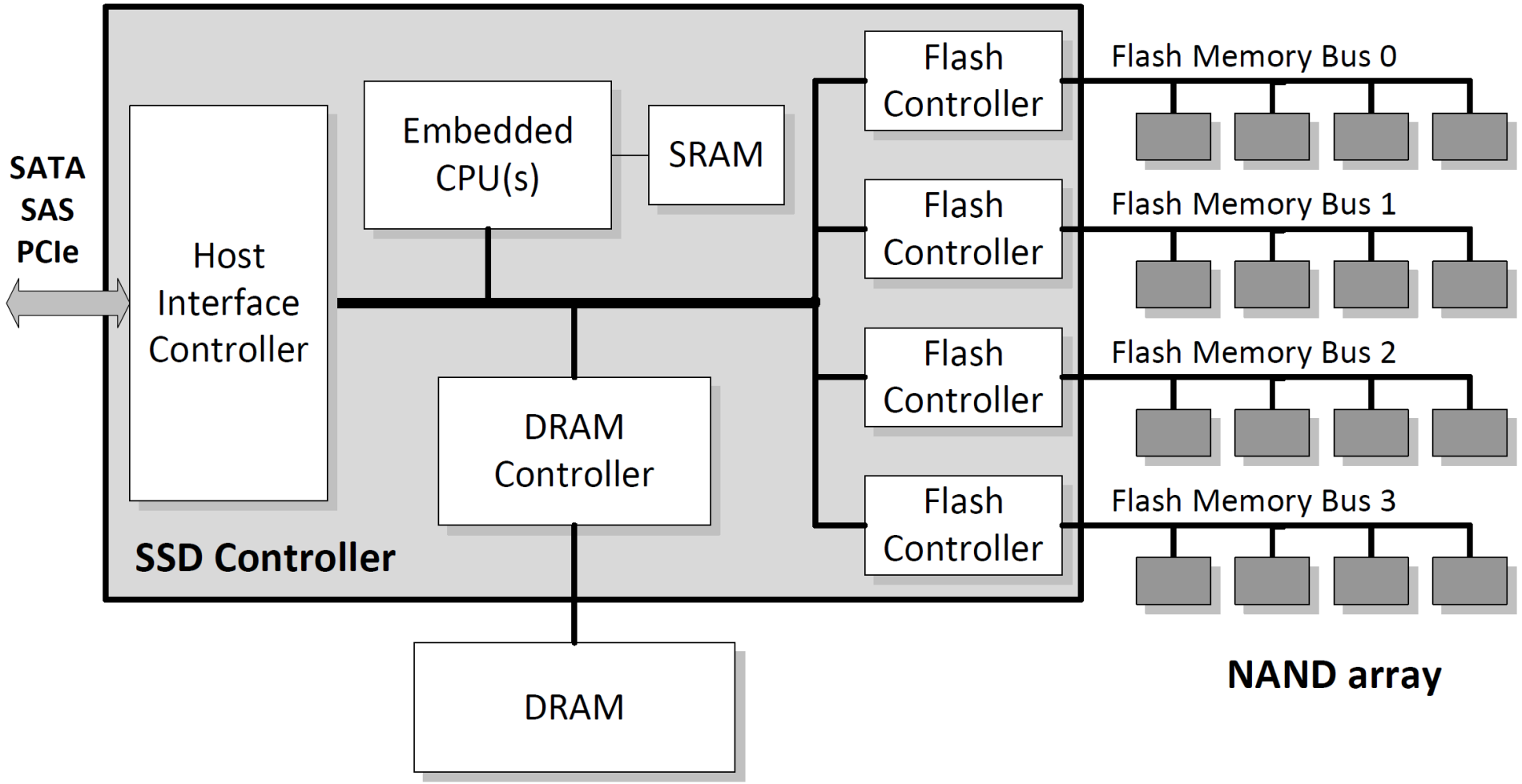
SAS



U.2



SSD Internals



The Unwritten Contract

- Several assumptions are no longer valid

Assumptions	Disks	SSDs
Sequential accesses much faster than random		
No write amplification		
Little background activity		
Media does not wear down		
Distant LBNs lead to longer access time		

Source: A. Rajimwale et al., "Block Management in Solid-State Devices," USENIX ATC, 2009.

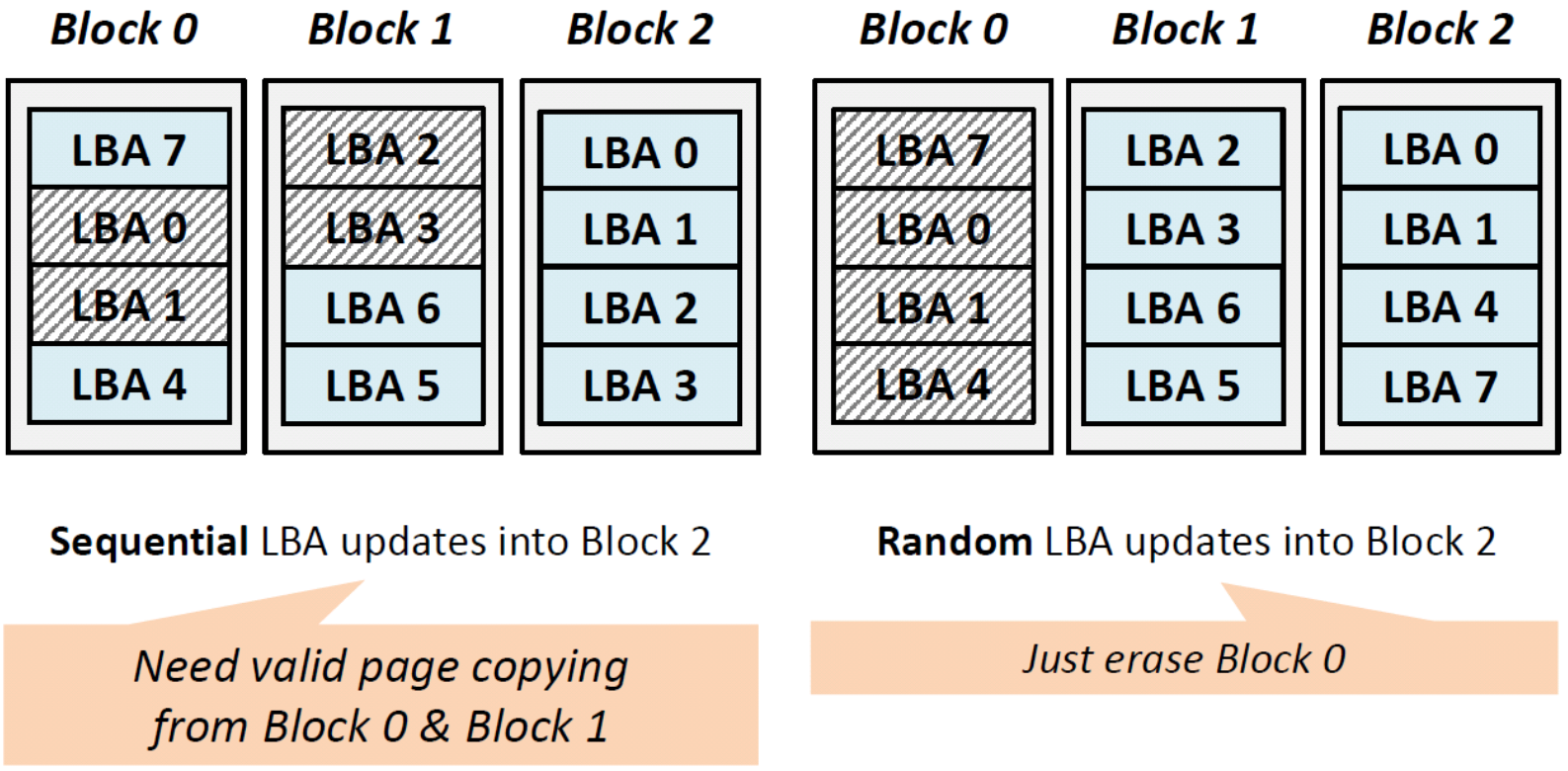
The Multi-streamed Solid-State Drive

(J.-U. Kang et al., HotStorage, 2014)

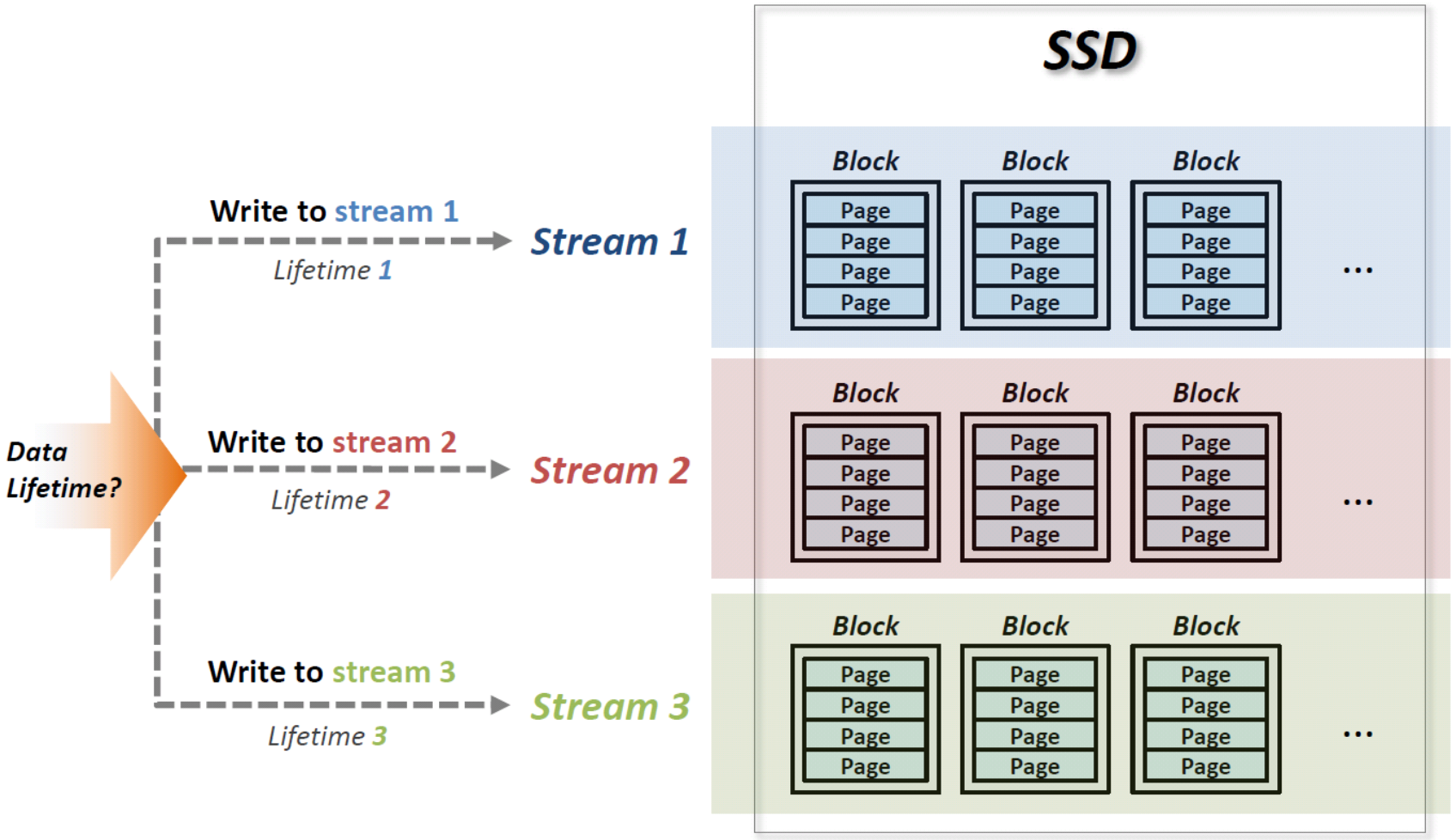
Some of slides are borrowed from the authors' presentation.

Effects of Write Patterns

- Previous write patterns (= current state) matter

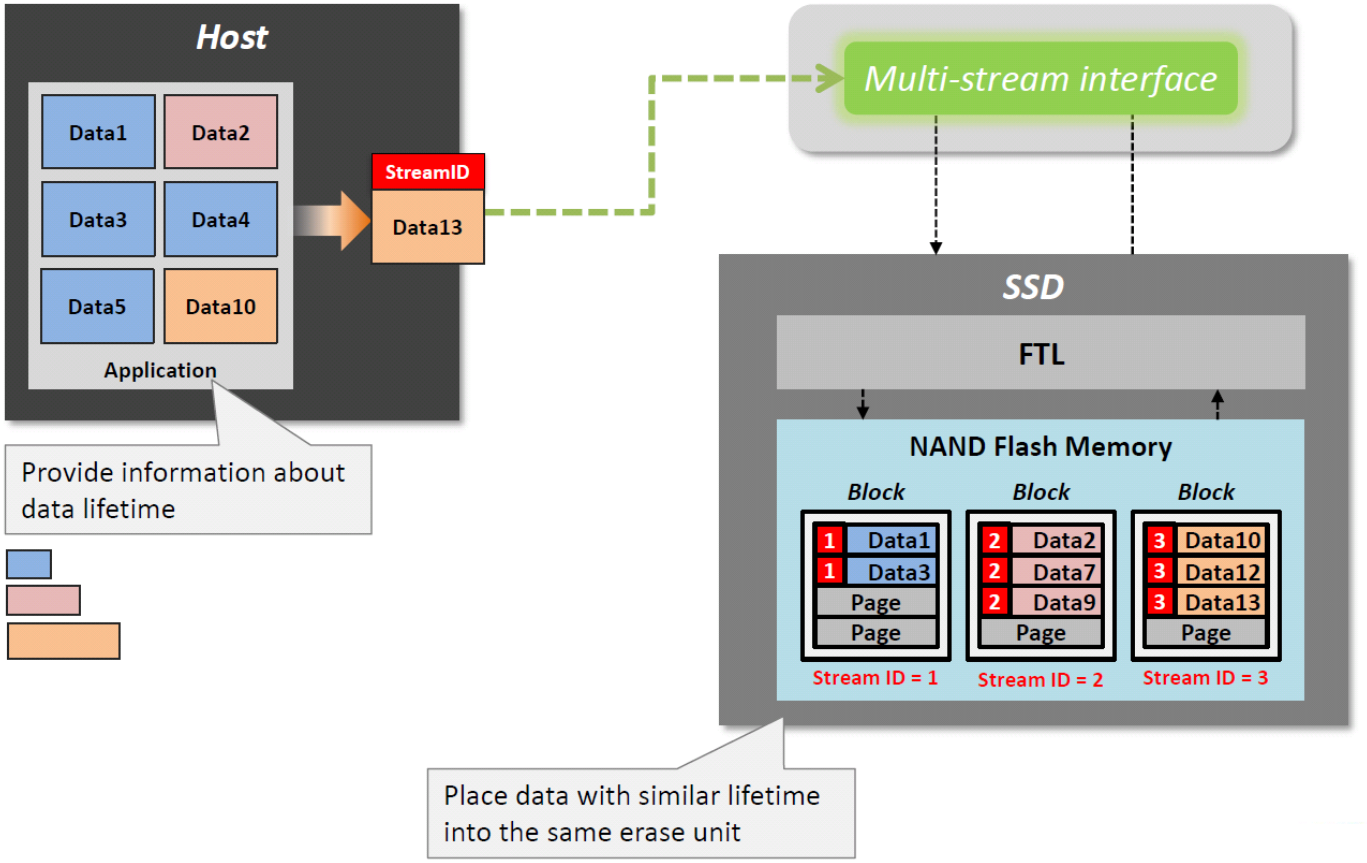


Stream


SSD
Data Lifetime?
Write to stream 1
Lifetime 1
Stream 1
Write to stream 2
Lifetime 2
Stream 2
Write to stream 3
Lifetime 3
Stream 3
Block
Page
...

