Western Digital

Introduction to Open-Channel Solid State Drives

Matias Bjørling
Director, Solid-State System Software

August 7th, 2018

Flash Memory Summit

Forward-Looking Statements

Safe Harbor | Disclaimers

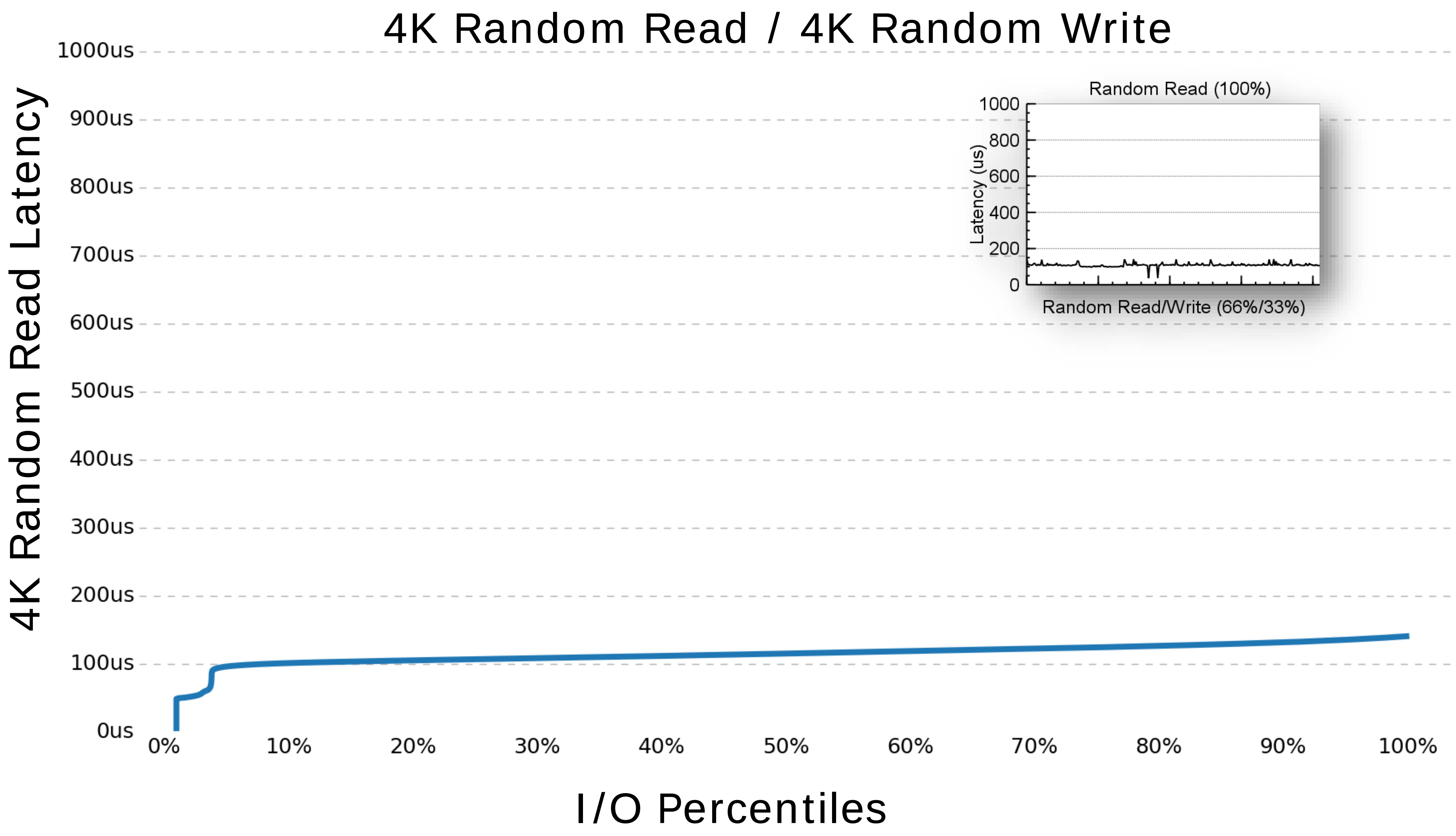
This presentation contains forward-looking statements that involve risks and uncertainties, including, but not limited to, statements regarding our solid-state technologies, product development efforts, software development and potential contributions, growth opportunities, and demand and market trends. 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.

