SDC 20

BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

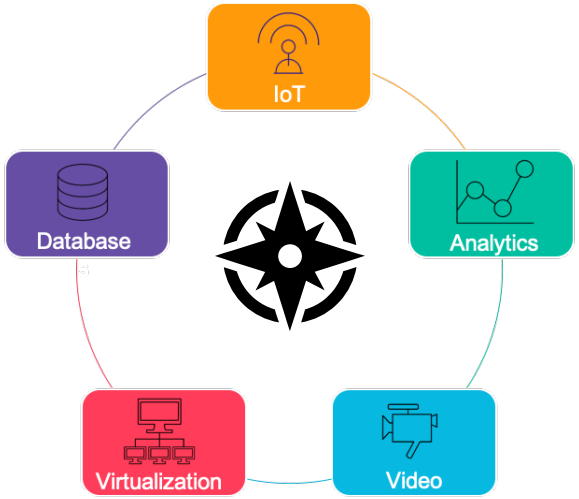
ZenFS, Zones and RocksDB

Who likes to take out the garbage anyway?

Hans Holmberg, Technologist
Western Digital

Data Growth

- IDC[1] expects that **103 zettabytes** of data will be generated worldwide by 2024



- **How do we scale?**

1) IDC Worldwide Global DataSphere IoT Device and Data Forecast, 2019–2023

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

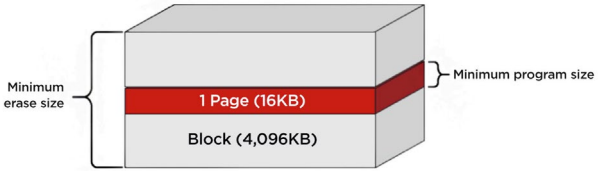
Solid State Drive (SSD)

- An SSD bundles 10-100s NAND chips and an SSD controller together
- The SSD controller manages NAND chips characteristics and expose the storage through a storage interface
- A NAND chip is composed of erase blocks, consisting of many pages
  - Within each erase block, you can **only write sequentially**
  - Erase block **must be erased** before new writes
  - **Limited number of erases** of an erase block

