Open. Together.

OCP
SUMMIT

HW/SW Co-Design for Predictable IO Latency

Nav Kankani, HW Solutions TPM
Vijayan Rajan, SW Engineer
Facebook, Inc.

OPEN PLATINUM™

OCP SUMMIT

Open. Together.

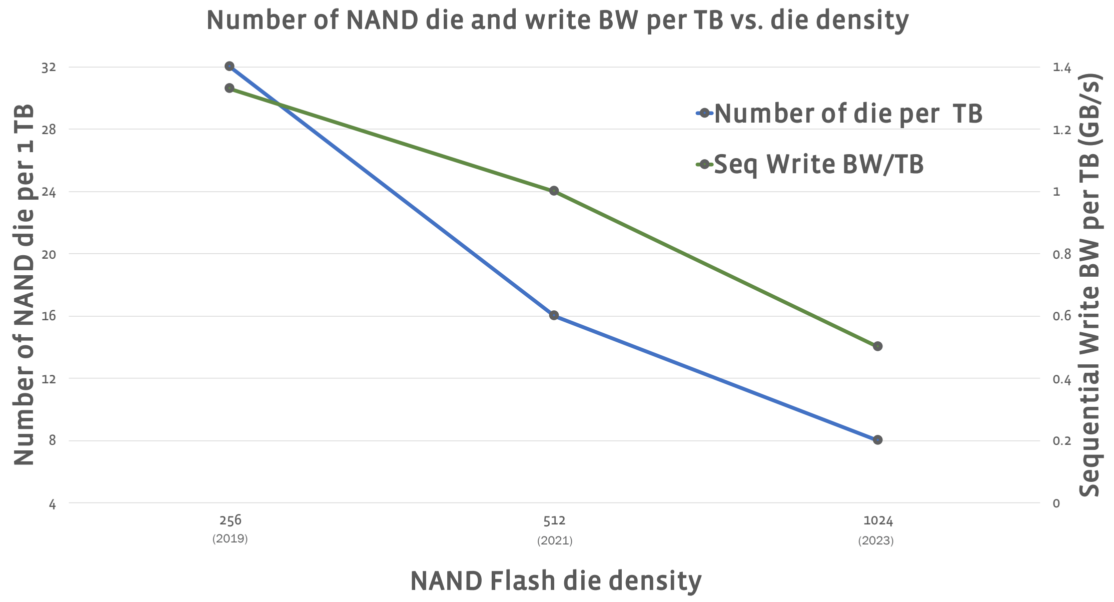
Context - Today's Storage Types

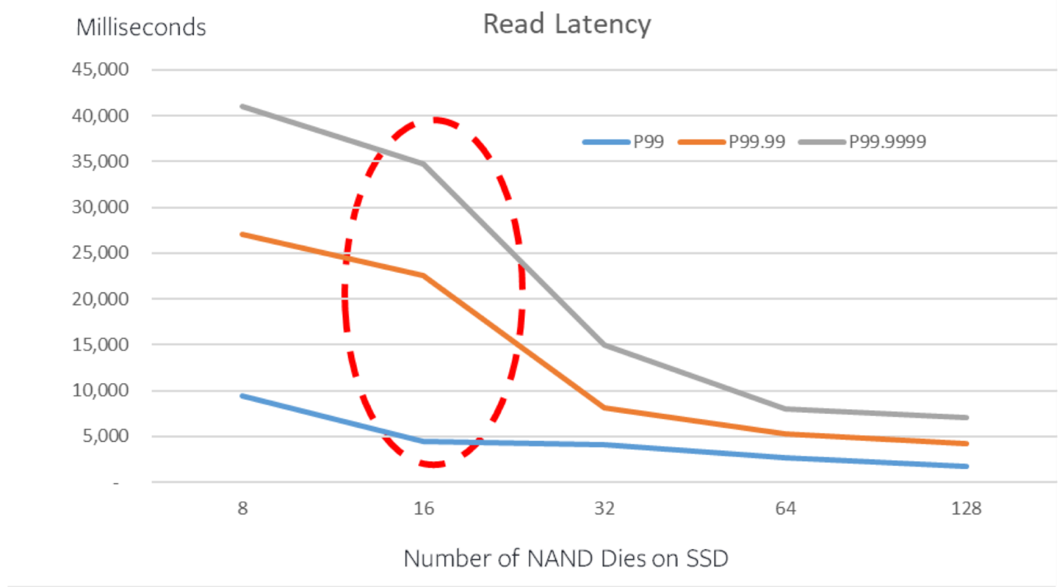
Transactional ⟷ **Archival**

Core Register	L1/L2/L3 Cache	DRAM/SCM	SCM/Flash	FLASH/HDD	HDD/TAPE

Size	64KB-2MB	16GB-128GB	128GB-1TB	512GB-4TB	4TB-16TB
Speed	1-20ns	50- 1us	250ns-50us	1-5ms	5-100ms
Storage Tier	S0 (Financial)	S1 (Transactional)		S2 (Warm non-transactional)	S3 (Archival/Cold)
Storage Architecture	Scale-up/ Block Store/High Resiliency/Availability			Scale-out/File-Object/Feature-Rich/Cost	
Storage Application	My-SQL/Oracle/SAP	Cassandra/MongoDB/Rocks-DB		Warm-storage/Cold-Store	