Agenda

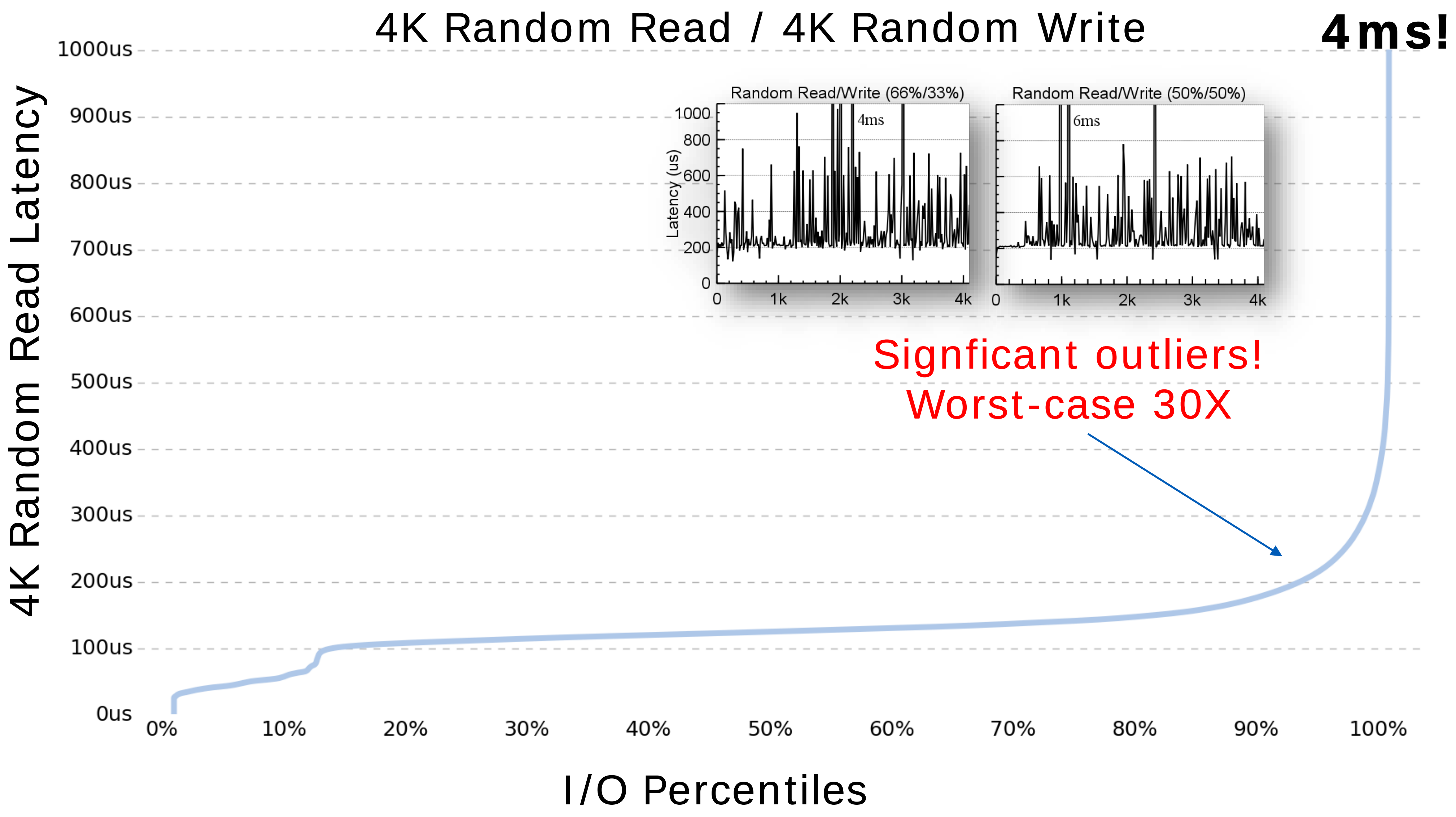
1. Motivation
2. Interface
3. Eco-system
4. Open-Source Contributions

0% Writes - Read Latency



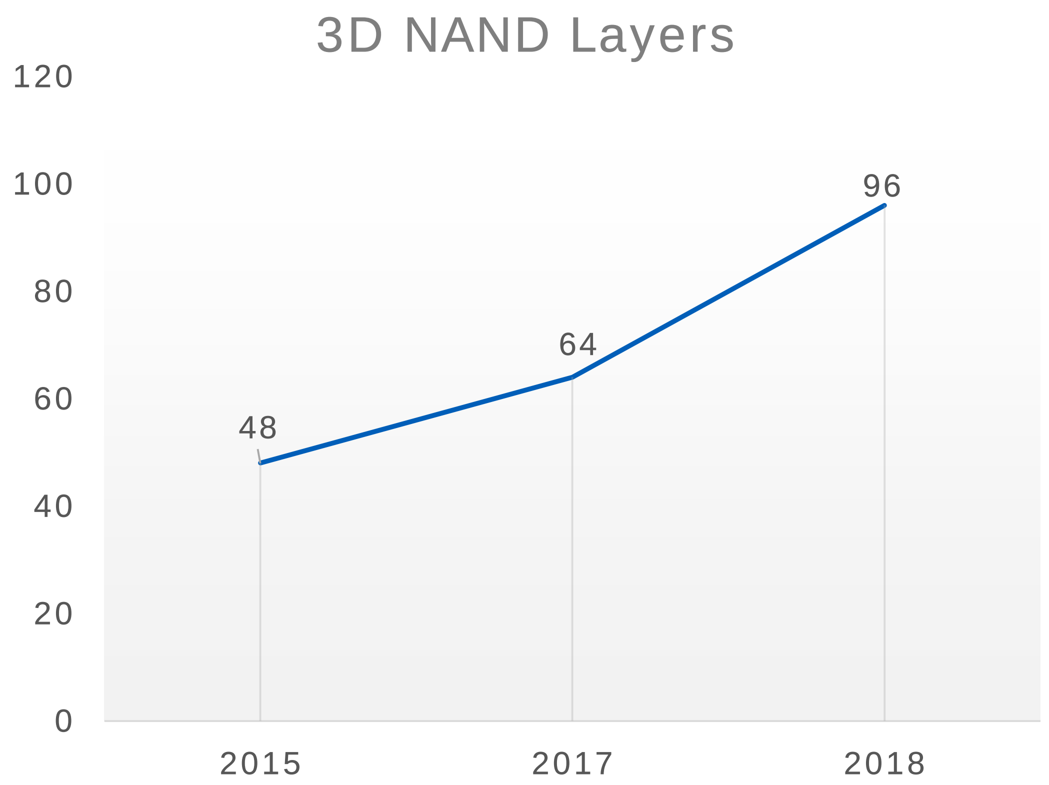
Western Digital.