SSD controller and
10-100x NAND chips

Massive Parallelism

NAND chip layout

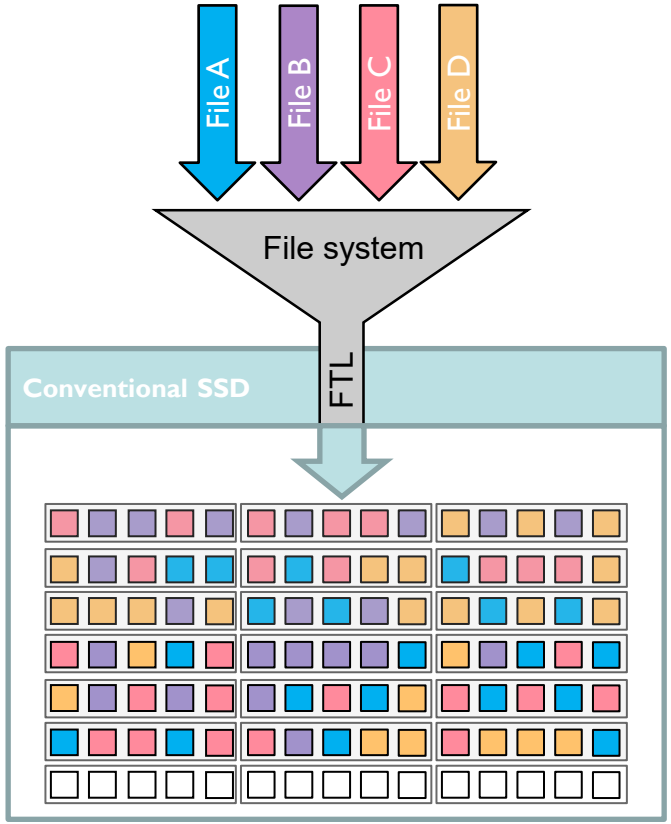


Performs a single operation (Read/Program/Erase) at a time

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Conventional SSDs

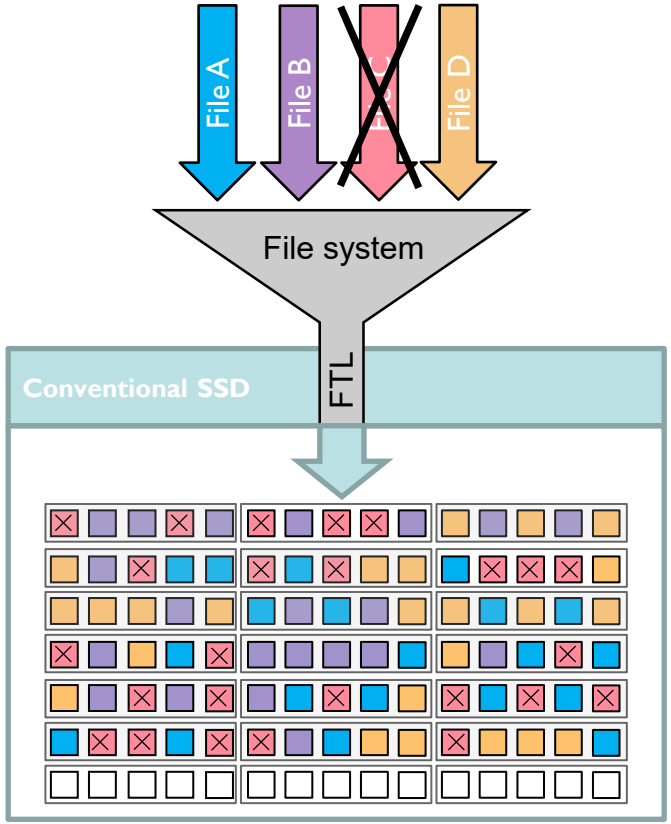
- A Flash Translation Layer (FTL) maps logical blocks to physical addresses
- Files can't be separated by the drive
- Different types of data gets co-located in the same erase units



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Space reclaim

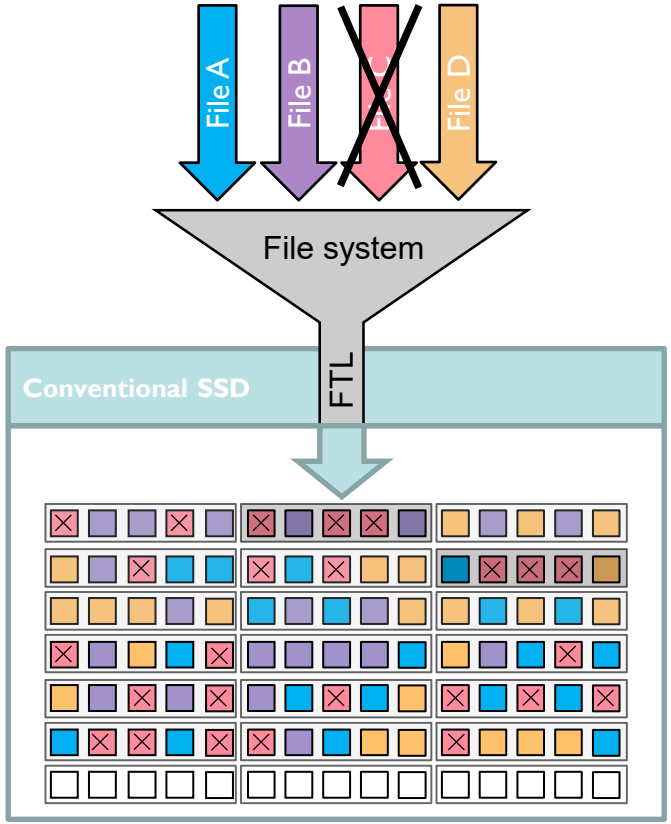
- When a file gets deleted the space occupied can't be reused without erasing an entire erase unit
- Garbage collection is needed to evacuate still-valid data from the erase unit



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Space reclaim

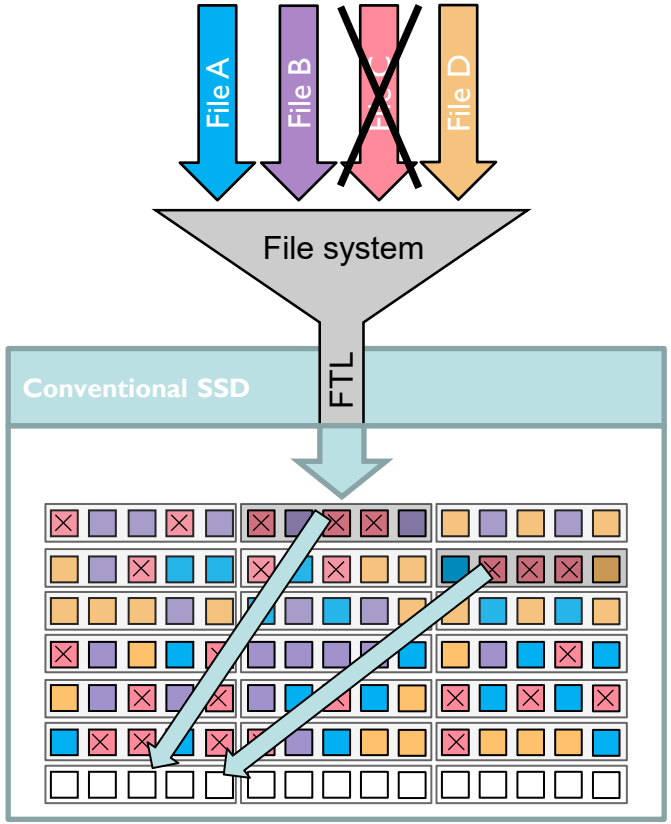
- The controller picks erase units to be evacuated



