STORAGE



QLC @ Hyperscale

Joseph Cirnigliaro *Meta, Fleet Systems Engineering*

Vineet Parekh *Meta, Fleet Systems Engineering*

OCP PLATINUM Member

A case for QLC SSDs @ Hyperscale



A case for QLC SSDs @ Hyperscale

- HDDs have been growing in density, but not performance
- QLC enables new workloads that are beyond HDD write throughputs
- QLC provides higher rack density and improved power efficiency over TLC SSDs

TLC, QLC & HDD Comparison

Specification	TLC	QLC	HDD (CMR)
CELL			
P/E Cycle Endurance (drive)	1,000–3,000	~1,000	—
DWPD (5-year)	1–3	0.25–0.6	—
PERF			
Sequential Read (GB/s)	7–14	7–14	0.30
Sequential Write, sustained (GB/s)	6–10	2.0–3.5	0.30
Random Read, 4K (IOPS)	2.0–3.3M	0.9–2.0M	168
Write IOPS, 4K	280–900K	25–120K	550
SCALE			
Max Drive Capacity, 2026 (TB)	61.44	122.88–245.76	26
TB per Watt	1.2–3.1	4.9–8.2	2.7–3.1
2U Rack Density (PB @24 drives)	1.47	2.95	0.31
USE CASE			
Target Workload	Write-mixed, latency-sensitive	Read-dominant, capacity-dense	Sequential / bulk
Storage Tier	Hot (Performance)	Warm (between TLC & HDD)	Cold/Mixed

QLC @ Hyperscale Overview

STORAGE



A Case for QLC SSDs Blog

OCP Endurance Storage System

Key Bottlenecks of QLC Storage

- Supply Chain
  - Dense Enterprise QLC have high TCO and are increasingly harder to come by
- Performance
  - QLC holds up under large-block, read-dominant traffic, but sustained writes cap at ~3 GB/s and small random writes quickly become the limiting factor
- Longevity
  - Endurance runs low on QLC with DWPD that is sensitive to small random writes
  - Density raises the stakes over block level error tolerances as these events have the potential of removing hundreds of terabytes worth of capacity

Early Challenges Onboarding QLC

Failure reason	Supplier QLC A	Supplier QLC B	All QLC
IO uncorrectable + media quality	0.35	8.8	1.2
PCIe signal integrity / drive missing	2.0	2.7	2.0
Tooling / non-drive signal	0.58	0.22	0.51
Total	**2.9**	**11.7**	**3.7**

Addressing Key Bottlenecks in QLC SSDs



Mitigating High TCO and Capacity Constraints

Problem

Singular block level failures often result in **full drive failures** raising annual swap rates

Mitigating High TCO and Unavailable Capacity

- The paradigm is shifting toward treating localized media failures and transitory errors as expected events to be managed in place
- Retiring a mostly healthy device over singular bad blocks wastes scarce capacity
- System fault tolerances must allow for added bit errors and still be safe

Mitigating High TCO and Capacity Constraints

Problem

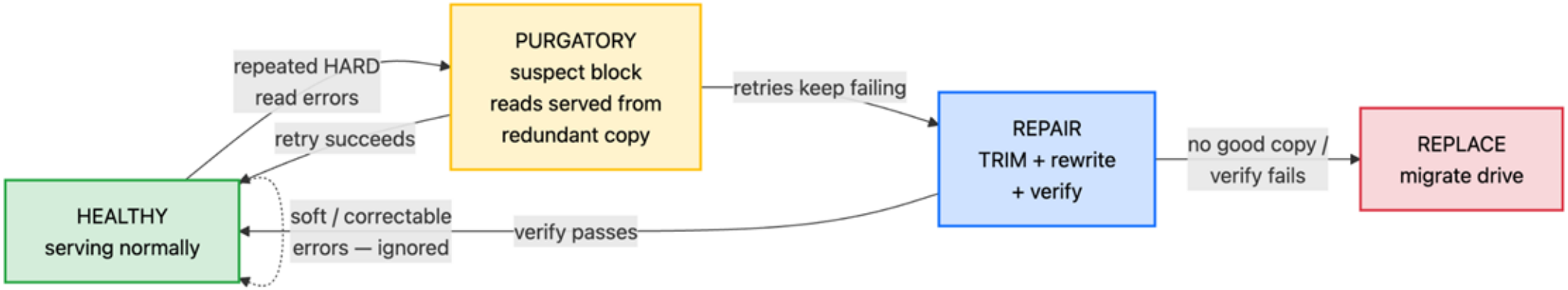
Singular block level failures often result in **full drive failures** driving annual swap rates

