facebook

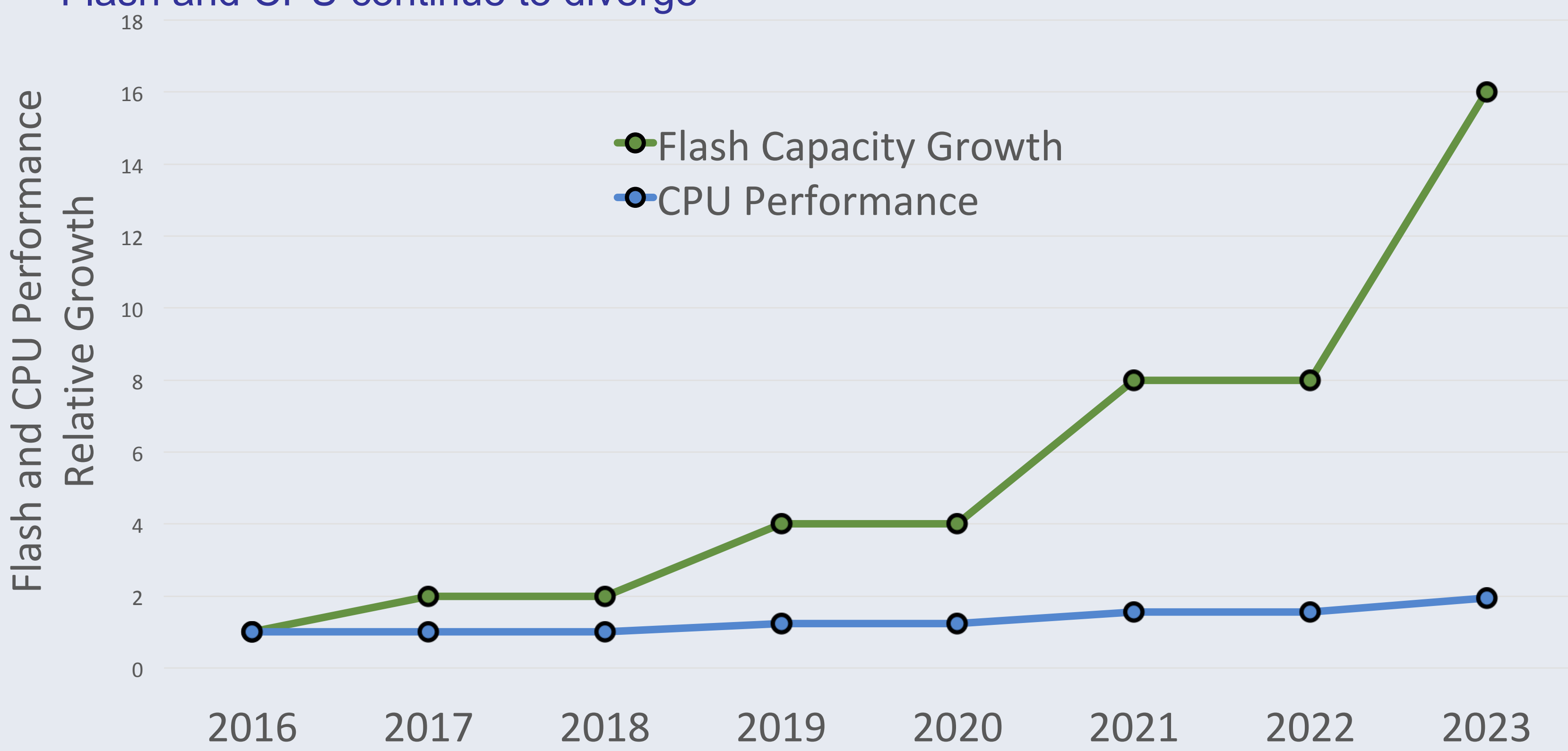
Enabling NVMe® I/O Determinism @ Scale

Chris Petersen, Hardware System Technologist
Wei Zhang, Software Engineer
Alexei Naberezhnov, Software Engineer
Facebook