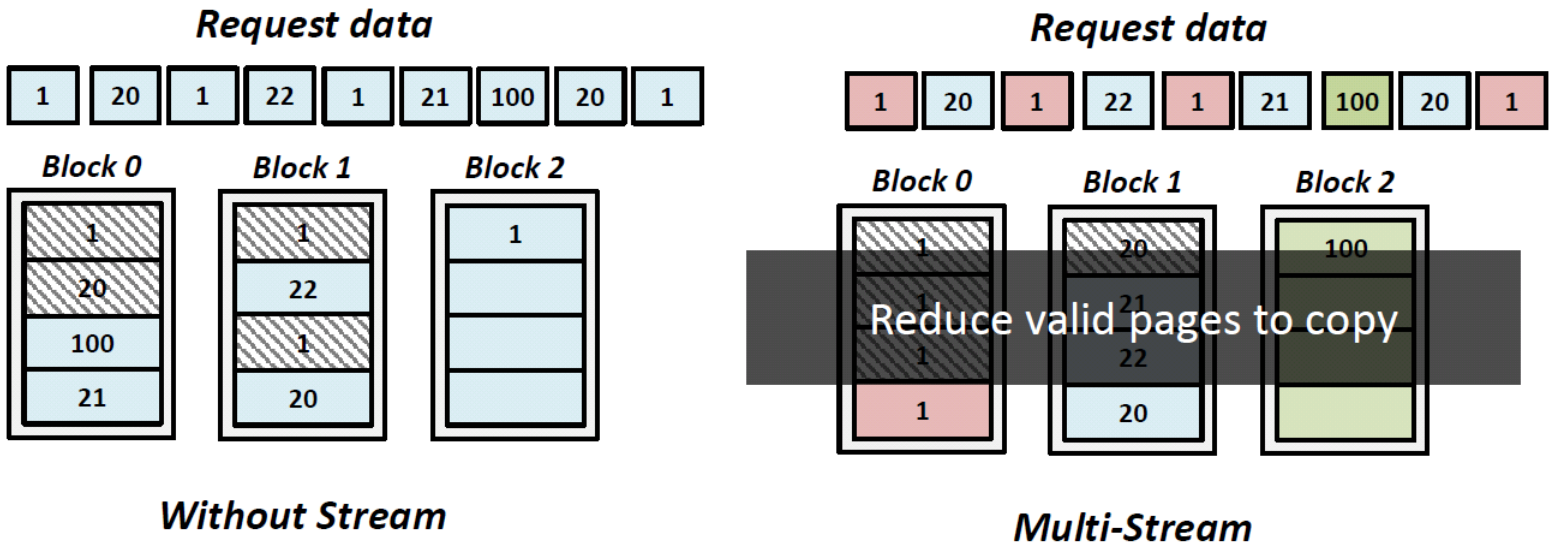
The Multi-streamed SSD

- Mapping data with different lifetime to different streams



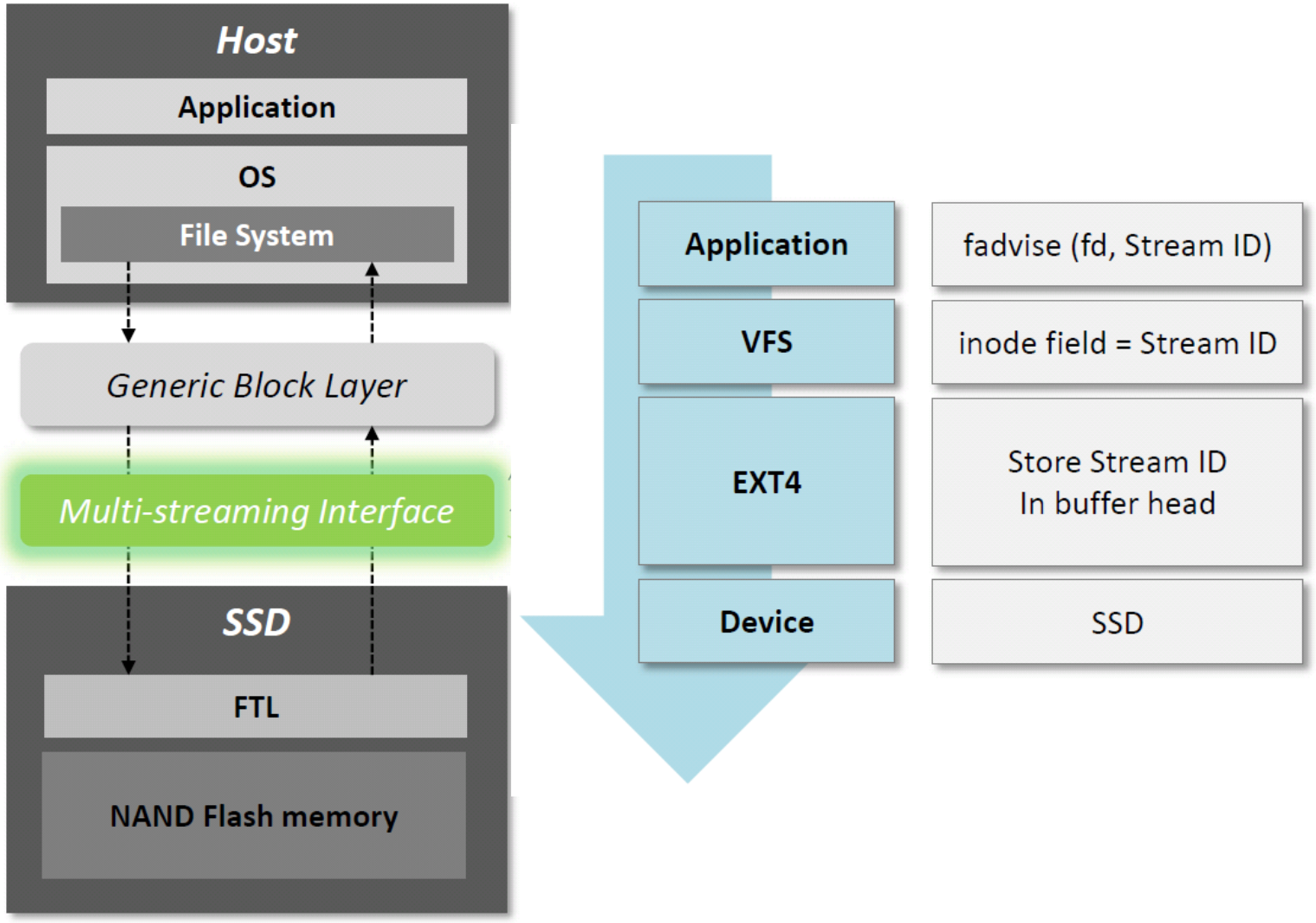
Working Example

- High GC efficiency → Performance improvement

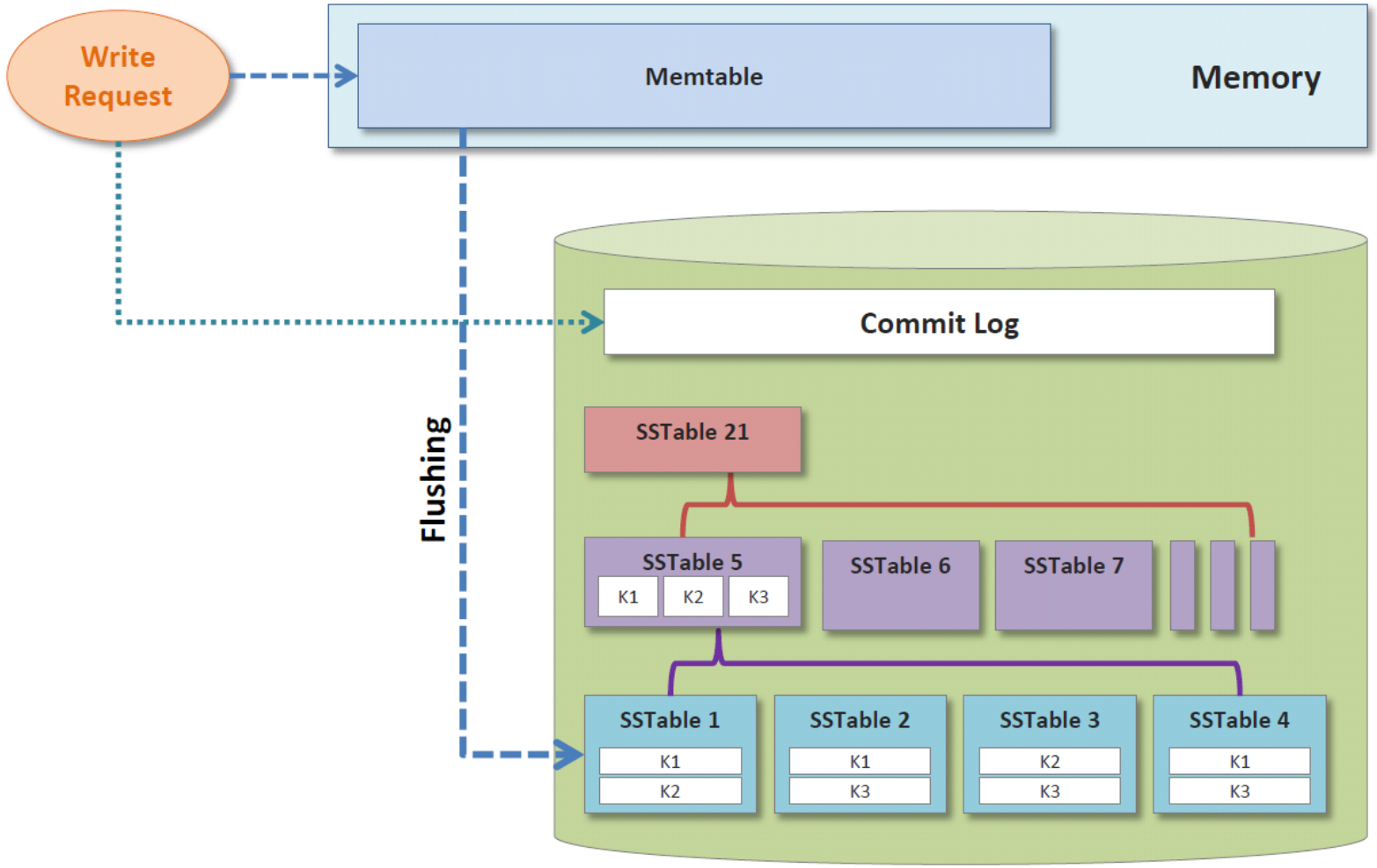


For effective multi-streaming,
proper mapping of data to streams is essential!

Architecture

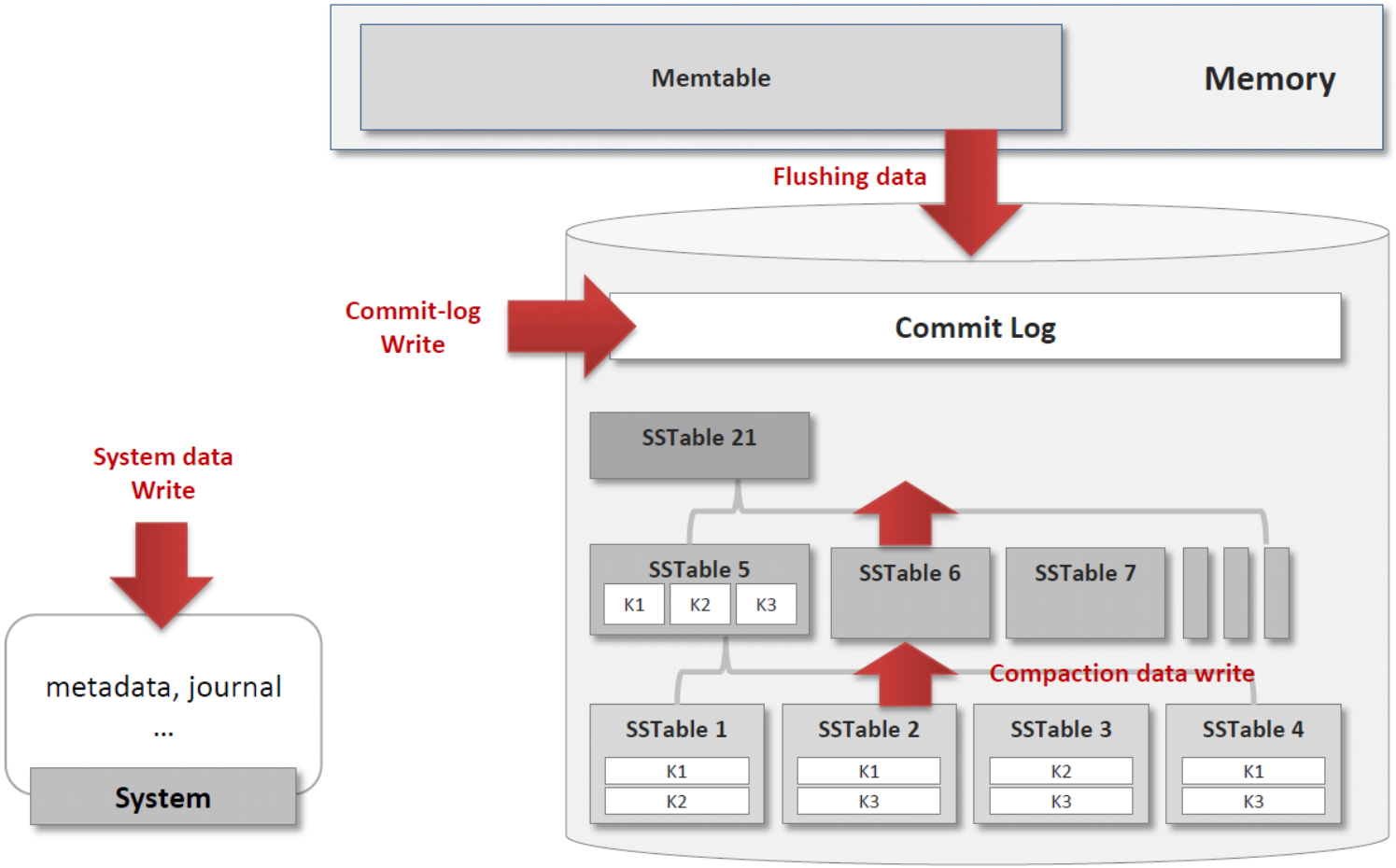


Case Study: Cassandra



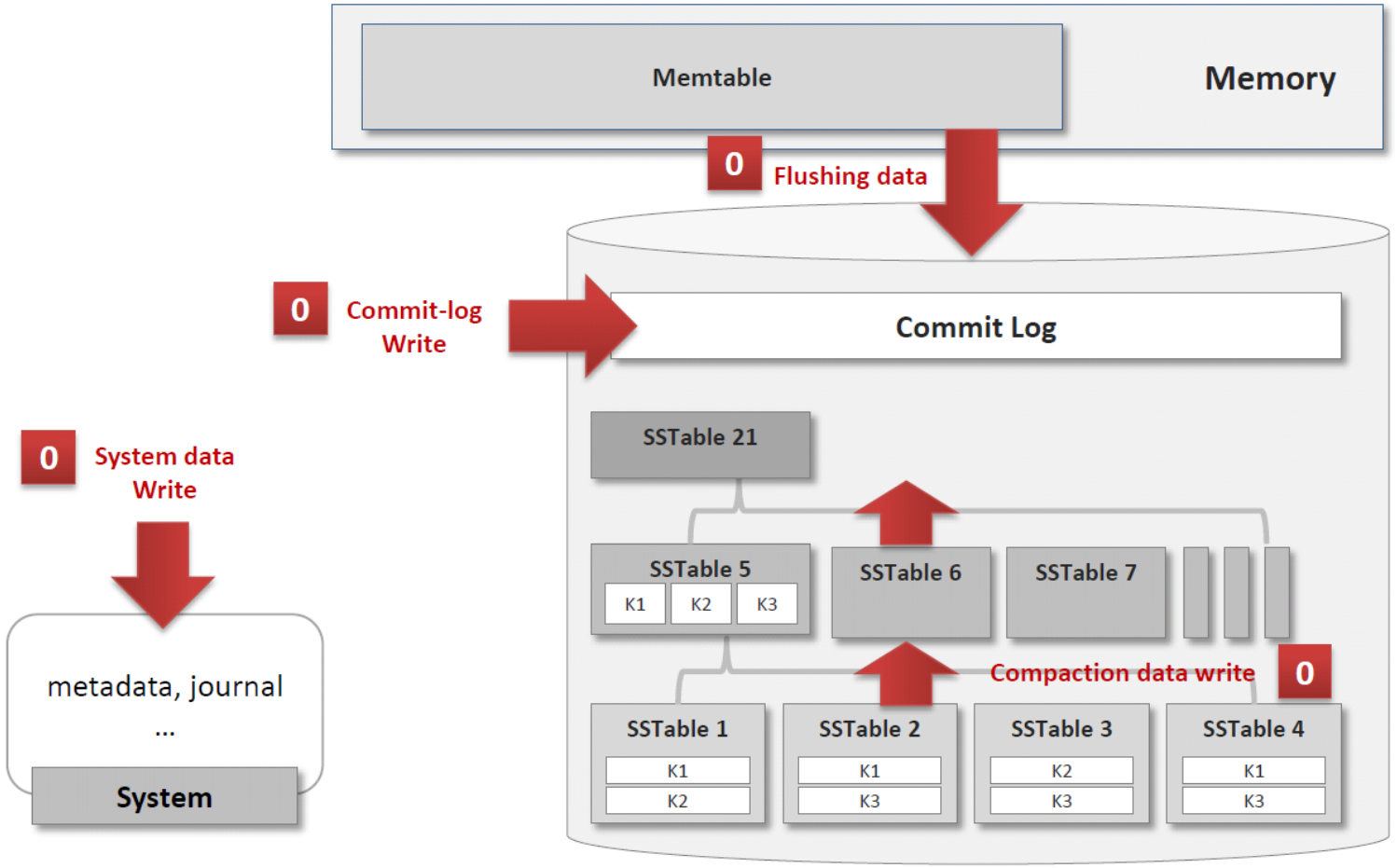
Cassandra's Write Patterns

- Write operations when Cassandra runs



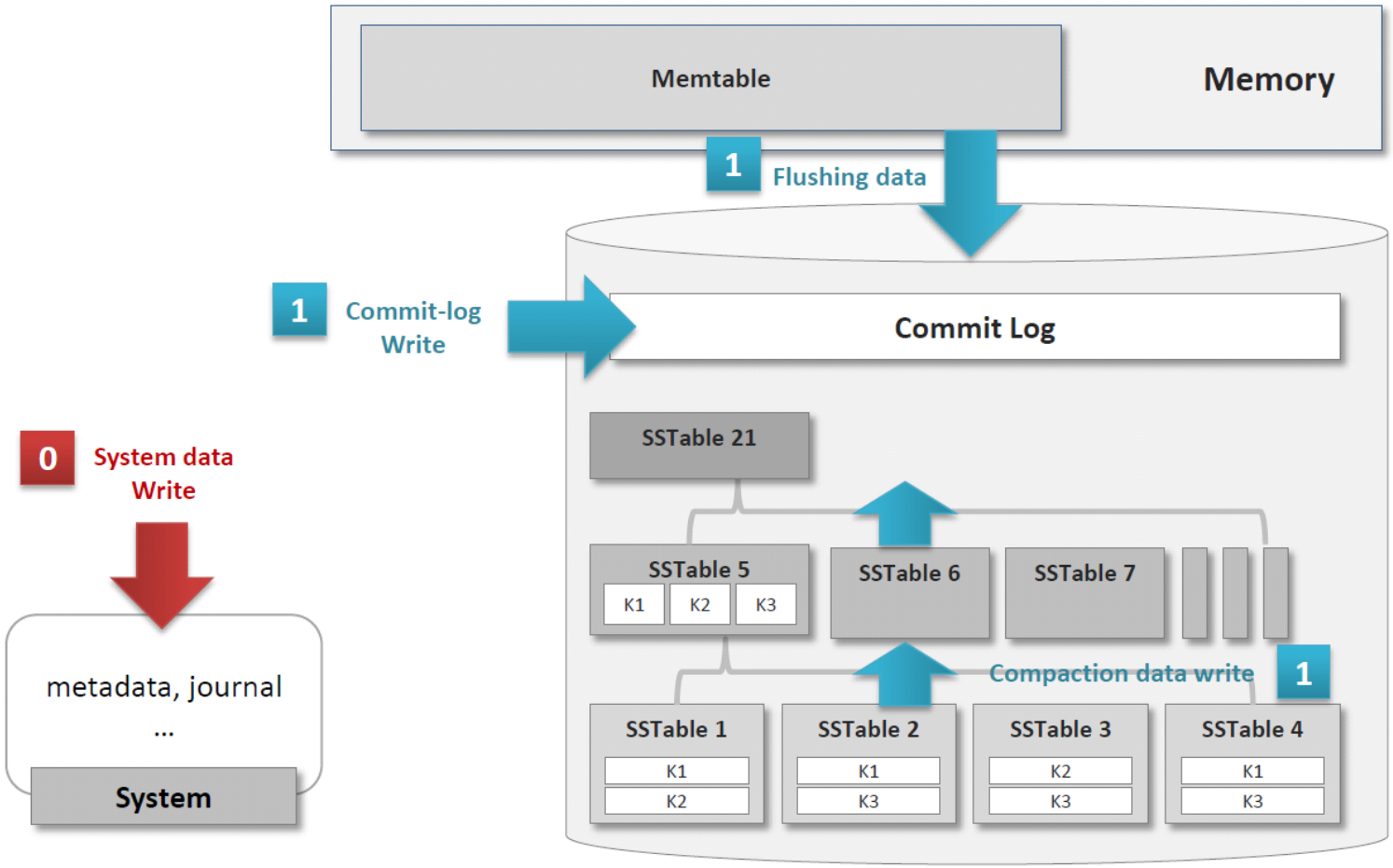
Mapping #1: Conventional

- Just one stream ID (= conventional SSD)



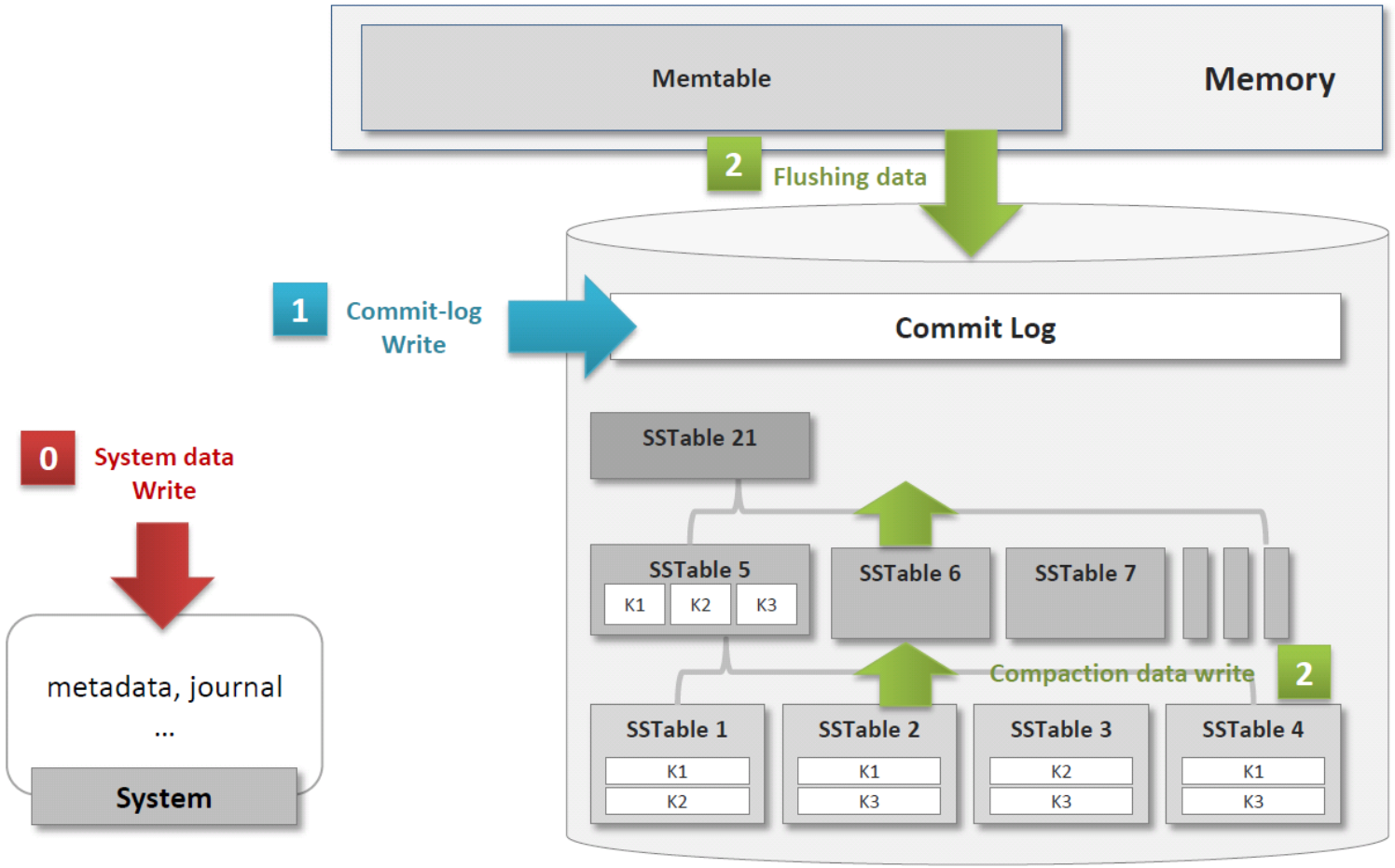
Mapping #2: Multi-App

- Separate application writes (ID 1) from system traffic (ID 0)