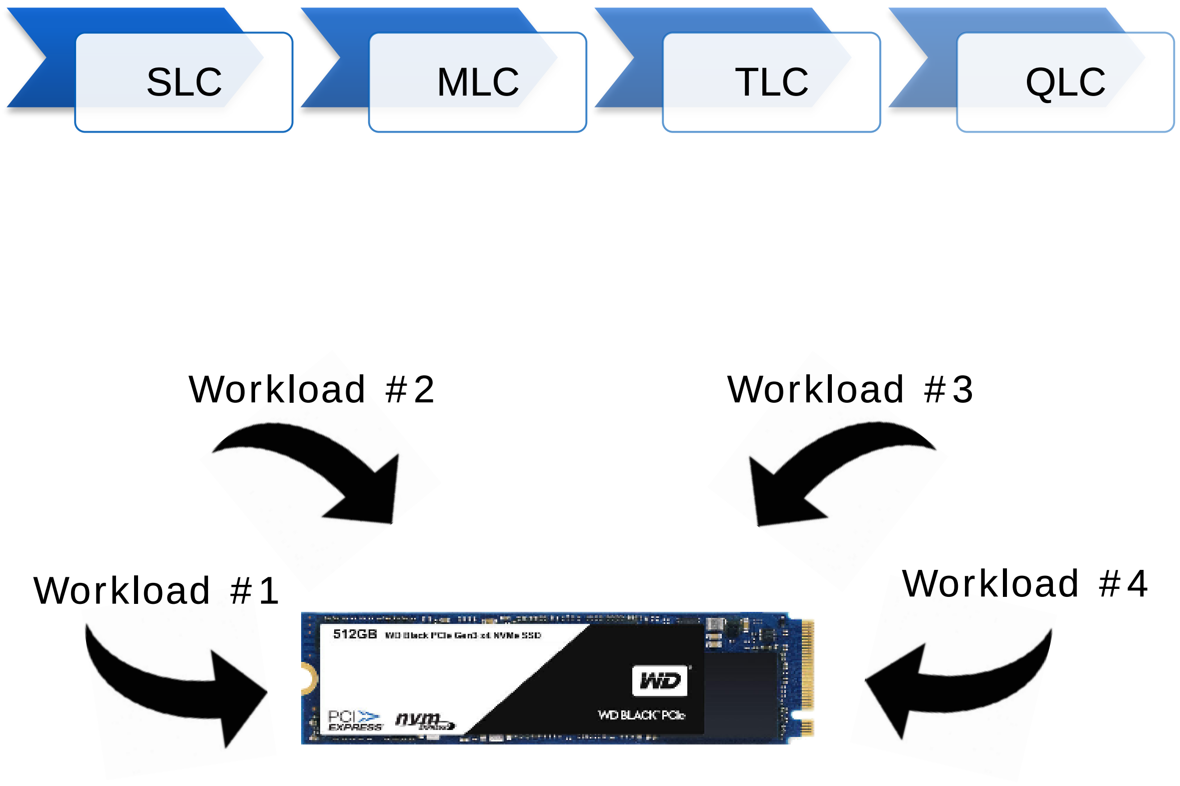
20% Writes - Read Latency



NAND Chip Density Continues to Grow

While Cost/GB decreases

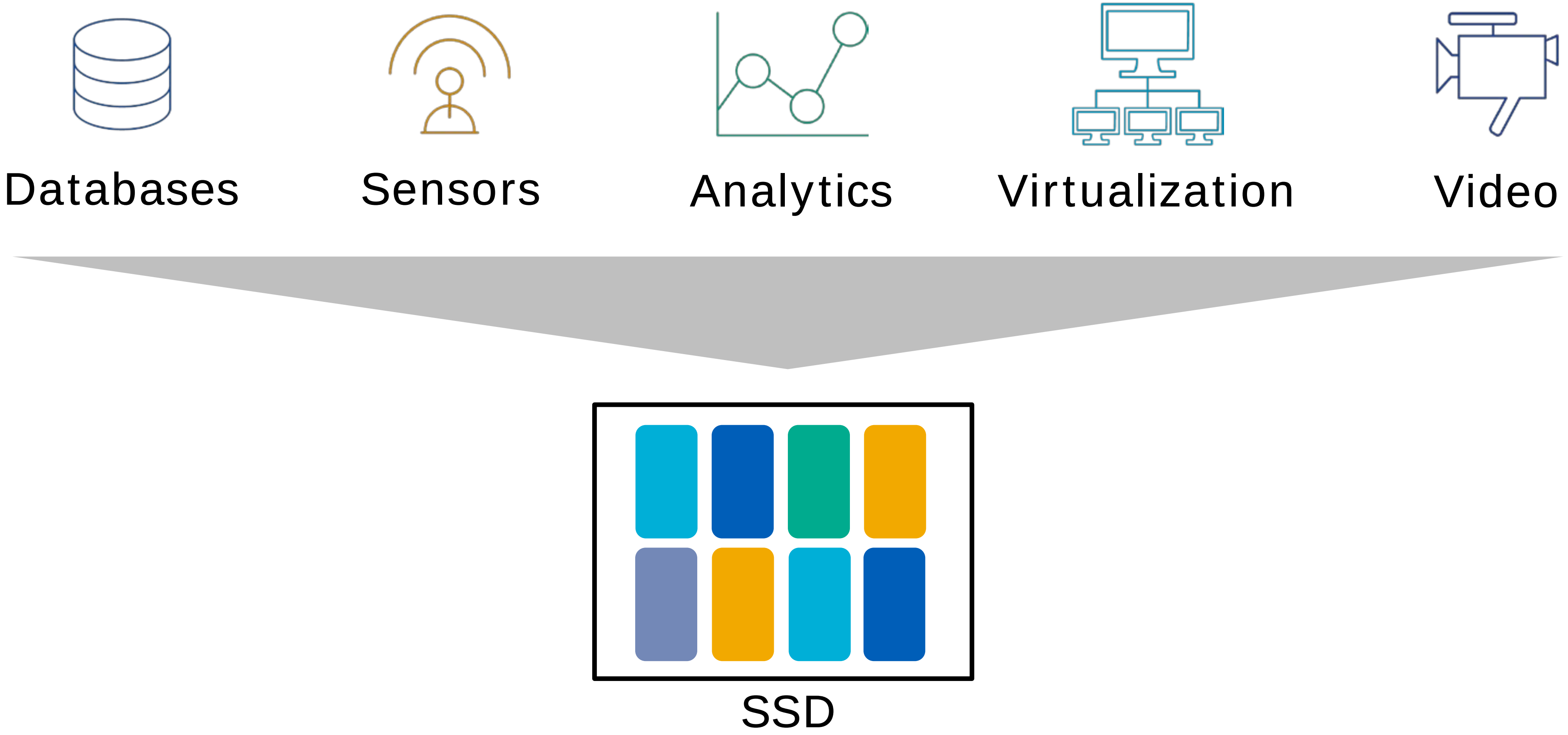




Flash Memory Summit 2018, Santa Clara, CA

Ubiquitous Workloads

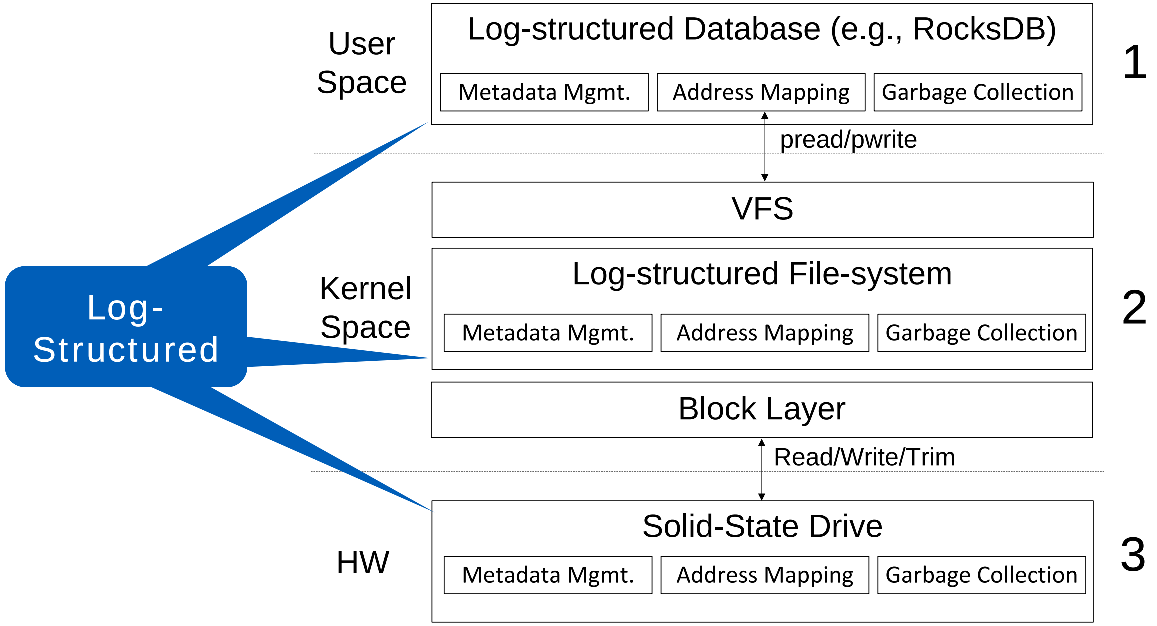
Efficiency of the Cloud requires many different workloads of a single SSD



Single-User Workloads

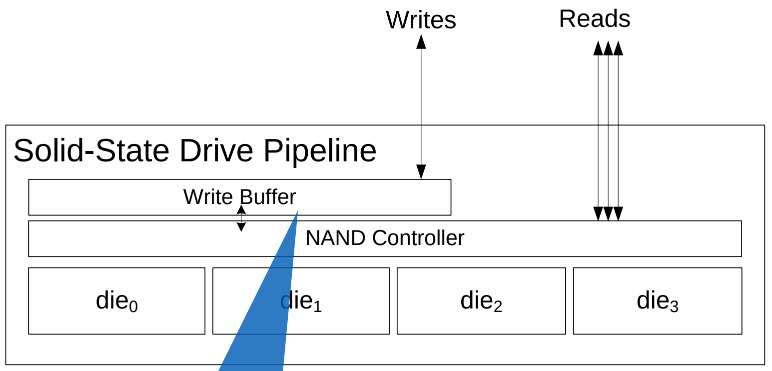
Indirection *and* ***Narrow Storage*** *interface is a main cause of outliers*

Host: Log-on-Log



Unable to align data logically = Write amplification increase + extra GC

Device: Write Buffering

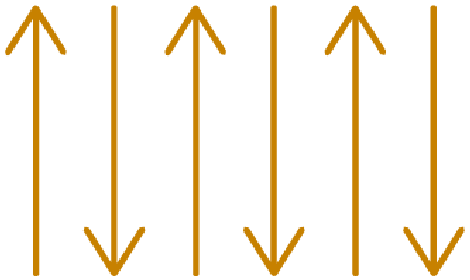


Buffered Writes

Drive maps logical data to the physical location with **Best Effort**

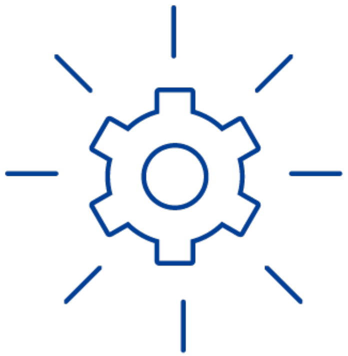
Host is oblivious to physical data placement due to indirection

