



# MQSim 2.0: A Framework for Realistic Studies of AI-Era SSDs and Disaggregated Storage

Arash Tavakkol<sup>1</sup>, Xin Huang<sup>2</sup>, Xinlei Jia<sup>2</sup>, Longfei Luo<sup>2</sup>, Jiaquan Zhang<sup>2</sup>, Shai Bergman<sup>1</sup>, Ji Zhang<sup>2</sup>

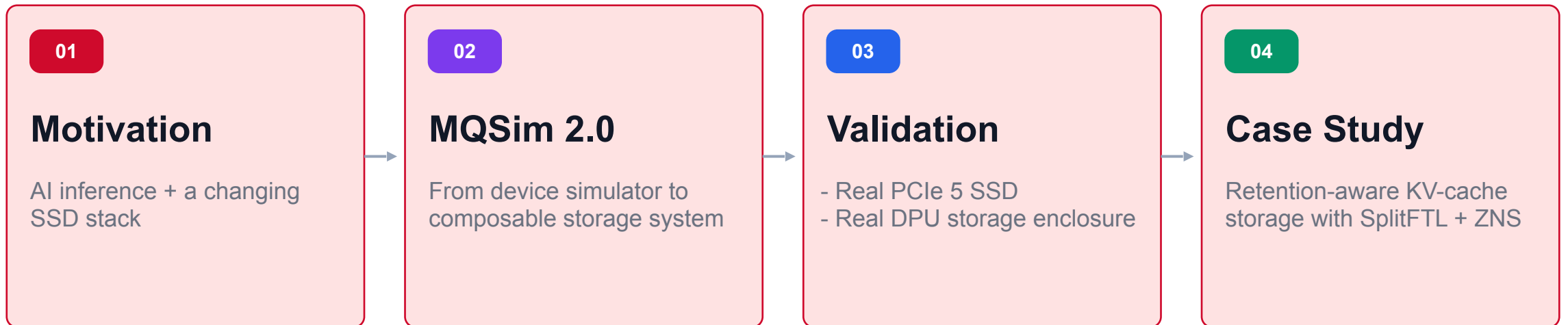
<sup>1</sup> Huawei Technologies Switzerland AG, Switzerland | <sup>2</sup> Huawei Technologies Co., Ltd, China

**Presenter: Arash Tavakkol**

18th ACM Workshop on Hot Topics in Storage and File Systems  
Prague, Czech Republic · September 28–29, 2026