Industry Trends – NAND Flash Storage

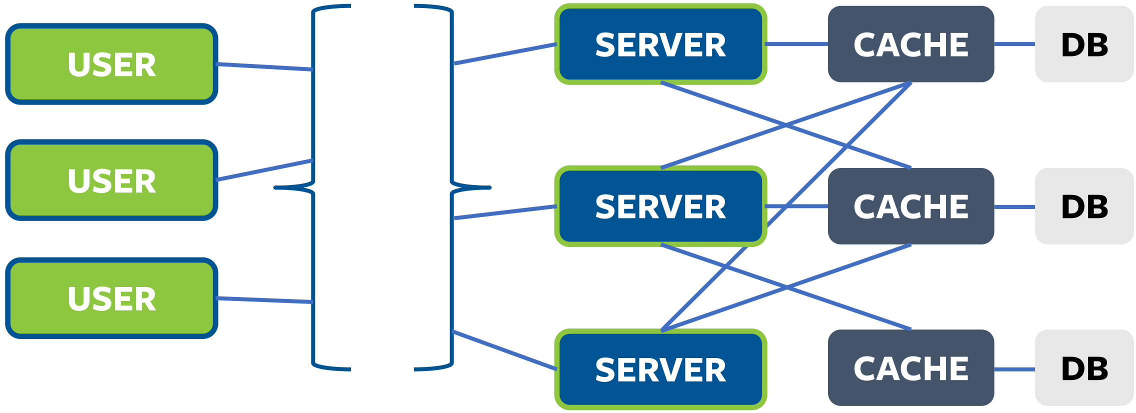




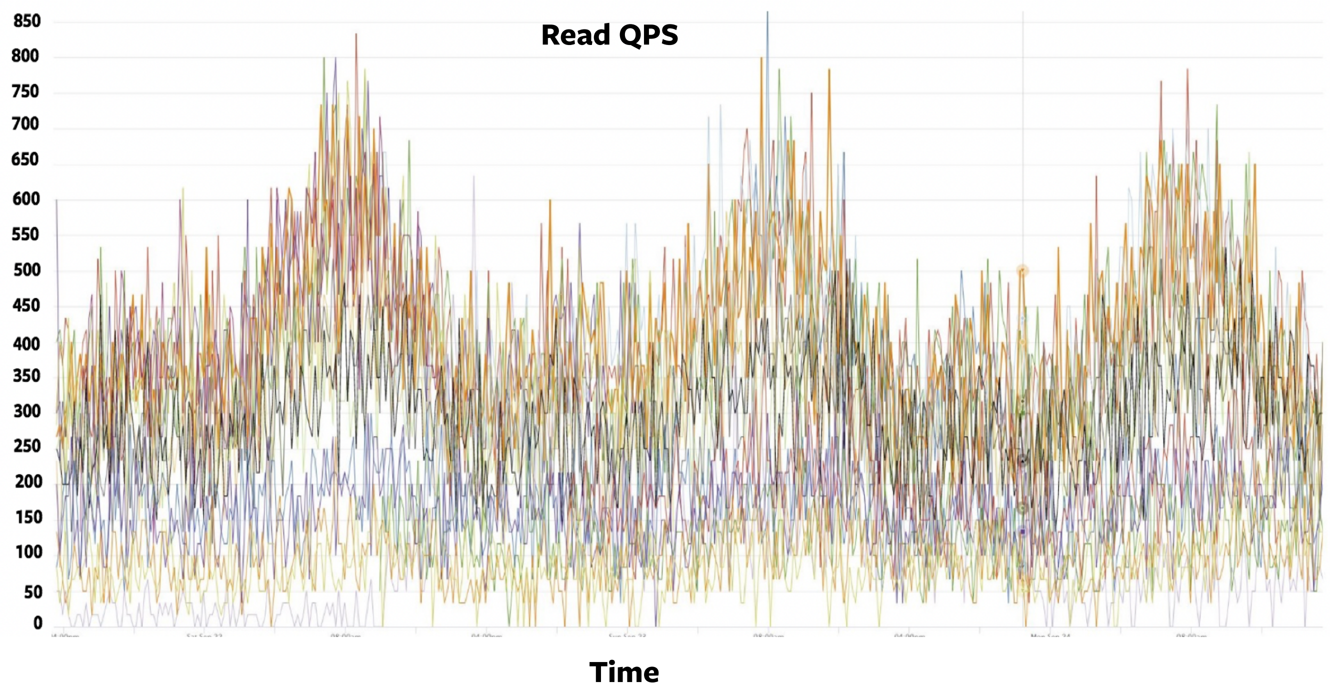
- **NAND Flash Densification**
  - **IOPS/TB decreasing**
- **Less NAND Die per TB**
  - **Increase in IO latency and unpredictability**

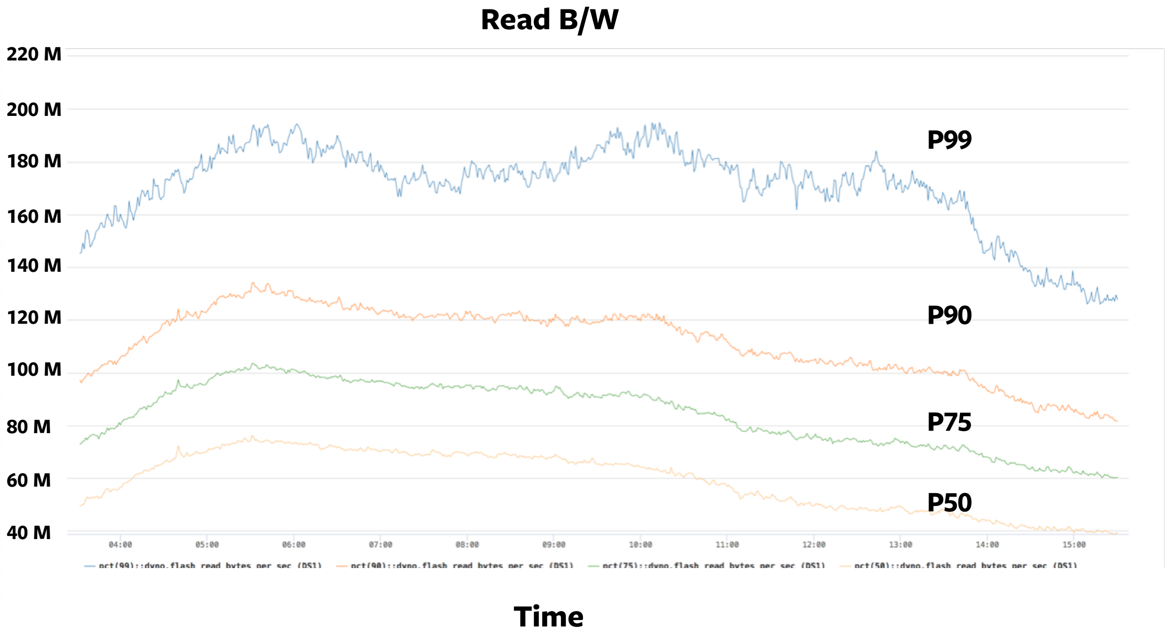
Facebook's Architecture

- **Massive levels of Sharding to connect users**
- **Fetch requests incur large fanout on the back-end**
- **Data read from many servers and multiple pieces from each**



