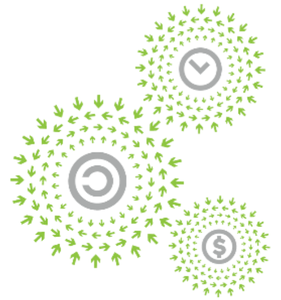
Open. Together.

OCP SUMMIT

STORAGE

From Open-Channel SSDs to Zoned Namespaces

Matias Bjørling, Director, Western Digital



Western Digital®

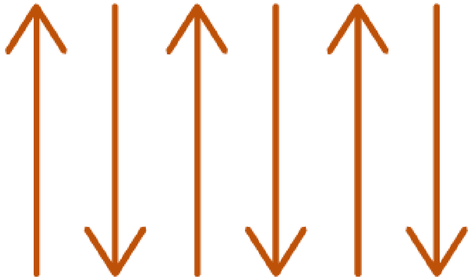
Forward-Looking Statements

Safe Harbor | Disclaimers

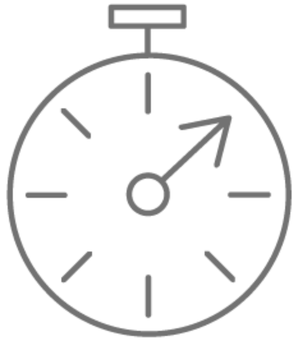
This presentation contains forward-looking statements that involve risks and uncertainties, including, but not limited to, statements regarding our solid-state technologies, product development efforts, software development and potential contributions, growth opportunities, and demand and market trends. Forward-looking statements should not be read as a guarantee of future performance or results, and will not necessarily be accurate indications of the times at, or by, which such performance or results will be achieved, if at all. Forward-looking statements are subject to risks and uncertainties that could cause actual performance or results to differ materially from those expressed in or suggested by the forward-looking statements.

Key risks and uncertainties include volatility in global economic conditions, business conditions and growth in the storage ecosystem, impact of competitive products and pricing, market acceptance and cost of commodity materials and specialized product components, actions by competitors, unexpected advances in competing technologies, difficulties or delays in manufacturing, and other risks and uncertainties listed in the company's filings with the Securities and Exchange Commission (the "SEC") and available on the SEC's website at www.sec.gov, including our most recently filed periodic report, to which your attention is directed. We do not undertake any obligation to publicly update or revise any forward-looking statement, whether as a result of new information, future developments or otherwise, except as required by law.