Facebook @ Scale
800 Million
1.3 Billion
2.2 Billion

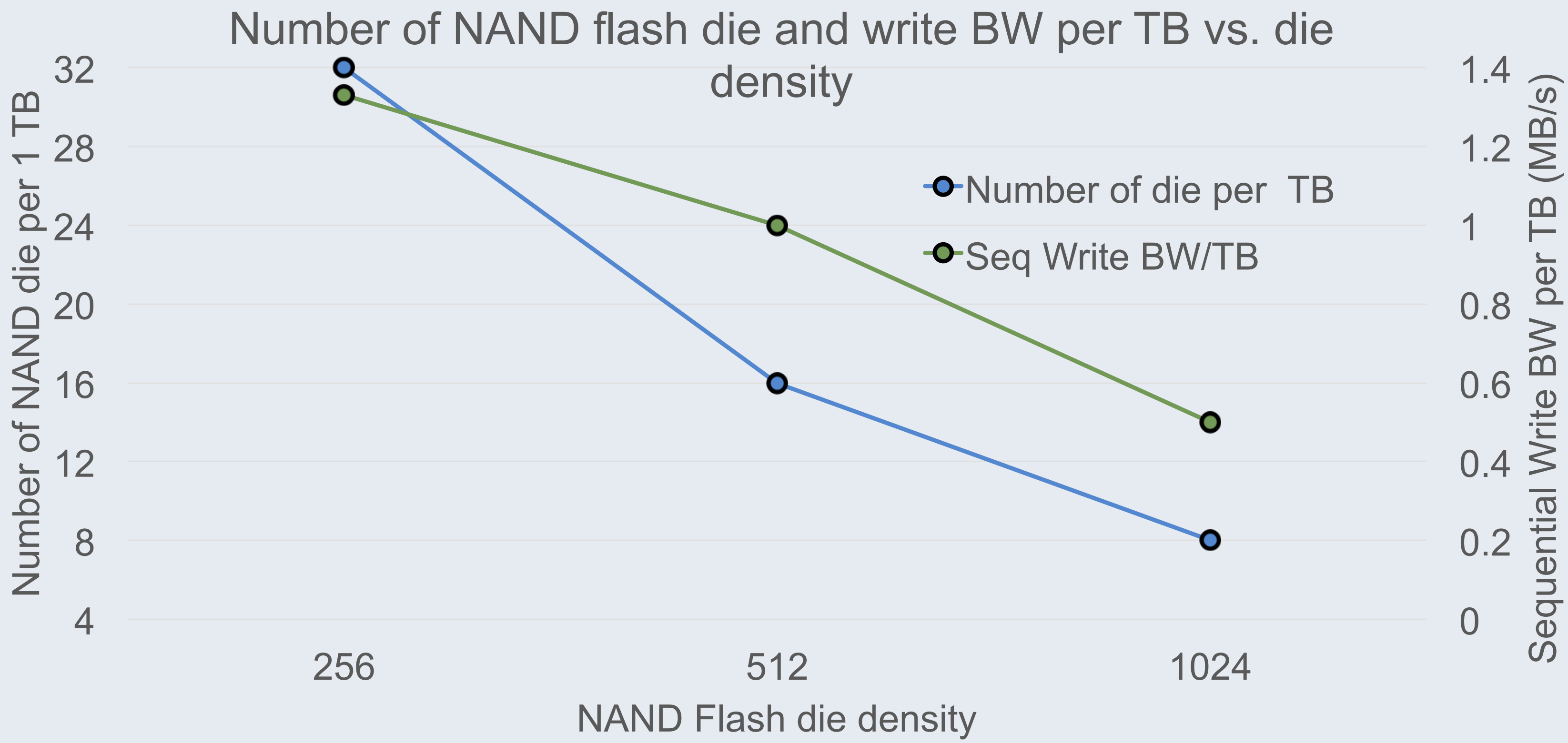
Industry Trends

Flash and CPU continue to diverge

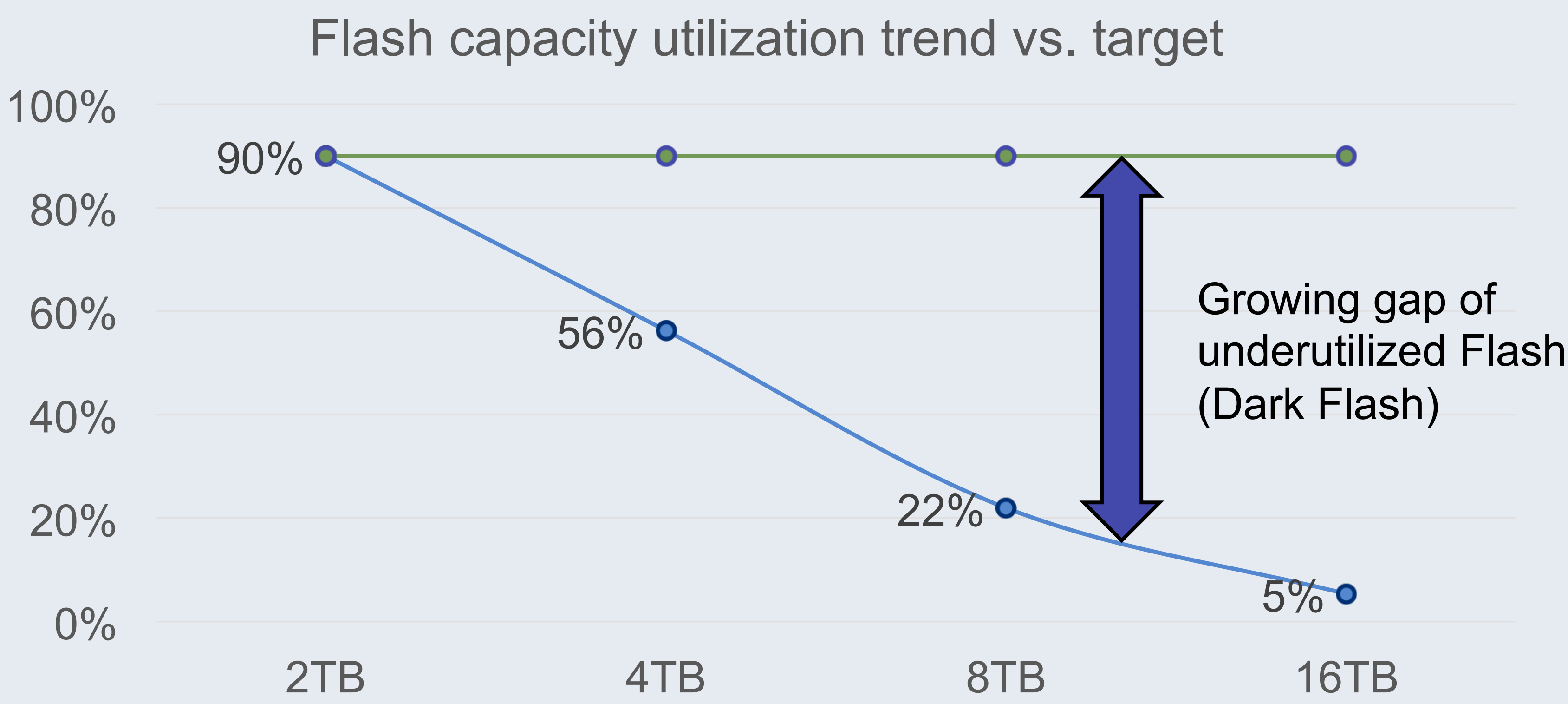


Industry Trends

NAND Flash and SSDs



Dark Flash



Note: Includes 25% generation over generation performance improvements

Flash Workloads @ Facebook

- Read Intensive with bursty writes
- Facebook flash applications are sensitive to read latency
  - Especially read latency outliers
- Multiple, concurrent instances