Variability in Hyperscale Workloads

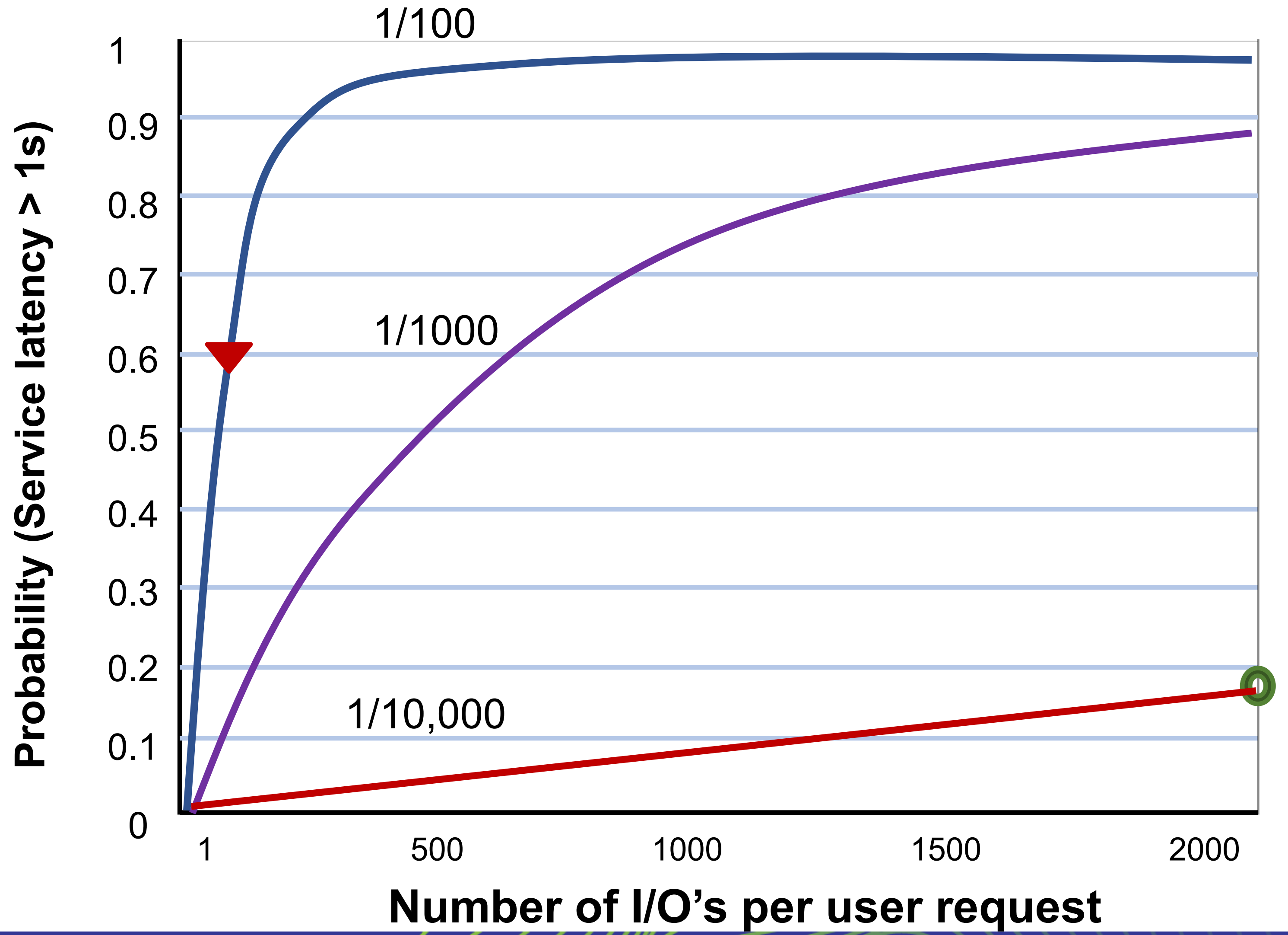




Asymmetricity in read and write access patterns across shards

Read Bandwidth Variation at different latency levels (P99 to P50)

Why Does Storage Latency Matter?



- 1 user request => ~10-1000's back-end requests
- Back-end requests have their own read and write amplification.
- Tail, rather than average latency is important
  - With N*M requests to N servers, probability of high latency is compounded.