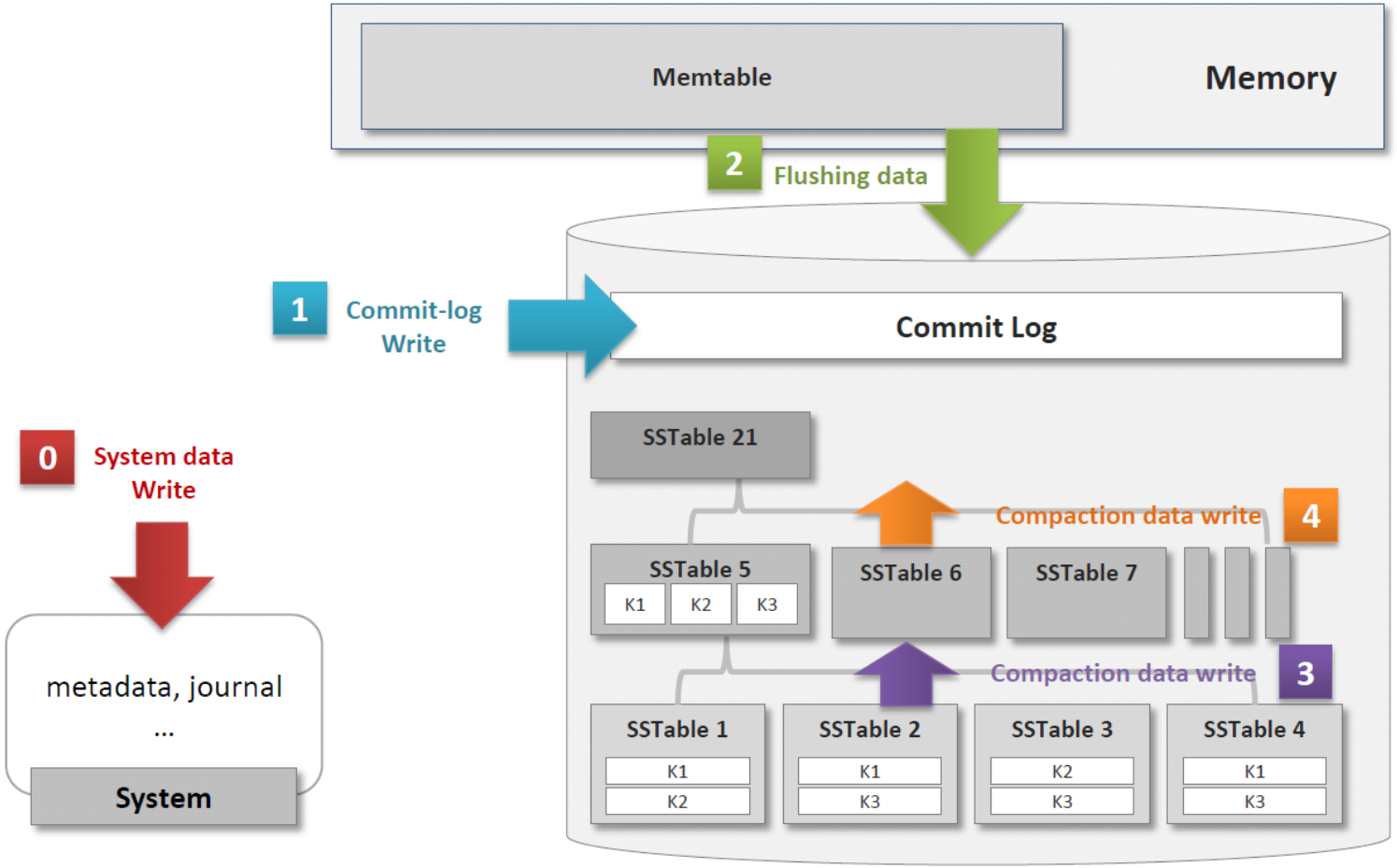
Mapping #3: Multi-Log

- Use three streams; further separate Commit Log



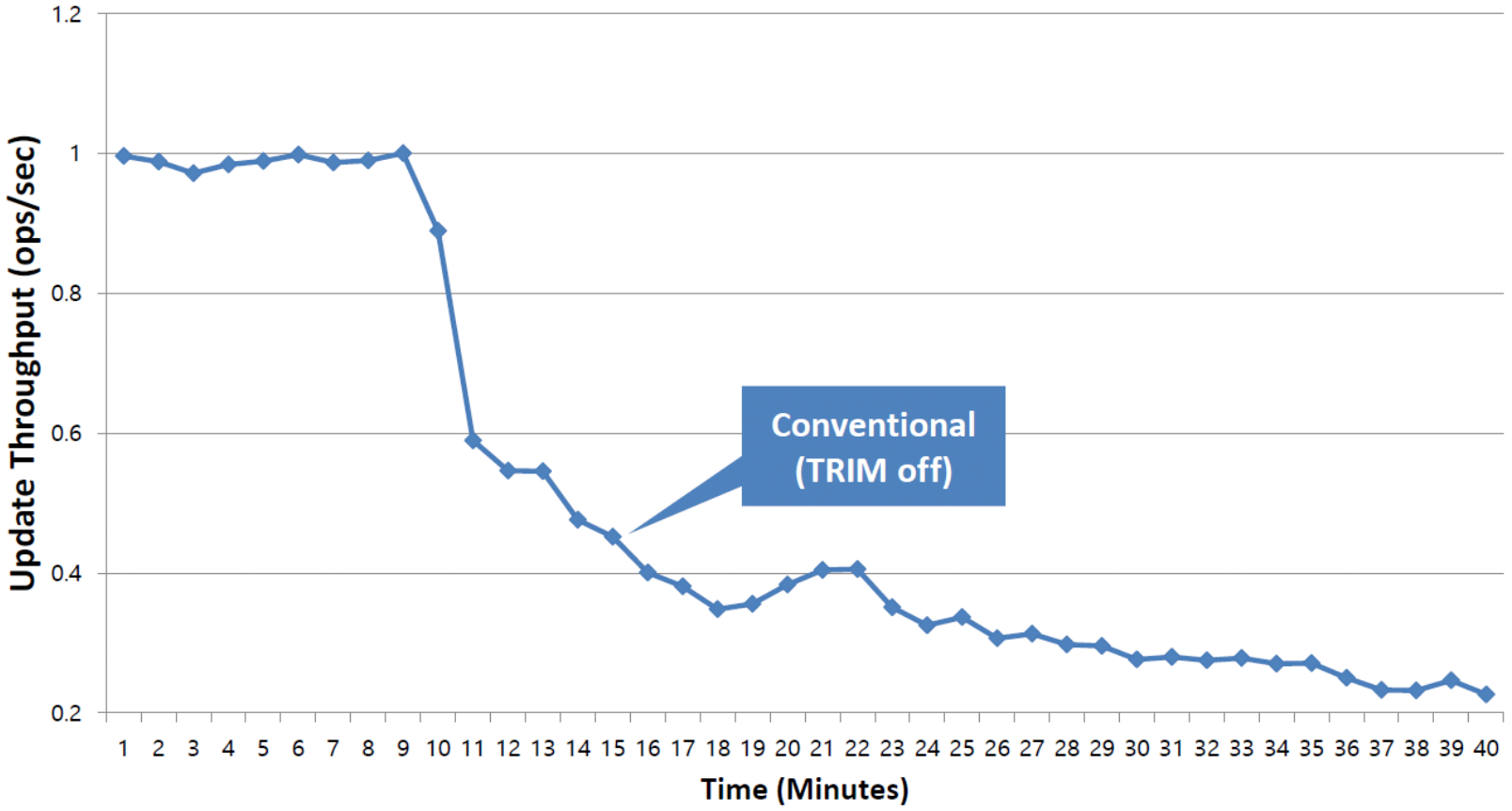
Mapping #4: Multi-Data

- Give distinct streams to different tiers of SSTables



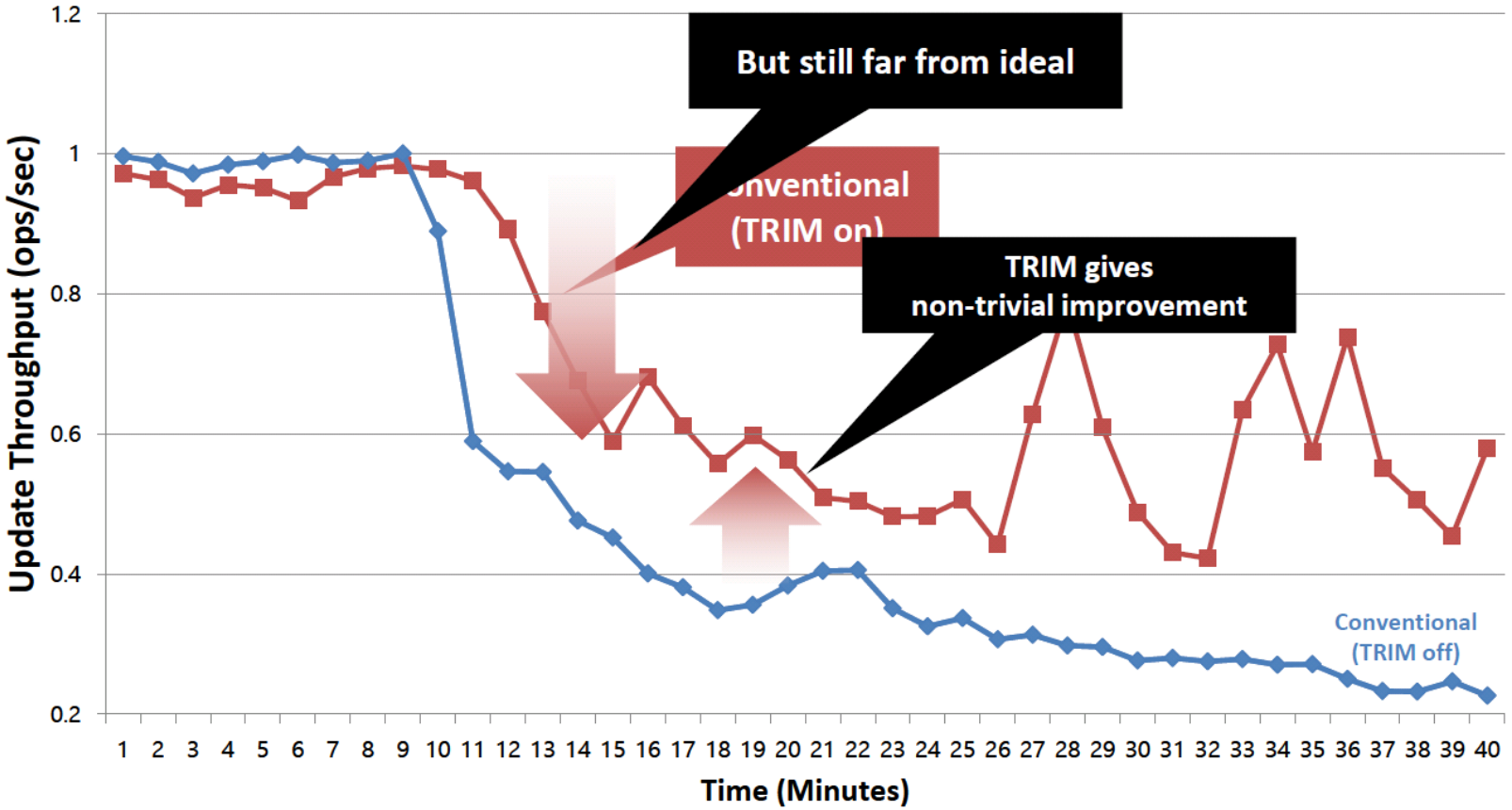
Results: Conventional

- Cassandra's normalized update throughput
  - Conventional "TRIM off"



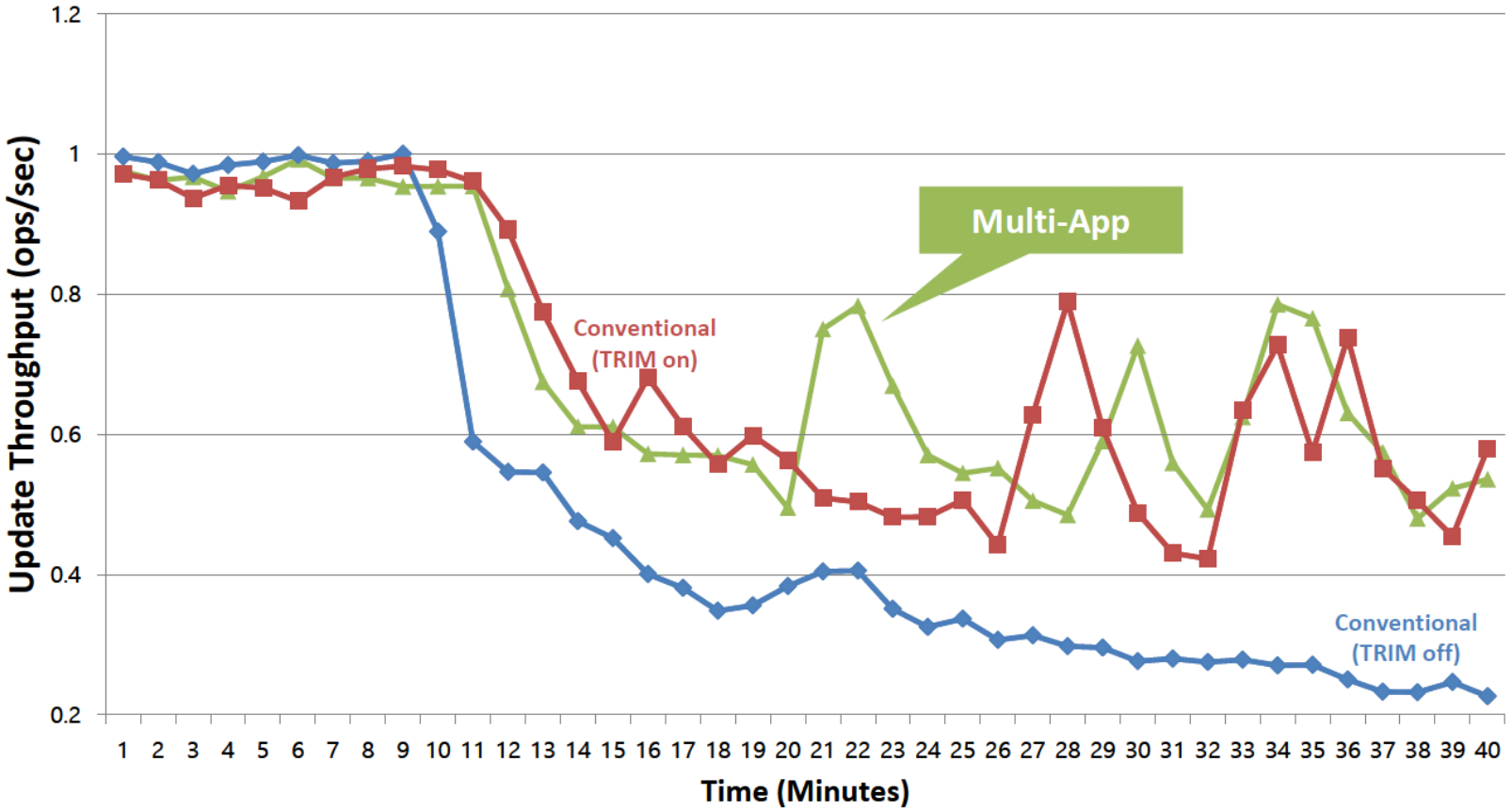
Results: Conventional with TRIM

- Cassandra's normalized update throughput
  - Conventional "TRIM on"



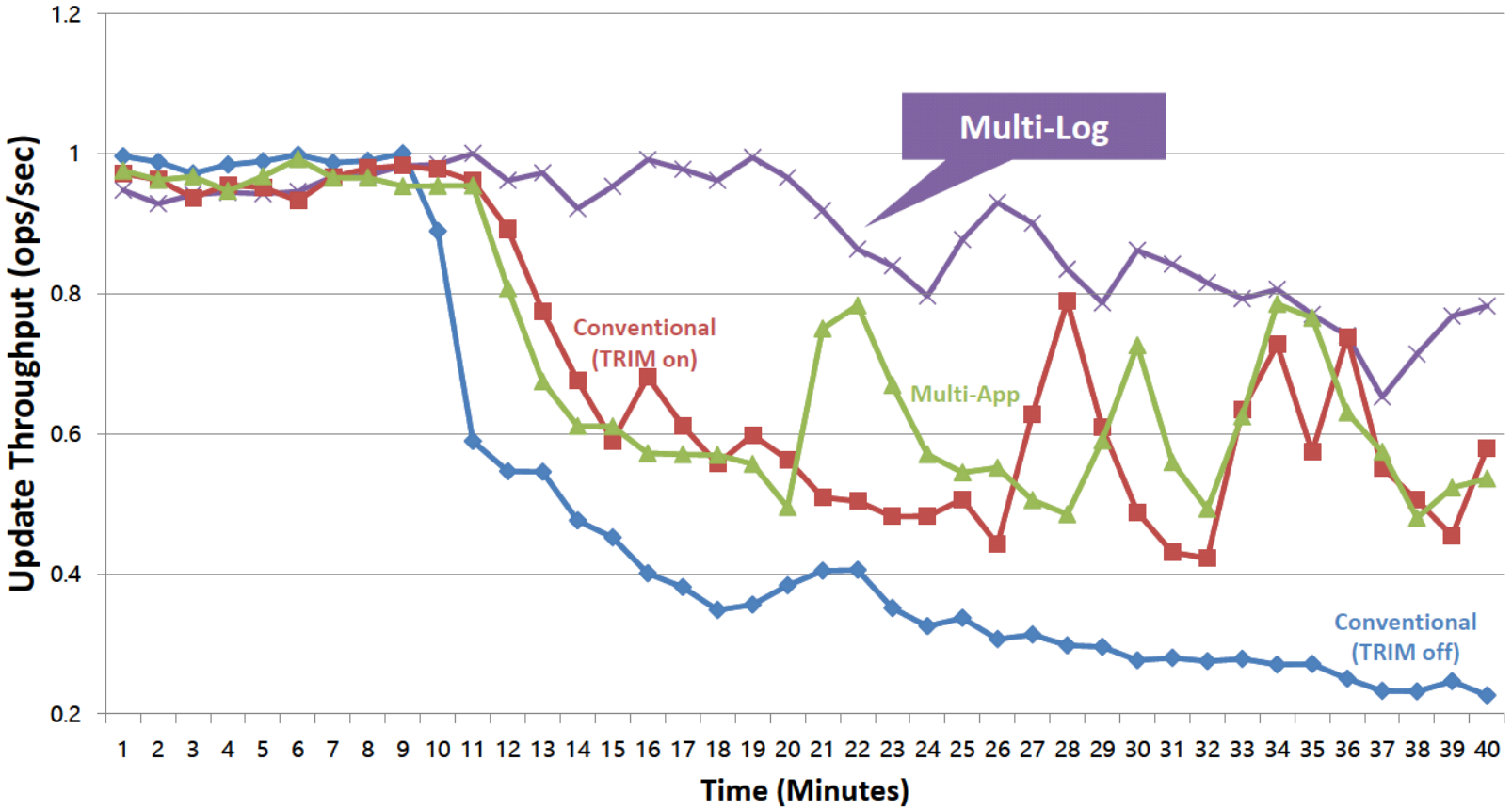
Results: Multi-App

- Cassandra's normalized update throughput
  - "Multi-App" (System data vs. Cassandra data)