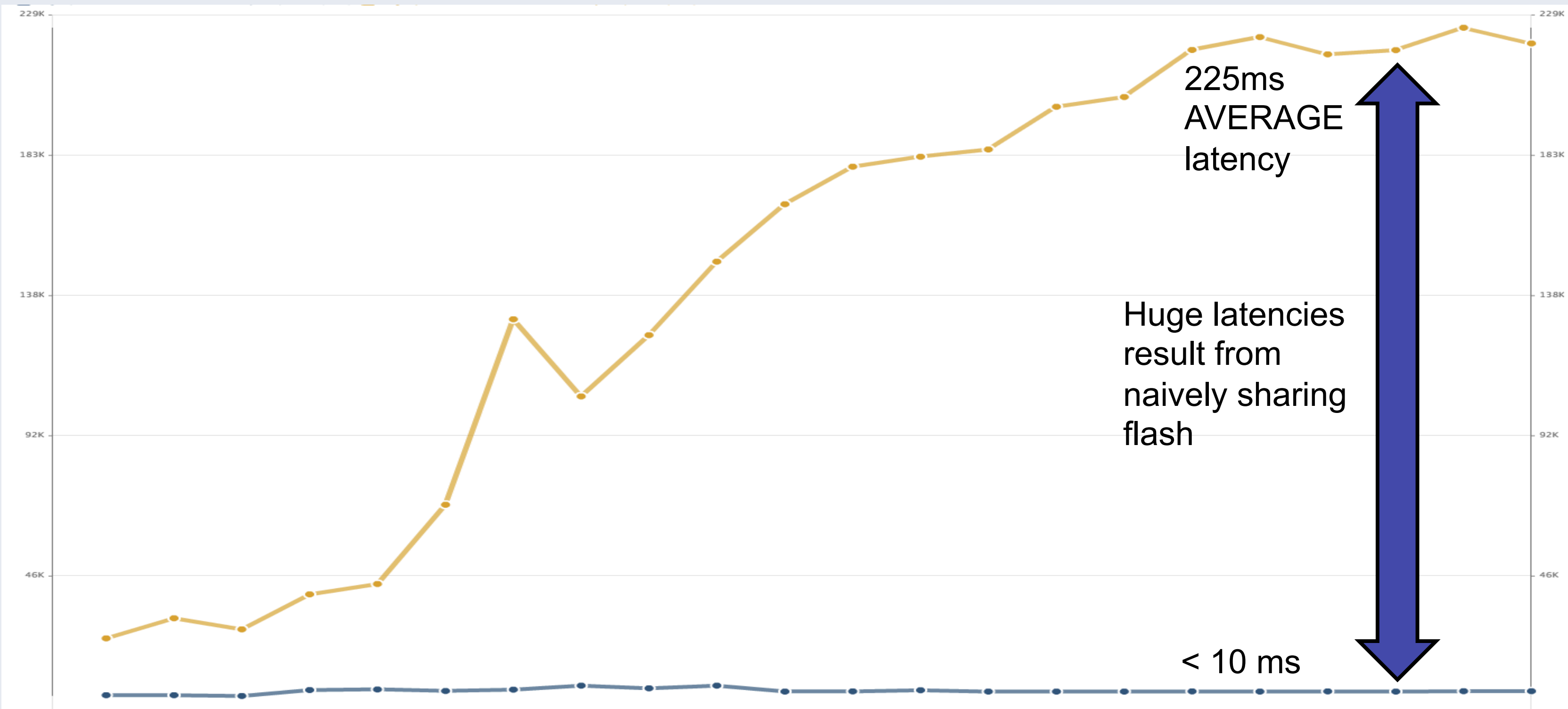
RocksDB

MyRocks

Caching

Sources of Read Latency

External (aka Noisy Neighbors)



Sources of Read Latency

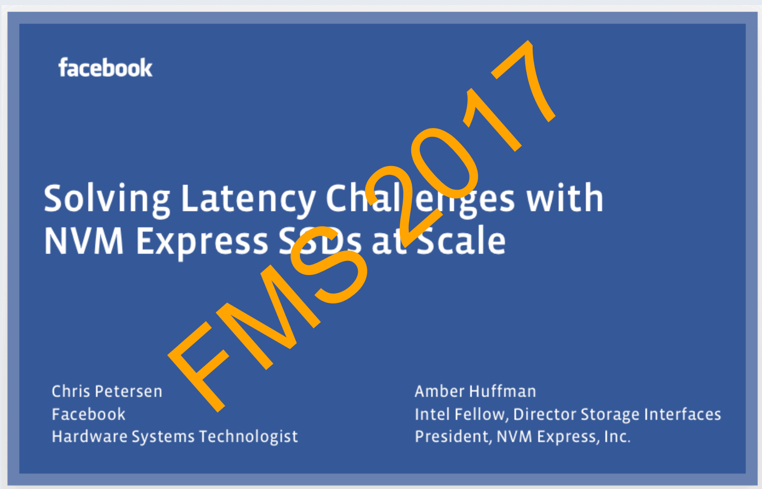
Internal (within the application)

- Read latency outliers are caused by “collisions” with
  1. Concurrent flash writes
  2. Flash background operations:
     - Garbage collection
     - Wear leveling
     - Read scrub
     - Block erase

Read time	~60– 100us
Program time	~1 – 1.5ms
Erase time	~10 – 15ms

- Error correction
- Exception handling (e.g. program/erase failures)

Solution: NVMe® I/O Determinism



NVMe standards have been ratified!

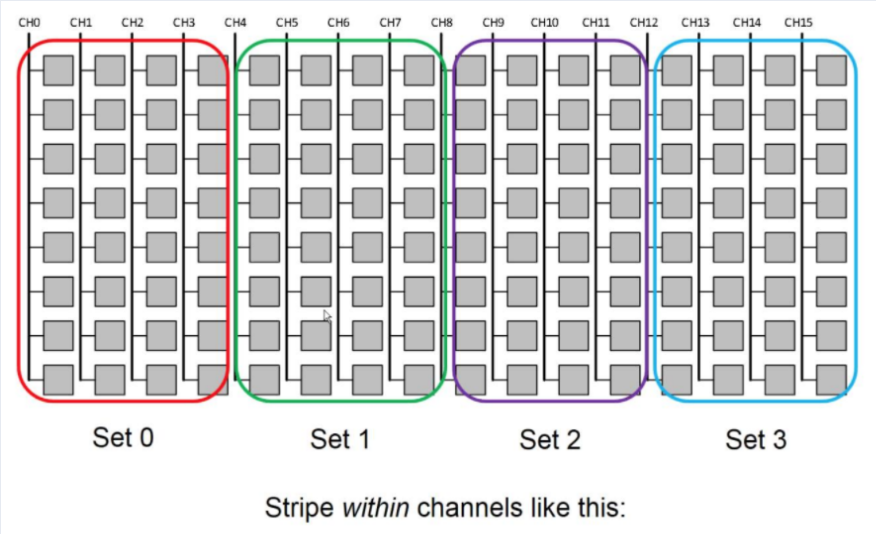
- NVM Sets and Read Recovery Level (TP 4018a)
- NVMe Predictable Latency Mode (TP 4003a)
- Additional improvements are a work in progress!