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Space reclaim

- The controller picks erase units to be evacuated
- Still-valid data is evacuated to another erase unit



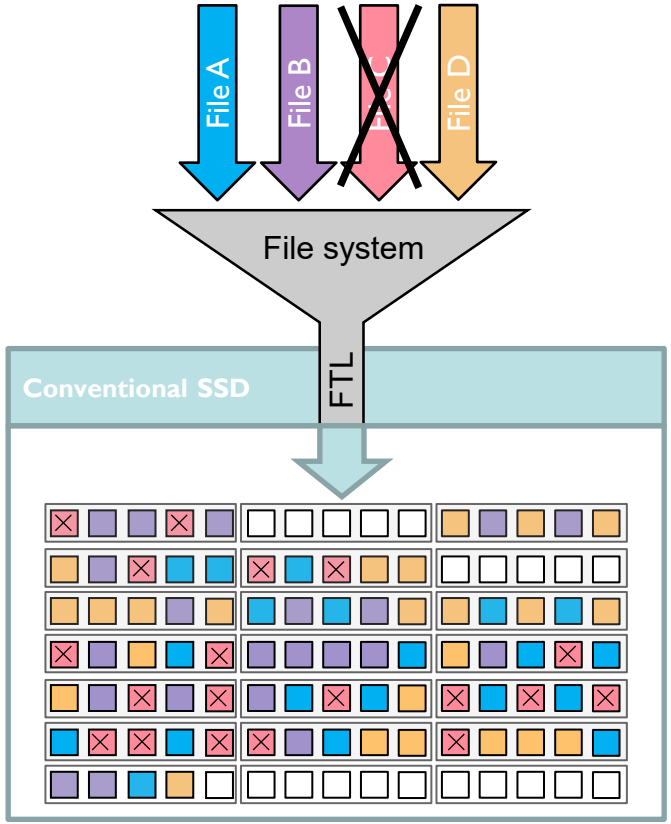
2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Space reclaim

- Once an erase unit is evacuated it can be erased and reused
- Garbage collection causes rewrites of user data

Write Amplification

$$WA = \frac{drive\ blocks\ written}{host\ blocks\ written}$$



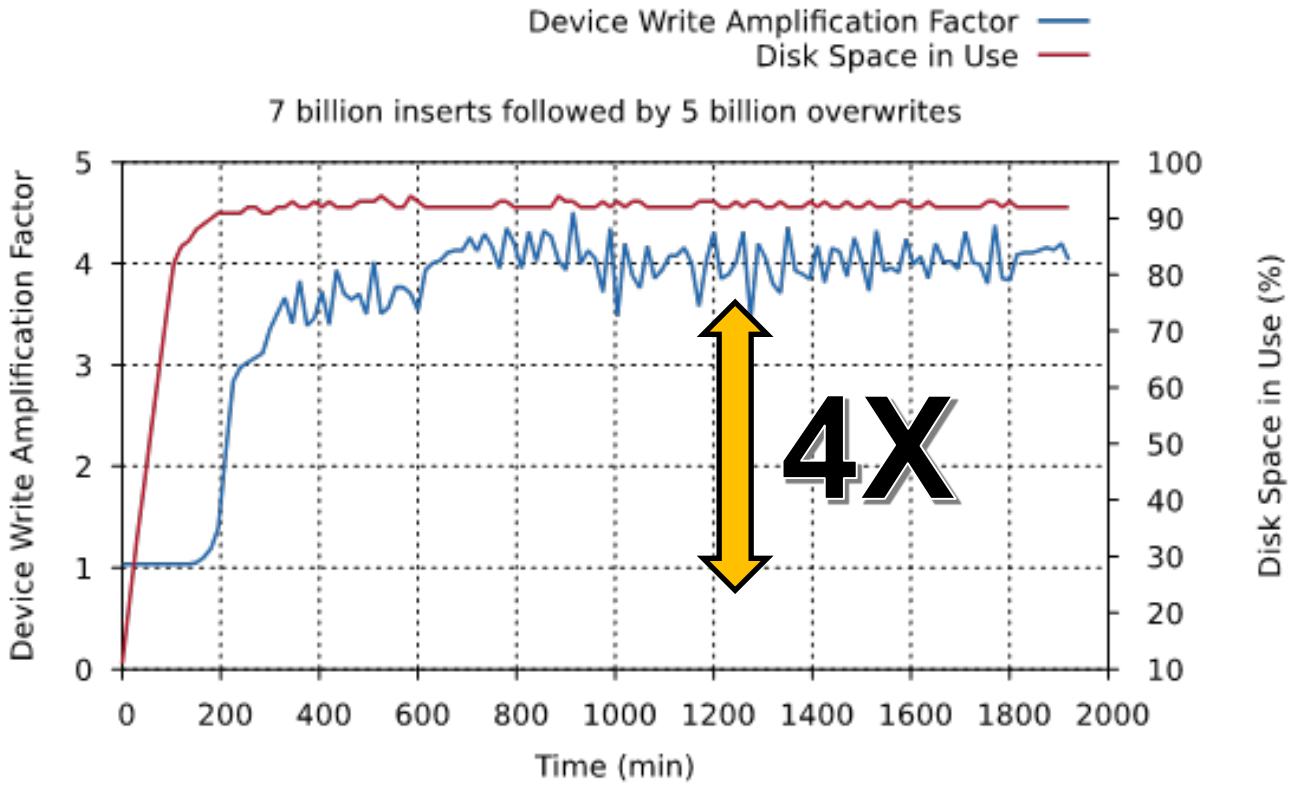
2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Write Amplification