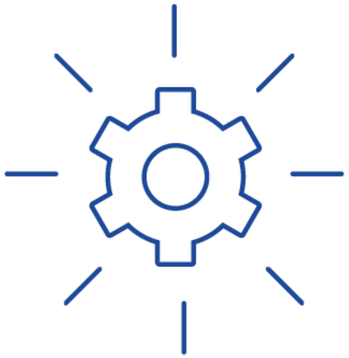
Open-Channel SSDs



I/O Isolation



Predictable Latency



Data Placement &
I/O Scheduling

Ubiquitous Workloads

Efficiency of the Cloud requires many different workloads to a single SSD

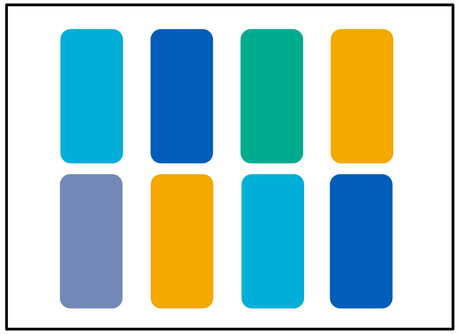
Databases

Sensors

Analytics

Virtualization

Video



Solid-State Drive

Open-Channel SSD Concepts

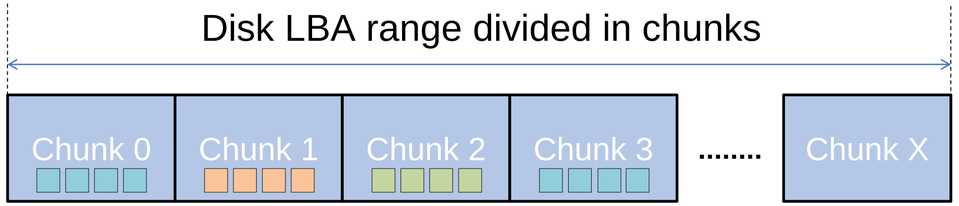
Chunks & Parallel Units

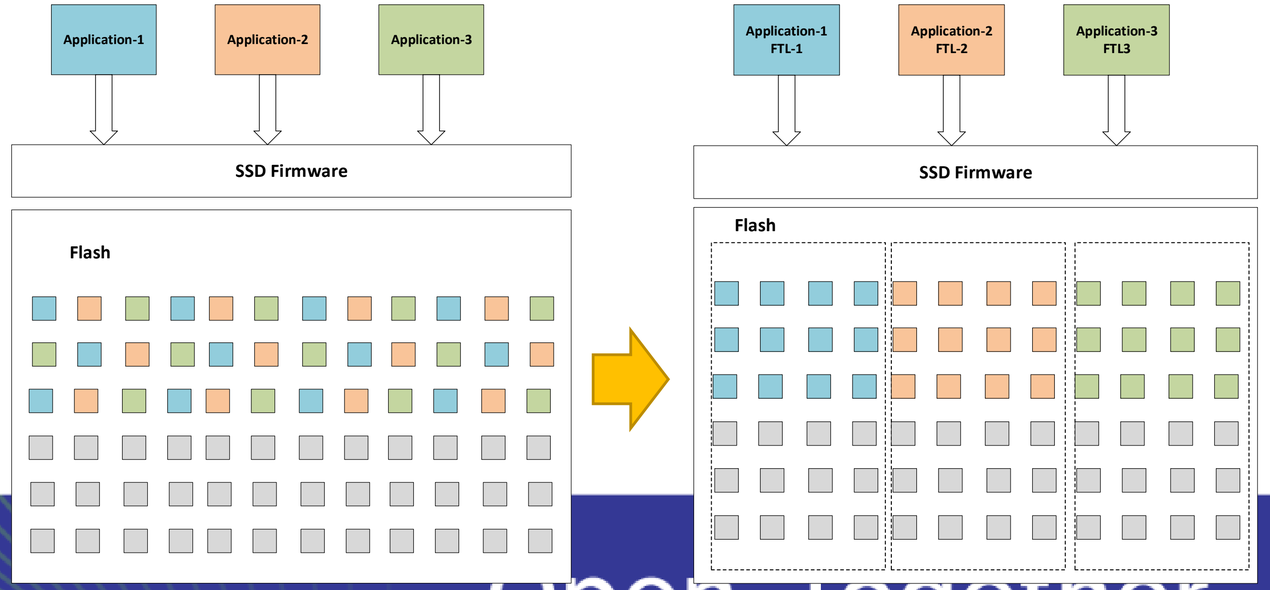
Chunks

- Write sequentially within an LBA range
- Requires reset for rewrites
- Borrows from HDD's SMR specification (ZAC/ZBC)
- Optimized for SSD physical constraints
  - Align writes to media layout

Parallel Units

- Host can direct I/Os to separate workloads
- Stripes across single or multiple dies.
- The parallel units inherits the throughput and latency characteristics of the underlying media
- Served by I/O determinism (NVMe™ 1.4)





Industry Adoption

Hyper-scalers, all-flash array vendors, and large storage system vendors that have been considering or uses Open-Channel SSD architectures can now benefit from standardization and a broader eco-system.

Key concepts to be introduced into the NVMe™ specification

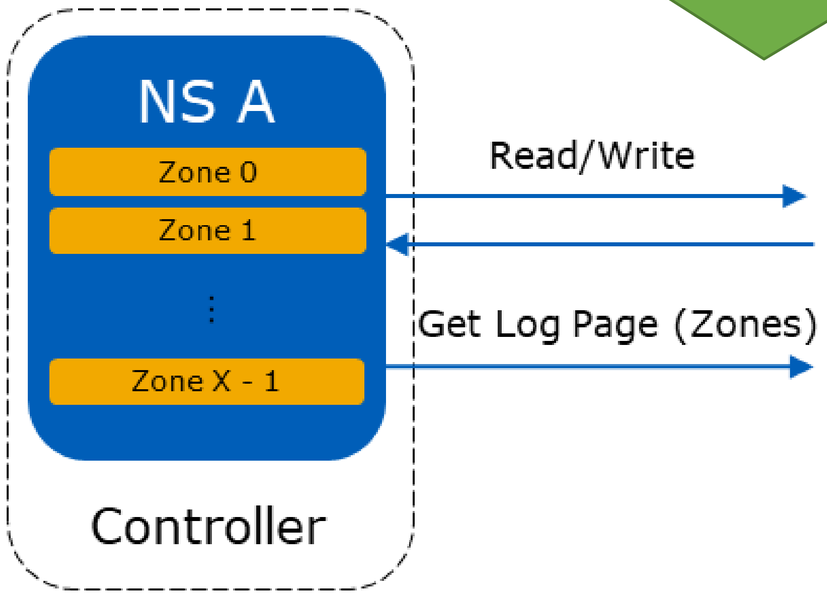
Zoned Namespaces (ZNS)