https://nvmexpress.org/resources/specifications/

NVMe® I/O Determinism: NVM Sets

NVM Sets

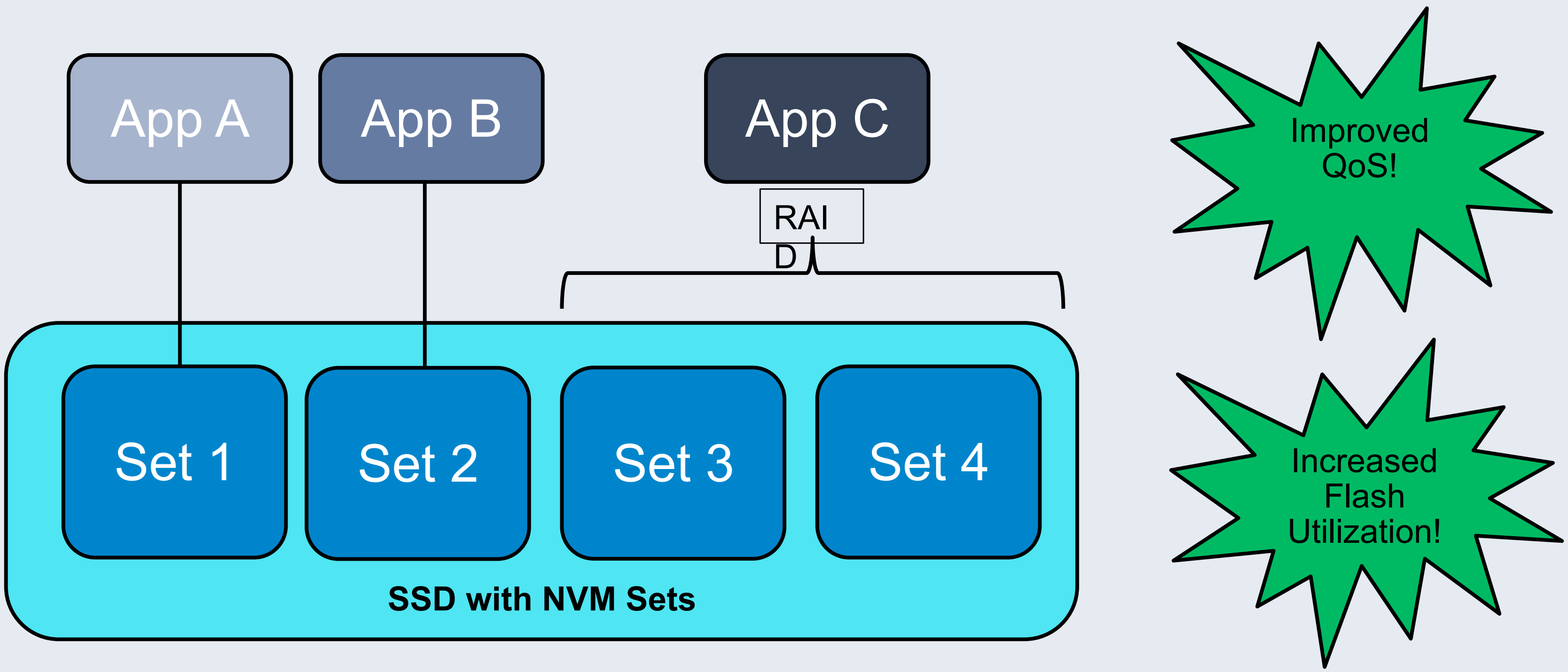
- An abstract allocation of SSD HW resources
- Each set has dedicated NAND resource
- Each set can have dedicated channels, depends on architecture
- Each set carries out its own writes and background operations
- Physically isolated to avoid “Collison” caused by the noisy neighbors



Benefits

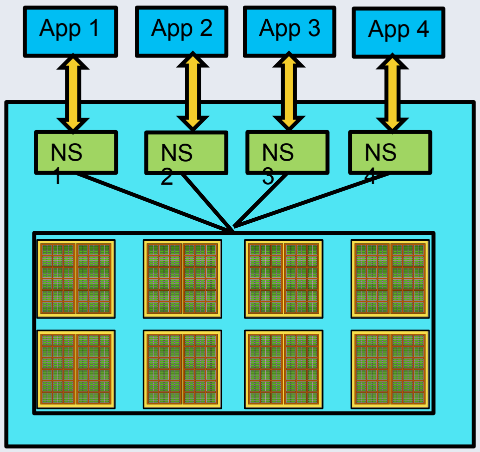
- Enables QoS Regions at the SSD level
  - Better support of multi-tenants on an SSD
- Host software can leverage sets as-is
  - Part of the NVMe Standard
  - Sets are exported as namespaces
  - Host OS does NOT need to be sets-aware

Use Cases @ Facebook

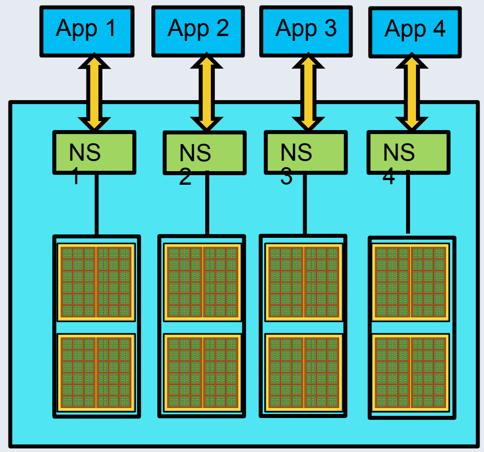


Aligns with Facebook's Disaggregated Flash Strategy!

Evaluation Setup



VS



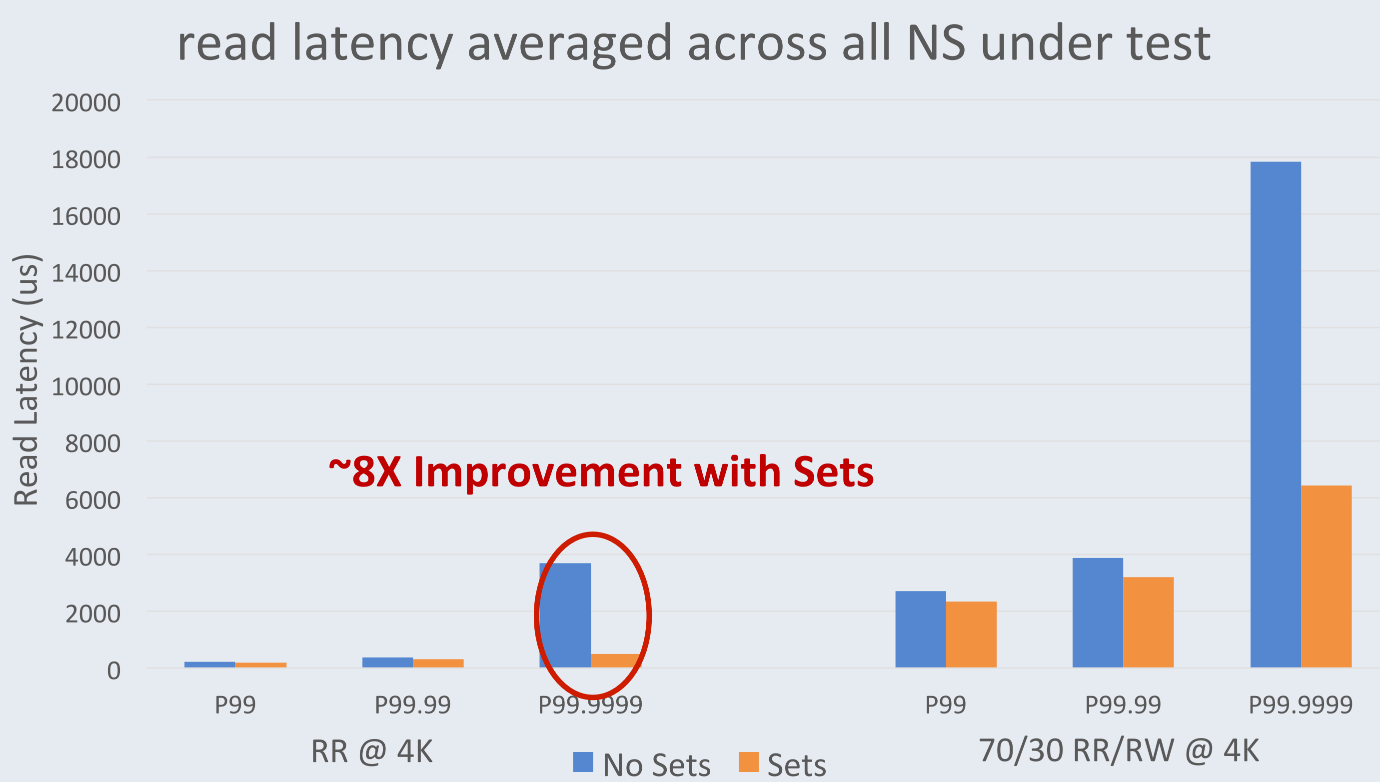
SSD with 4 Namespaces (No Sets)

