**ZONED NAMESPACES (ZNS)**

Concepts, Use Cases & Open Ecosystem

---

### Why a New Interface?

**Log-on-log Problem**

- **TCO**
  - Device WAF ~ 1
  - Less DRAM

- **Multi-tenancy**
  - QLC NAND
  - Less device OP

---

### ZNS Concepts

**Zoned LBA Space**

- **# Zone state machine**
  - Zone transitions managed by host
  - Zone excursions triggered by device (AER)

- **# Append Command (Nameless Write)**
  - Increase QD within zone by offloading LBA map
  - Require major changes to host (FS / Applications)
  - Full reimplementation of LBA-based features
    - E.g., snapshotting

---

### ZNS Extensions

**ZRWA**

- **# Concept**
  - Zone Random Write Area (ZRWA) exposes host write buffer in front of a given zone

- **# Why**
  - Enable QD>1 writes without requiring changes to host (as in append case)
  - Support in-place updates to further reduce WAF in real workloads (metadata)

---

### Use Cases

**Archival**

- **Goals**
  - TCO, WAF, DRAM
  - Facilitate QLC consumption

- **Operation**
  - Store cold, mostly immutable data
  - Build on top of SMR use cases
  - Very large zones
  - Reduce metadata
  - Need for large QD (ZRWA or Append)

---

### I/O Predictability

- **# Goals**
  - Inherit TCO, WAF, DRAM
  - Enable multi-tenancy - Noisy Neighbor
  - Cover Open-Channel SSD use cases

- **# Operation**
  - Provide **Isolation Domains** for zones
  - Avoid 100s of namespaces (NVM Sets)
  - Use existing zone properties
  - Host data placement and I/O Scheduling

---

### Upstream Linux Ecosystem

**Linux Stack**

- **# Build on top of zoned block**
  - Add support for ZNS features
  - NVMe driver
  - Block Layer
  - Log FS: F2FS, XFS
  - User Space support
    - Libraries, cli too
    - fio
  - QEMU Emulation

- **xNVMe**
  - Enable non-block NVMe
  - Simplify application support
  - Multiple backends, no changes
    - Linux
    - iocctng, libaio*, psync*
  - ioctct
  - SPDK
  - FreeBSD
  - Windows*