Under Standardization

Technical Proposal in the NVMe™ working group

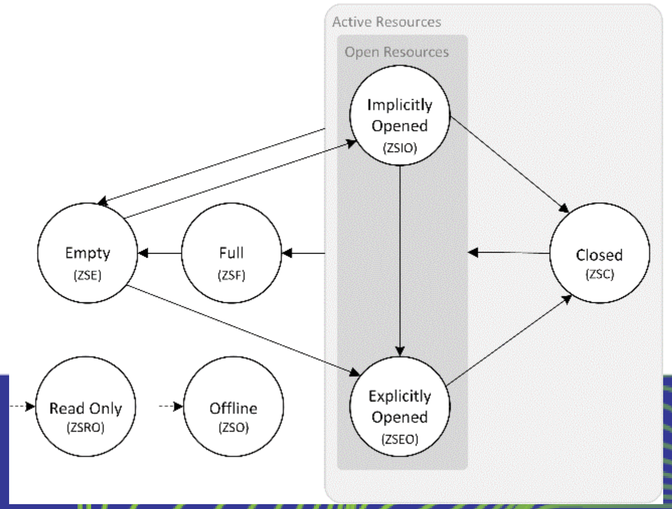
Standardizes zone interface as an approach to:

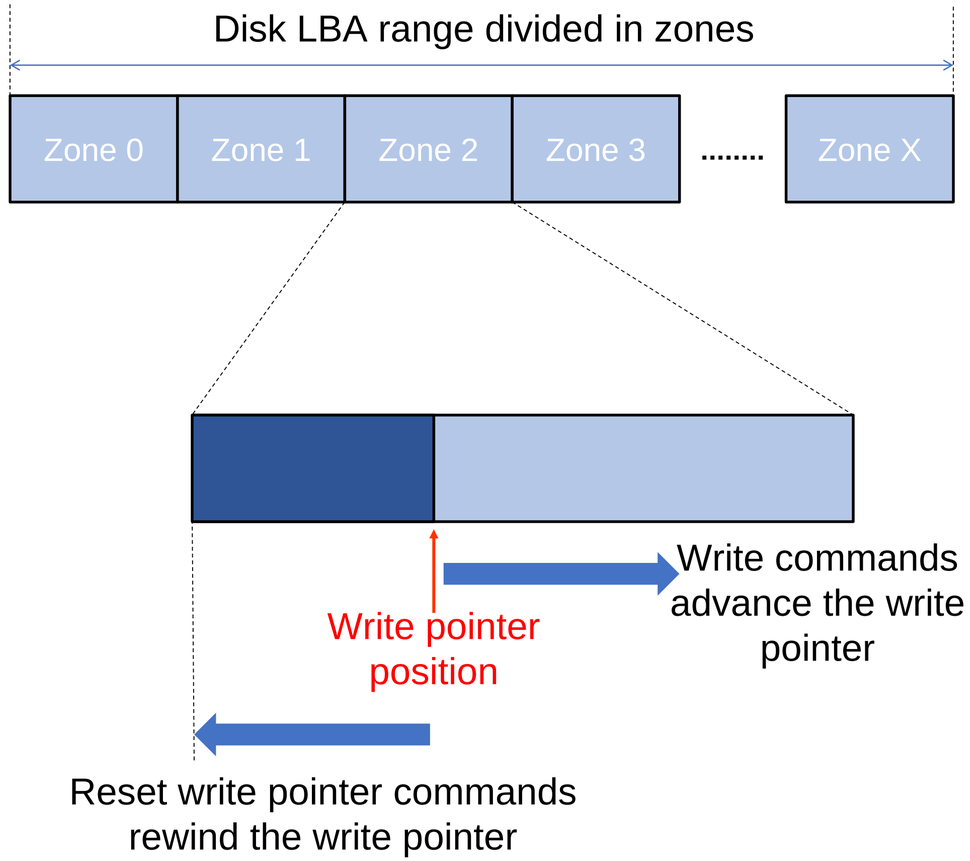
- Reduce device-side write amplification
- Reduce over-provisioning
  - *"Note that excessive over-provisioning is similar to early replacement -- in both cases you buy more devices."*
    - Mark Callaghan, Facebook
- Reduce DRAM in SSDs
  - Highest cost after NAND itself
- Improve latency outliers and throughput
  - Reduces device-side data movement
  - The tail at scale
- Enable software eco-system.
  - Everyone benefits from software improvements!



Zoned Namespaces similar to ZBC/ZAC for SMR HDDs

- Storage capacity is divided into zones
- Each zone is written sequentially
- Interface optimized for SSDs
  - Align with media characteristics
    - Zone size aligned to media (E.g., NAND block sizes)
    - Zone capacity aligned to physical media sizes
  - Reduce NAND media erase cycles (Write amp.)