# Agenda

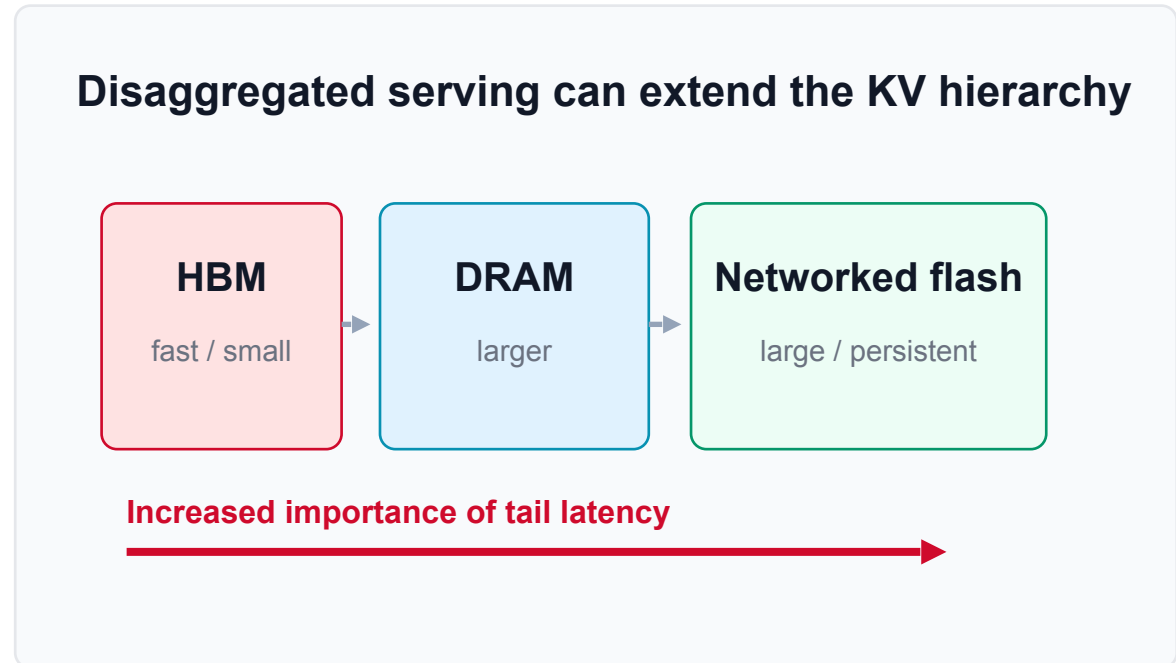
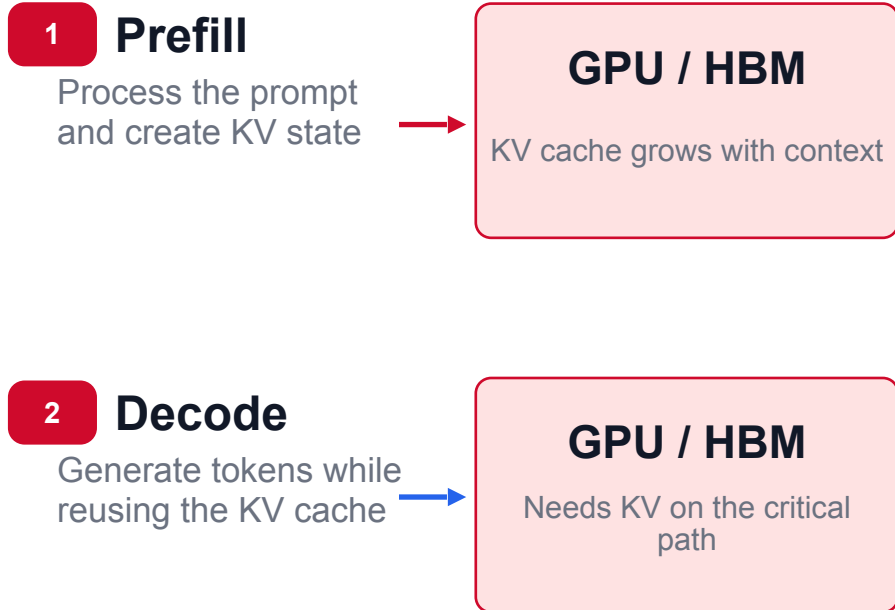
A 15-minute tour: why the model must expand beyond a single SSD



**How do we study modern storage as one coupled system?**

# AI inference is making storage latency-critical

KV-cache state increasingly moves beyond GPU memory in disaggregated serving



**Key point:**  
Once KV state reaches networked flash, the storage subsystem becomes part of the inference latency path.

# The storage stack changed on multiple coupled fronts

A quick SSD refresher

## Minimal SSD model



### Writes are out-of-place

Old pages become invalid; space is reclaimed later.

### Garbage collection (GC)

- Move valid pages + erase blocks
- Consumes the same internal bandwidth as foreground I/O
- Write amplification + latency interference.