- More drive writes per user write
  - ⇨ Shorter life span of the drive
  - ⇨ Decreased write performance
  - ⇨ Worse read tail latencies

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

RocksDB WA on a conventional SSD



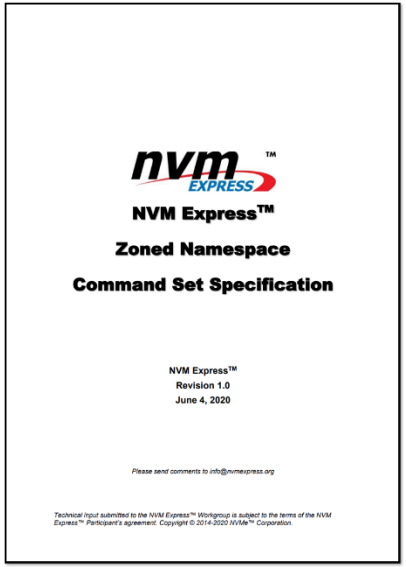
2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Who likes to take out the garbage anyway?

Can’t we do better?

NVMe™ Zoned Namespace Command Set

- Introduces the Zoned Storage Model for NVMe™
- Introduces a new namespace type (Zoned Namespaces)
  - Exposes a set of zones of fixed size to be written sequentially and reset for new writes (matches the NAND media characteristics)
  - Implements the Zoned Namespaces Command Set
  - The command set inherits the NVM Command Set
    - i.e., Read/Write/Flush commands are available.
- Optimized for SSDs
  - Adds a more efficient interface for flash media
  - Improves system-level performance

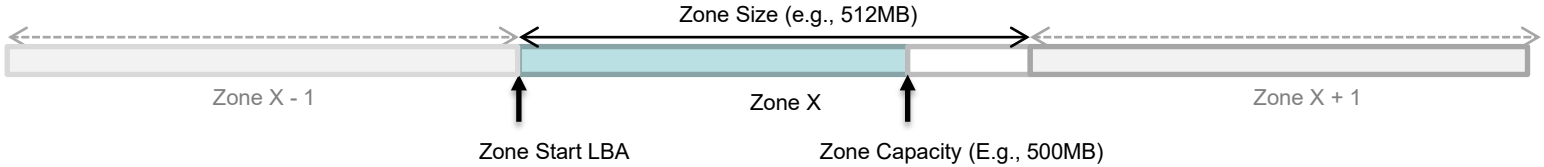


https://nvmexpress.org/developers/nvme-specification/
(Available in the 1.4 TP package)

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Zoned Storage Model Overview

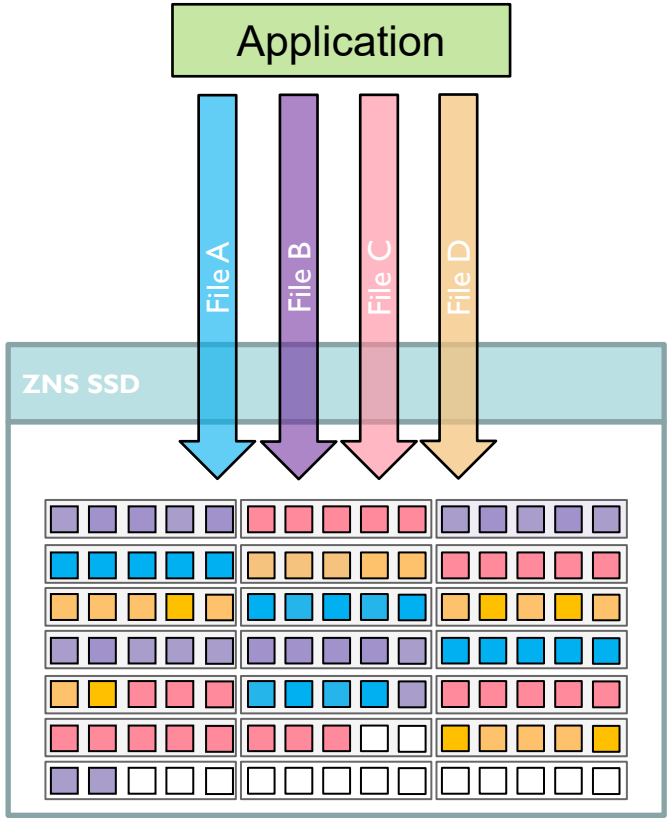
- Zone States
  - Empty / Open / Closed / Read Only / Offline
- Zone Management
  - Open Zone, Close Zone, Finish Zone, and Reset Zone
- Zone Size & Zone Capacity
  - Zone Size is fixed
  - **Zone Capacity** is the writeable area within a zone
- **Active** and Open Resources associated to a zone
  - Limits the maximum active and open zones