Open-Channel SSDs



I/O Isolation

Predictable Latency



Data Placement &
I/O Scheduling

Concepts in an Open-Channel SSD

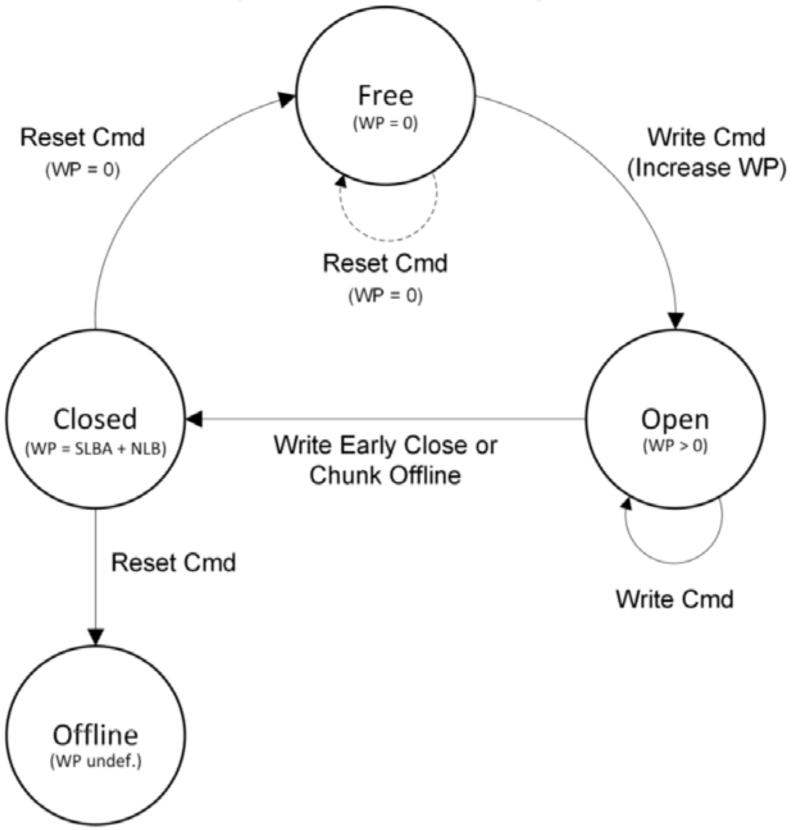
Interface Blocks

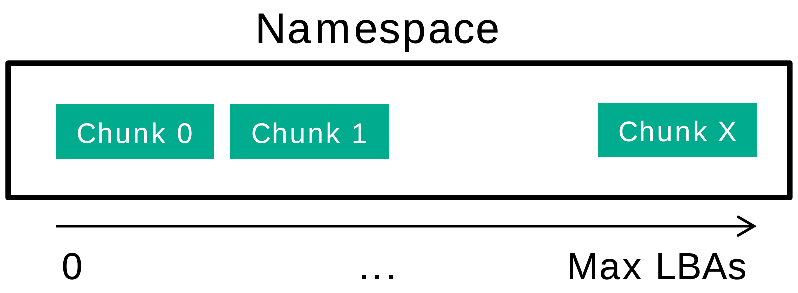
- Chunks
  - Sequential write only LBA ranges
- Hierarchical addressing
  - A sparse addressing scheme projected onto the NVMe™ LBA address space
- Direct-to-media min/opt write size
  - Eliminate write buffering
- Host-assisted Media Refresh
  - Improve I/O predictability
- Host-assisted Wear-leveling
  - Improve wear-leveling

Chunks

Enable orders of magnitude reduction of device-side DRAM

- A chunk is a range of LBAs where writes are required to be sequential.
- Reduces DRAM for L2P table by orders of magnitude
- Hot/Cold data separation
- Rewrite requires a reset
  - A chunk can be in one of four states (free/open/closed/offline)
  - If a chunk is open, there is a write pointer associated.
- Follows the same model as in the ZAC/ZBC standards.