Zone Information

Zone State

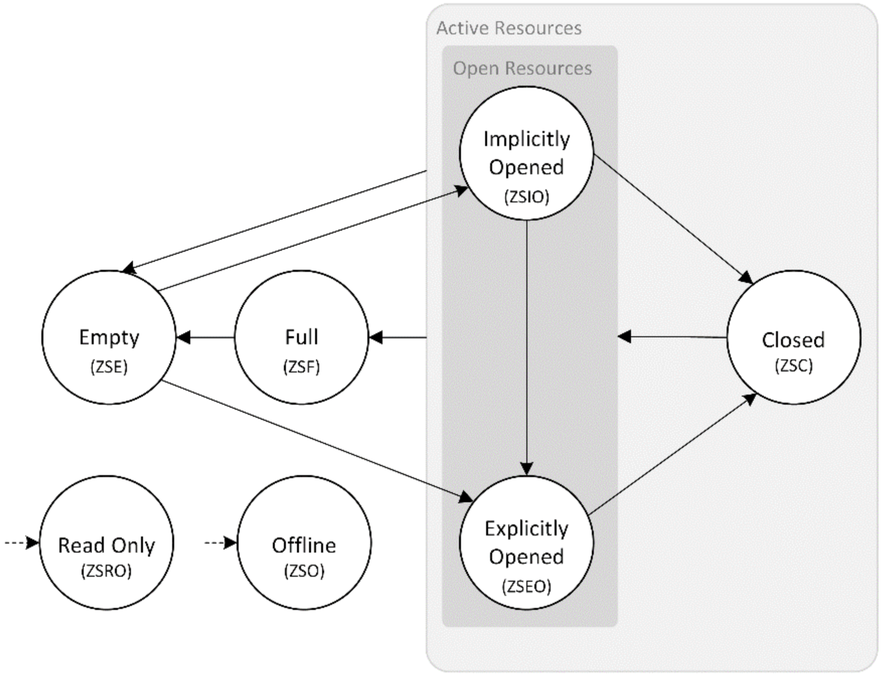
- Empty, Implicitly Opened, Explicitly Opened, Closed, Full, Read Only, Offline
- Empty -> Open -> Full -> Empty -> ….

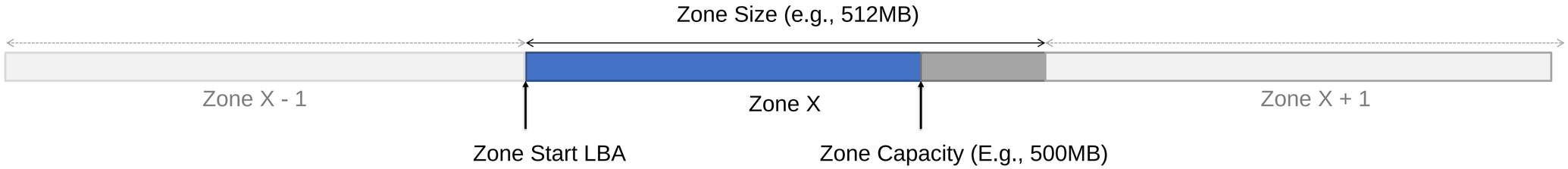
Zone Reset

- Full -> Empty

Zone Size & Zone Capacity

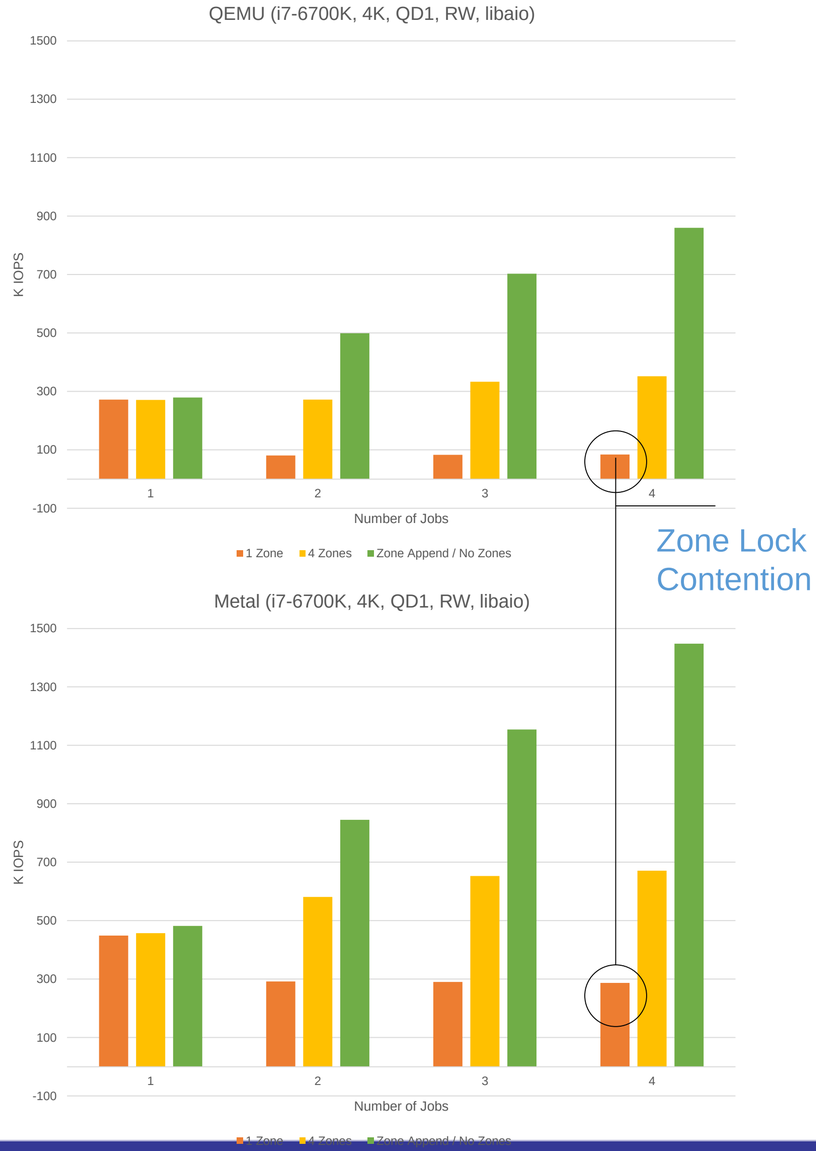
- Zone Size is fixed
- Zone Capacity is the writeable area within a zone