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

ZNS enables smart data placement

- Smart data placement enables better media utilization, lower WA
- Files can be mapped directly to erase units (zones)
  - **Minimal garbage collection**
  - **Lower write amplification**
- No longer a need to reserve over-provisioned media
  - **~7-28% more storage capacity**
- Enables QLC media for TLC use cases
- Reduces controller DRAM requirements

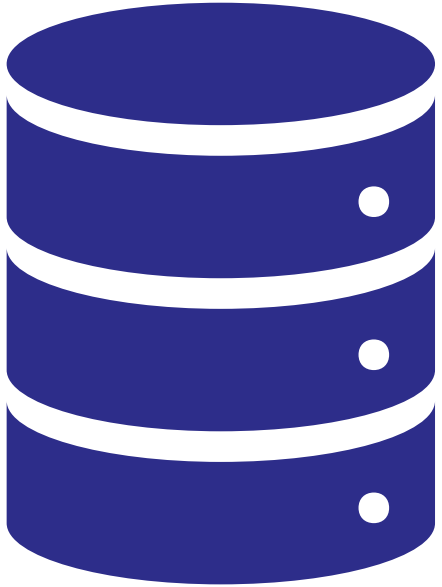


2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

RocksDB

A great fit for ZNS adaptation!

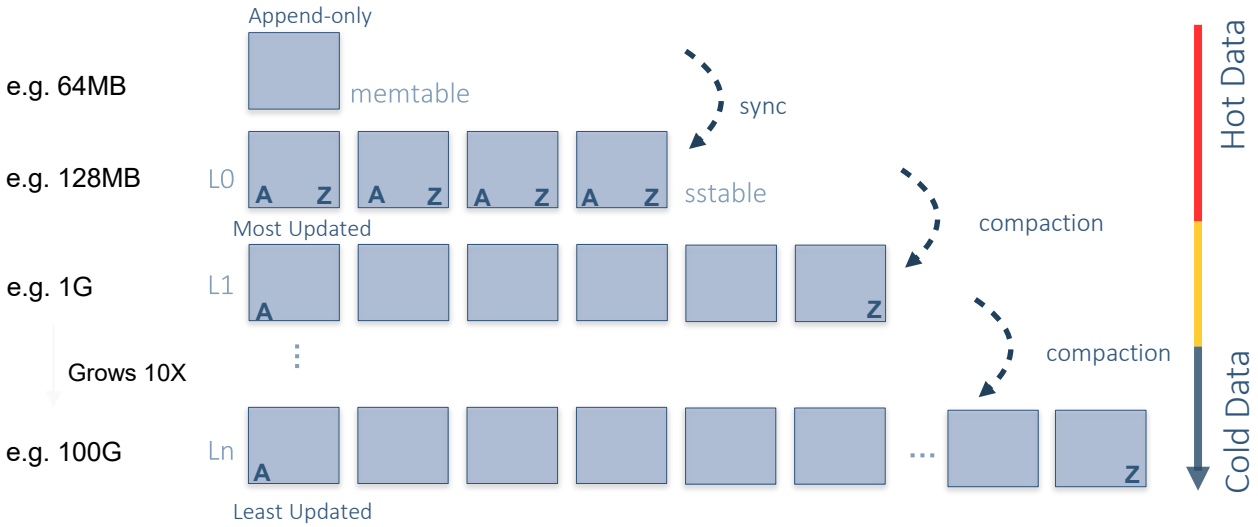
- Widely used persistent key-value store for fast storage environments
- Log-structured, flash friendly
  - Clear separation of hot and cold data
- Open source
- Portable
- Active community
  - Up to date with latest storage stack improvements (e.g. io_uring)
- Pluggable storage backends



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

RocksDB on-disk data structures

Clear separation of hot and cold data



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

ZenFS

- Goal: end-to-end ZNS integration
  - Add native support for ZNS as a RocksDB FileSystem class
  - Use as much as possible of RocksDB data knowledge to do smart data placement
- Minimize write amplification
  - Avoid garbage collection
  - Minimize wear
  - Maximize write throughput
  - Improve read performance
- Minimize integration effort for users