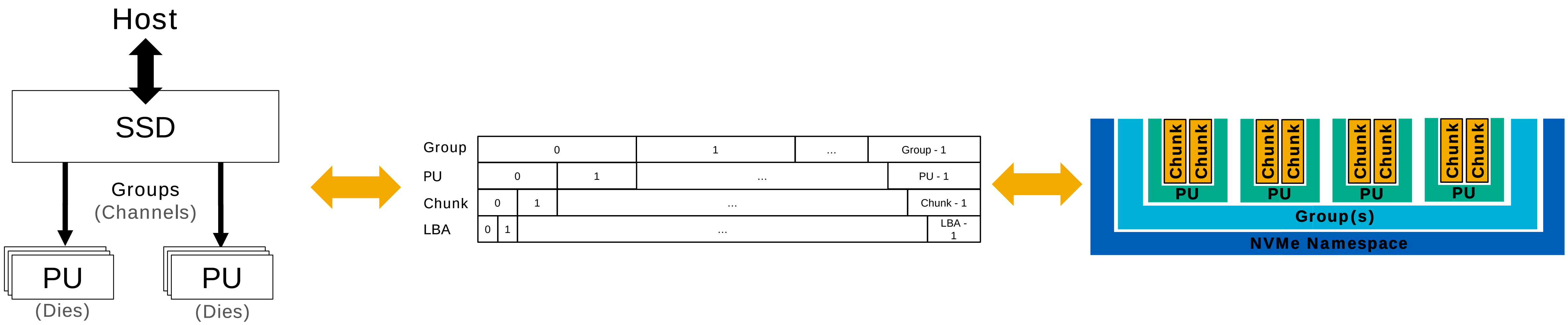


Hierarchical Addressing

Channels and Dies are mapped to Logical Groups and Parallel Units

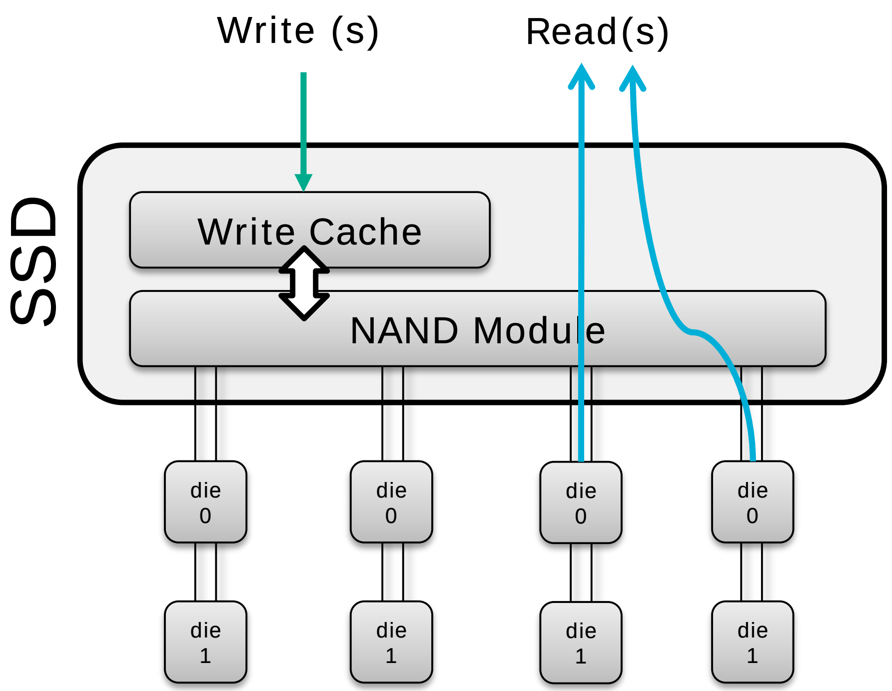
- Expose device parallelism through Groups/Parallel Units
  - One or a group of dies are exposed as parallel units to the host
  - Parallel units are a logical representation

Physical | **LBA Address Space** | **Logical**

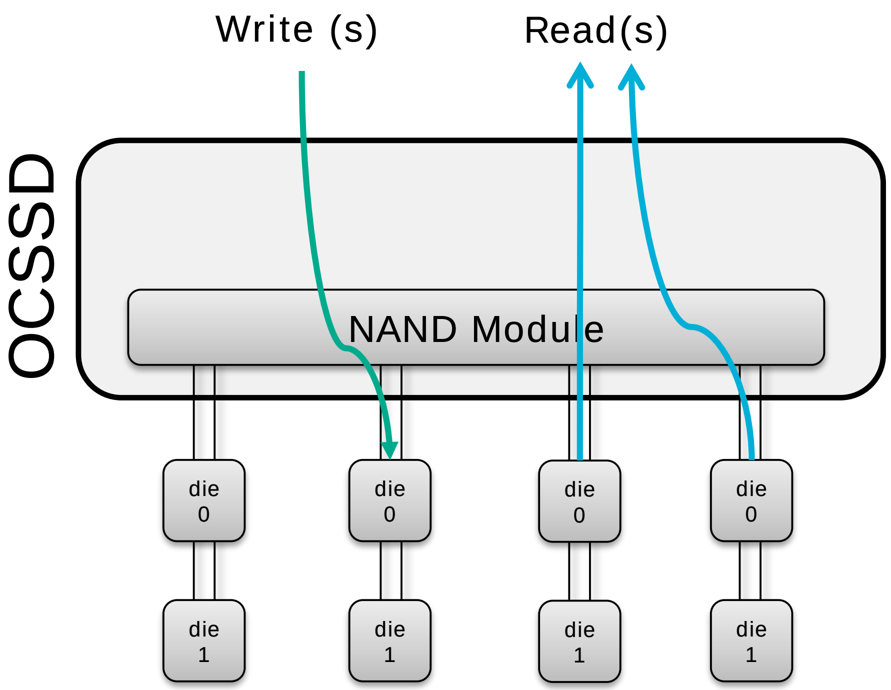


Direct-to-Media

Bypass device write cache and move data directly to NAND



Add unpredictability. When to flush?
Limited streams due to DRAM/SRAM requirements

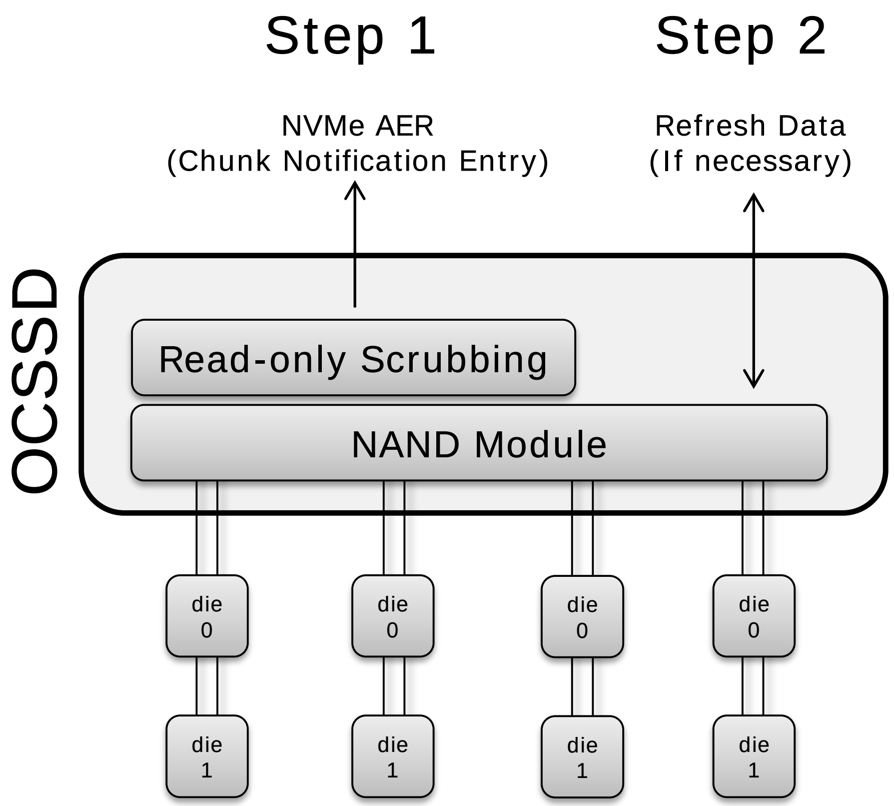


Synchrounous Write
Large Writes (Flash page size)
Large number of streams / open chunks