## What changed around the SSD since 2018?

### Denser media

QLC + pSLC buffering + one-shot programming

### Host-directed placement

ZNS and FDP expose placement semantics to software

### Control decomposition

Flash-management logic moves to host / array controller / DPU

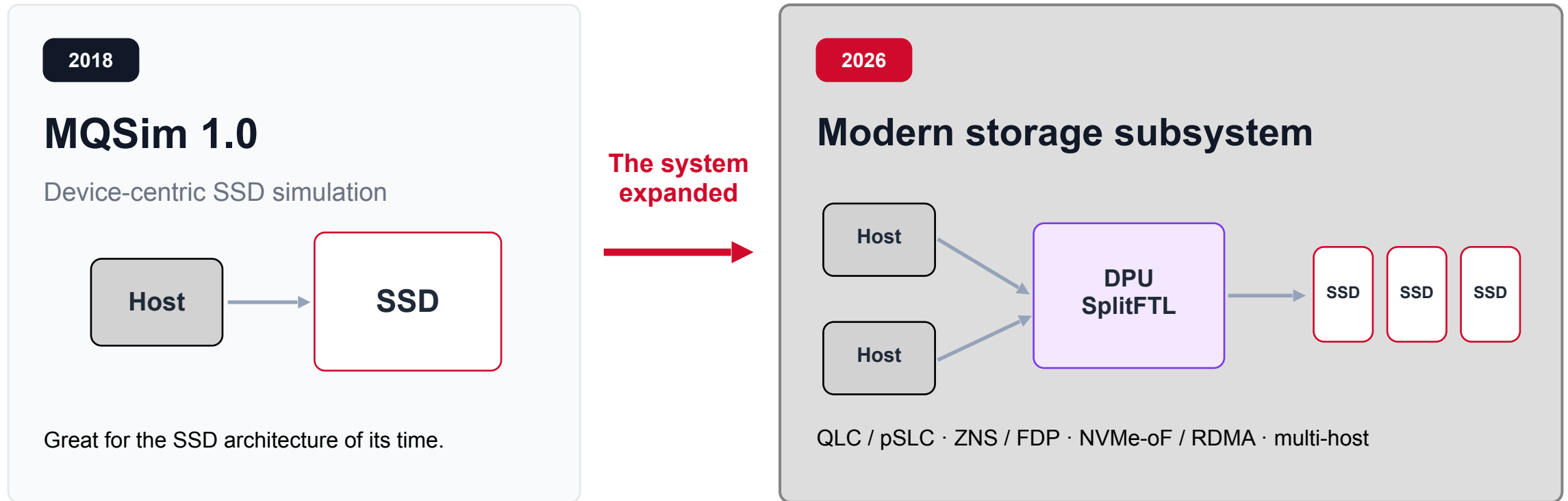
### Disaggregated transport

NVMe-oF / RDMA extend the storage path across the fabric

**These are coupled mechanisms.  
Studying one in isolation can miss the real bottleneck.**

# MQSim was built for a different storage era

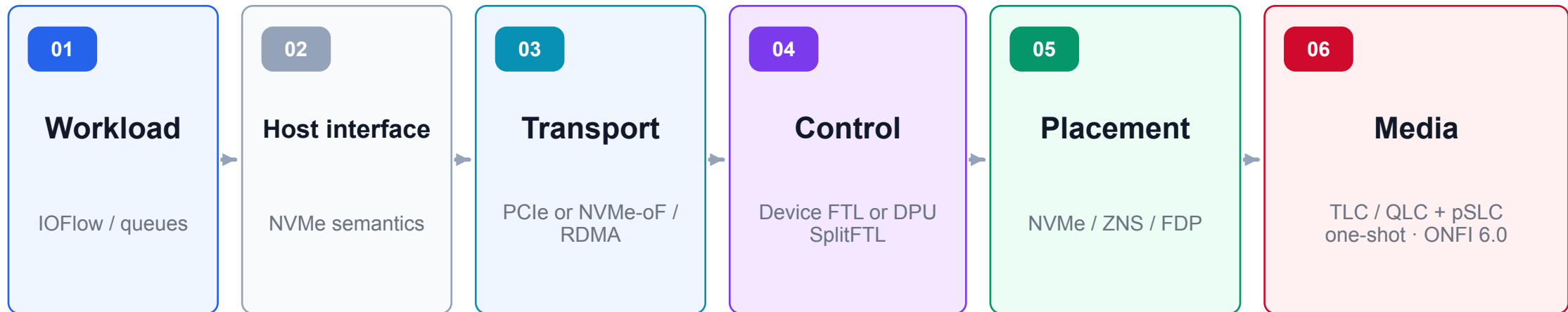
The abstraction must evolve with the system being modelled