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

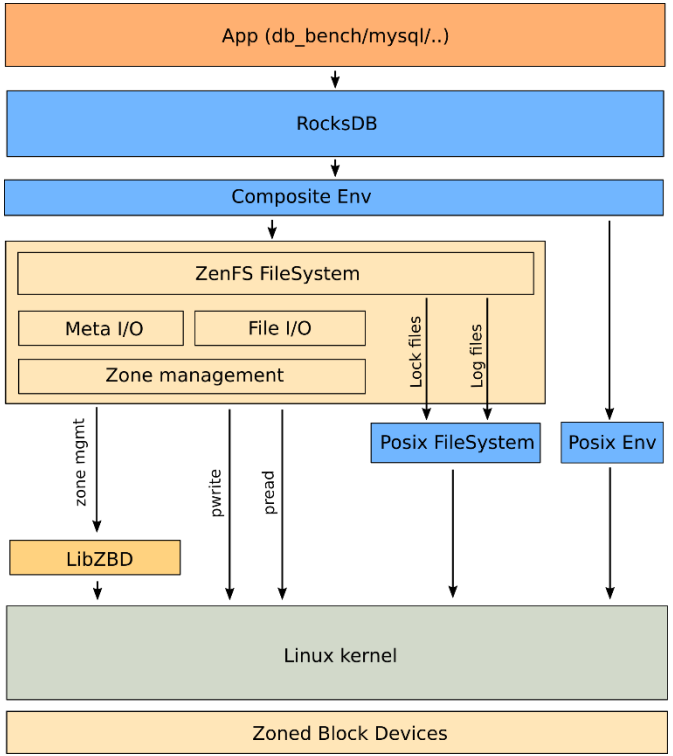
Architecture considerations

- Maximize reuse of existing components
- Leverage existing zone eco-system
- Keep complexity down to a minimum
- Flexibility in the right places
  - Disk IO path
  - Allocation algorithm

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

ZenFS Architecture

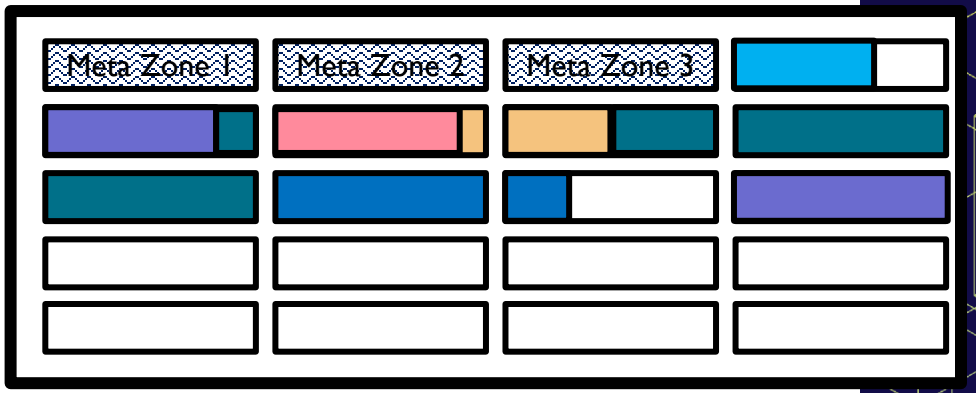
- Minimal impact on RocksDB itself
  - Adds a new ZenFS FileSystem class
  - **No changes** to the database code
- Stores all data and metadata files
  - Manifest
  - Write-ahead log
  - Sorted string tables (SSTs)
- Minimal external dependencies
  - Only relies on libzbd for zone operations
  - IO flows through conventional interfaces (pwrite/pread/libaio/io_uring..)
- ~ 2000 lines of code



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

ZenFS on-disk storage format

- File data is stored in a set of extents
  - Extents are a contigous part of the address space **within a zone**
  - Files can be split over zones
    - Allows files of arbitrary size
  - Files can share zones
    - Enables full zone capacity usage
- ZenFs Metadata
  - File and global file system metadata
  - Stored incrementally in a rolling log
  - Occupies the first three zones



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Zone Allocation

ZNS Constraints

- Sequential writes only
- Limited number of open zones
- Valid data must be evacuated before we can reuse a zone
  - Garbage collection is an option, but this leads to write amplification

Help from RocksDB

- Write life time hints
  - Metadata, WAL and SSTs
  - Different life time hints for SST levels
- File size hints
  - RocksDB provides file size hints before writing
- Append-only writes