Host-assisted Media Refresh

Enable host to assist SSD data refresh

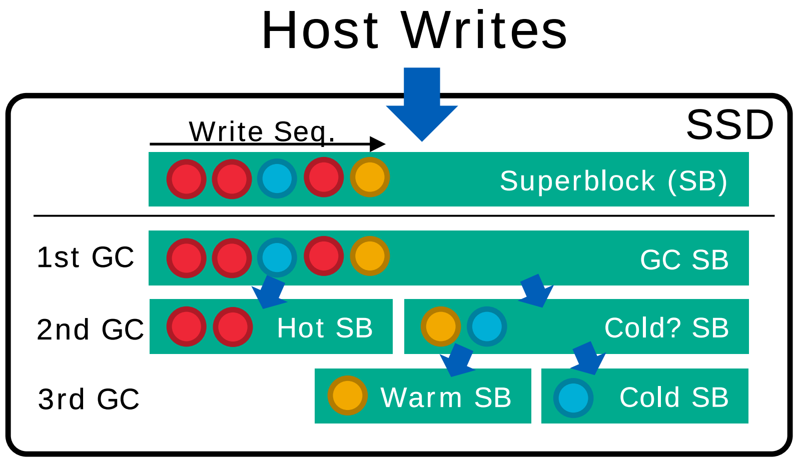
- SSDs refreshes its data periodically to maintain reliability. It does this through a data scrubbing process
  - Internal read and writes make the drive I/O latencies unpredictable.
  - Writes dominates I/O outliers
- 2-step Data Refresh
  - Device to only perform the data scrubbing read part - Data movement is managed by host
  - Increases predictability of the drive. Host manages refresh strategy
    - Should it refresh? Is there a copy elsewhere?

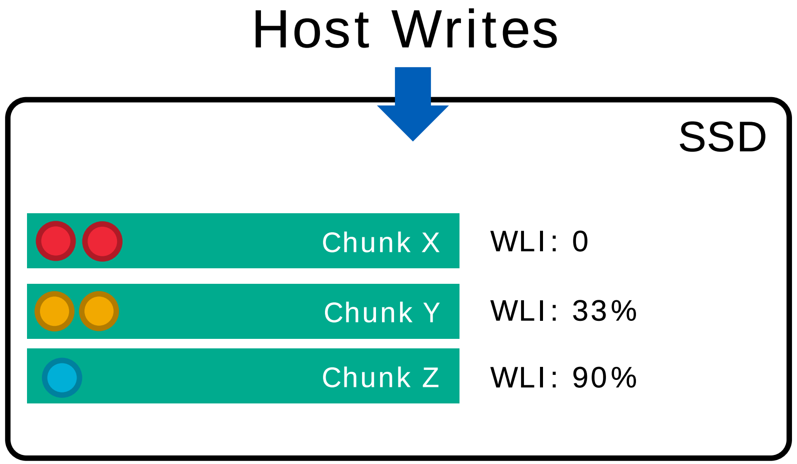


Host-assisted Wear-Leveling

Enable host to separate Hot/Cold data to Chunks depending on wear

- SSDs typically does not know the temperature of newly written data
  - Placing hot and cold data together increases write amplication
  - Write amplication is typical 4-5X for SSDs
- Chunk characteristics
  - Limited reset cycles (as NAND blocks has limited erase cycles)
  - Place cold data on chunks that are nearer end-of-life and use younger chunks for hot data
- Approach
  - Introduce per-chunk relative wear-level indicator (WLI)
  - Host knows workload and places data w.r.t. to WLI
  - No need to needlessly garbage collect chunks → **Increases lifetime, I/O latency, and performance**

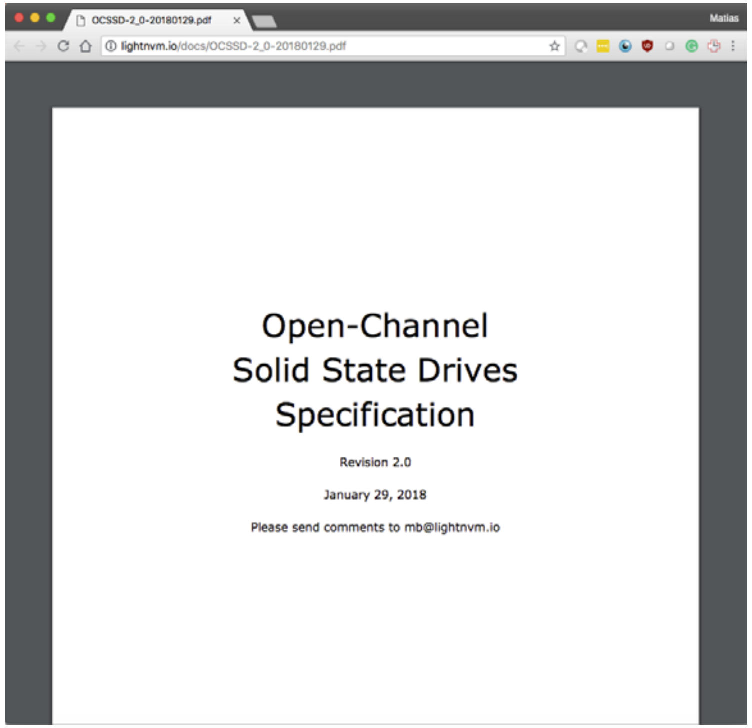




Interface Summary

The concepts together provide

- **I/O Isolation** through the use of Groups & Parallel Units
- **Fine-grained data refresh** managed by the host
- **Reduce write amplification** by enabling host to place hot/cold data efficiently
- **DRAM & Over-provisioning** reduction through append-only Chunks
- Direct-to-media to **avoid expensive internal data movement**