SSD with 4 Sets

Workload Patterns

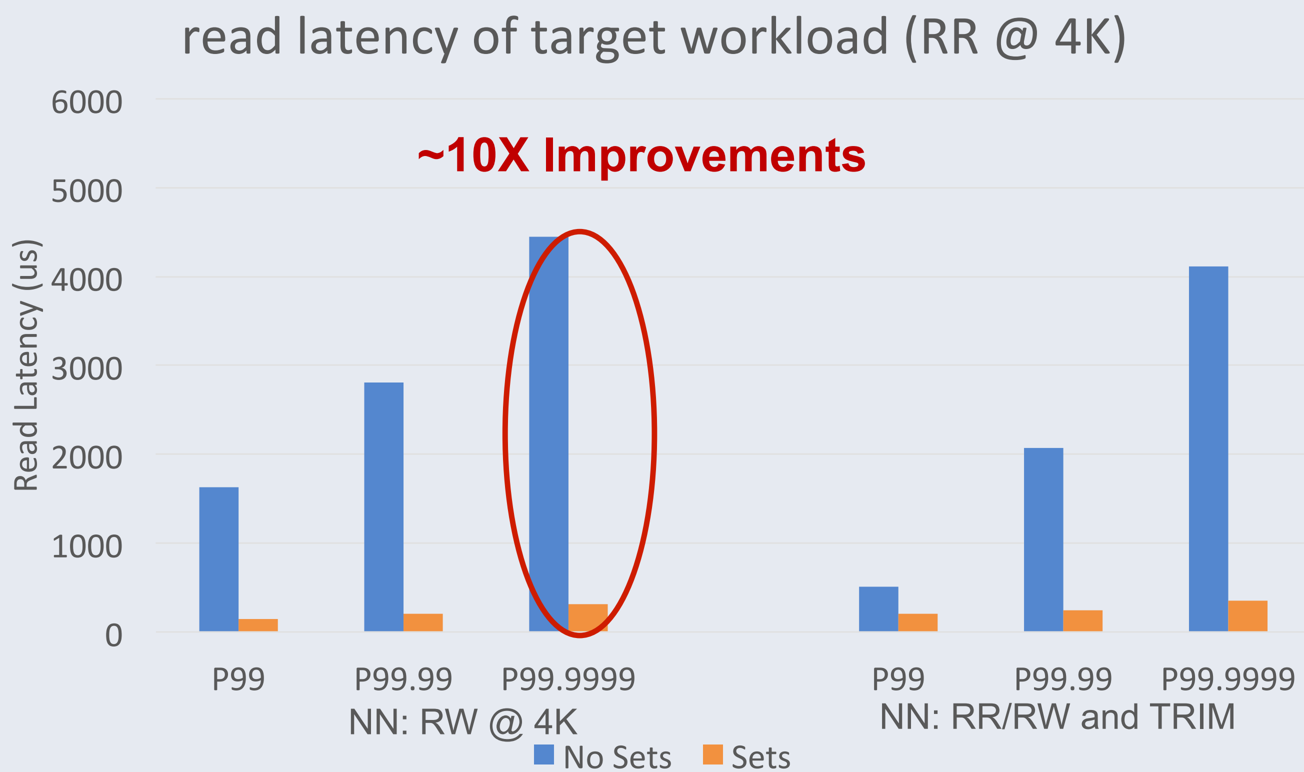
- All namespaces run the same workloads
- Noisy Neighbors
  - One namespace runs the target workload (NS1)
  - The rest of three namespaces act as noisy neighbors (NS2-4)

Neighbors with Same Workloads



NS: Namespace
RR: Random Read
RW: Random Write

Noisy Neighbors



NS: Namespace
NN: Noisy Neighbor
RR: Random Read
RW: Random Write

Current Implementation Limitations

- NVM Sets is non-trivial to implement with the current SSD architecture
  - Hard to partition all resource in the current generation of controllers
  - Design the NextGen sets-aware SSD controller
- Lack of per-set endurance group info
  - TP 4050: Endurance Group Information Enhancements
- SQs are not associated with Sets
  - TP 4045: SQ Associations

facebook

NVMe® I/O Determinism: Predictable Latency Mode

Effects of Internal Activities on Latency

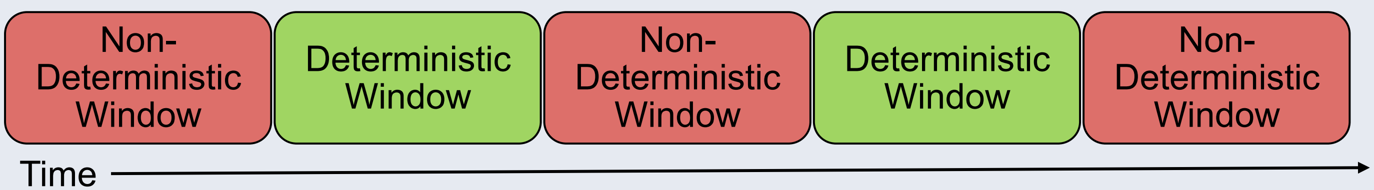
- Internal operations account for most of NAND activity:
  - 7% OP results in 14.3x worst-case WA
  - 28% OP results in 3.6x worst-case WA
- Internal operations usually happen in batches
- Scheduling of internal operations is a black box to the host

Latency Improvement Approaches

- Load limiting (e.g. queue depth, bandwidth)
- Over-provisioning
- Program/erase suspend
- Open-channel
- NVMe Predictable Latency Mode (PLM)