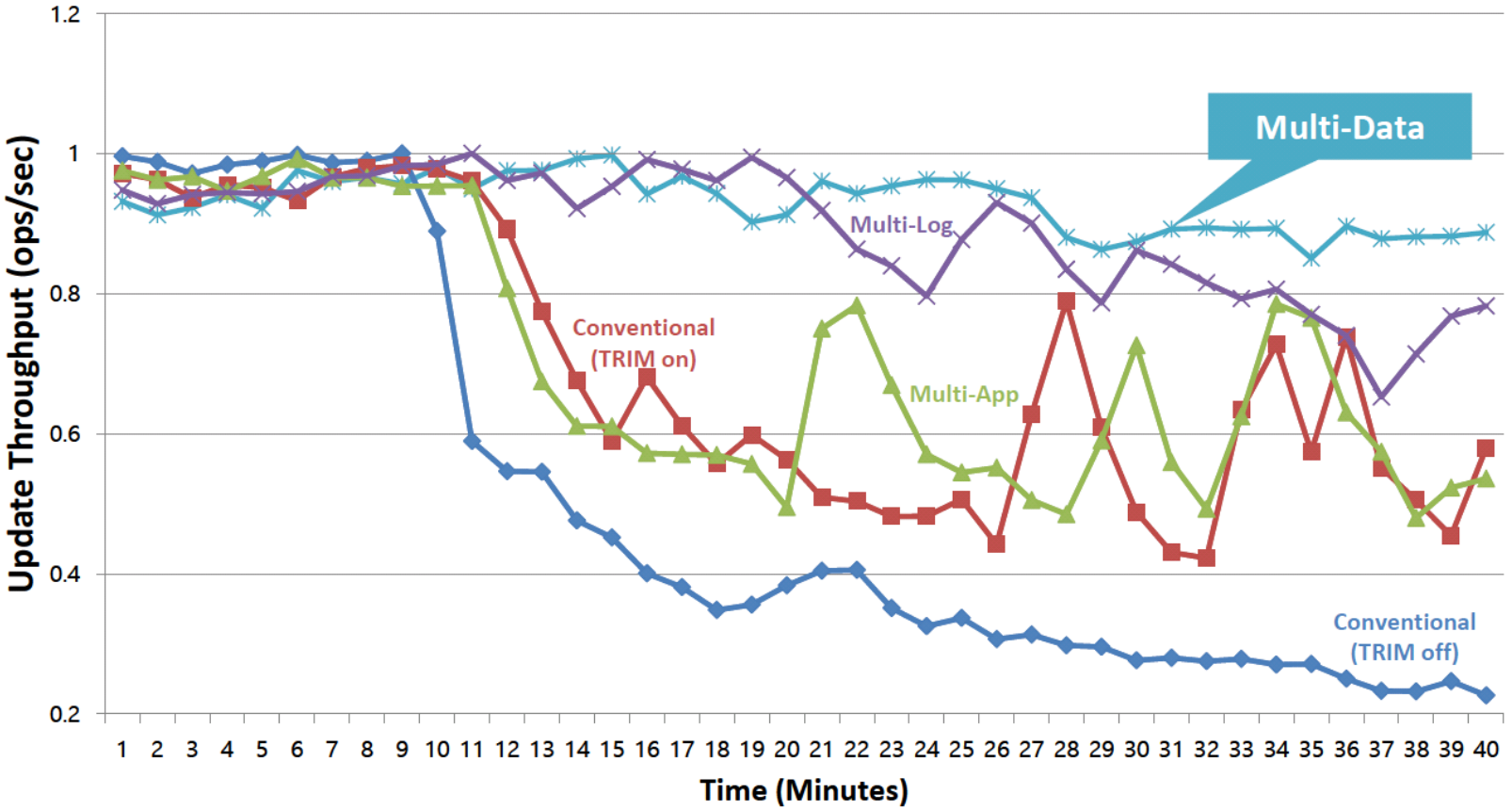
Results: Multi-Log

- Cassandra's normalized update throughput
  - "Multi-Log" (System data vs. Commit-Log vs. Flushed data)



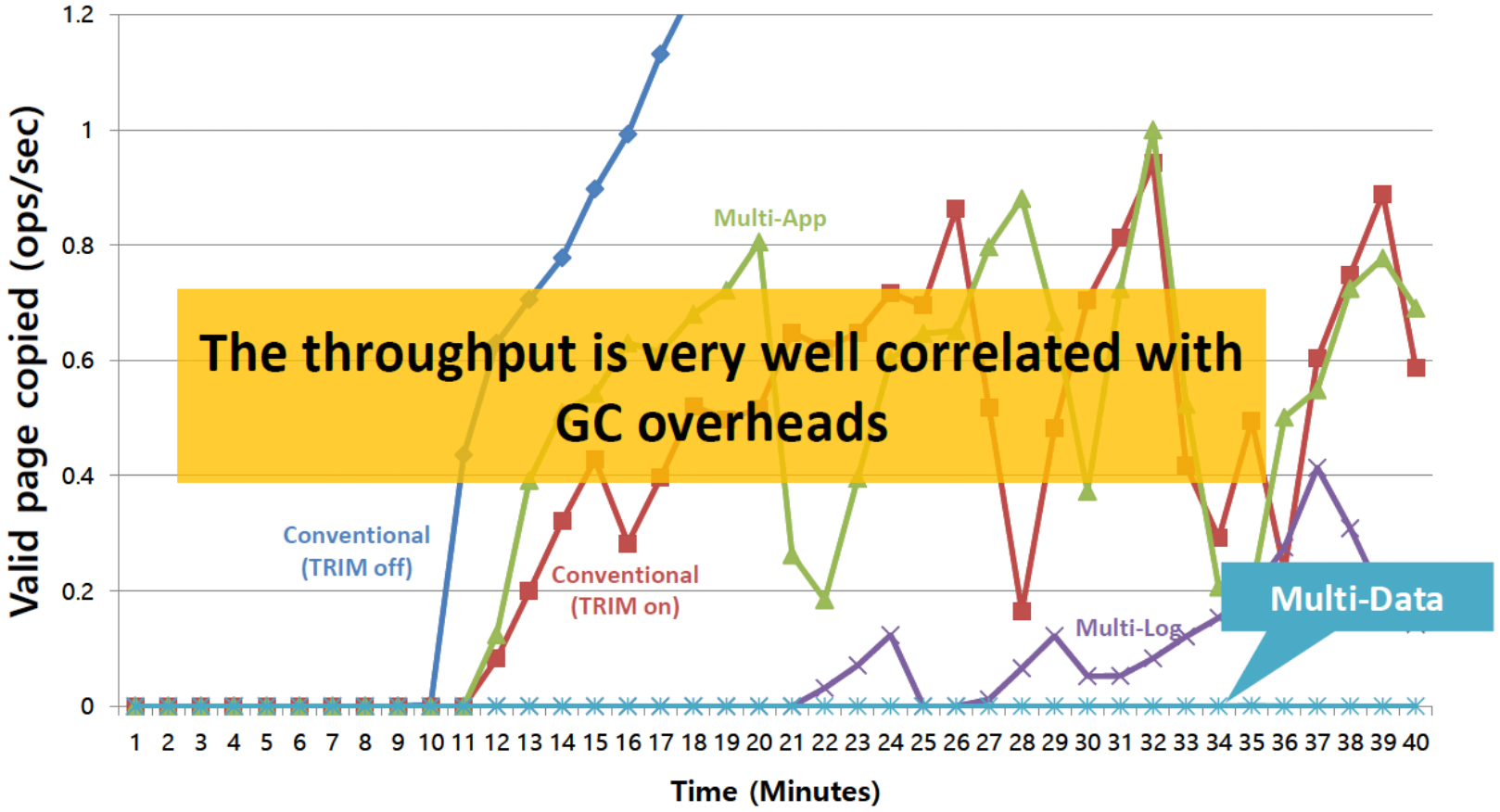
Results: Multi-Data

- Cassandra's normalized update throughput
  - "Multi-Data" (System data vs. Commit-Log vs. Flushed data vs. Compaction Data)



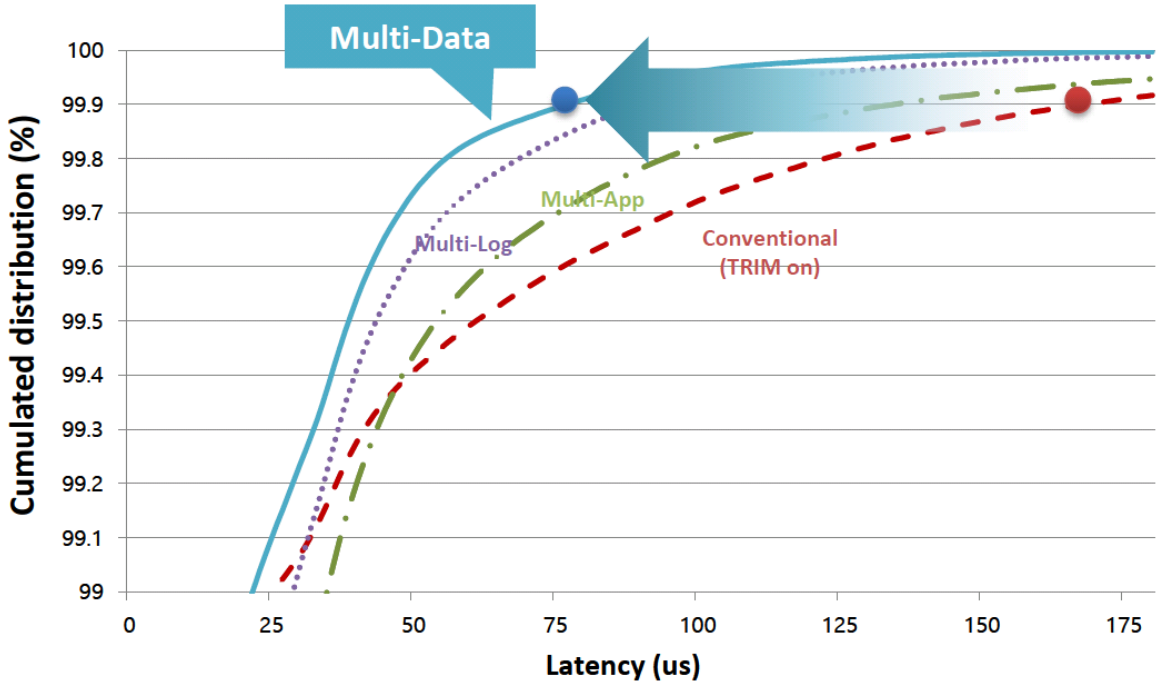
Results: GC Overheads

- Cassandra's GC overheads



Results: Latency

- Cassandra's cumulated latency distribution
  - Multi-streaming improves write latency
  - At 99.9%, Multi-Data lowers the latency by 53% compared to Normal



Summary

- Mapping application and system data with different lifetimes to SSD streams
  - Higher GC efficiency, lower latency
- Multi-streaming can be supported on a state-of-the-art SSD and co-exist with the traditional block interface
- Standardized in T10 SCSI (SAS SSDs) in 2015
- Standardized in NVMe 1.3 in 2017

Open-Channel SSD (OCSSD)

Why OCSSD?

I/O Isolation

Enable I/O isolation between tenants by allocating your SSD into separate parallel units.

Predictable Latency

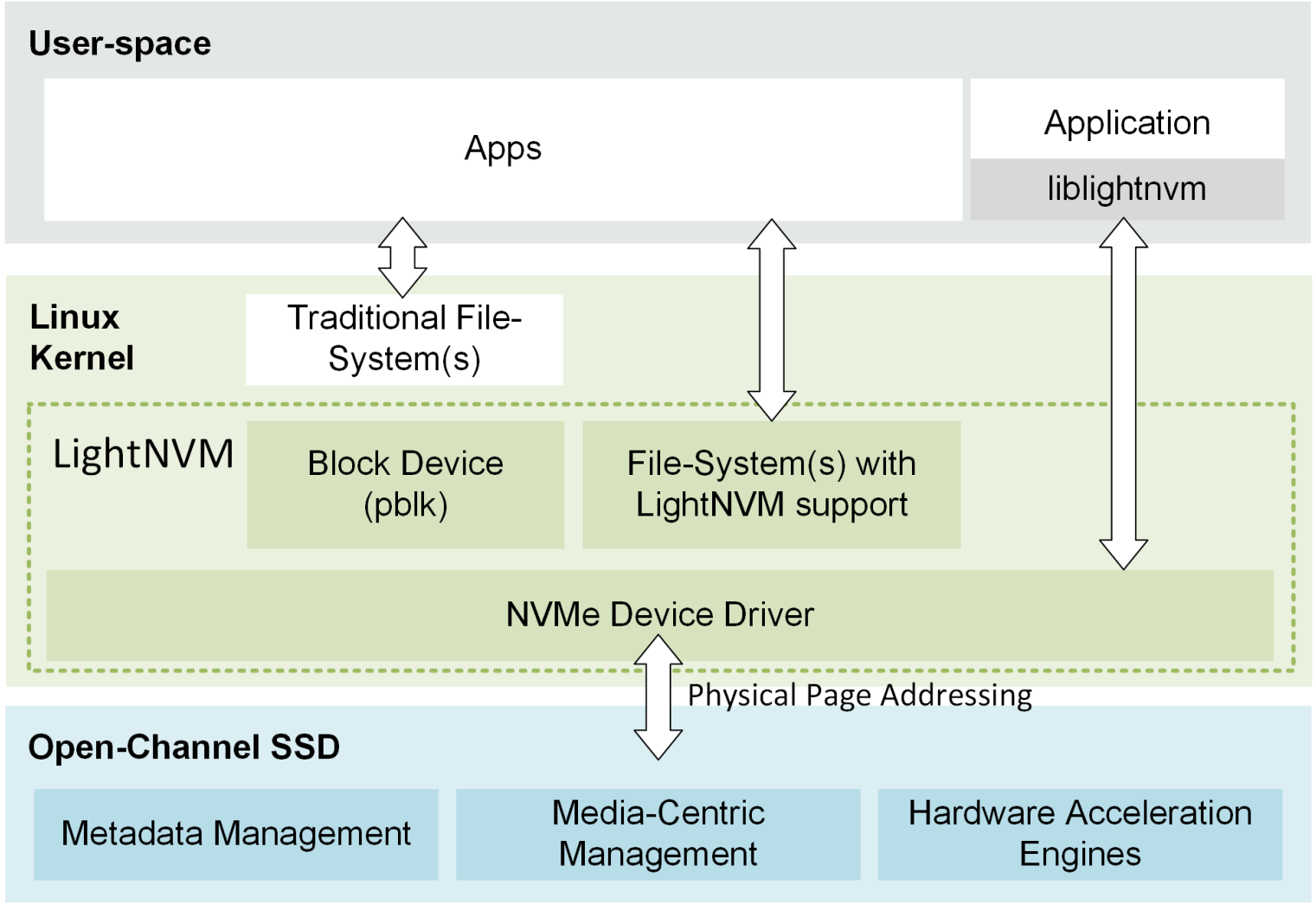
No more guessing when an IO completes. You know which parallel unit is accessed on disk.

Data Placement & I/O Scheduling

Manage your non-volatile memory as a block device, through a file-system or inside your application.

https://platformlab.stanford.edu/Seminar%20Talks/Matias_Bj_rling___Javier_Gonzales.pdf

OCSSD Architecture

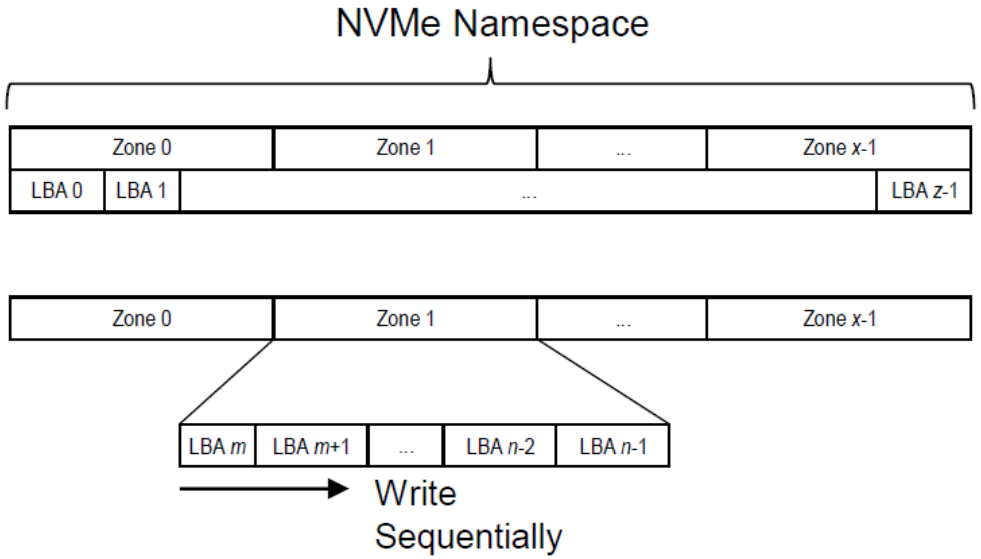


Zoned Namespace (ZNS) SSDs

Source: Matias Bjørling, How Zoned Namespaces Improve SSD Lifetime, Throughput, and Latency, FMS, 2020.