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

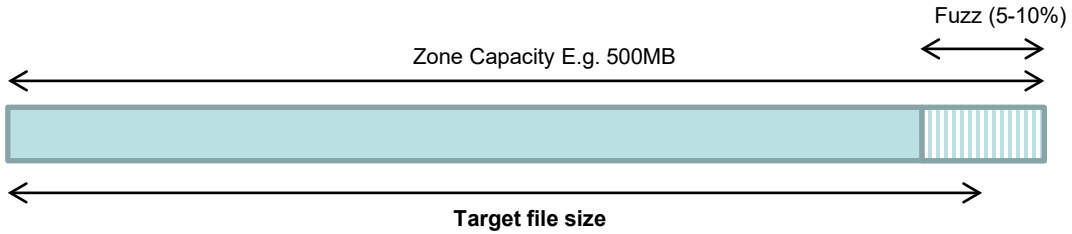
ZenFS extent allocator

- Pick a zone for a new extent:
  - When a file is first written to
  - When there is no capacity left in the zone
- Current zone picking scheme:
  - If there is space left in a zone, fill it with warmer data to maximize capacity usage
    - Pick the first zone that fullfills: **Zone_lifetime > File_lifetime** OR is empty
    - If the picked zone is empty: **Zone_lifetime := File_lifetime**
- Files hold a write lock on the zone they are writing to
- Zones are marked as free when all files using the zone has been deleted
  - **No GC, No write amplification**

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

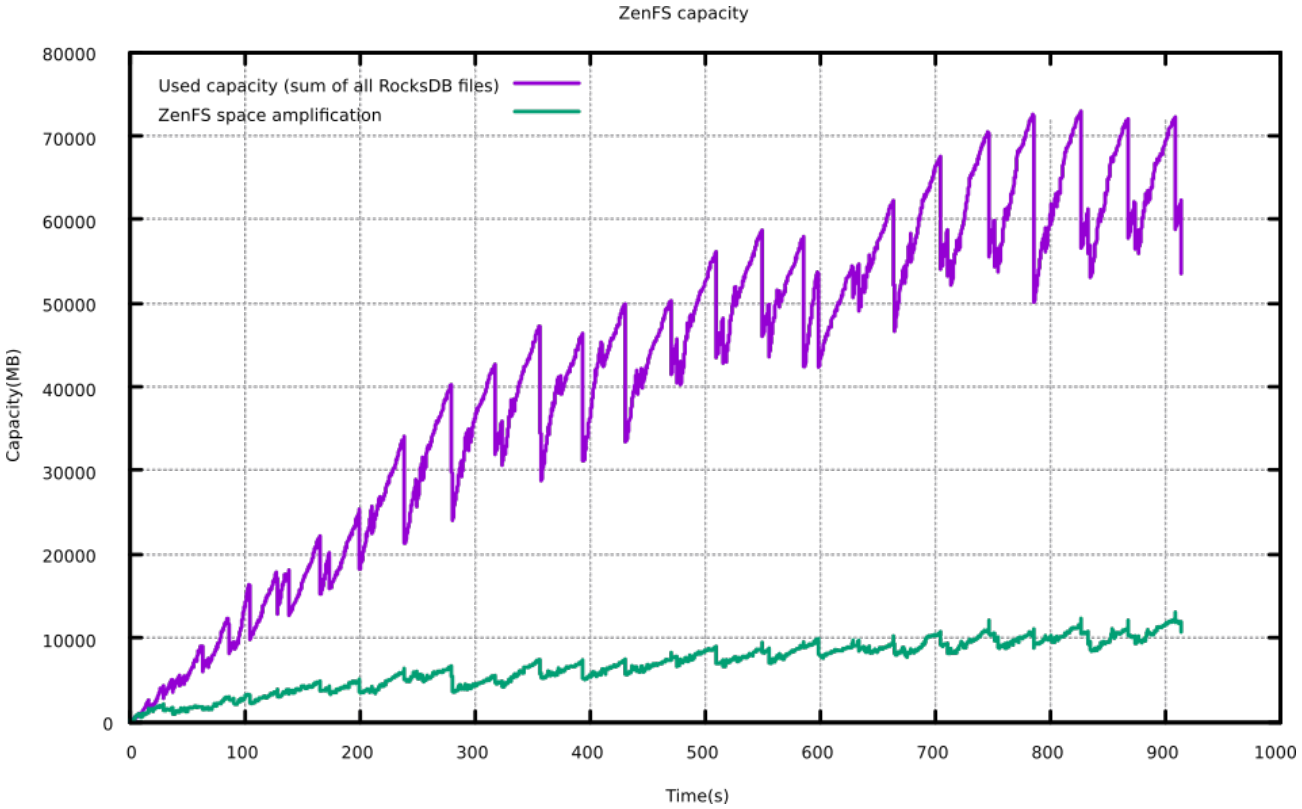
Fuzzy file size alignment

- The best capacity utilization is achieved if file sizes are aligned with zone capacity
- Target file size can be configured - but is just a hint to RocksDB
  - Allow for approximate/fuzzy zone capacity alignment
  - Set target file size in the middle of the fuzz area
  - Finish zones when they have reached the fuzz area