Solution

Enable your storage application to be **recoverable and repairable** from block level failures

Mitigating High TCO and Unavailable Capacity

Self-Healing Block Recovery



Addressing QLC Endurance

Problem

Inefficient data placement results in **WAF of 3-5x** causing QLC endurance to be **20% of TLC**

Drive-managed placement

Application
writes carry no placement hints

Block layer
lifetime information is lost in the stack

SSD controller
packs writes into open blocks; guesses grouping

inside the drive



Garbage collection
copies live pages out before erasing

WAF ↑ tail latency ↑ endurance ↓

Host-directed placement

Application
tags each write: hot / warm / cold

NVMe write command
placement ID rides along (dtype = 2)

SSD controller
assign each Placement ID to its own reclaim unit

inside the drive



Host deallocates
frees a whole reclaim unit at once

WAF ↓ steady latency endurance ↑

Addressing QLC Endurance

Problem

Inefficient data placement results in **WAF of 3-5x** causing QLC endurance to be **20% of TLC**

Solution

Adopt NVMe features to enable application to **flexibly place data**. Effective DWPD improves from **0.17 to 0.96**

Call to Action

Flexible Drive Configurations

Open to collaborating with storage partners on innovative solutions to allow flexible, dynamic adjustment of SSD specifications to better fit evolving workload demands.

Support OCP Data Center NVMe® SSD Specification

- Real-time reporting of NVMe buffer fill level
- GC active/idle state + progress (bytes folded/sec)
- Live WAF reporting
- Host I/Os impacted by GC