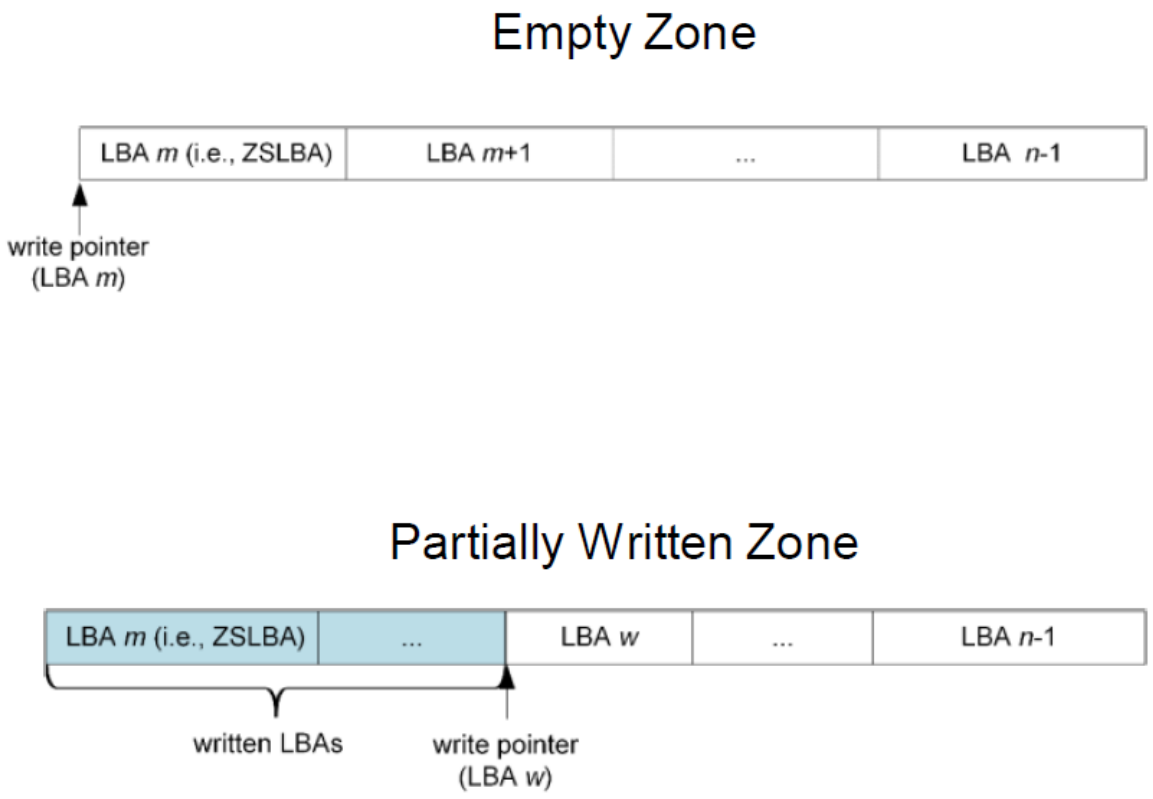
Zoned Storage Model

- Zones are laid out sequentially in an NVMe namespace
- The zone size is fixed and applies to all zones in the namespace
  - e.g., 512 MiB
- The command set inherits the NVMe Command Set
  - Built upon the conventional block interface (Read, Write, Flush and other commands)
  - Adds rules to collaborate on host and device data placement



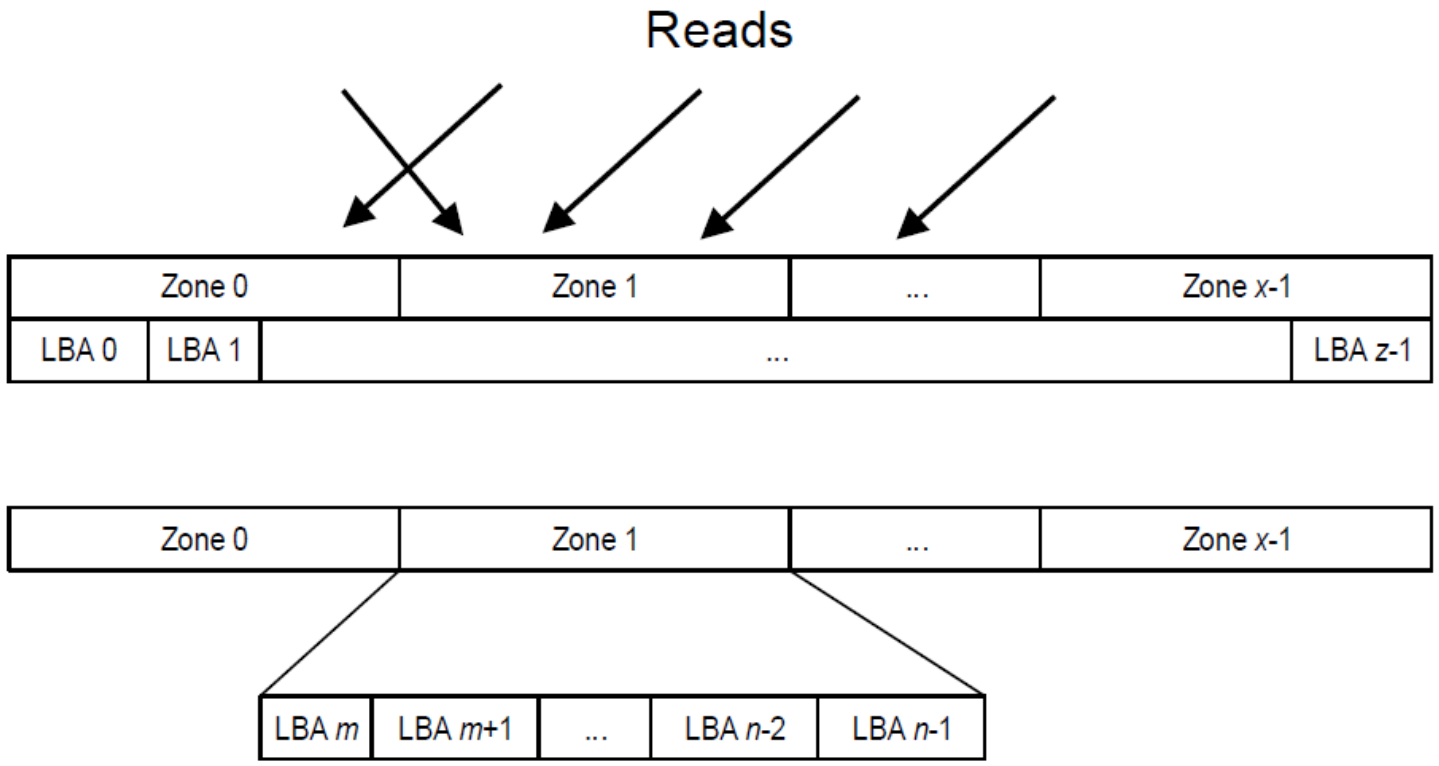
Writing to a Zone

- "Sequential Write Required"
  - Must be written sequentially
  - Must be reset if written to again
- Each zone has a set of associated attributes:
  - Write pointer
  - Zone Starting LBA
  - Zone Capacity
  - Zone state
- Very similar to writing zones with host-managed SMR HDDs



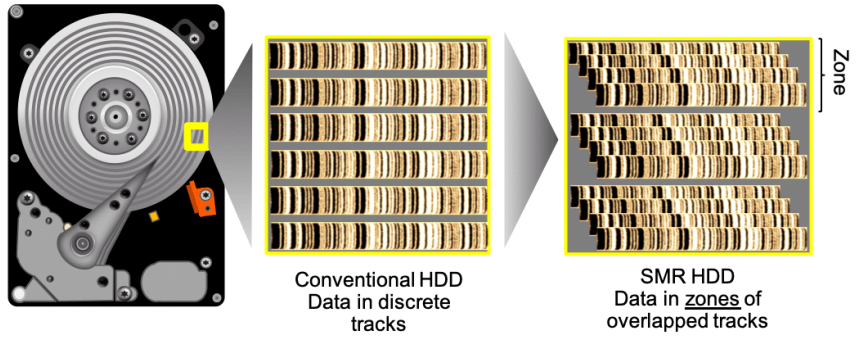
Reading from a Zone

- Writes are required to be sequential within a zone
- Reads may be issued to any LBA within a zone and in any order



SMR HDDs and ZNS SSDs

- Host-managed SMR HDDs
  - Implements the SMR (ZAC/ZBC) specifications
  - ZAC: Zoned Device ATA Command Set in T13/SATA
  - ZBC: Zoned Block Commands in T10/SAS

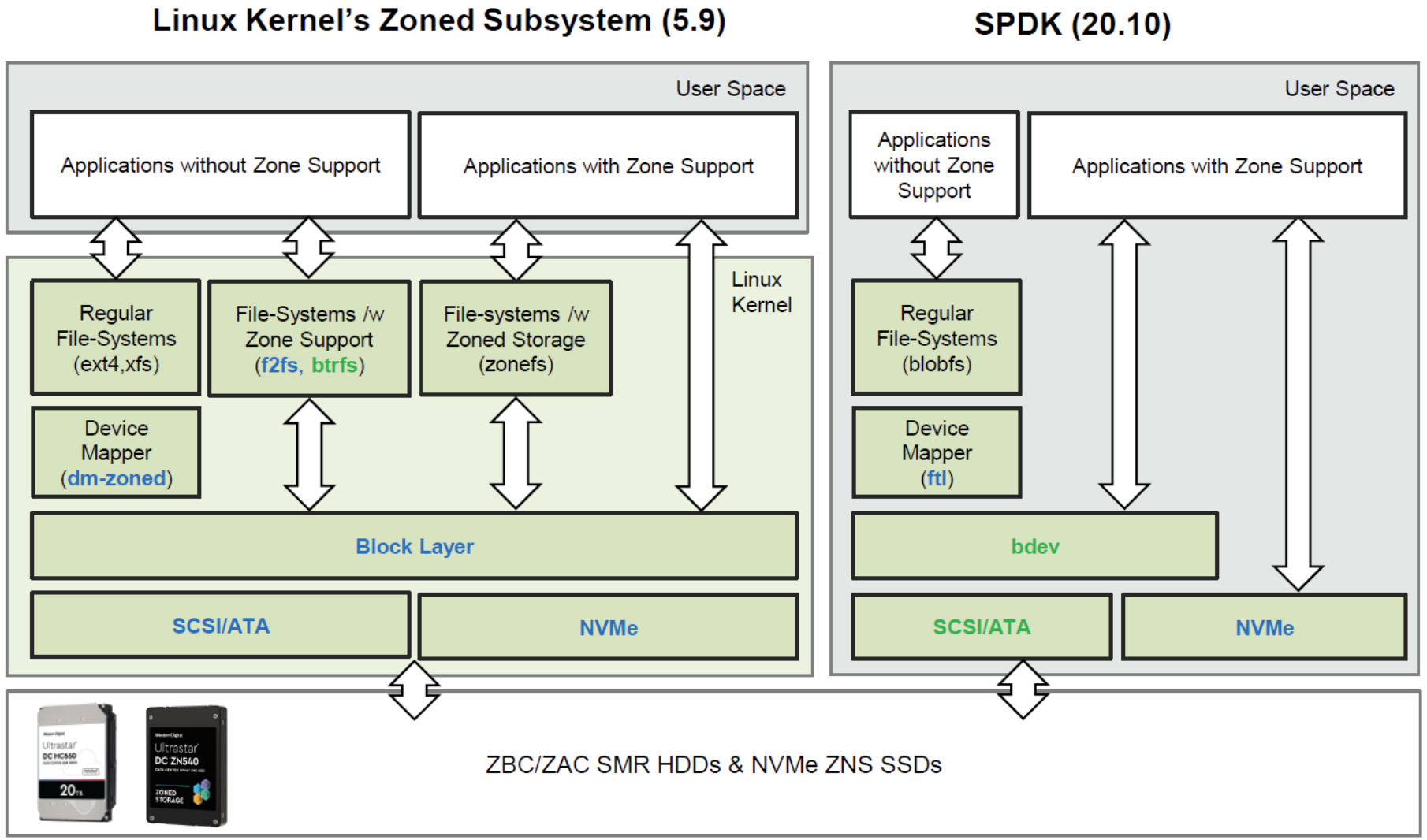


- NVMe ZNS SSDs
  - Implements the Zoned Namespace Command Set specification
  - Aligns with ZAC/ZBC to allow interoperability

- A single unified software stack support both storage types
  - Utilizes the already mature Linux storage stack built for SMR HDDs

Image from https://blog.westerndigital.com/what-is-zoned-storage-initiative/

Zoned Storage Software Stack



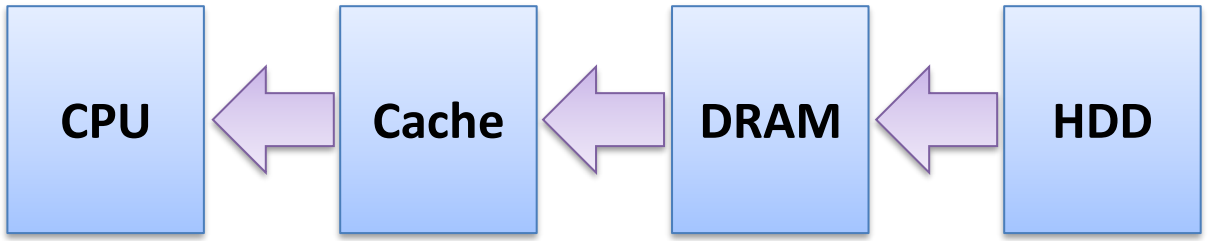
Biscuit: A Framework for Near-Data Processing of Big Data Workloads

(B. Gu et al., ISCA, 2016)

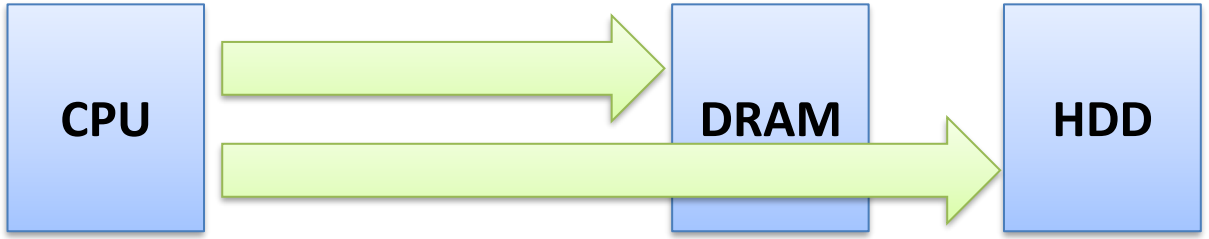
Some of slides are borrowed from the authors' presentation.

Moving Data

- In memory hierarchy
  - Move data toward ALU to remedy long latency while accessing high-locality data

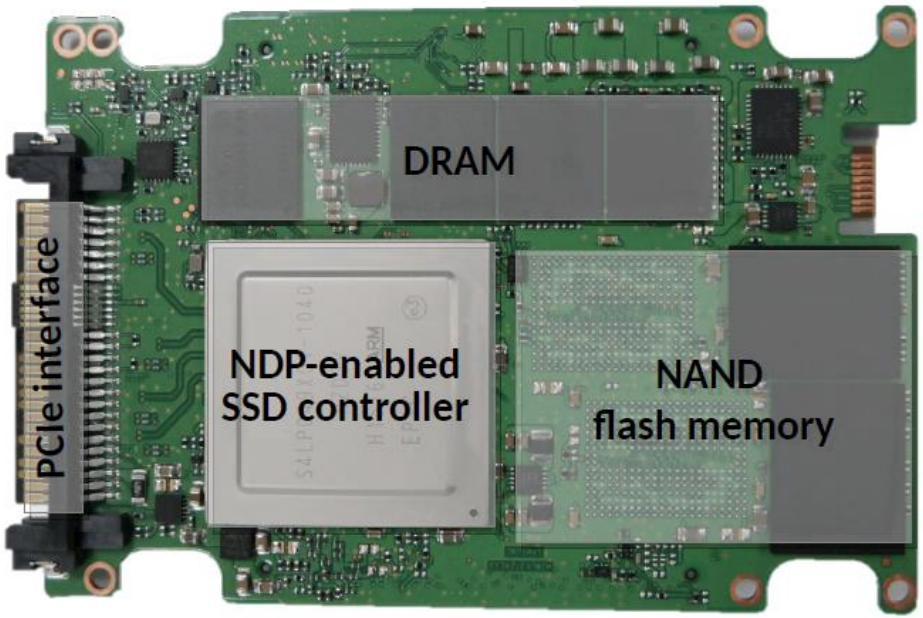


- In computation hierarchy
  - Move computation toward memory to remedy long latency while accessing low-locality data



Hardware

- SSD with a user-programmable NDP (Near-Data Processing) framework



[Inside of PM1725]

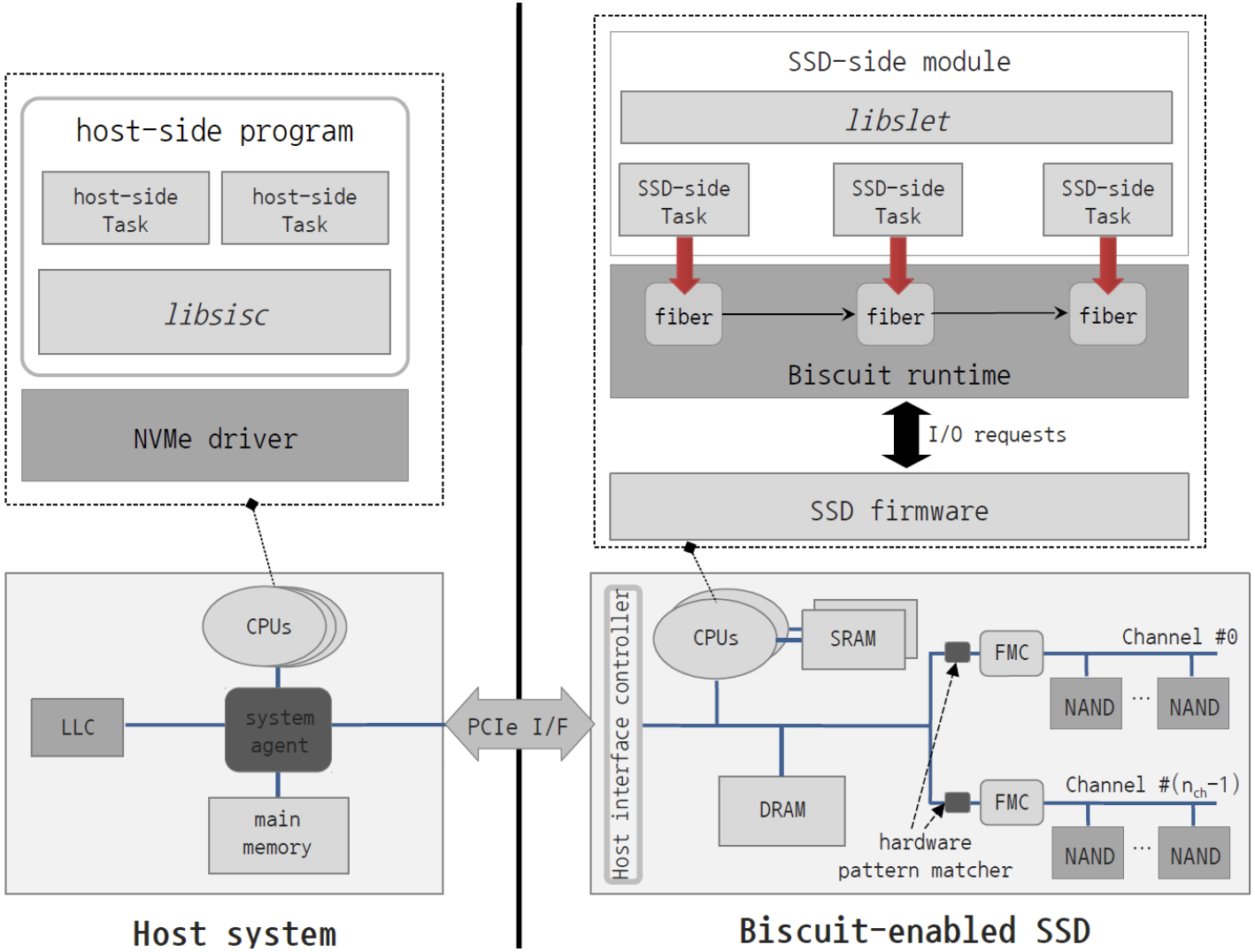
Item	Description
Host interface	PCIe Gen.3 ×4 (3.2 GB/s)
Protocol	NVMe 1.1
Device density	1 TB
SSD architecture	Multiple channels/ways/cores
Storage medium	Multi-bit NAND flash memory
Compute resources for Biscuit	Two ARM Cortex R7 cores @750MHz with MPU
On-chip SRAM	< 1 MiB
DRAM	≥ 1 GiB