**Research gap:**  
Existing tools cover pieces of this stack, but not the coupled end-to-end system in one framework.

# MQSim 2.0: from SSD simulator to composable storage-system simulator

Components have explicit timing and communication paths



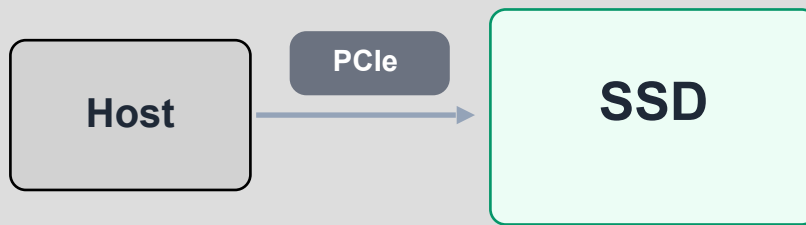
**This is the central architectural shift in MQSim 2.0.**

**Compose the same pieces differently to model direct-attached or disaggregated storage, without changing frameworks.**

# One framework, two deployment topologies

Device interface and deployment topology are orthogonal choices

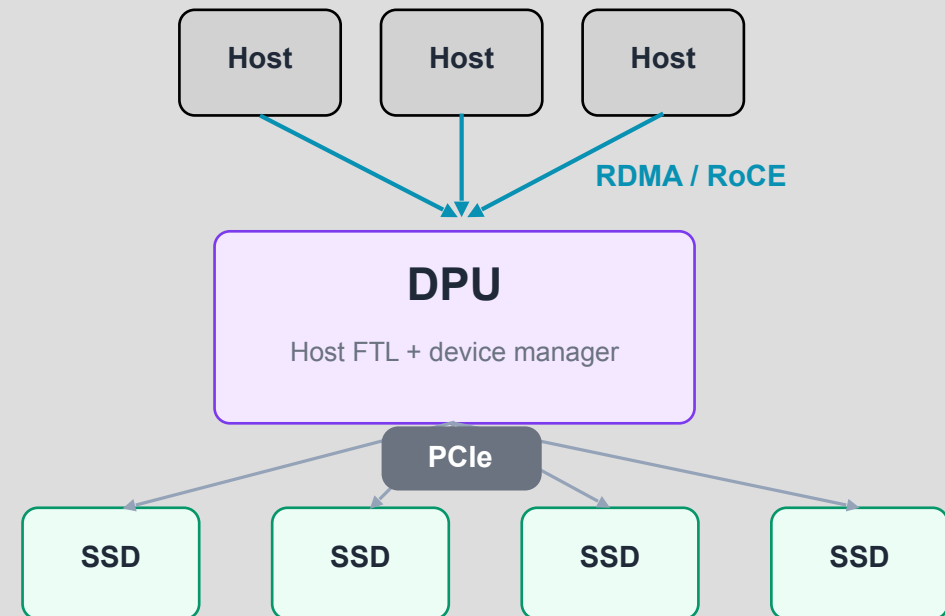
## A Direct-attached local SSD



Backward-compatible with MQSim 1.0 workflows

Updated device models and execution engine

## B Disaggregated SplitFTL



Conventional NVMe, ZNS, or FDP SSDs can be used in either topology.

# What MQSim 2.0 brings under one roof

The point is not any one feature, it is their composition

## Media behavior

- QLC + pSLC cache
- Detailed one-shot programming
- ONFI 6.0 NAND-I/O timing

## Placement

- ZNS semantics
- FDP placement identifiers
- Zone / reclaim-unit isolation

## Control

- Device FTL
- DPU-resident SplitFTL
- Cross-device placement policies

## Transport + topology

- PCIe
- NVMe-oF / RDMA
- Multi-host shared storage

### Secondary contribution

### AI-assisted development

Structured contributor rule files + C++17 refactoring; an extensibility aid, not a correctness mechanism.