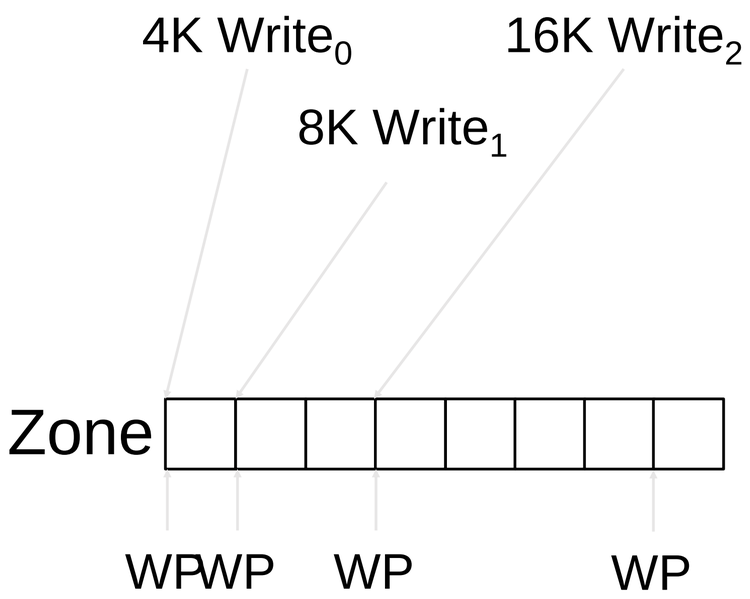
Zone Append

- Low scalability on multiple writers to a zone
  - Write Queue Depth per Zone = 1
  - IOPS: *80K vs 880K using Qemu* and *300K vs 1400K on bare metal*
- ZAC/ZBC requires strict write ordering
- Limits write performance and increases host overhead
- Big challenge with software eco-system, HBAs, etc.
- Introducing Zone Append
- Append data to a zone without defining offset
  - Drive returns where data was written in the zone



Zone Write Example

3x Writes (4K, 8K, 16K) – Queue Depth = 1



Zone Queue Depth = 1
Only one Write outstanding per zone

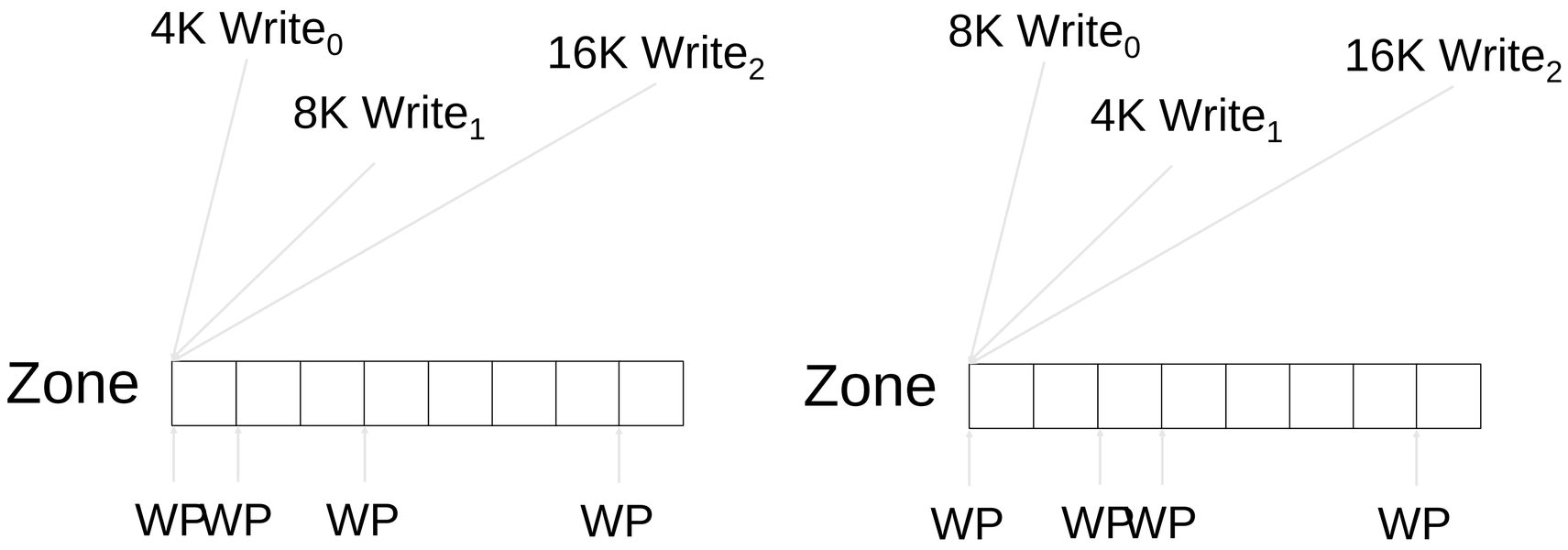
Host takes on the overhead of serializing I/Os.