- Hardware pattern matcher on each flash memory channel

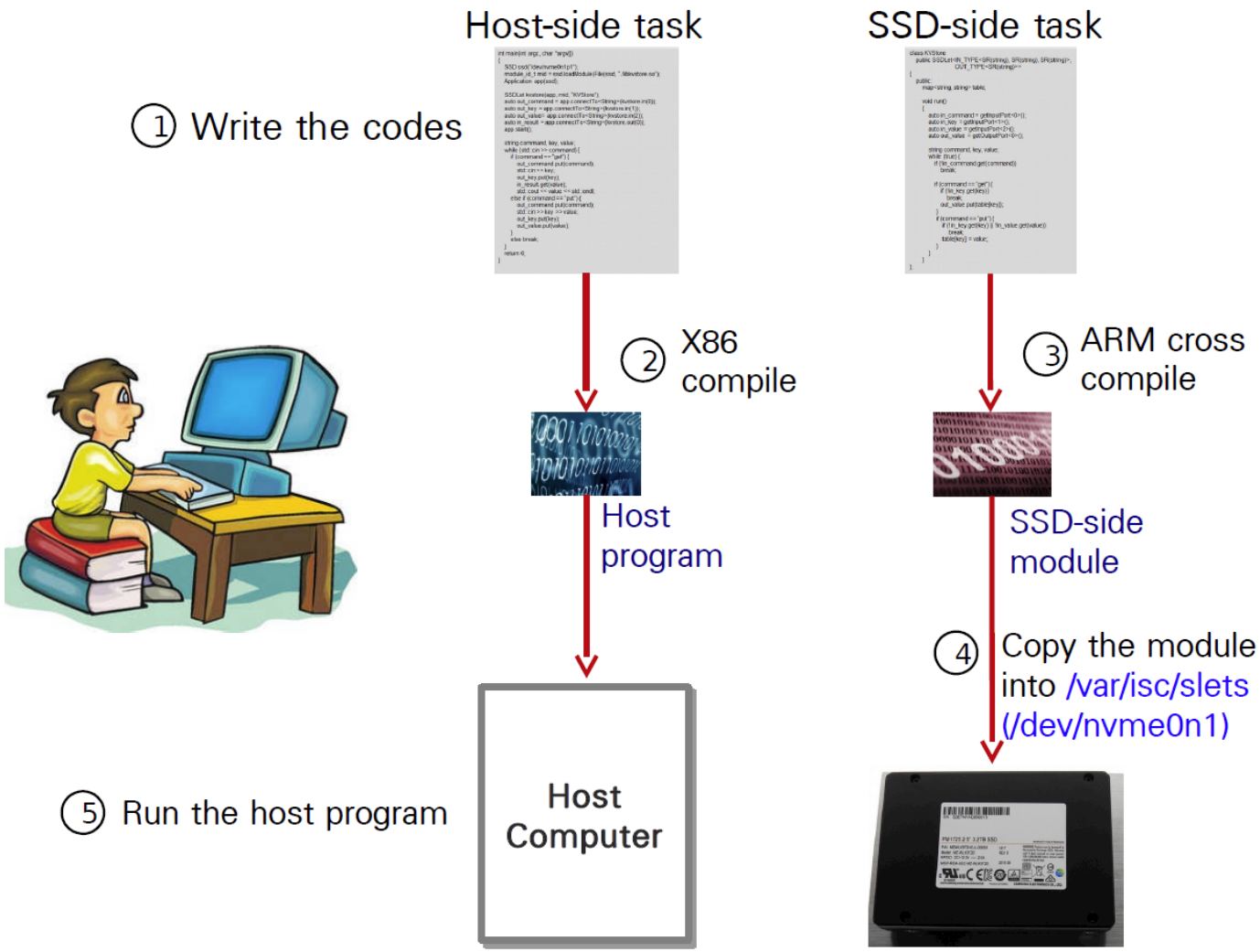
Runtime

- Cooperative multithreading
  - A limited form of multithreading (fiber as a scheduling unit)
  - Less context switch overhead
  - Safe resource sharing without locking
- Shared nothing architecture
  - All data transmission among threads through I/O ports
  - Enforced by the programming model and APIs
- Dynamic loader for user programs
  - User program as position-independent code (PIC)
  - Symbol relocation to locate each program in a separate address space

Biscuit System Architecture

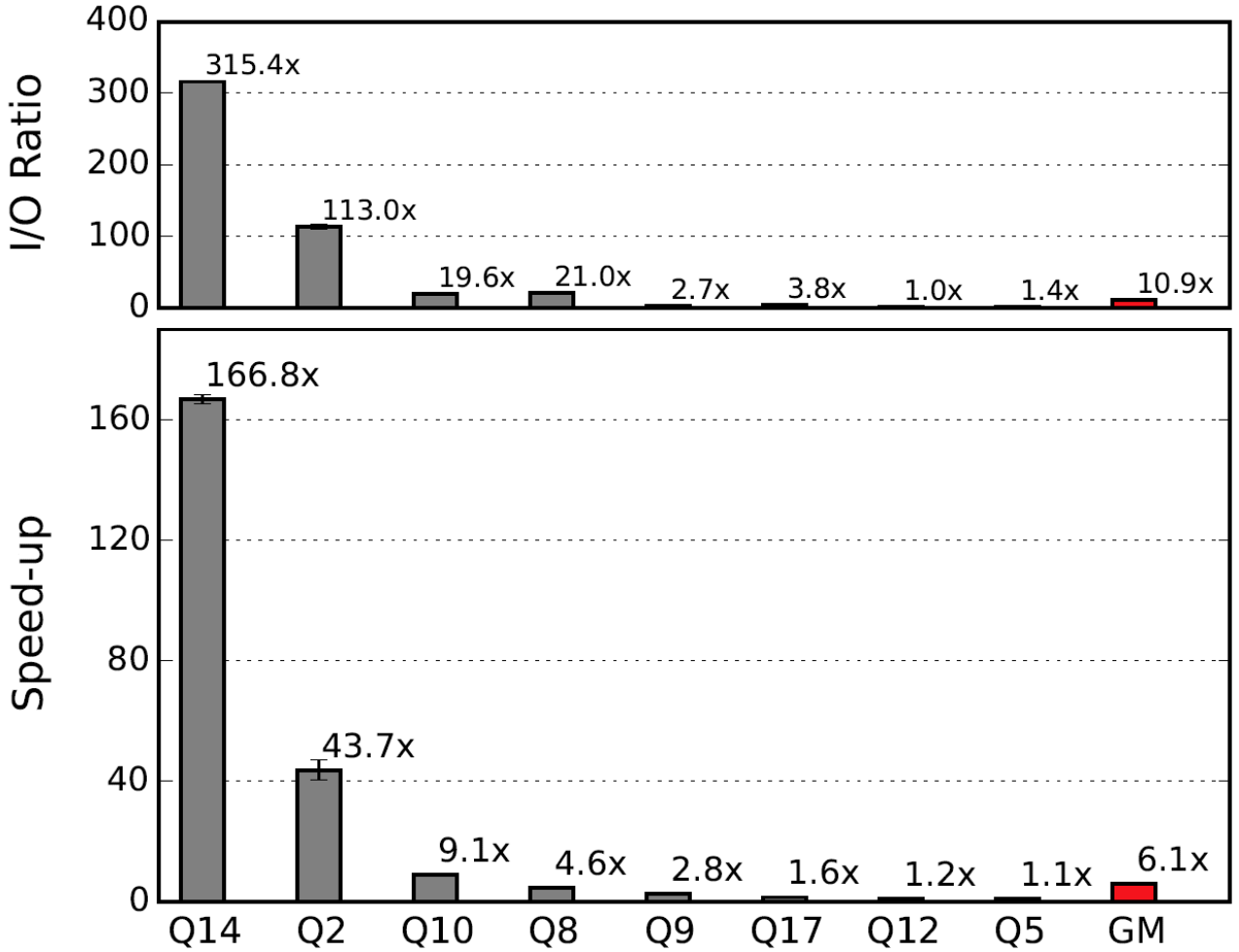


Development Process



TPC-H Results

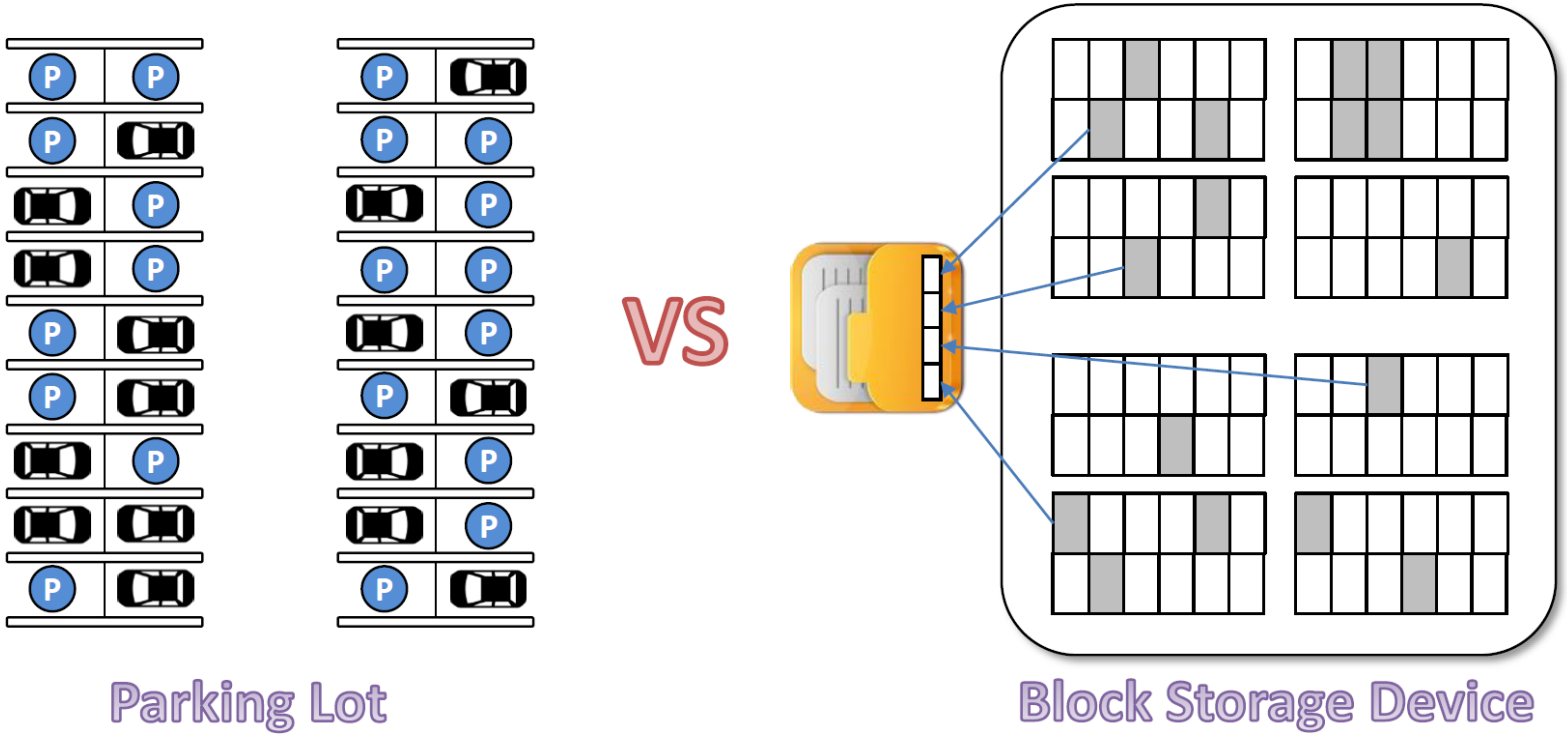
- Data analytics on MariaDB
  - TPC-H dataset at a scale factor of 100 (160GiB)
  - 6.1x on average
  - 3.6x speedup for running all queries (two days vs. 13 hours)



Key-Value SSD (KVSSD)

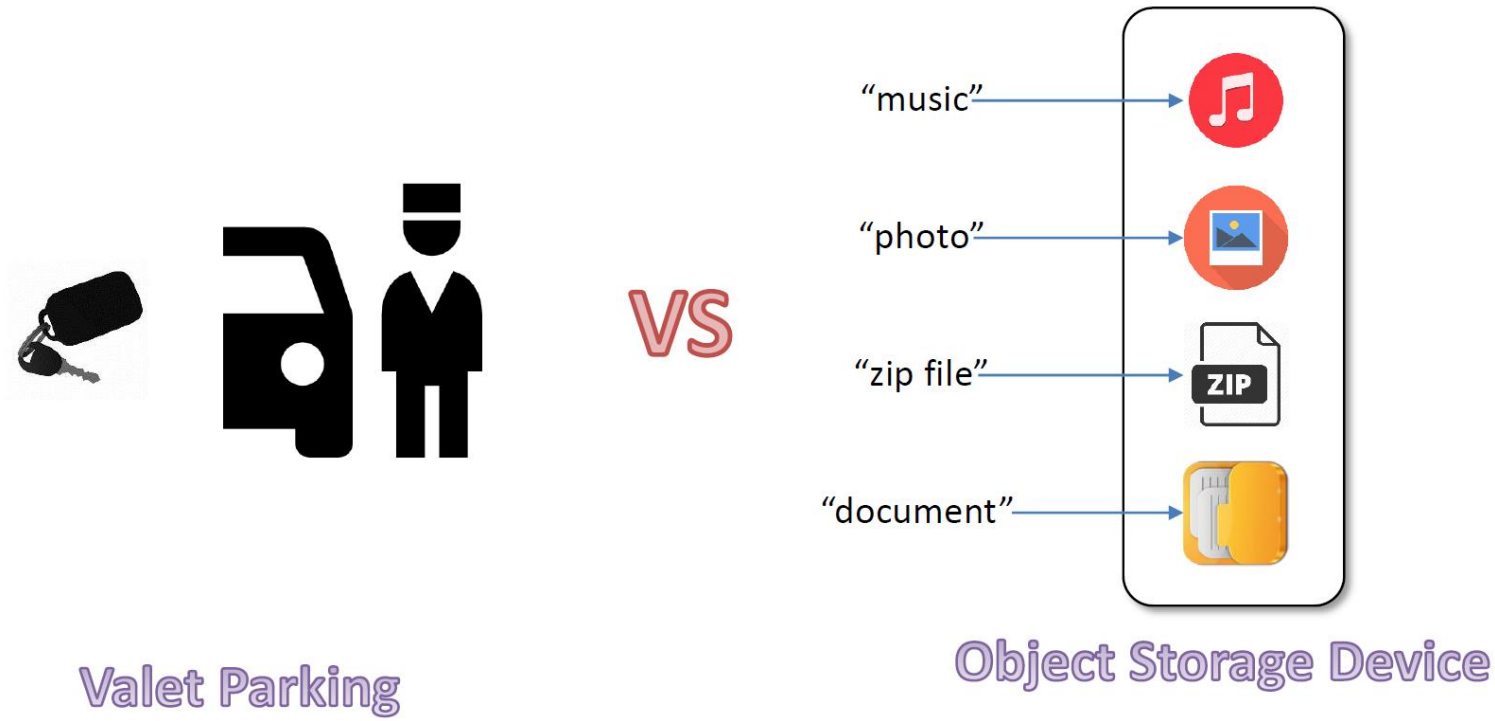
Block: Parking Lot Structure

- A driver (host) is responsible for parking (data management)

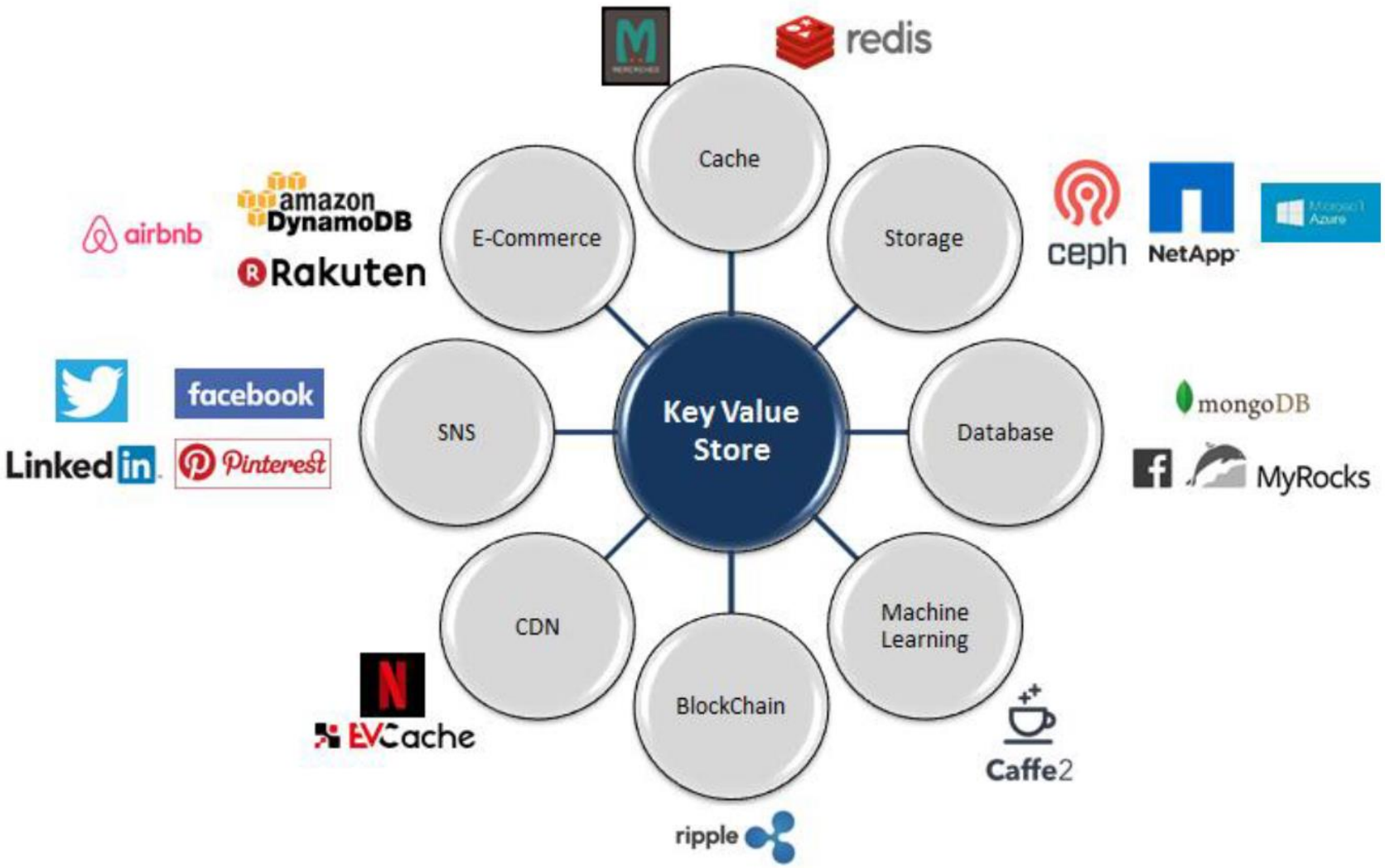


Object: Valet Parking

- A parking facility (storage) is responsible for parking (data management)