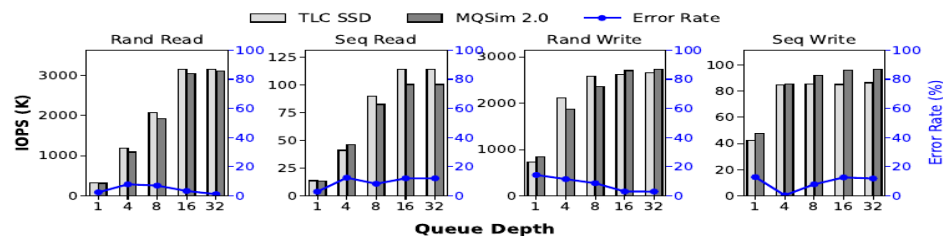
**A researcher can vary media, placement, control, and transport in one event-driven model.**

# Validation: device-level and end-to-end

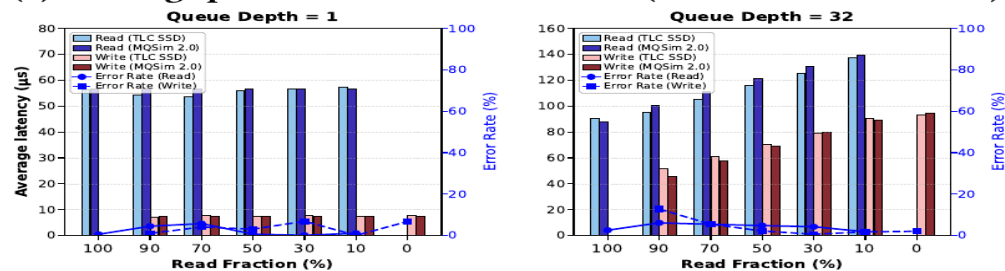
Two real hardware targets exercise different parts of the framework

## TLC NVMe Gen5

### Direct-attached device path



(a) Throughput across four workloads (overall MAPE = 7.9%).



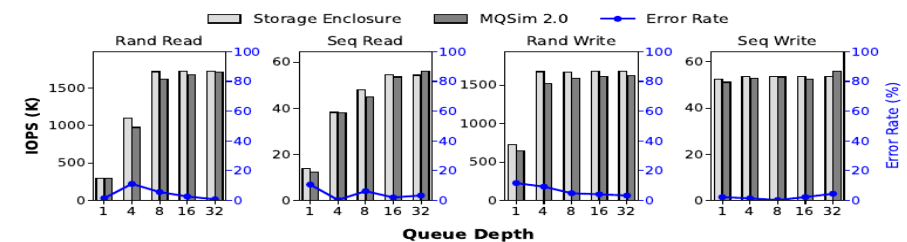
(b) Latency against mixed R/W workloads at different I/O queue depth (overall latency MAPE = 3.5%).

7.9% throughput MAPE

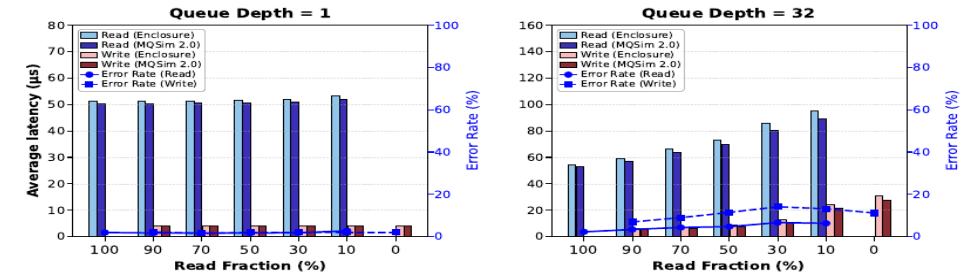
3.5% latency MAPE

## 24 SSD + 4 DPU

### End-to-end disaggregated path



(a) Throughput across four workloads (overall MAPE = 4.4%).



(b) Latency against mixed R/W workloads at different I/O queue depth (overall latency MAPE = 4.8%).

4.4% throughput MAPE

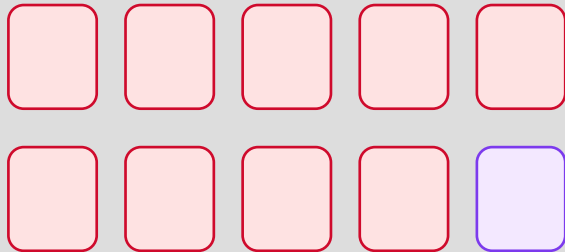
4.8% latency MAPE

The enclosure validation includes DPU SplitFTL + backend SSDs + NVMe-oF/RoCE transport.