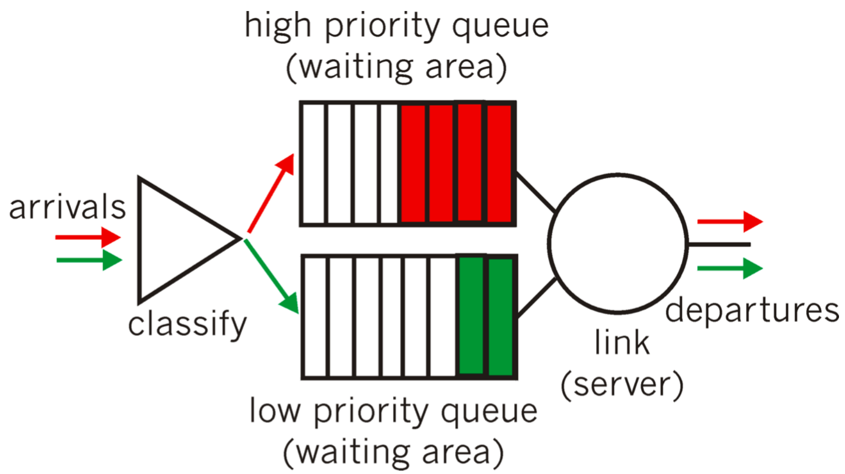
Problem Definition

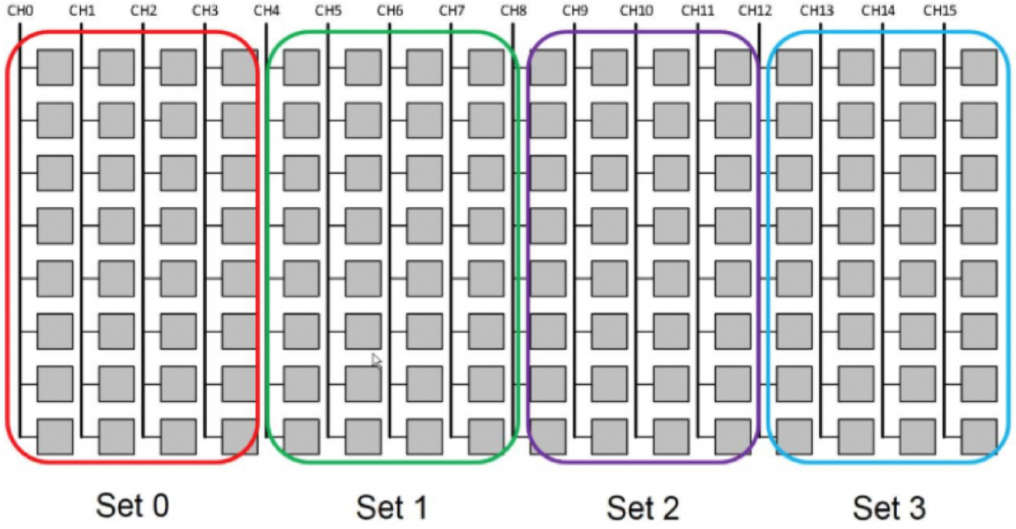
- Unpredictable latency in storage stack exists.
- Large scale distributed system’s need predictable latency regardless of unpredictable latency in the storage stack.

Optimizing for predictable latency

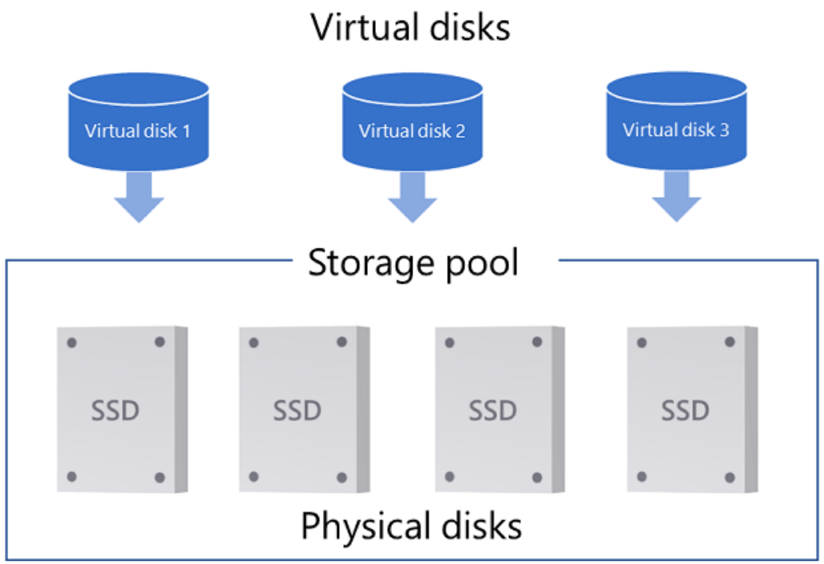
HW-Layer

- Parallel operation paths
- Priority Queues
- New Device Features
  - Write/Erase suspends
- Isolation
  - Streams
  - NVMeSets
- Predictable Latency Modes
- Max Read Recovery Limits