Insignificant when using HDDs

Significant when using SSDs

Zone Append Example

3x Writes (4K, 8K, 16K) – Queue Depth = 3



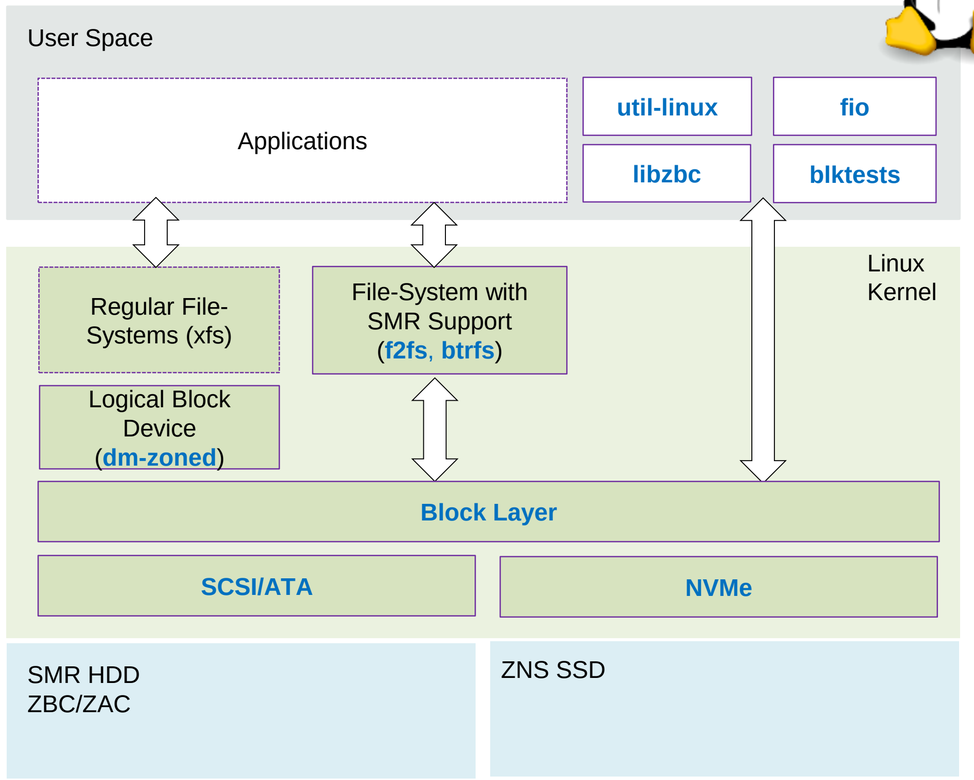
Drives takes on the responsibility of serializing I/Os.

Scalable for both HDDs and SSDs.

Zone Queue Depth >= 1
Multiple writes outstanding per zone

ZNS: Synergy w/ ZAC/ZBC software ecosystem

- ZAC/ZBC is the interface for SMR hard-drives
- Reuse existing work already applied for ZAC/ZBC hard-drives
- Existing ZAC/ZBC-aware file systems & device mappers “just work”
  - Few changes to support to ZNS
- Integrate directly with file-systems or applications
  - No host-side FTL
  - No 1GB DRAM per 1TB Media requirement
- Code for ZAC/ZBC already in production at technology adopters and broadly available in the Linux® eco-system.