# KV-cache case study: mixed lifetimes create flash-management work

Illustrative workload: 90% short-retention objects, 10% long-retention objects

## Case-study retention skew



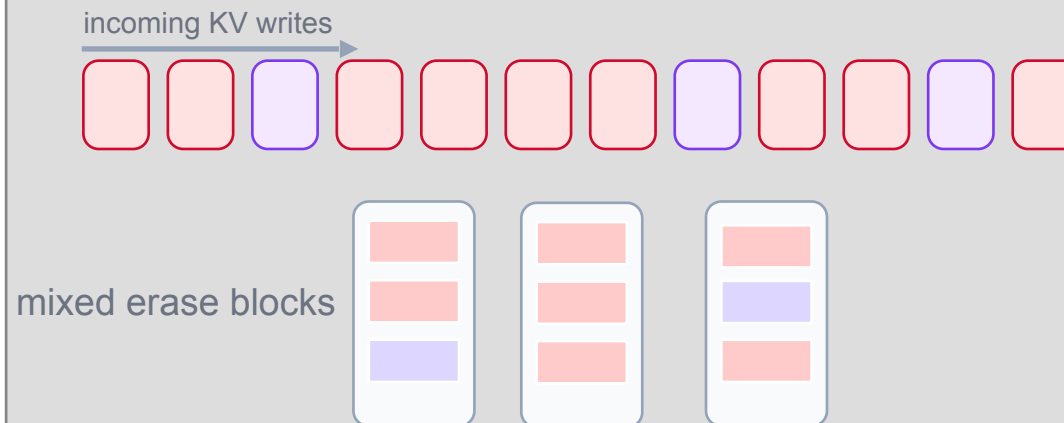
9 short-retention objects

1 long-retention object

Illustrative 90 / 10 parameter based on our analysis of production behaviour

## Conventional JBOF baseline

### Blind striping mixes short- and long-lived KV objects



short-lived objects expire

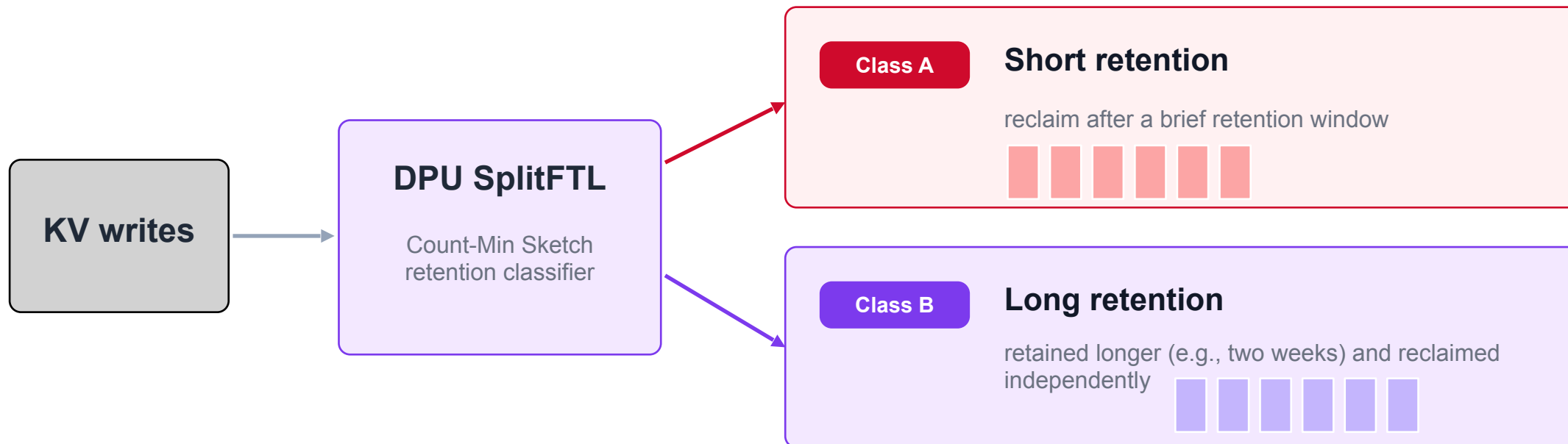


GC must copy surviving long-lived pages

**Result: extra NAND writes + interference with foreground I/O**

# Design: classify by lifetime, place into separate ZNS zones

SplitFTL provides the DPU-resident control point; ZNS provides lifetime isolation