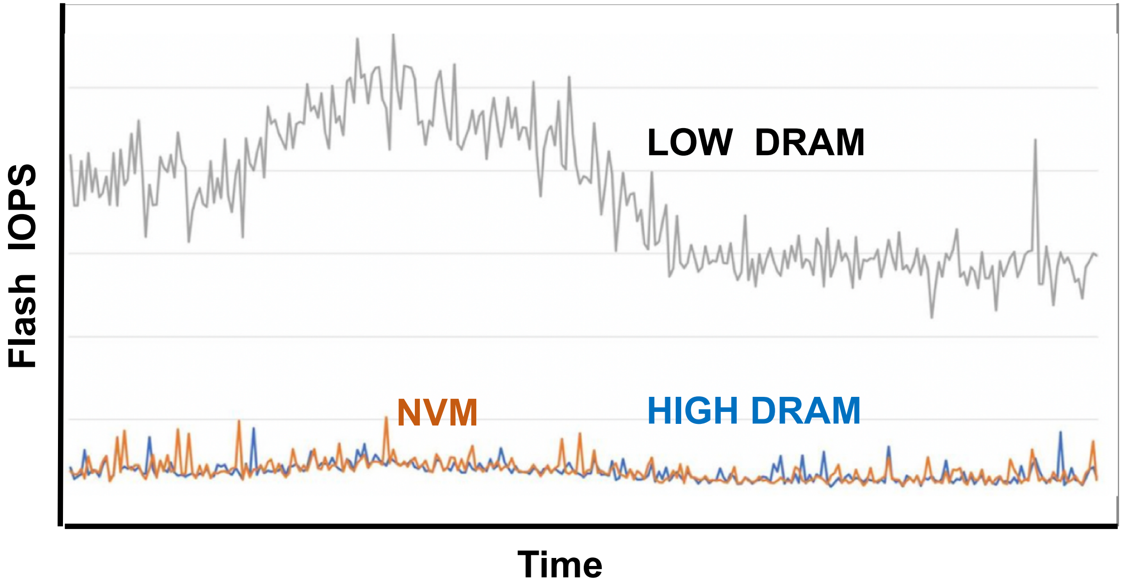


Optimizing for predictable latency



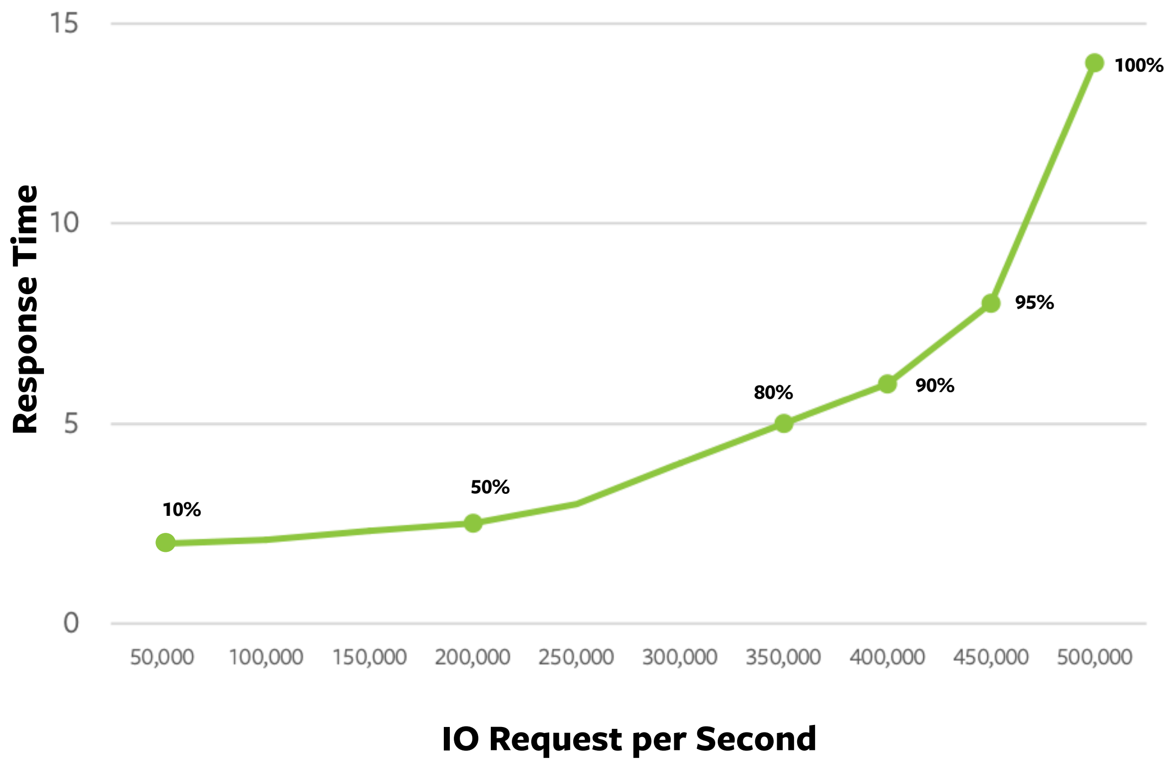
SW-Layer

- Shard Management & Rebalancing
- Pooling & Striping
- Block and Page Caching
- Tiering using SCM
- Write coalescing
- Dynamic Re-sizing



Key Trade-offs to be made to buy latency credits:

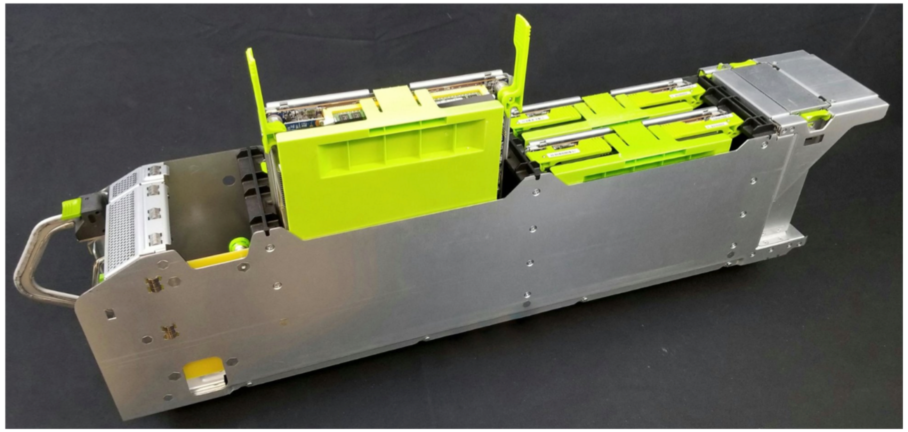
- Restricted Resource Sharing
- Reduced workload & scalability
- Lower queue depths
- Throttled Performance
- Inefficient power management



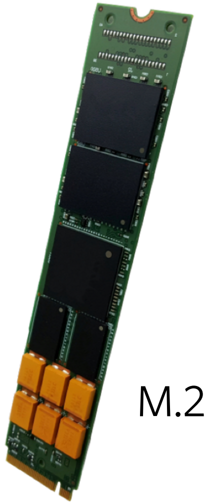
Why HW/SW Co-design for Predictable Latency?

- Impractical to eliminate IO stack latency at HW layer alone.
- Leverage existing latency trade-offs in HW & SW development.
- Knowledge of Application Domain opens new optimization opportunities & architectures.