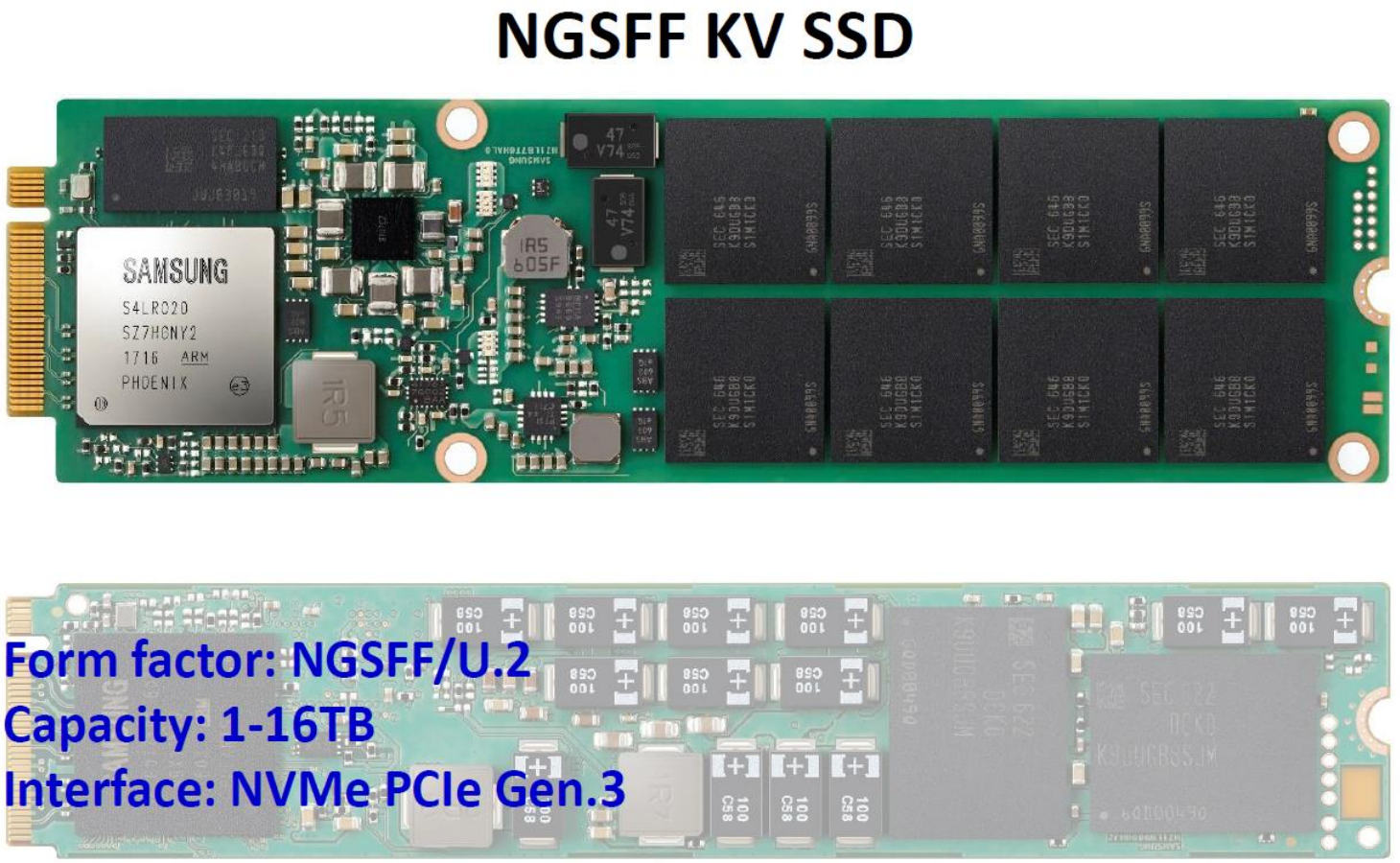


KV Stores Common in Systems at Scale

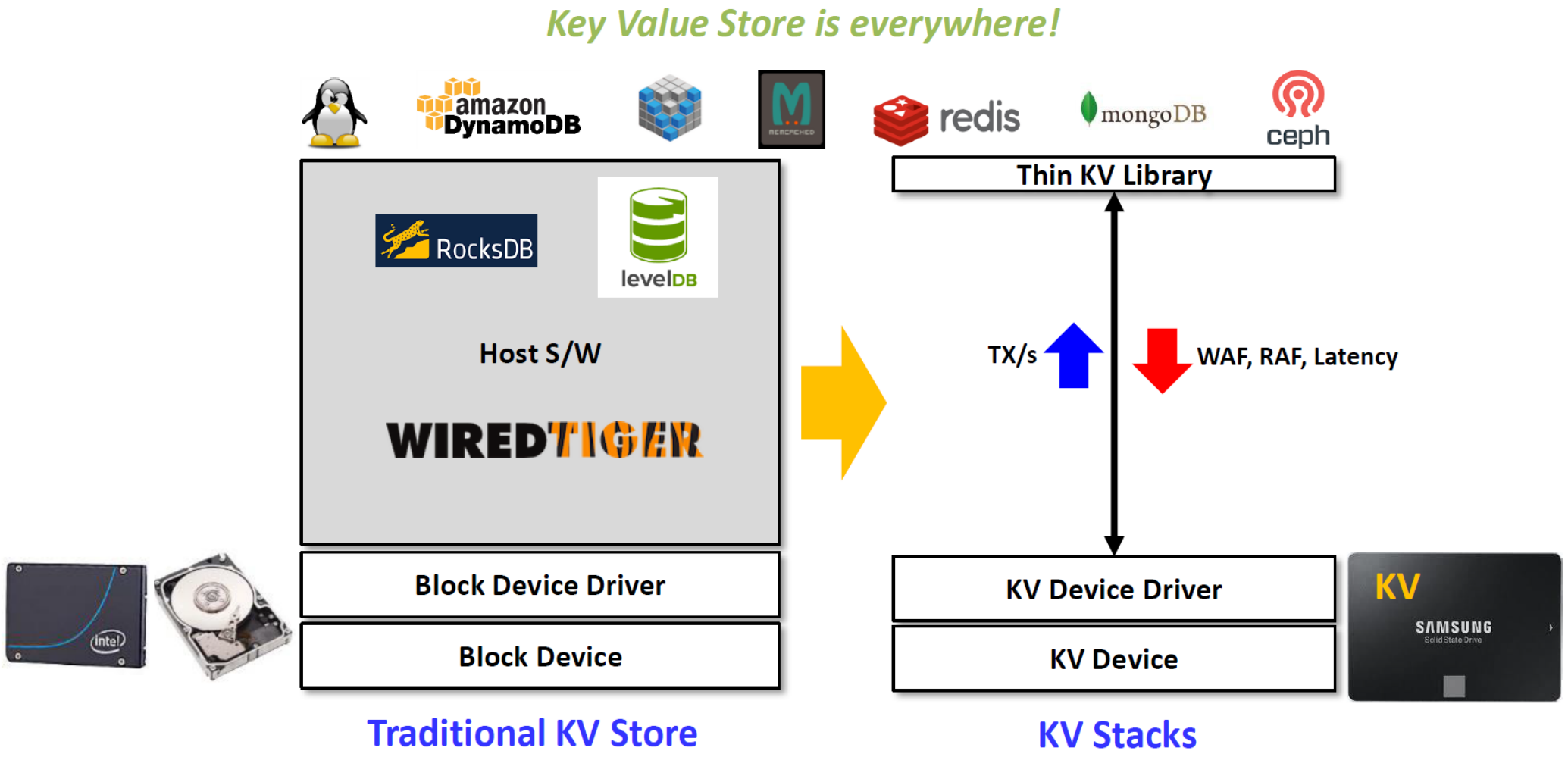


KVSSD Prototype

- Samsung KV-PM983

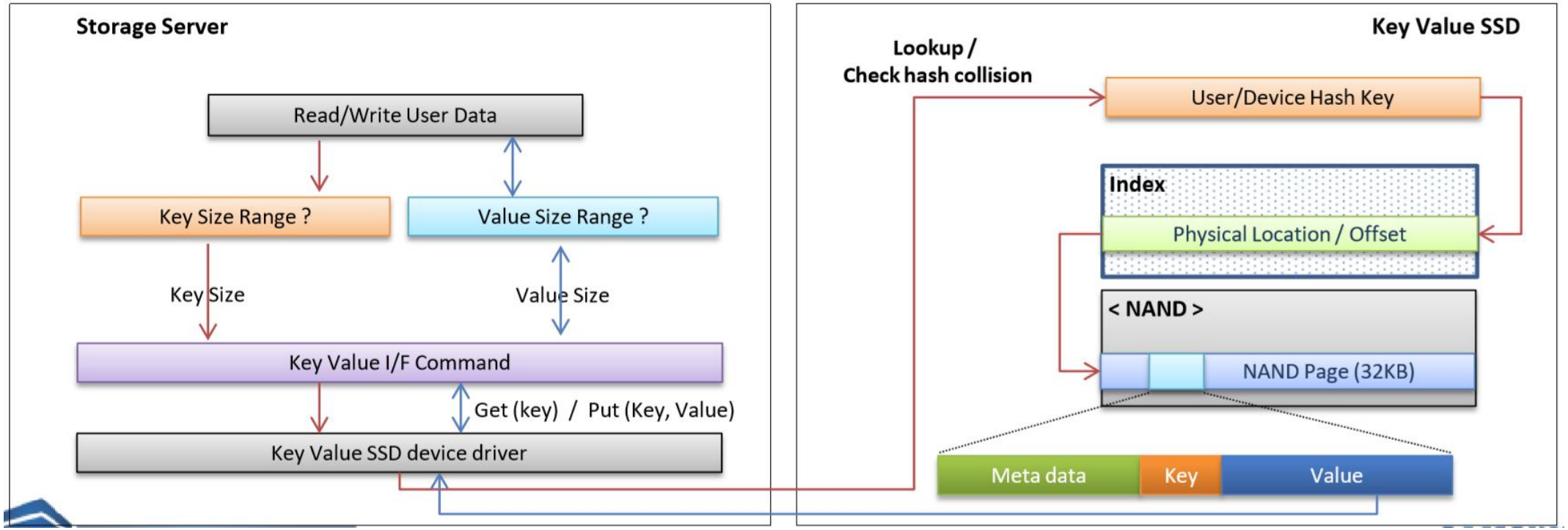


Key Idea


Key Value Store is everywhere!
amazon DynamoDB
redis
mongoDB
ceph
RocksDB
levelDB
Host S/W
WIREDTIGER
Block Device Driver
Block Device
Thin KV Library
TX/s
WAF, RAF, Latency
KV Device Driver
KV Device
KV
SAMSUNG
Solid State Drive
Traditional KV Store
KV Stacks

KVSSD Design

- Key size: up to 255B
- Value size: up to 2MB
- https://github.com/OpenMPDK/KVSSD



KVSSD Software Stack

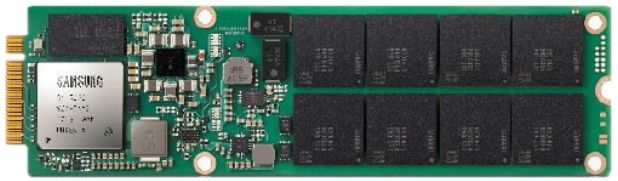
Key Value Library & Tools

Cache | AIO | Multi-Queue | Multi-Device | Memory Manager | Tools

KV Abstract Device Interface (ADI)

store/retrieve/delete/exist | KV Pair | namespace

Linux Kernel Device Driver | **Linux User-space Device Driver** | **Windows Device Driver**

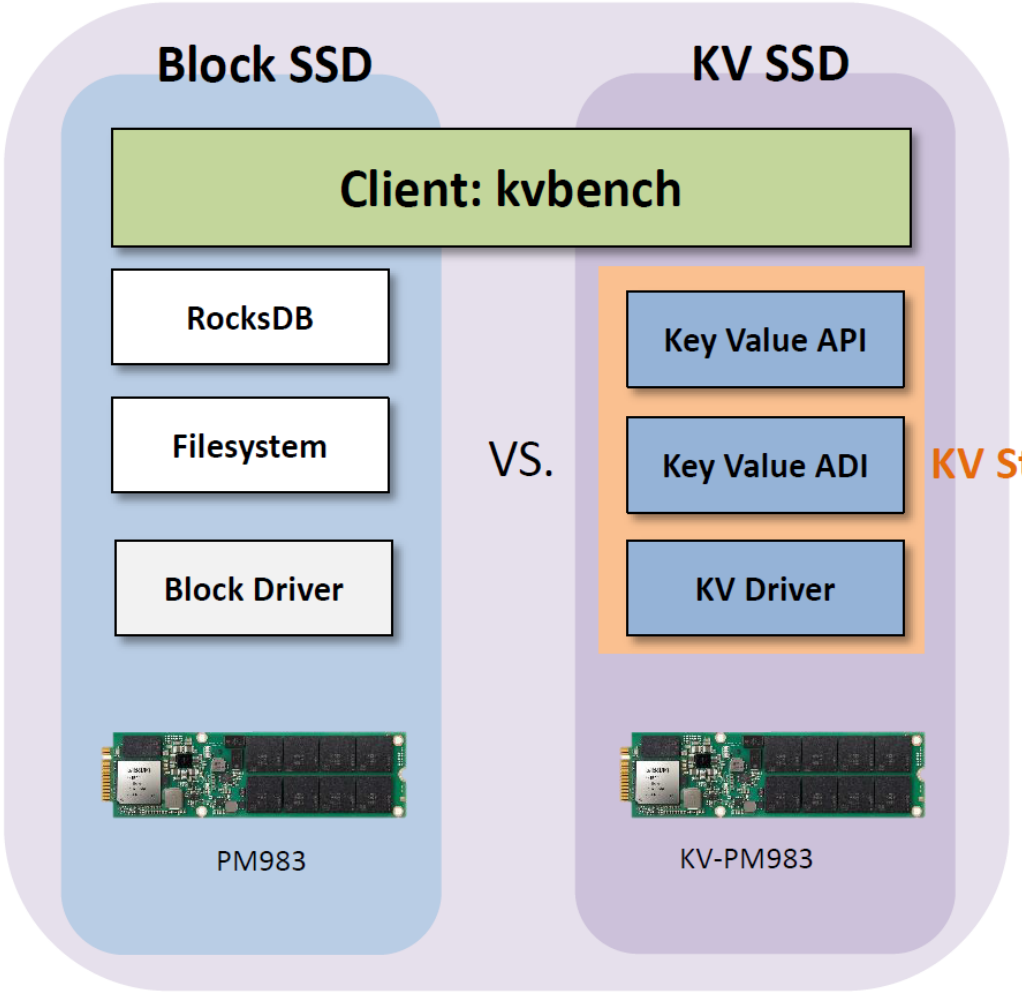


RocksDB vs. KVSSD

- RocksDB
  - Originated by Facebook and actively used in their infrastructure
  - Most popular embedded NoSQL database
  - Persistent Key-Value store
  - Optimized for fast storage (e.g., SSD)
  - Uses Log-Structured Merge (LSM) Tree architecture

- KV Stacks on KVSSD
  - Benchmark tool directly operates on KVSSD through KV Stacks

RocksDB vs. KVSSD: Evaluation

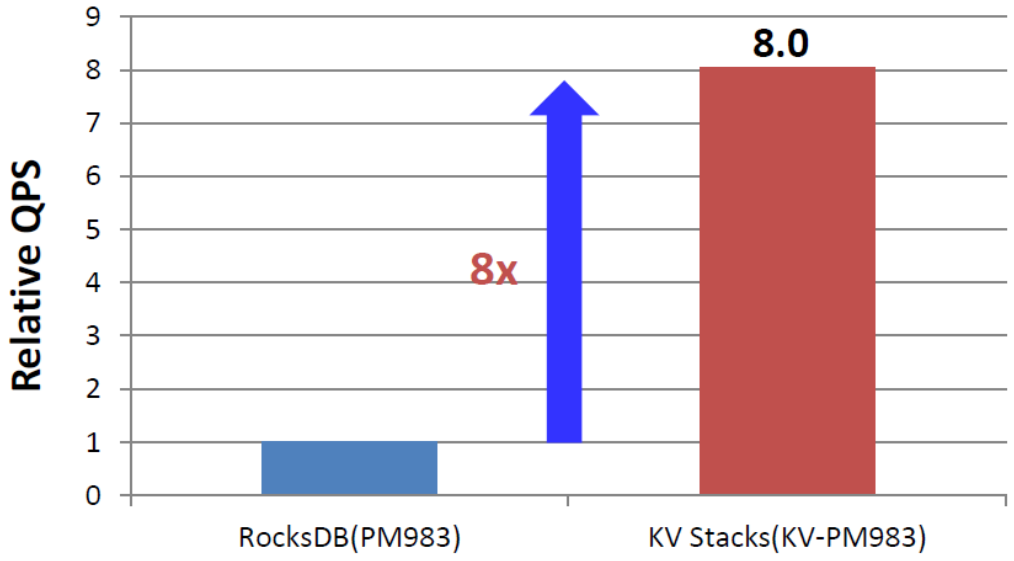


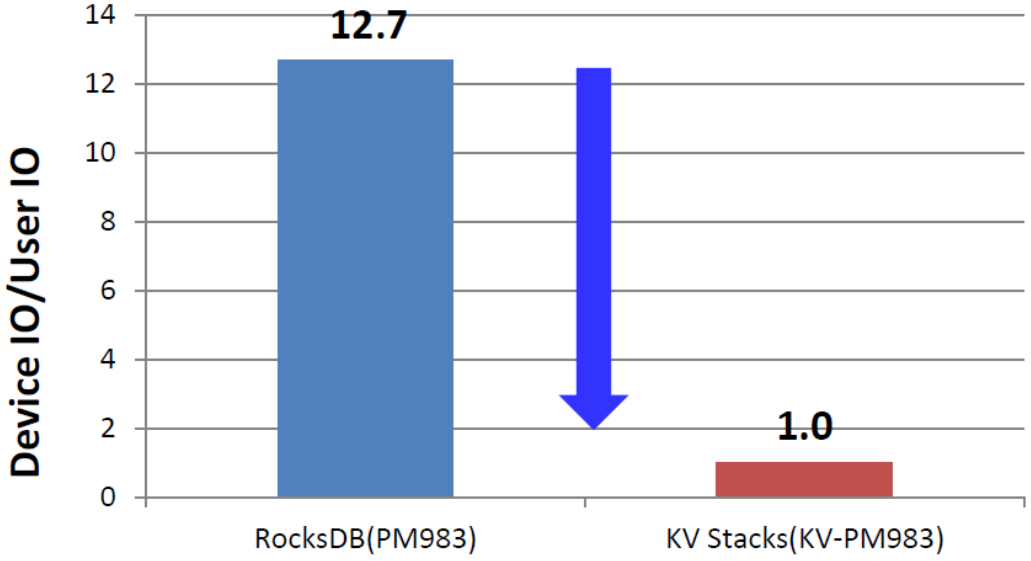
- **Better Performance**
  - Lean software stacks
  - Overhead moved to device
- **IO Efficiency**
  - Reduction of host traffic to devices

Hardware	Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz 96 GB RAM PM983(Block) & KV-PM983 SSD
Software	Ubuntu 16.04 RocksDB v5.0.2 on XFS 50M records, 16B Key, 4KB value

RocksDB vs. KVSSD: Results

- Random PUT performance
  - 8x more QPS (Query Per Second) with KV Stacks than RocksDB on block SSD
  - 90+% less traffic goes from host to device with KV SSD than RocksDB on block device



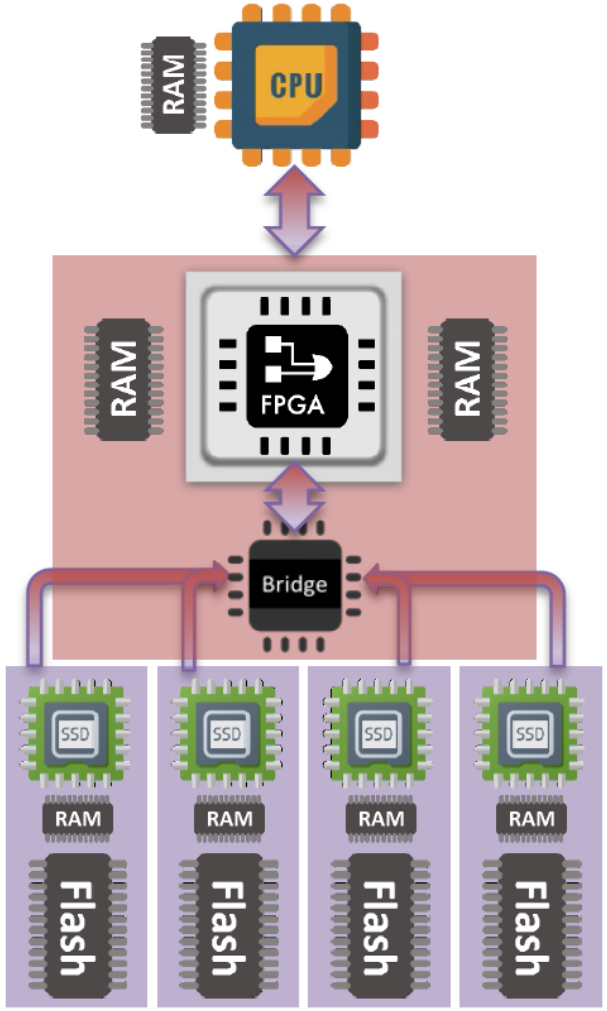


** Workload: 100% random put, 16-byte keys of random uniform distribution, 4KB-fixed values on single PM983 and KV-PM983 in a clean state*