**Separate lifetimes → reclaim zones independently → avoid copying long-lived KV data when short-lived data expires.**

Important nuance: the gains come primarily from ZNS lifetime separation; SplitFTL provides the upstream classification and placement path.

# Case-study result: less internal work, lower tail latency

Compared with the evaluated conventional JBOF baseline

## Write amplification

**48%** lower

Normalized to JBOF = 1.0



Less GC copying and backend work

## P99 latency

**31%** lower

Normalized to JBOF = 1.0



Less GC interference on the write path

Evaluated design point — not an intrinsic performance benefit of SplitFTL alone.

# Takeaways

MQSim 2.0 is a platform for studying the storage system that exists now, and the one being built next

1

## The SSD boundary moved outward

Media, placement, control, and transport now interact across devices, DPUs, and fabric.

2

## MQSim 2.0 models the composition

One event-driven framework covers local and disaggregated storage with modern flash behavior.

3

## The interactions can change design outcomes

In the KV case study, lifetime-aware ZNS placement cut WAF by 48% and P99 latency by 31%.

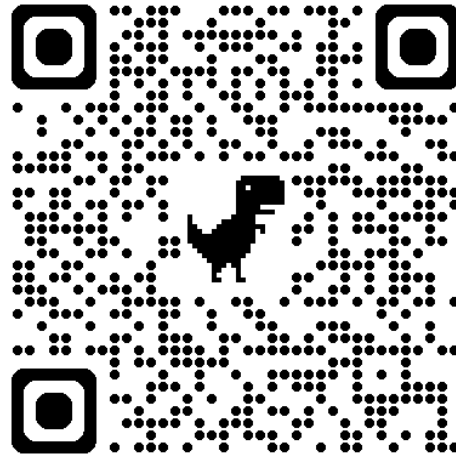
### What comes next:

- We are already extending MQSim 2.0 toward richer AI-inference workloads and deeper cross-layer studies.
- More to share soon.

**HUAWEI** Thank you

**GitHub Repo:**

**<https://github.com/HUAWEI-Storage-System-Lab/MQSim2.0>**



**Time for Questions**

**HUAWEI** Thank you

# Huawei invites researchers in the storage and systems domain to attend the Olympus Mons challenge

The exact timeline would be announced in the month of October

**Challenge 1: Ultra-High-Density Storage and Efficient Retrieval Technology for Mass Data**

**Challenge 2: Memory-Class Storage Media and Near-Storage Acceleration Technology for State Data**

**Challenge 3: Efficient Data Supply Technology Centered on Token Production Efficiency**

**Challenge 4: Highly Generalized and Evolvable Knowledge Representation Technology Based on Ontology-driven Knowledge Graphs**

**HUAWEI** Thank you

# Backup: feature coverage across SSD evaluation platforms

| Feature                       | MQSim 1.0 | NVMeVirt | FEMU | Amber | ConfZNS | FDPVirt | MQSim 2.0 |
|-------------------------------|-----------|----------|------|-------|---------|---------|-----------|
| ZNS                           | —         | ✓        | ✓    | —     | ✓       | ✓       | ✓         |
| FDP                           | —         | —        | —    | —     | —       | ✓       | ✓         |
| QLC + pSLC cache              | —         | —        | —    | —     | —       | —       | ✓         |
| Detailed one-shot programming | —         | —        | —    | —     | —       | —       | ✓         |
| ONFI 6.0 NAND-I/O timing      | —         | —        | —    | —     | —       | —       | ✓         |
| DPU-resident SplitFTL         | —         | —        | —    | —     | —       | —       | ✓         |
| NVMe-oF / RDMA transport      | —         | —        | —    | —     | —       | —       | ✓         |
| Multi-host shared storage     | —         | —        | —    | —     | —       | —       | ✓         |
| Structured AI-coding rules    | —         | —        | —    | —     | —       | —       | ✓         |