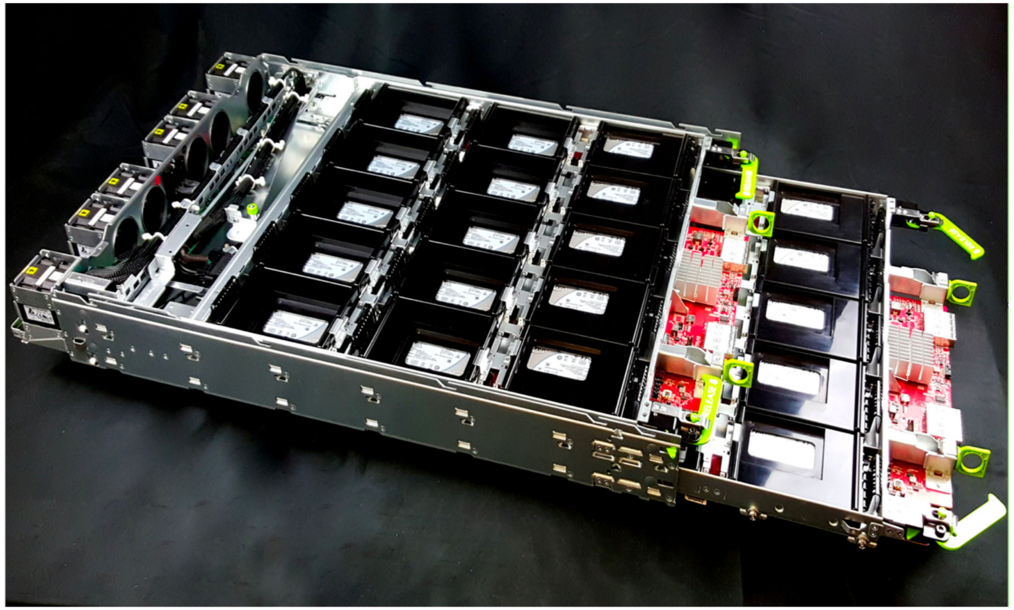
Facebook's OCP HW – Flash Based



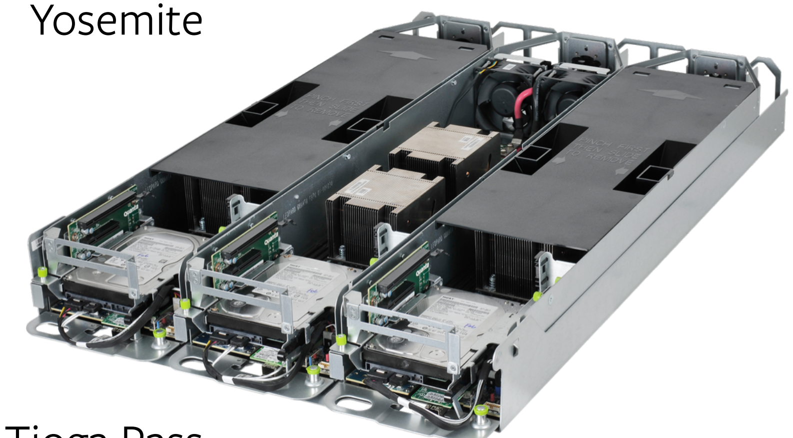
Yosemite



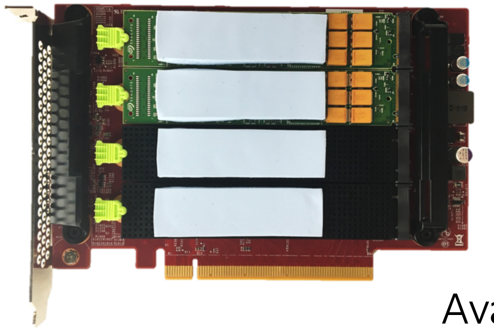
M.2 card



Lightning JBOF

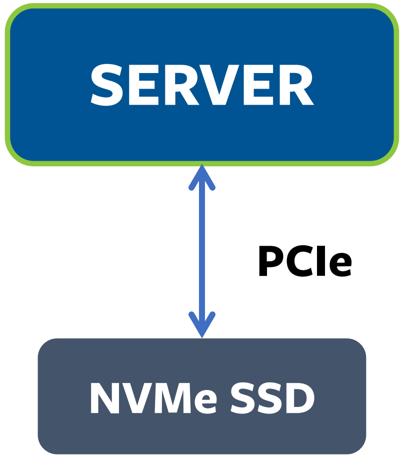


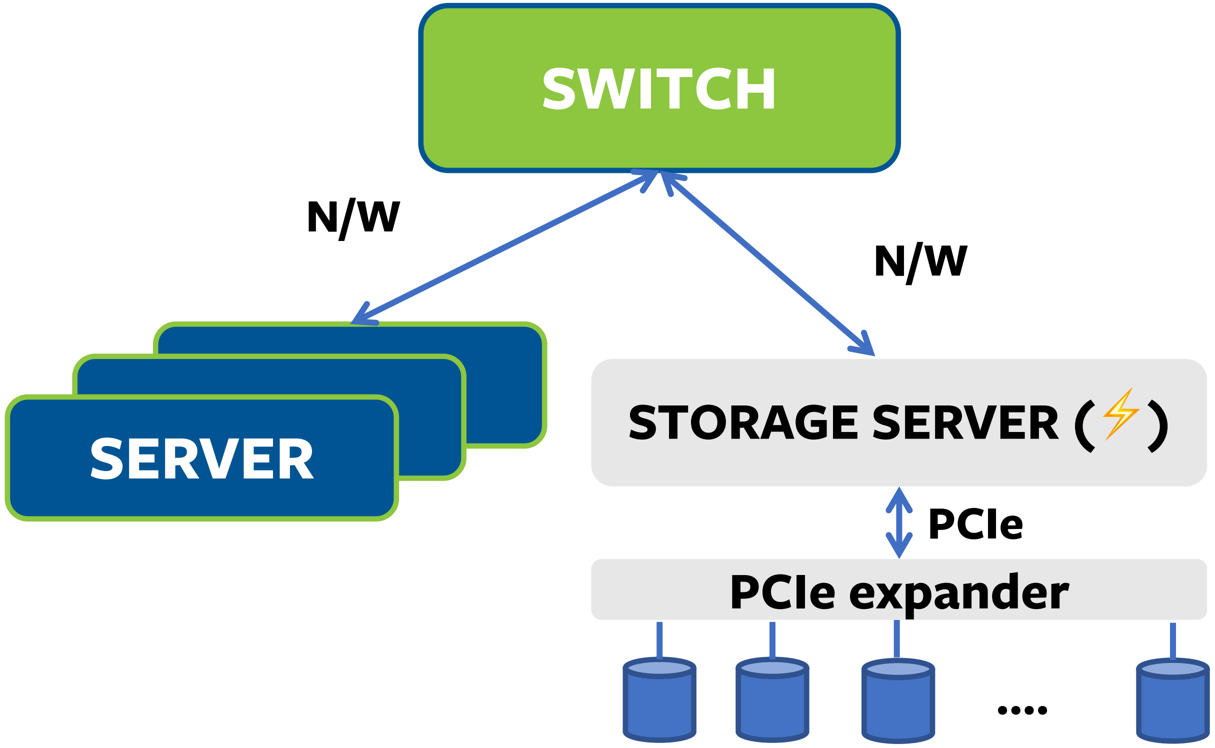
Tioga Pass



Ava card