*= Enhanced data paths for SMR drives

ZNS Support in Linux

Shows up as a host-managed Zoned Block Device

```
zns@zns-2:~$ cat /sys/block/nvme0n1/queue/zoned
host-managed
```

```
zns@zns-2:~$ cat /sys/block/nvme0n1/queue/chunk_sectors
2097152
```

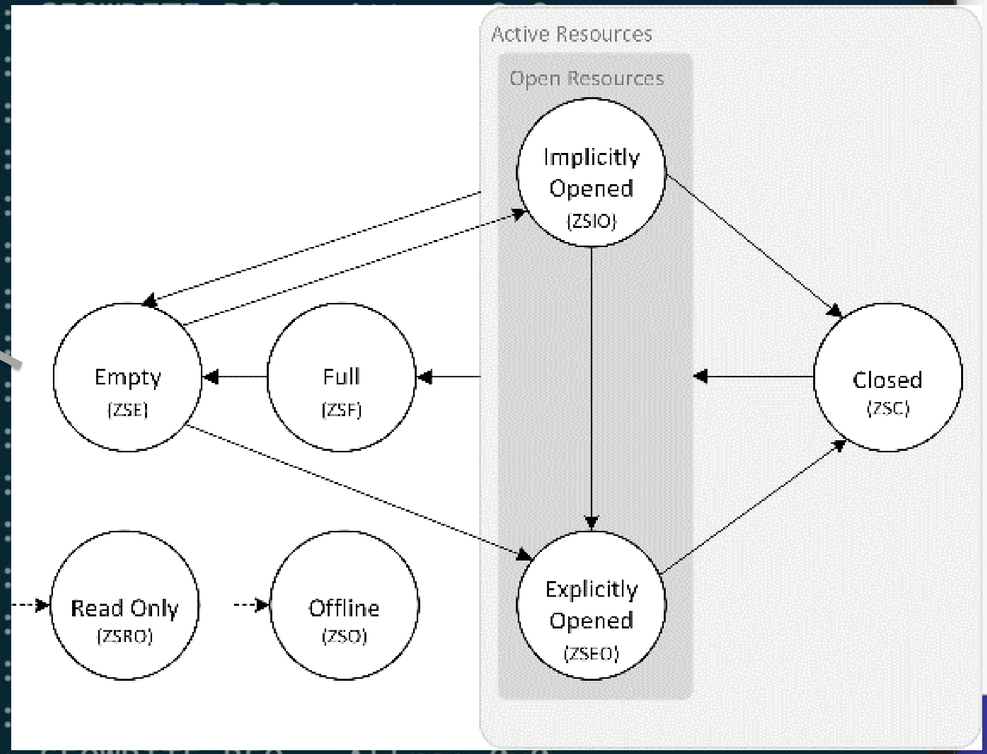
Zone Size = 1GB
0x200000/2097152 (512B Logical block size)