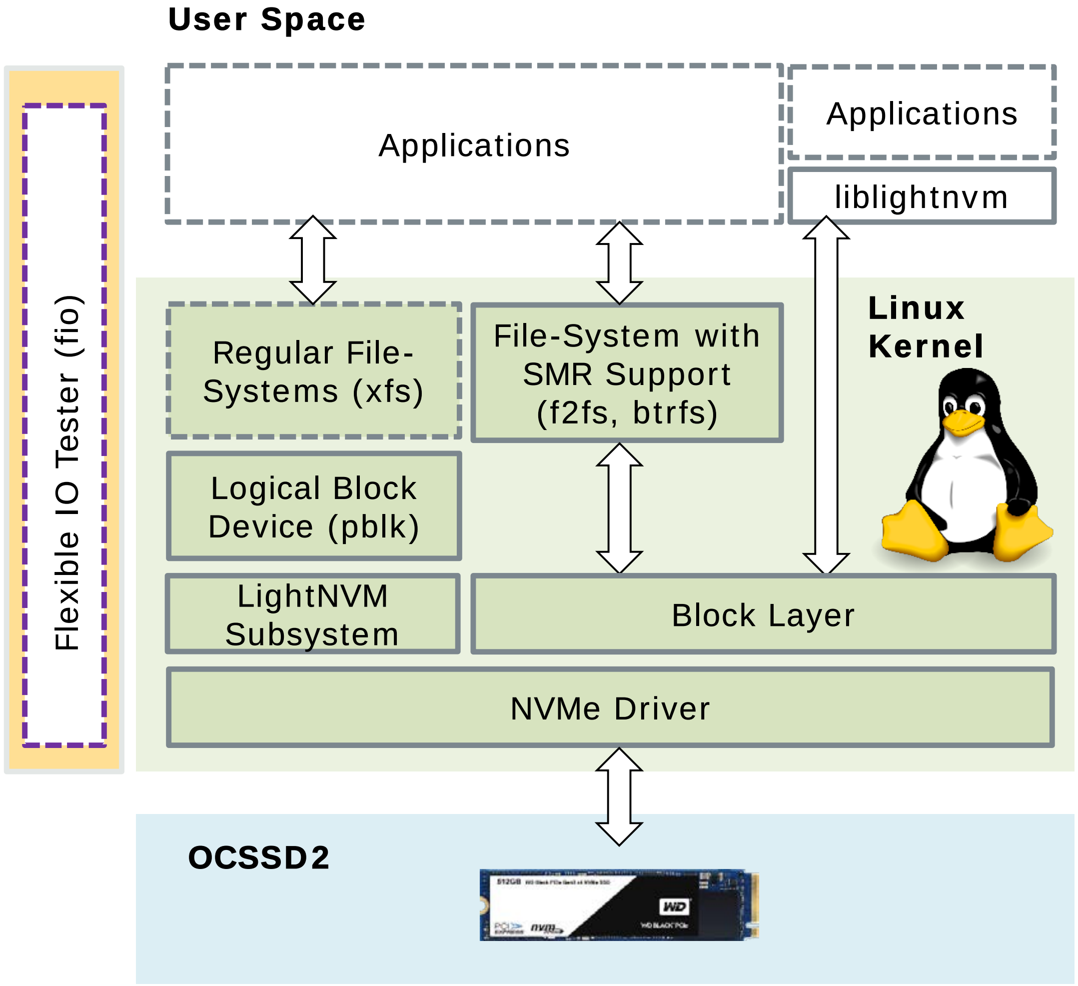


Specification available at http://lightnvm.io

Eco-system

Large eco-system through Zoned Block Devices and OCSSD

- Linux Kernel®
  - NVMe Device Driver
    - Detection of OCSSDs
    - Implements 1.2 and 2.0 specification
    - Registers as a ZBD and with LightNVM
  - LightNVM Subsystem
    - Provides core functionality
    - Target management
  - Target interface
    - Enumerate, get geometry, I/O interface, etc.
    - pblk host-side FTL – Map OCSSD to Block Device
- User-space
  - Libzbc, fio (ZBD support), liblightnvm
  - SPDK



Open-Source Software Contributions

- Initial release of subsystem with Linux kernel 4.4 (January 2016).
- User-space library (liblightnvm) support upstream in Linux kernel 4.11 (April 2017).
- pblk available in Linux kernel 4.12 (July 2017).
- Open-Channel SSD 2.0 specification released (January 2018) and support available from Linux kernel 4.17 (May 2018).
- SPDK Support for OCSSD (June 2018)
- Upcoming
  - OCSSD as Zoned Block Device (Patches available)
  - Fio with Zone support (Patches available)
  - 2.0a revision with fixes

Path to Standardization

The OCSSD interface combines several concepts

- Groups / Parallel Units = NVM Sets / Endurance Groups
- Direct-to-Media - Proposal in NVMe WG
- Chunks - Not yet defined
- Host-assisted Data Refresh
  - I/O Determinism: Deterministic Windows
  - **AER Feedback mechanism** – Fine-grained Host controlled Data Refresh
- Host-assisted WL
  - Endurance Groups with extended attributes
  - **Finer granularity through Chunks.** Improve hot/cold data placement → Dependent on the concept of chunks

OCSSD 2.0	NVMe
Parallelism Groups/PUs	NVM Sets + Endurance Groups
Direct-to-Media	In progress
Chunks	
Host-assisted Data Refresh	Deterministic Windows
Host-assisted WL	Endurance Groups

T

LightNVM: The Linux Open-Channel SSD Subsystem

https://www.usenix.org/conference/fast17/technical-sessions/presentation/bjorling

LightNVM

http://lightnvm.io

LightNVM Linux kernel Subsystem

https://github.com/OpenChannelSSD/linux

liblightnvm

https://github.com/OpenChannelSSD/liblightnvm

QEMU NVMe with Open-Channel SSD Support

https://github.com/OpenChannelSSD/qemu-nvme

Western Digital®