NVMe Predictable Latency Mode (PLM)

- Allows host to decide when internal operations may happen
- Drive encapsulates scheduling algorithm and all media details
- Drive advertises only required details about scheduling capabilities:
  - Estimates of time, # of reads and writes until maintenance is required



NVMe PLM: Contract

Host agrees:

- Not to send writes or trims during D-window
- Respect window estimates advertised by the drive

Drive agrees:

- Not to do operations unrelated to reads during D-window
- Drive may switch back to ND-window if contract is broken

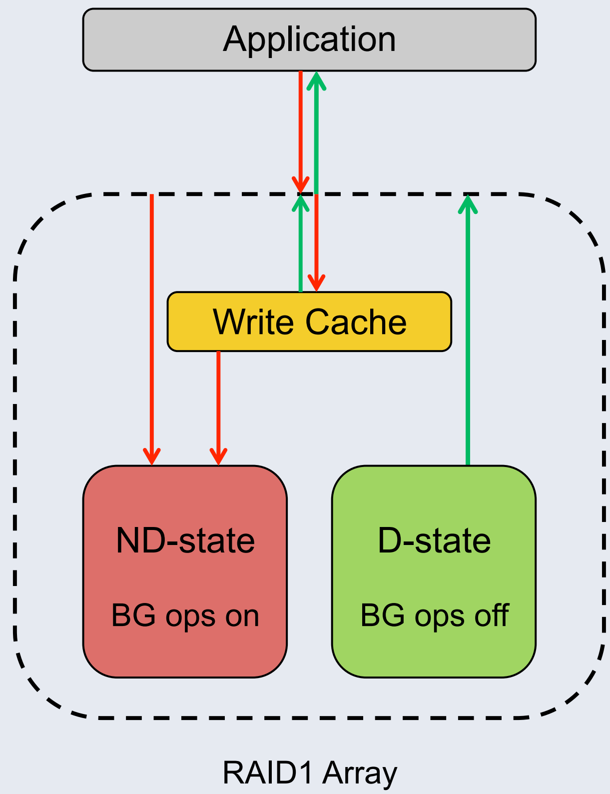
NVMe PLM: Prototype

<u>Goals:</u>

- Improve consistency of read latency
- Achieve read-only like latency for mixed workloads

<u>Approach:</u>

- Leverage data redundancy & PLM to segregate reads from other operations

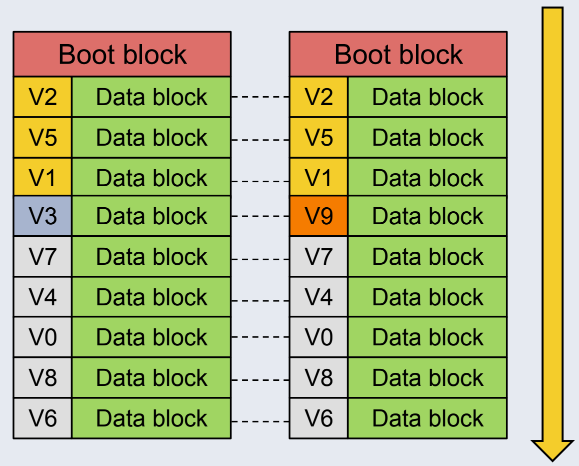


NVMe PLM: Write Cache

Prototype uses RAM cache that relies on NVMe meta-data for power fail recovery

Benefits:

- Flexible configuration
- Minimal impact on performance
- No need for additional hardware
- Allows R/W access during recovery*



Recovery:
1. Check boot blocks
2. Clean shutdown?
3. Check data block pairs for version mismatch

NVMe PLM: Kernel support

MD:

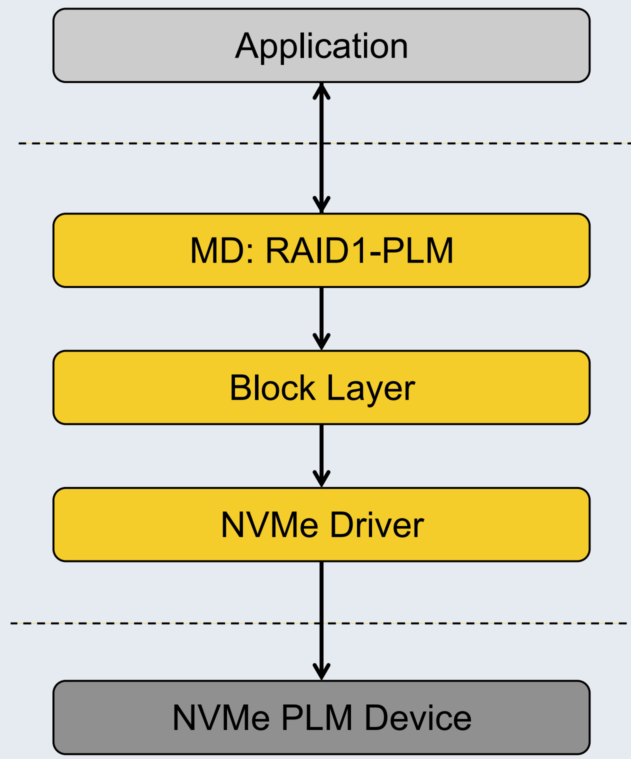
- New Raid1-PLM personality

Block Layer:

- Expose PLM interface
- Expose generic metadata interface

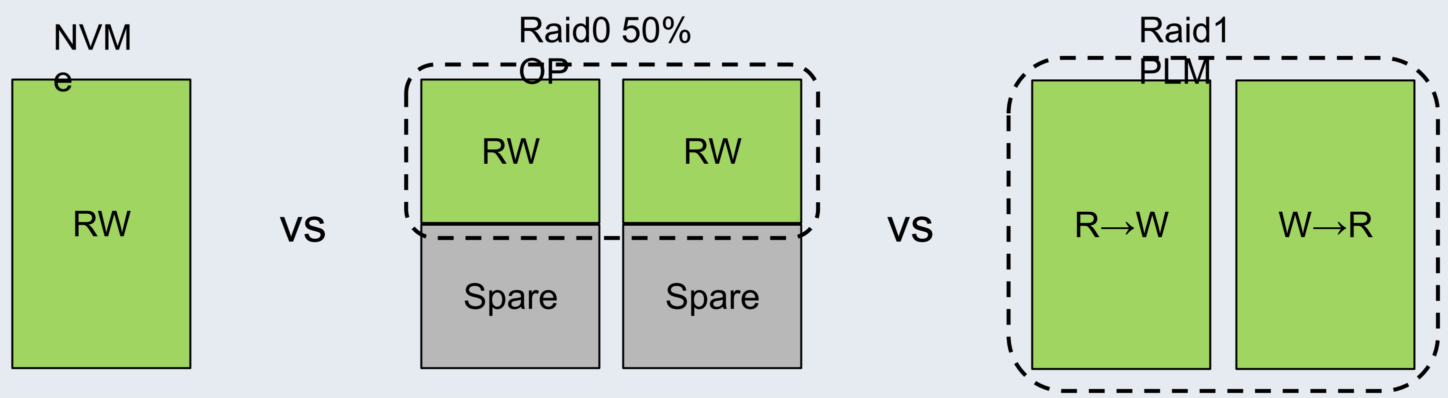
NVMe Driver:

- Implement PLM interface
- Add support for generic metadata
- Add support for set associated SQs

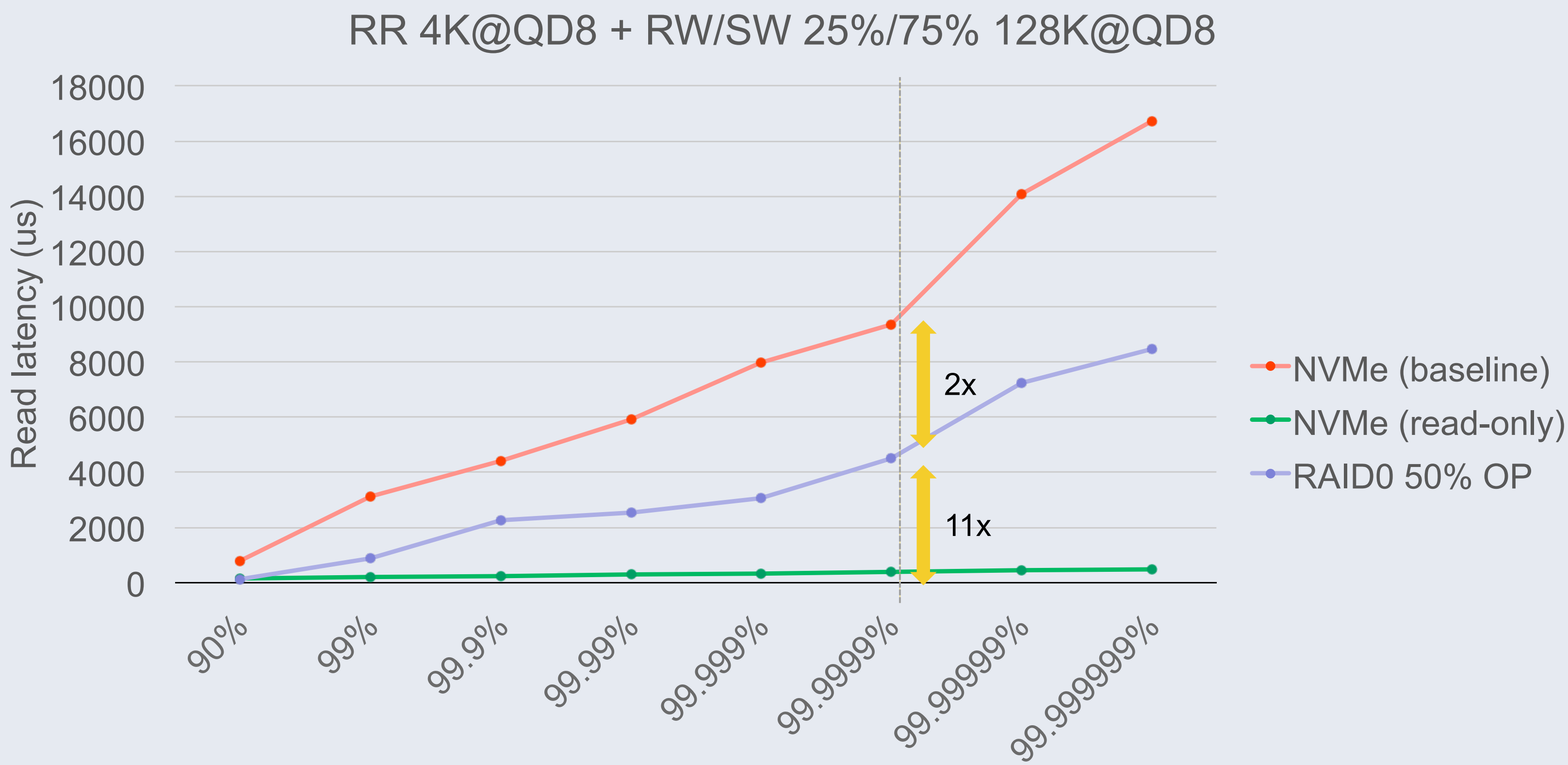


NVMe PLM: Test setup

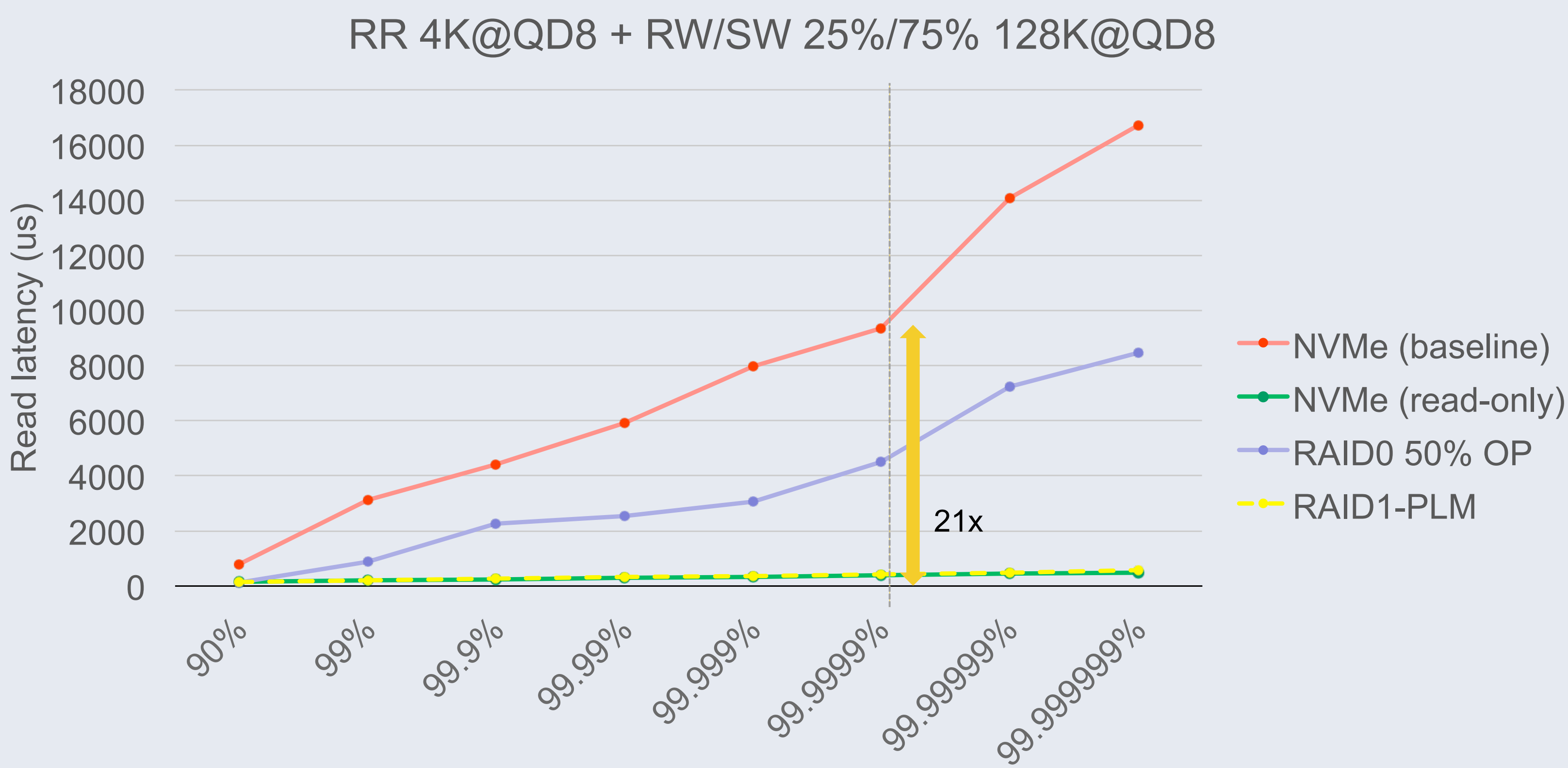
- Same usable capacity, read & write rate
- Random read 4K @ QD8
- 1:2 mix of random and sequential writes 128K @ QD8
- Initialized with 2 passes of mixed writes