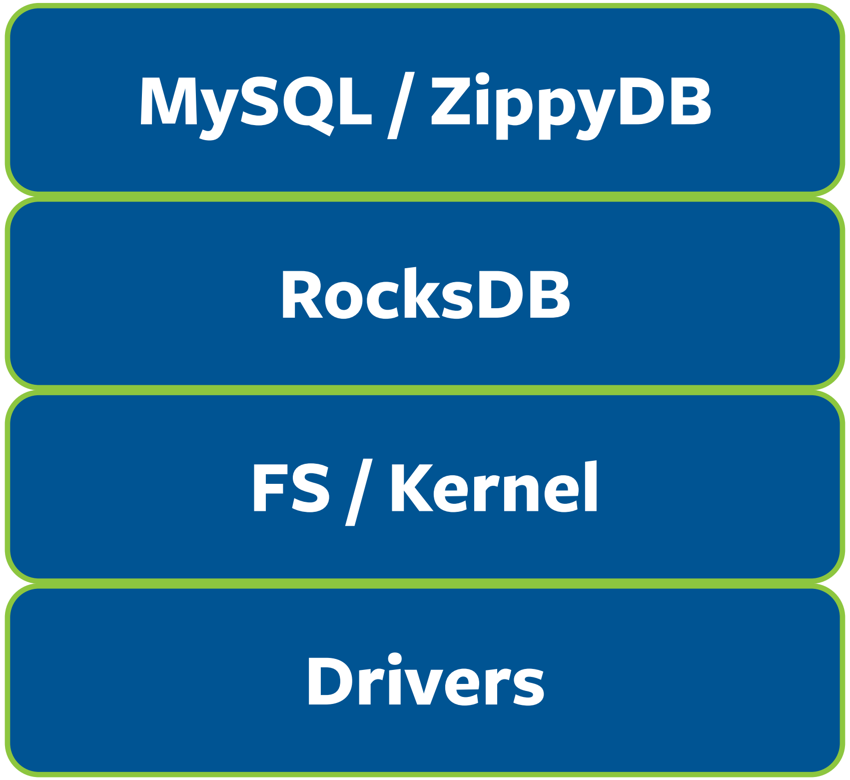
Leveraging OCP HW for Efficiency and Latency



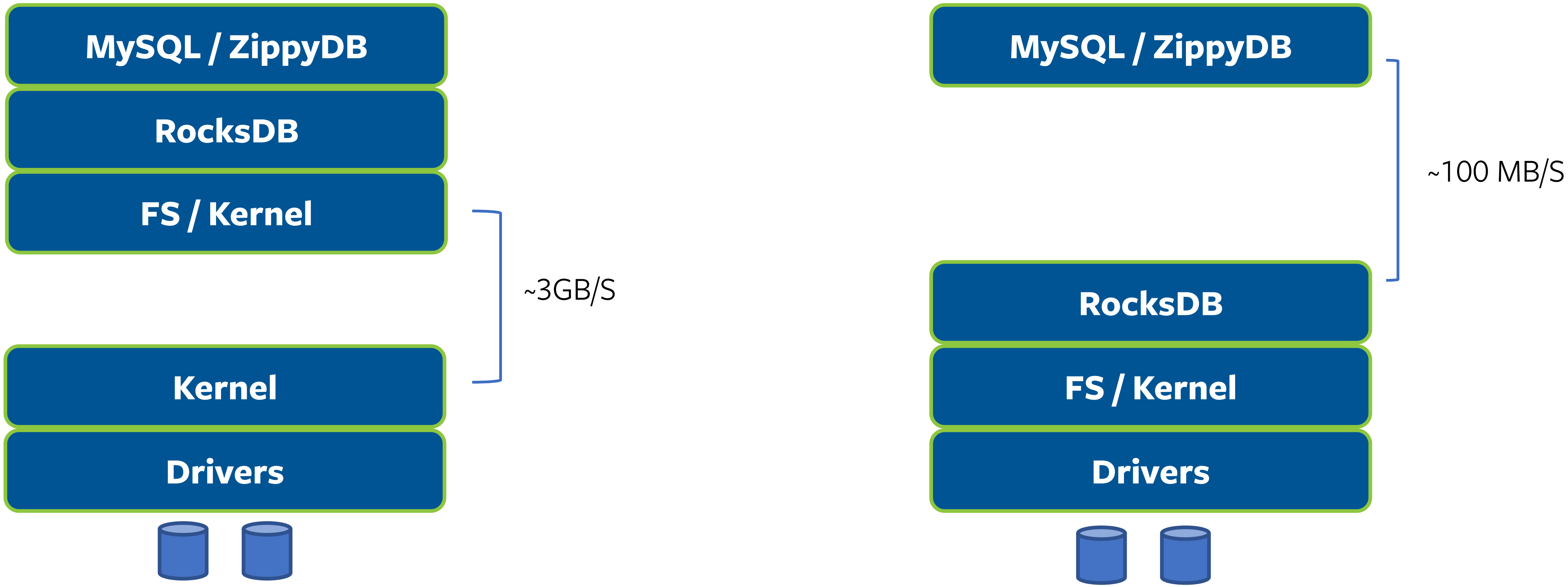


RocksDB at Facebook

- Most database technologies at Facebook use RocksDB
  - ZippyDB: Replicated, Consistent Key-value as a service
  - MySQL: :Local Key-value store
- Each service is sharded (very) widely.