Thank You

Joseph Cirnigliaro - *Meta, Fleet Systems Engineering*
jcirni@meta.com

Vineet Parekh - *Meta, Fleet Systems Engineering*
vineetparekh@meta.com

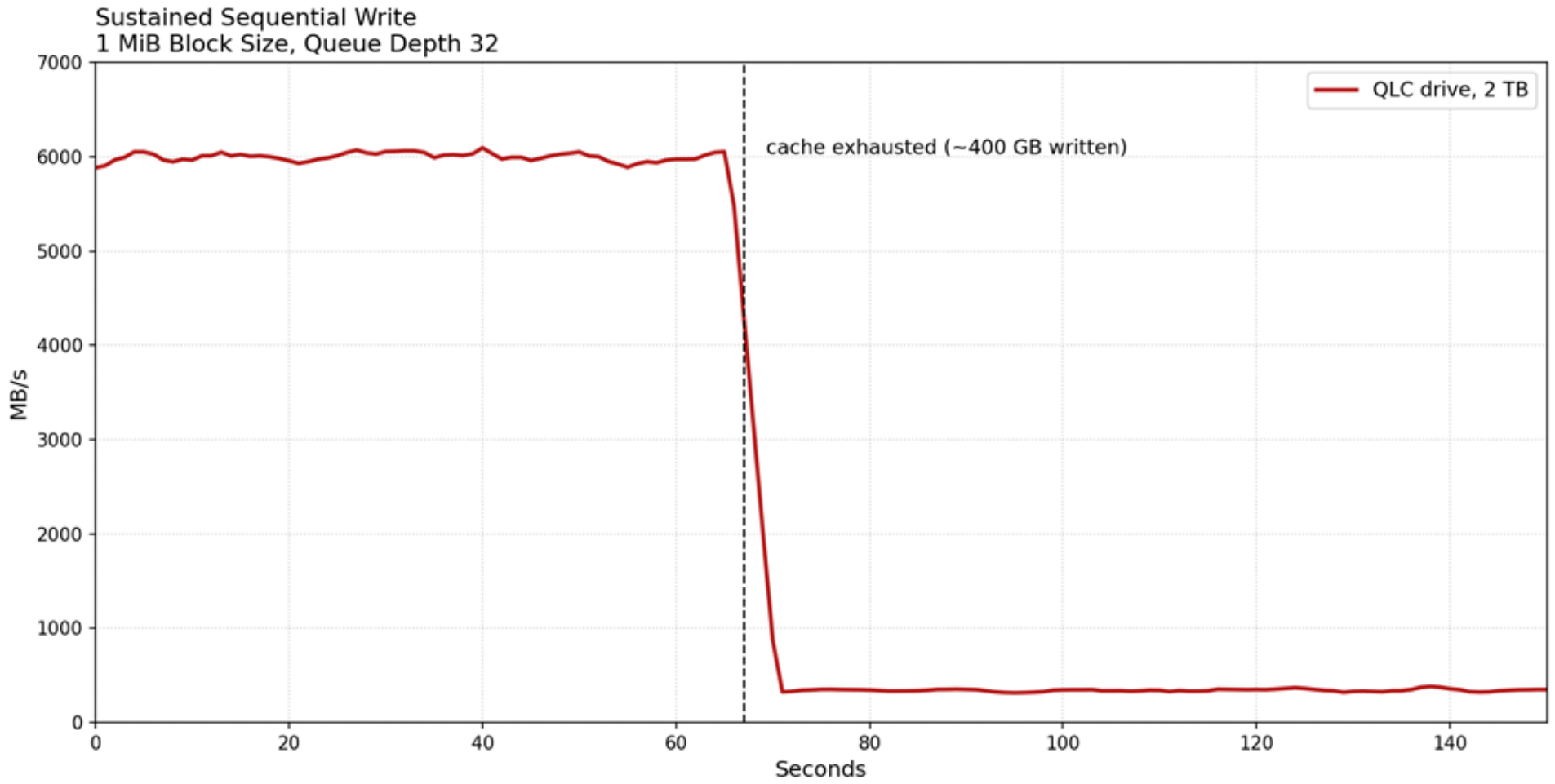
Appendix

Mitigating QLC Sustained Write Performance

Problem

Sustained writes can drop **3–10x** after initial SLC or pSLC buffer fills

Mitigating QLC Sustained Write Performance



Representative QLC Workload Profile



Figure 1. Workload signature of QLC host, note the bursty write pattern in Bytes/s across a 24 QLC drive host

Mitigating QLC Sustained Write Performance

Problem

Sustained writes can drop **3–10x** after initial SLC or pSLC buffer fills

Solution

Make the application **cache-aware** and allocate burst write traffic to **prioritized cooled off drives**

Cache Aware Allocation

Burst Write Traffic

SSD Cache Aware Allocator

READY: cooled off, cache available
H1·D2 H2·D1 H2·D3 H3·D2 H3·D3

IN USE: absorbing, cache partly consumed
H1·D5 H2·D6 H3·D4

COOLING OFF: consumed, timer running
H1·D1 H1·D3 H2·D4 H3·D5 H3·D6

Host-Side Bookkeeping
We do not know the drive's internal GC state but we can de-prioritize recently written to drives in the face of new write traffic

We can benchmark and classify a given drives cache based on its performance cliff to inform our application logic

Host 1 6 drives	D1 COOLING 100% consumed	D2 READY 0% consumed cooled off	D3 COOLING 100% consumed	D4 READY 0% consumed cooled off	D5 IN USE 60% consumed	D6 COOLING 95% consumed
Host 2 6 drives	D1 READY 0% consumed cooled off	D2 COOLING 100% consumed	D3 READY 5% consumed cooled off	D4 COOLING 100% consumed	D5 READY 15% consumed cooled off	D6 IN USE 75% consumed
Host 3 6 drives	D1 READY 5% consumed cooled off	D2 READY 0% consumed cooled off	D3 READY 0% consumed cooled off	D4 IN USE 70% consumed	D5 COOLING 100% consumed	D6 COOLING 90% consumed

READY: cache available | IN USE: absorbing | COOLING OFF: deprioritized