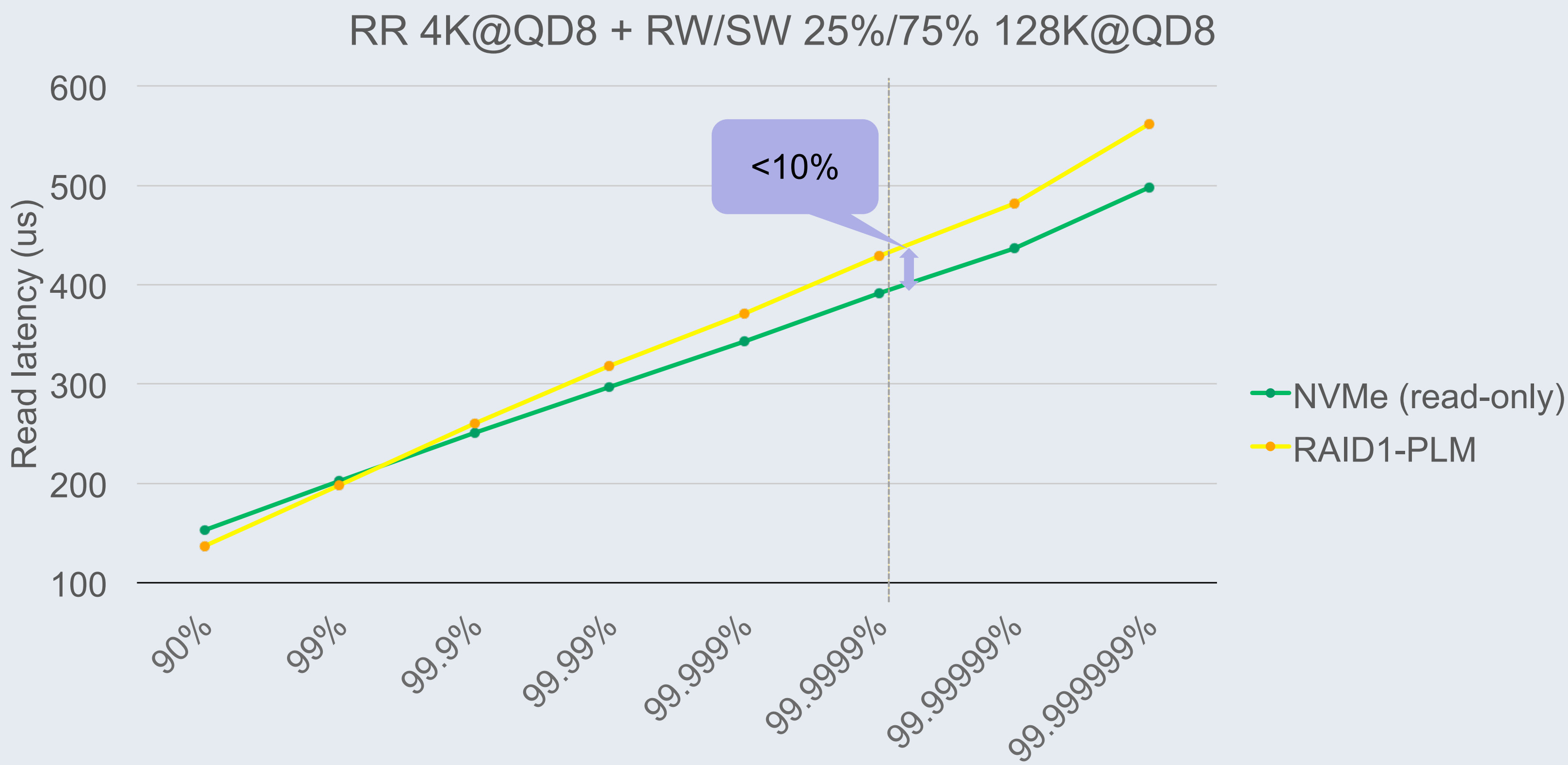
NVMe PLM: Test results



NVMe PLM: Test results



NVMe PLM: Test results



NVMe PLM: Proposition

	RAID1 (2 drives)		RAID5 (4 drives)	
	Array vs 1 Drive	Utilization	Array vs 1 Drive	Utilization
Capacity	100%	50%	300%	75%
Write BW	50%	25%	75%	18.75%
Read BW	100%	50%	200%	50%
Extra Hardware	none		NVRAM	
Read Latency	read-only like		almost read-only like	

NVMe PLM: Future work

- Explore other redundancy schemes
- Explore other power fail recovery schemes
- Explore multi-set configurations (requires TP4045)

facebook

Thank You!

Check out this I/O Determinism talk too:

The Reality of an NVMe IO Deterministic Drive Using QLC
Steven Wells, Fellow – SSD Data Center Architecture
August 7th, 4:50pm