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

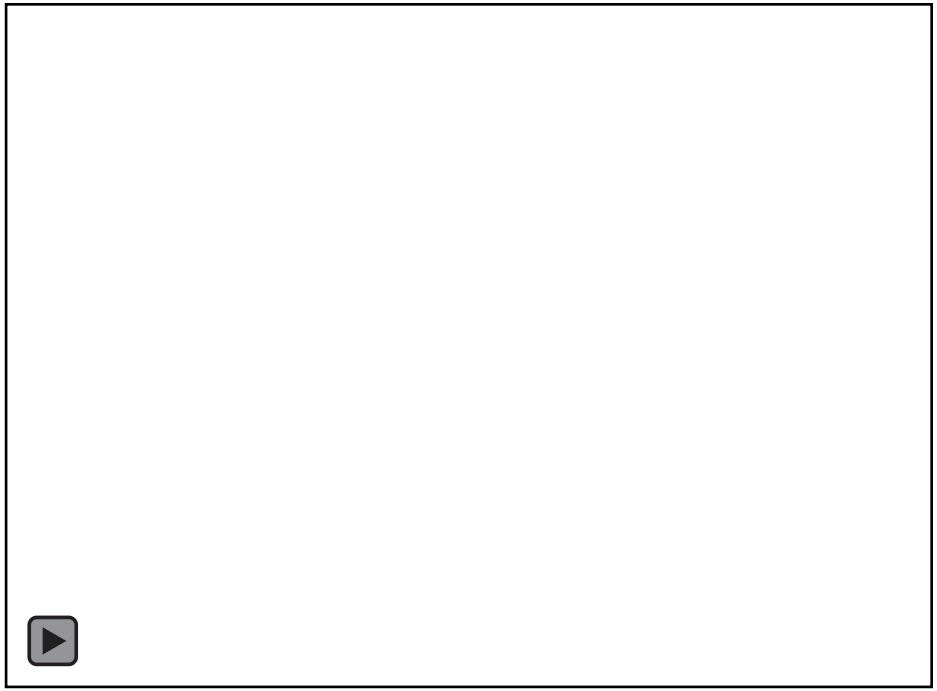
Space amplification



Write amplification = ~1.0

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Allocation visualization



2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

ZenFS Development

- Debugging/testing
  - db_bench, db_stress
  - gdb
- Tracing
  - A ZenFS log file is generated for each run on debug builds
    - /tmp/zenfs_<blockdevname>_*.log
  - Traces file operations and zone allocations
  - Great for debugging and evaluating extent allocation algorithms

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Current state

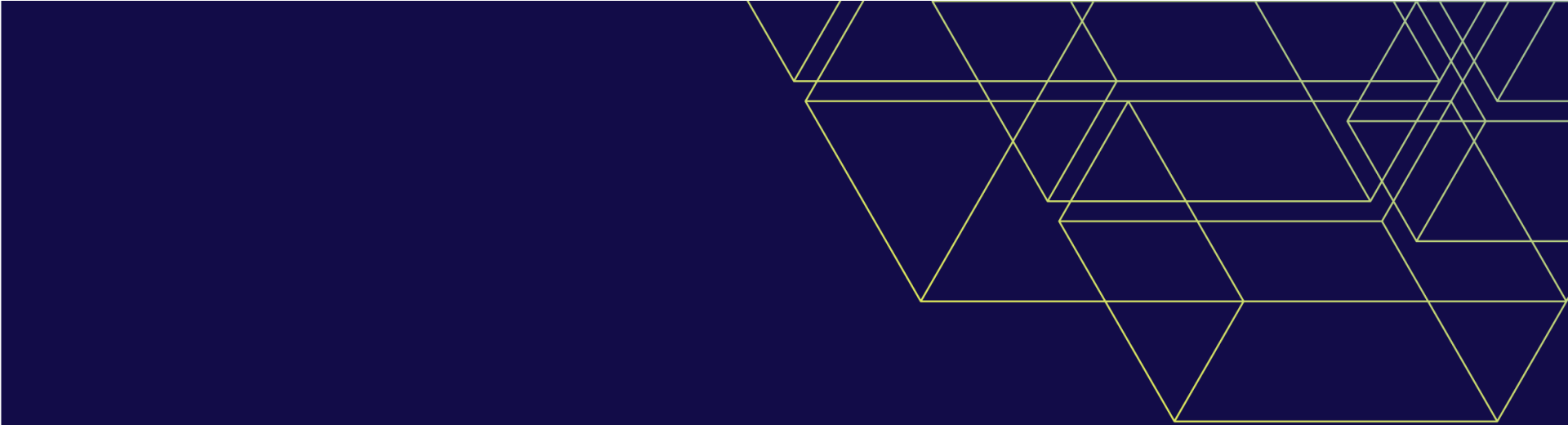
- Plumbing done
- Works with all Zoned Block Devices
  - ZNS SSDs, SMR Drives
  - Emulation/test devices: nullblk, QEMU
- Upstreaming in progress

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.

Try it out!

- Easy to get started
  - Use a memory-backed nullblk as a zoned block device
- Latest code with instructions(see README.md):
  - https://github.com/westerndigitalcorporation/rocksdb
- For more information on zones and the eco-system
  - https://zonedstorage.io

2020 Storage Developer Conference. © Western Digital Corporation. All Rights Reserved.



Thanks!