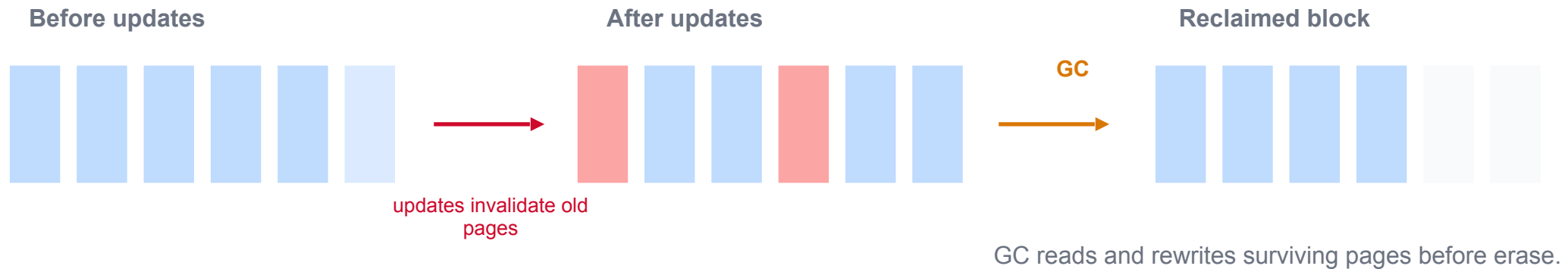
Table reflects publicly available versions as of the June 2026 submission date.

BACKUP

# Backup: why garbage collection creates interference

## Out-of-place update

A page cannot be overwritten until its erase block is erased.



**That background work consumes the same flash resources as foreground I/O — which is why placement and data lifetime matter.**

# Backup: ZNS vs. FDP — the high-level distinction

## ZNS

### Host-managed sequential zones

- Append-only write discipline
- Explicit zone lifecycle + reset
- Can largely eliminate device-side GC within zones
- Stronger host responsibility

## FDP

### Host-guided placement identifiers

- Normal writes carry placement IDs
- SSD groups data into reclaim units
- Separates lifetimes without append-only semantics
- Less restrictive host model

**The KV case study uses ZNS because lifetime-isolated zones map naturally to born-together / die-together KV objects.**

BACKUP

# Backup: validation configurations and scope

## TLC NVMe Gen5 SSD

3.84 TB

Datasheet: 3.47M read IOPS / 787K write IOPS

7% over-provisioning

Purpose: validate the direct-attached device path under the evaluated workloads.

Result: 7.9% throughput MAPE; 3.5% overall latency MAPE.

## DPU-based storage enclosure

24 U.2 NVMe QLC SSDs

4 active-active DPUs (16 cores, 64 GB DDR5 each)

NVMe-oF over RoCE v2; 40 GB/s per DPU

~2  $\mu$ s unloaded fabric RTT

Single remote node: observed aggregate bandwidth capped at ~7 GB/s

Purpose: validate the end-to-end disaggregated path.

Result: 4.4% throughput MAPE; 4.8% overall latency MAPE.

Scope caveat: these experiments validate the evaluated configurations end to end — not every modeled mechanism independently.

BACKUP

# Backup: case-study assumptions and limitations

## Retention distribution

The 90% short-retention fraction is an illustrative workload parameter, not a universal production measurement.

## Trace realism

The workshop case study is parameterized rather than a replay of a production KV-cache trace.

## Validation scope

The TLC SSD and 24-SSD enclosure validate evaluated end-to-end paths, not every mechanism independently.

## Media-model validation

The paper does not provide separate hardware validation of the pSLC and one-shot QLC models.

## What is left open

Broader workload sensitivity and additional device-level validation are future work.

These limitations are why the paper presents the KV result as an illustrative end-to-end design point.

BACKUP