Zone Information

List of zones including metadata

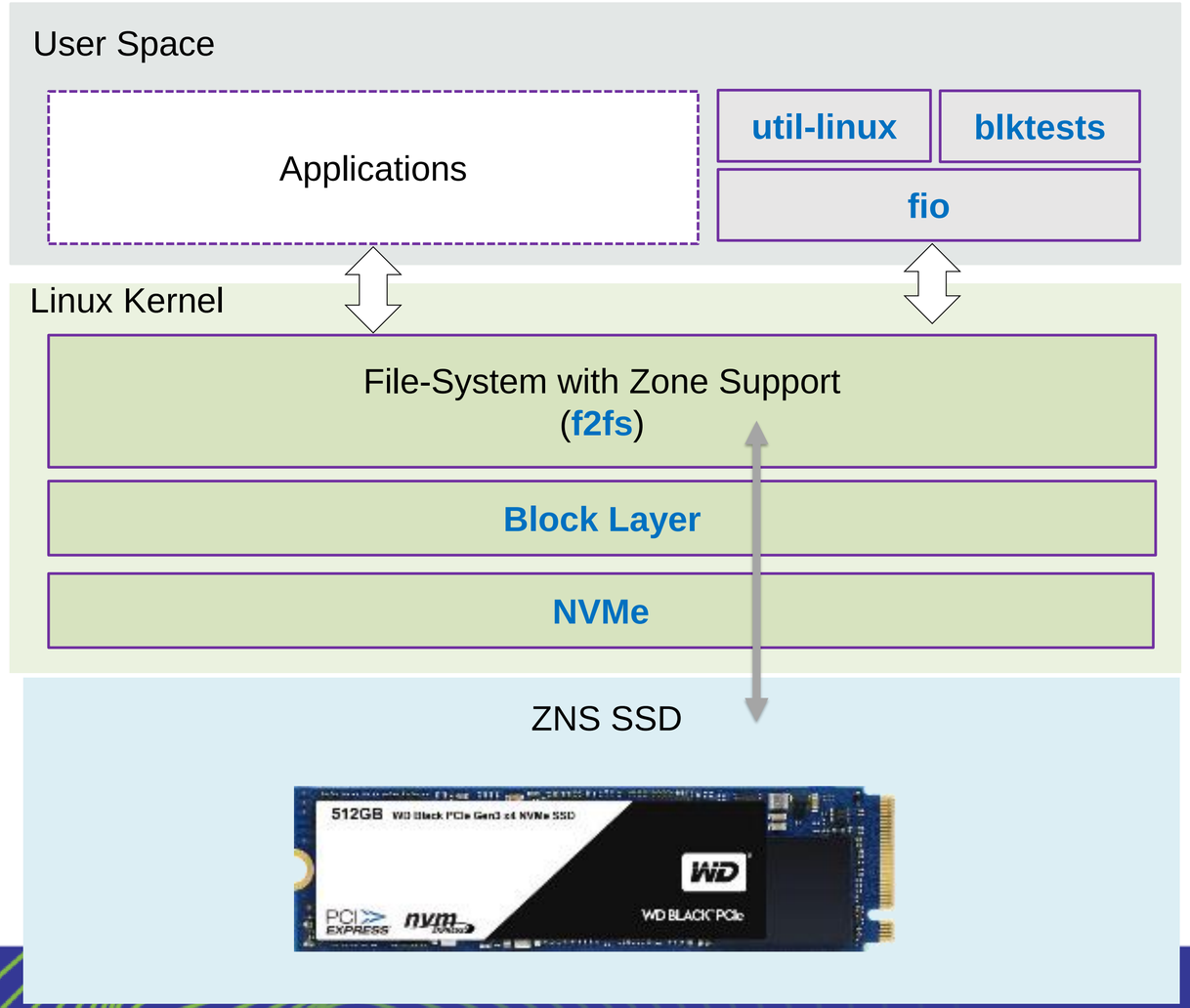
3. zns@zns-2: ~/zns-demo/nvme-cli (ssh)

```
zns@zns-2:~/zns-demo/nvme-cli$ sudo ./nvme zone-log /dev/nvme0n1 -l 4096 -o 0 -H
SLBA: 0x0        WP: 0x0        Cap: 0x200000   State: EMPTY        Type: SEQWRITE_REQ   Attrs: 0x0
SLBA: 0x200000   WP: 0x200000   Cap: 0x200000   State: EMPTY        Type: SEQWRITE_REQ   Attrs: 0x0
SLBA: 0x400000   WP: 0x400000   Cap: 0x200000   State: EMPTY        Type: SEQWRITE_REQ   Attrs: 0x0
SLBA: 0x600000   WP: 0x600000   Cap: 0x200000   State: EMPTY        Type: SEQWRITE_REQ   Attrs: 0x0
SLBA: 0x800000   WP: 0x800000   Cap: 0x200000   State: EMPTY        Type:
SLBA: 0xa00000   WP: 0xa00000   Cap: 0x200000   State: EMPTY        Type:
SLBA: 0xc00000   WP: 0xc00000   Cap: 0x200000   State: EMPTY        Type:
SLBA: 0xe00000   WP: 0xe00000   Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x1000000  WP: 0x1000000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x1200000  WP: 0x1200000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x1400000  WP: 0x1400000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x1600000  WP: 0x1600000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x1800000  WP: 0x1800000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x1a00000  WP: 0x1a00000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x1c00000  WP: 0x1c00000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x1e00000  WP: 0x1e00000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x2000000  WP: 0x2000000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x2200000  WP: 0x2200000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x2400000  WP: 0x2400000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x2600000  WP: 0x2600000  Cap: 0x200000   State: EMPTY        Type:
SLBA: 0x2800000  WP: 0x2800000  Cap: 0x200000   State: EMPTY        Type: SEQWRITE_REQ   Attrs: 0x0
SLBA: 0x2a00000  WP: 0x2a00000  Cap: 0x200000   State: EMPTY        Type: SEQWRITE_REQ   Attrs: 0x0
SLBA: 0x2c00000  WP: 0x2c00000  Cap: 0x200000   State: EMPTY        Type: SEQWRITE_REQ   Attrs: 0x0
```



File-system Integration

The File-System is the "FTL" – Manages mapping table, OP, and GC strategy.



Format f2fs file-system

```
root@zns-2:~/zns# ./create_f2fs.sh

        F2FS-tools: mkfs.f2fs Ver: 1.10.0 (2018-01-30)

Info: Disable heap-based policy
Info: Debug level = 0
Info: Label =
Info: Trim is enabled
Info: Host-managed zoned block device:
      2080 zones, 30 randomly writeable zones
      4096 blocks per zone
Info: Segments per section = 8
Info: Sections per zone = 1
Info: sector size = 512
Info: total sectors = 68157440 (33280 MB)
Info: zone aligned segment0 blkaddr: 4096
Info: format version with
  "Linux version 5.0.0-rc4-custom+ (parallels@ninja) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04
Info: [/dev/dm-0] Discarding device
Info: Discarded 33280 MB
Info: Overprovision ratio = 3.120%
Info: Overprovision segments = 1073 (GC reserved = 576)
Info: format successful
```

Read and Write with from f2fs with an ZNS drive