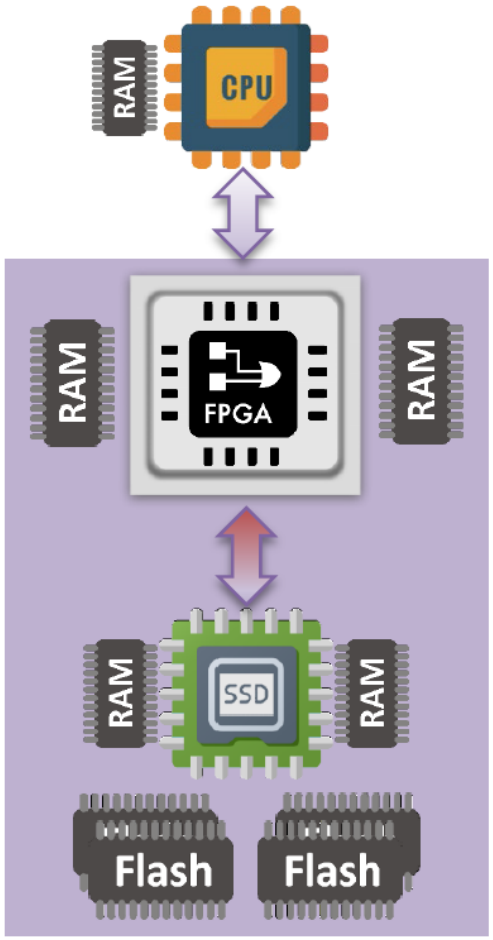
Computational Storage

SNIA: Computational Storage Instances

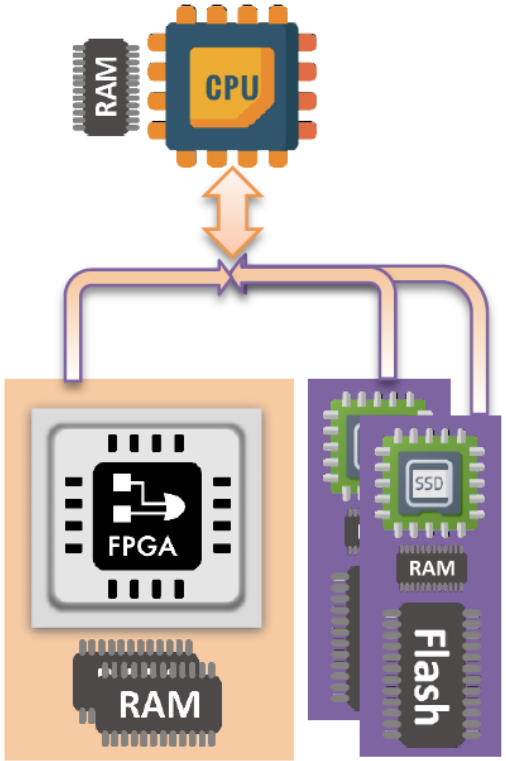
FPGA to Multi-SSDs



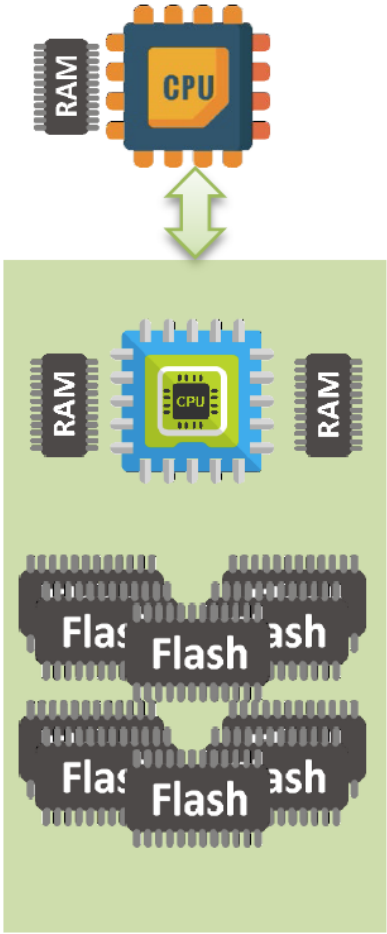
FPGA+SSD Controller



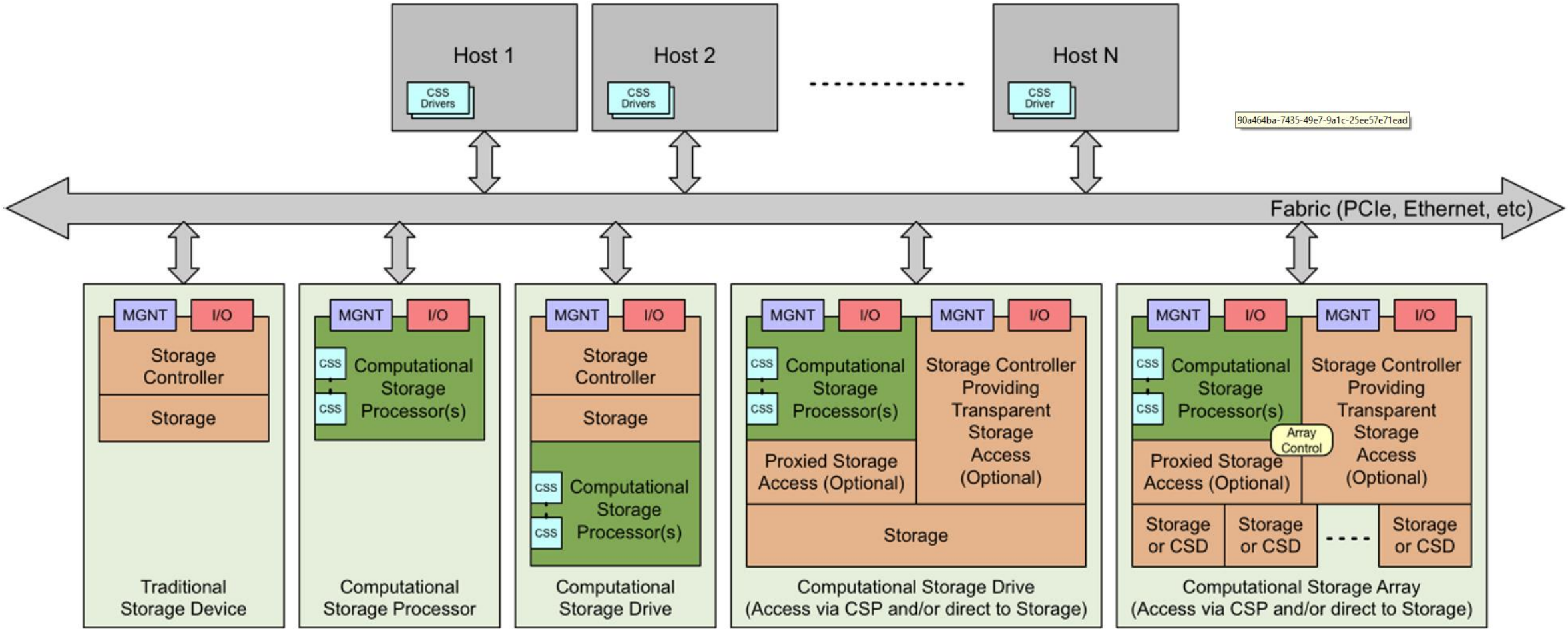
FPGA Only



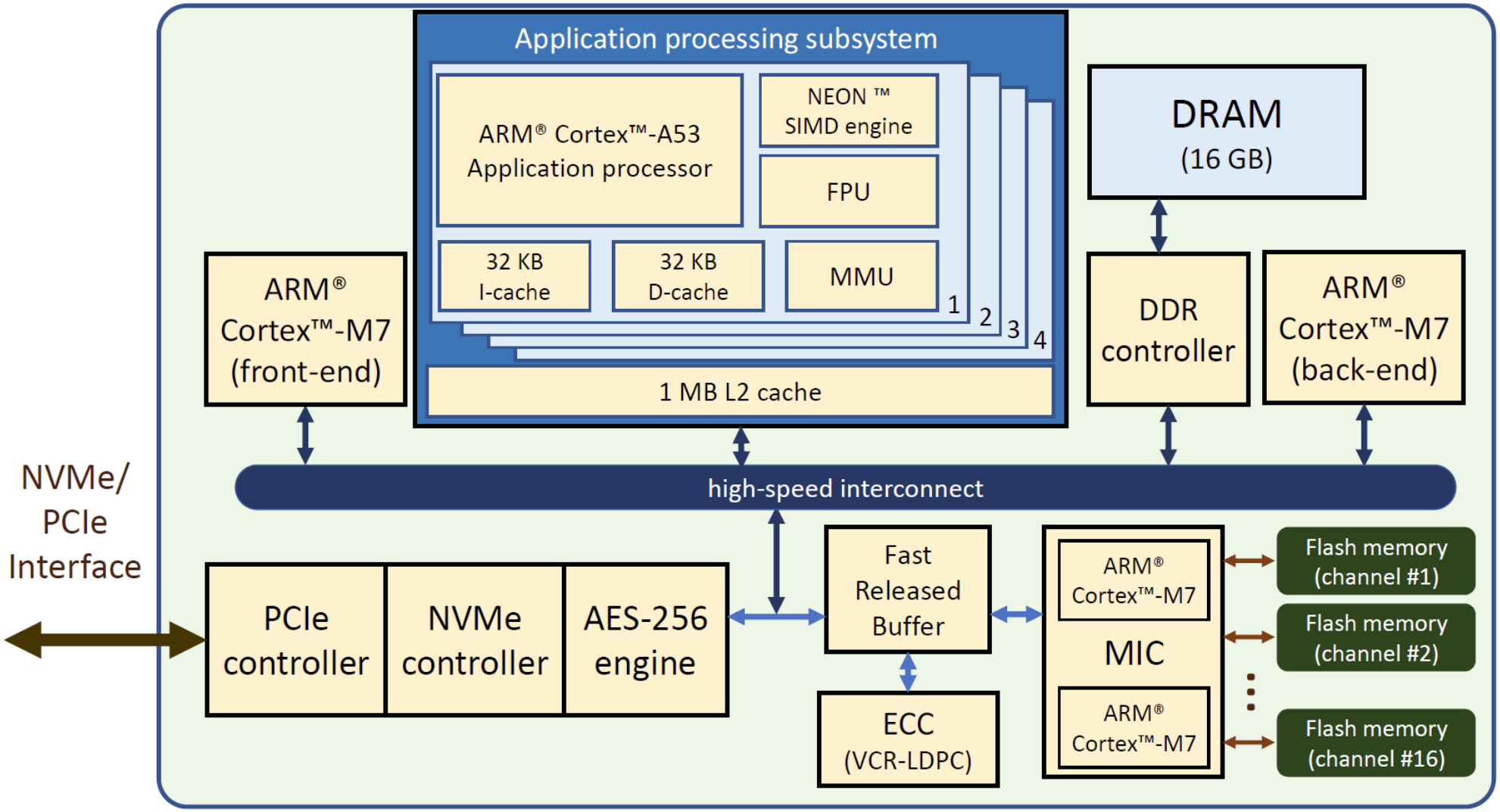
ASIC

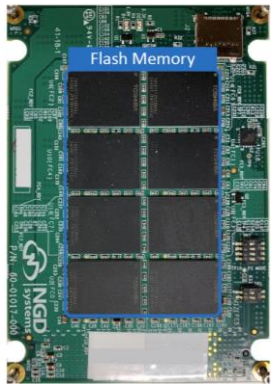


SNIA: Computational Storage Devices (CSx)



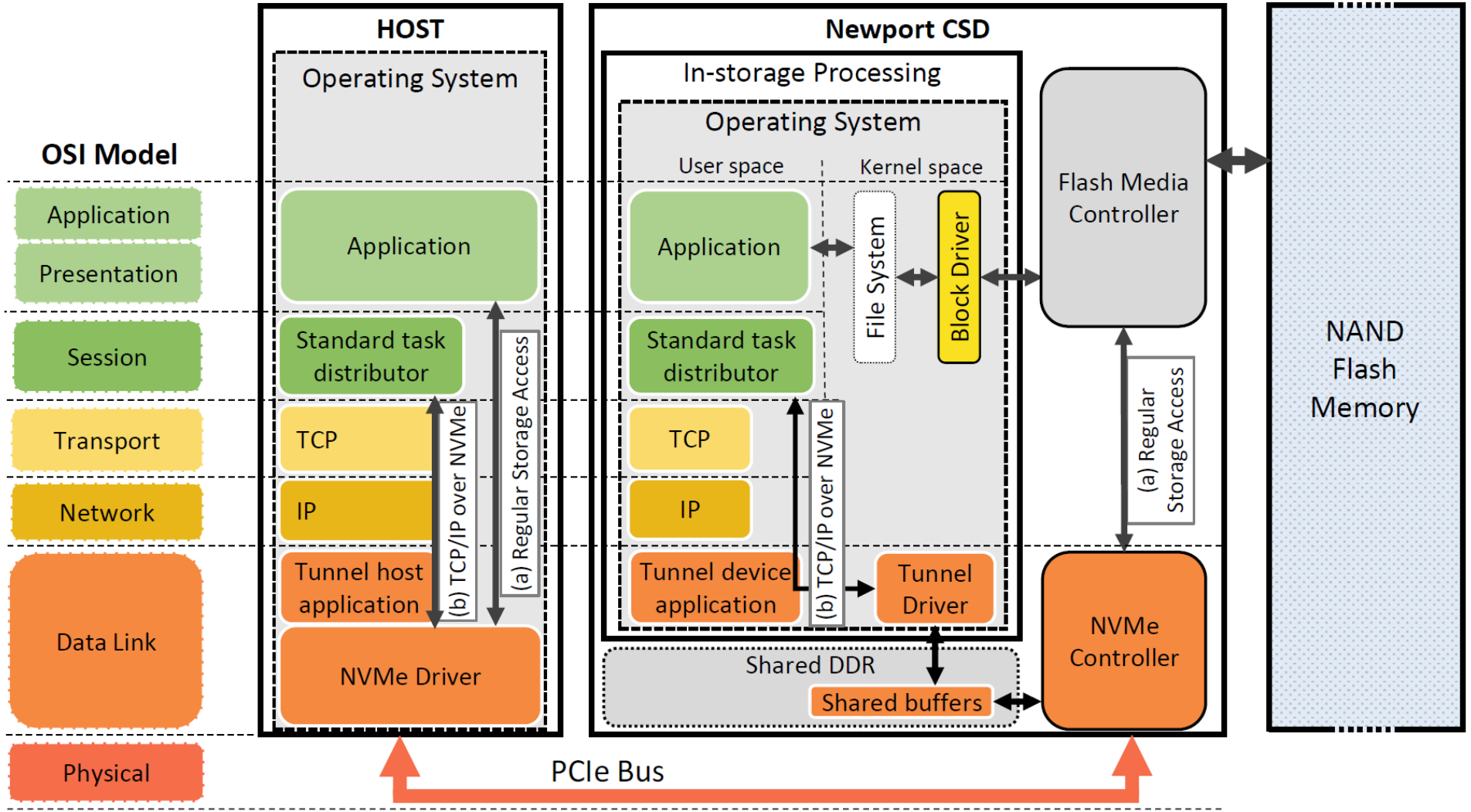
NGD Newport SSD





Source: J. Do et al., "Cost-effective, Energy-efficient, and Scalable Storage Computing for Large-scale AI Applications, ACM ToS, 2020.

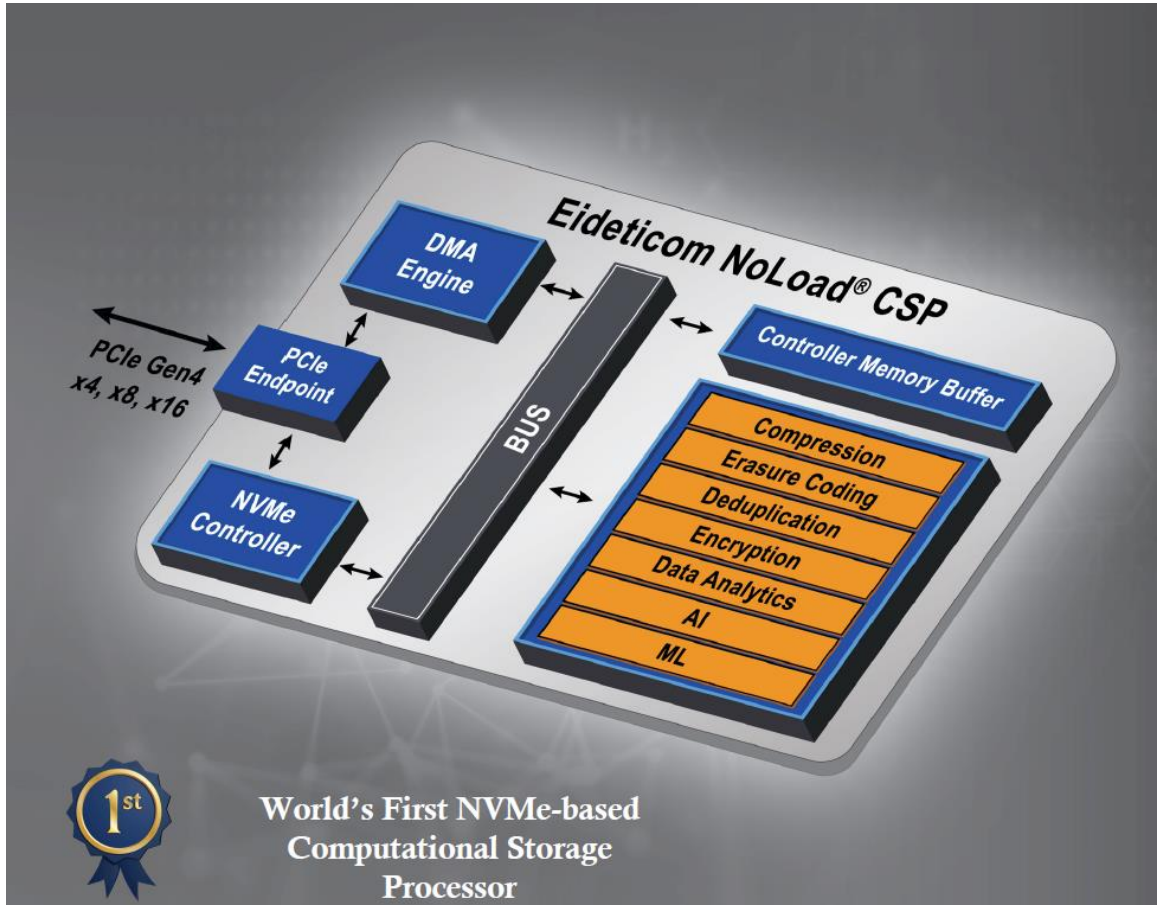
NGD Newport SSD: Software Stack



Source: J. Do et al., "Cost-effective, Energy-efficient, and Scalable Storage Computing for Large-scale AI Applications, ACM ToS, 2020.

Eideticom NoLoad CSP

- NVMe computational accelerators
  - Compression
  - Encryption
  - Erasure coding
  - Deduplication
  - Data analytics
  - AI and ML



Samsung SmartSSD

- PM983F
- Xilinx FPGA