Architectural Options



Networking within & across racks

- Key-Value stores incur high write amplification.
  - RocksDB is better, but is no exception.
- Huge difference in bandwidth:
  - Compare: 120 MB/s reads/writes of small keys & values (256 bytes) vs. 3000 MB/s disk reads and writes.
- Keeping amplified I/O local saves networking, improves latency, especially tail latency.
  - PCIe; sled-local networks; rack-local networks.

Flexible Hardware for Efficient Software

- Key/Value stores are CPU- and DRAM-hungry.
- Lightning JBOF-based designs achieve good sharing, and great capacity management.
- Perfect for Blocks protocols; but difficult to run RocksDB
  - 1 JBOF + 5-15 DB + RocksDB hosts: works perfectly.
  - 1 JBOF with 5-15 RocksDB instances + 5-15 DB hosts: extremely imbalanced.
- Need for a flexible combination of CPU+DRAM+SSD.

Leveraging Yosemite HW as Shared-Storage

- For RocksDB use cases, achieves better ratios.
  - With two NVMe SSD per server in Yosemite Chassis:
    1 CPU + X DRAM + 4- 8TB SSD
- Compare this against JBOF based design:
  2 CPU + Y DRAM + 60-240TB SSD
- This is a comparison available with today's OCP choices.
  Better designs and faster networking always welcome!

Design Imperatives: Flexible Ratios

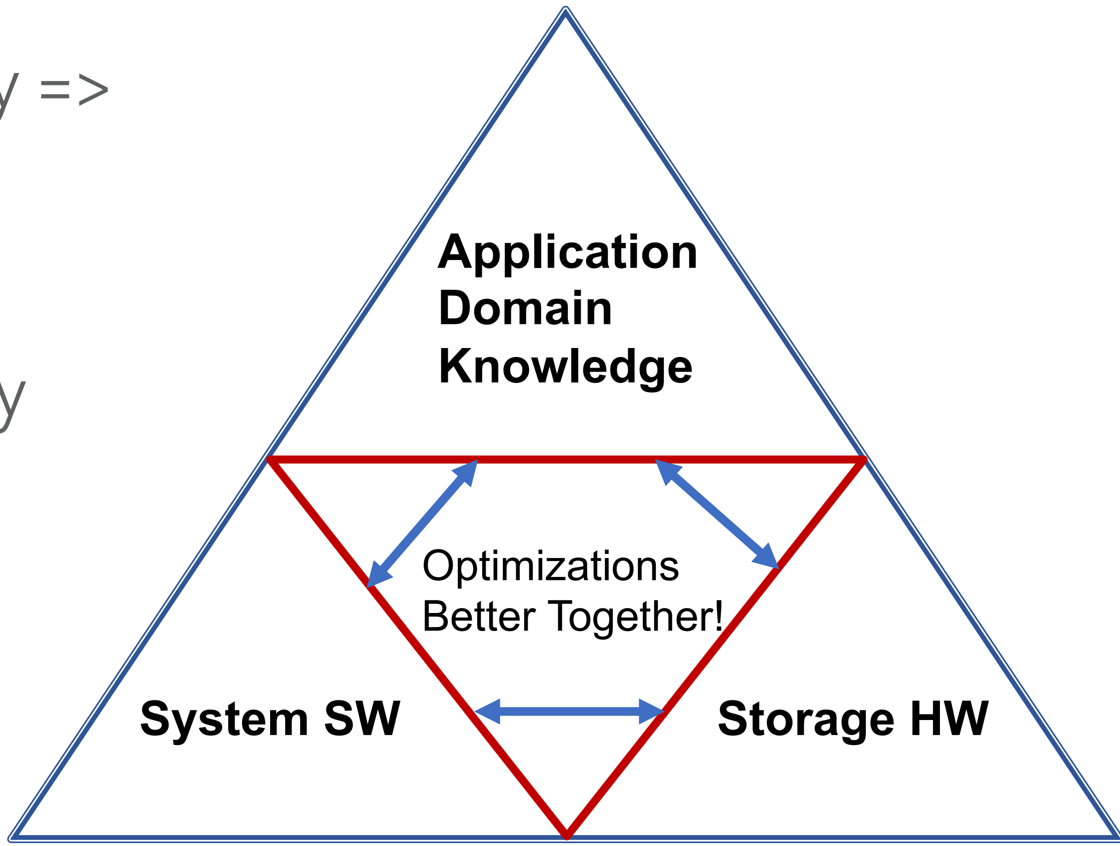
- Hardware rearchitecting goes hand-in-hand with software reconfiguration.
- At scale, getting efficiency is hard.
- We need a flexible set of building blocks: the right ratios of CPU, DRAM and SSD within each server
  ... connected with low-cost, high-speed networks.

Conclusion:

- HW/SW co-design for predictable IO latency => **Better together!**

- Leverage FB's OCP components for flexibility to build multiple balanced solutions

- Customize architectures to be application aware.



Please Visit Booths for more information on OCP HW !

Open. Together.

OCP Global Summit | March 14–15, 2019

OCP SUMMIT