STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers
Virtual Conference
September 28-29, 2021
A SNIA Event

Boosting Performance and QoS of MySQL™ with NVMe™ Zoned Namespace SSDs.

Aravind Ramesh,
Western Digital Corporation.

Agenda

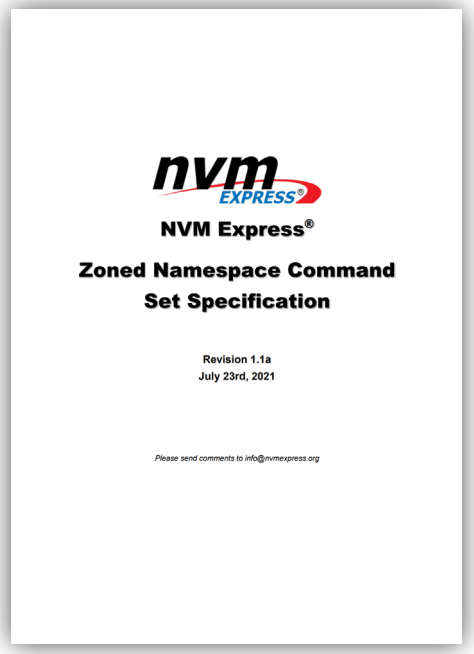
- Conventional SSDs
- NVMe™ Zoned Namespaces SSDs.
- MySQL™ Storage Engine Model.
- Introduction to ZenFS.
- MySQL™ Stack on ZNS SSDs.
- Benchmarks.
- Ecosystem Enablement.
- Q & A.

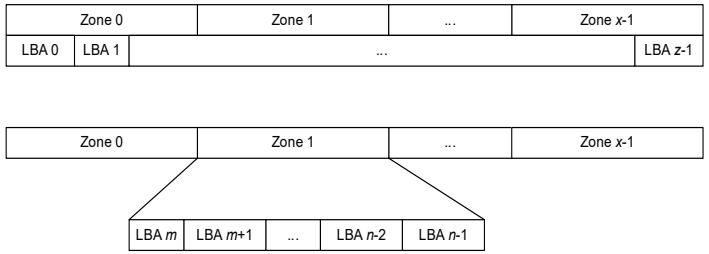
Conventional SSDs

- SSDs are fundamentally different from HDDs.
  - Cannot over write, needs to erase blocks before writing again.
  - Expected to work like a HDD.
- Firmware on SSDs do a lot of work.
  - Manage overwrite requests, do garbage collection to reclaim old blocks.
  - Causes space amplification and write amplification.
  - Over provision is used to maintain performance.
- Increased write amplification can deteriorate life of NAND media.
  - Pre-set P/E cycles for the media.
- Most of these issues are rooted from the fact that the SSDs are expected to work like HDDs.

NVMe™ Zoned Namespaces SSDs.

- NVM Express™ Zoned Namespace Command Set specification introduces the Zoned Storage Model to SSDs.
  - Aligns host data placement to the characteristics of flash-based SSDs.
  - The media within the SSD is exposed as a set of zones and these zones can be written to sequentially only.
  - A zone can be reset to reclaim the space and can be written from the start of the zone.
  - Data can be placed on to the zones directly from the host on to the SSD.
- Benefits of ZNS
  - Significantly better I/O latency QoS.
  - Higher throughput and endurance of SSD (3-4x)
  - More capacity (7%-28%) as the conventional SSD over-provisioning for well-performent device-side garbage collection no longer is necessary
- Challenges:
  - Requires host to understand the layout of zones and specific write requirements. E.g., Write sequentially within a zone.
  - Well-suited for flash-friendly workloads (log-structured, logging, caching, etc.).



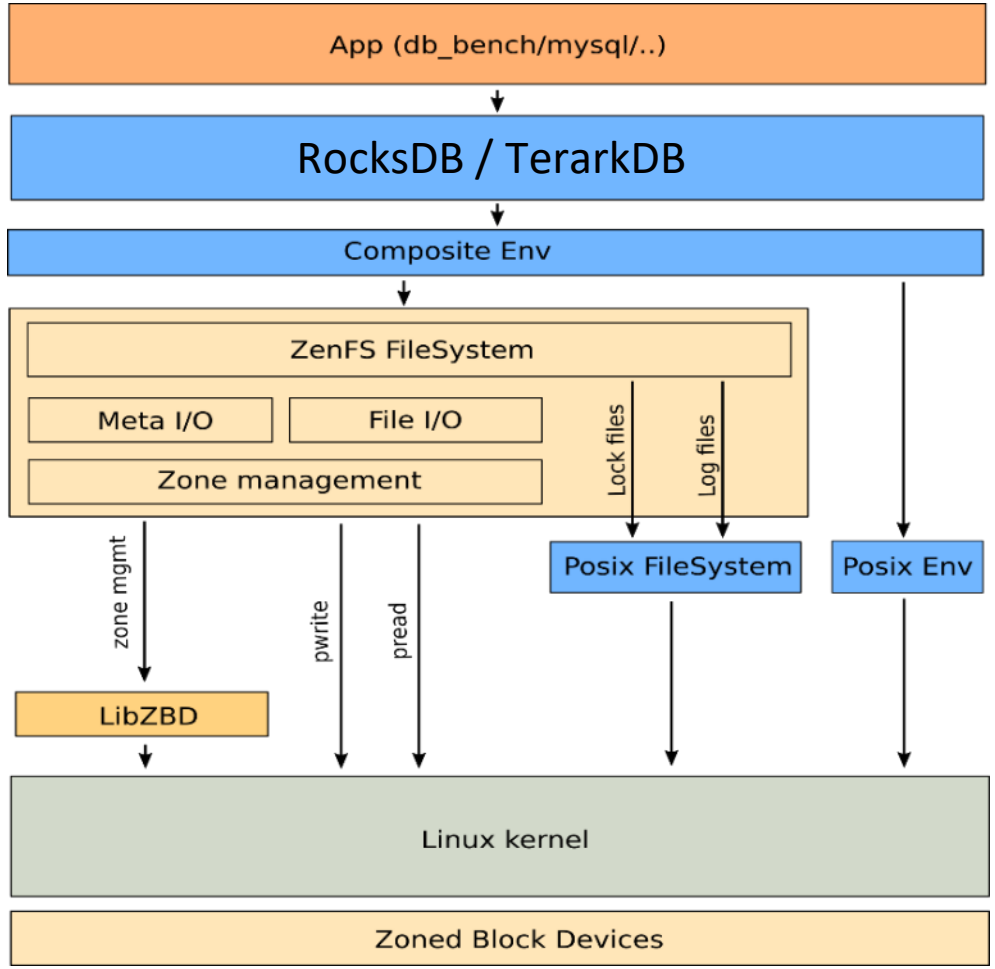


MySQL™ Storage Engine Model

- MySQL™ is an open source relational database system.
- MySQL supports pluggable model of storage engines. A storage engine is responsible for managing different table types and reading and writing data from/to persistent storage. The pluggable storage engine architecture also provides a standard set of management and support services that are common among all underlying storage engines.
- Storage engines:
  - InnoDB: Default storage engine, uses btree data structures, more read friendly.
  - MyRocks: MySQL with RocksDB as the storage engine. Write-optimized storage backend.
  - Other storage engines (MyISAM, Memory, CSV, Archive, …)

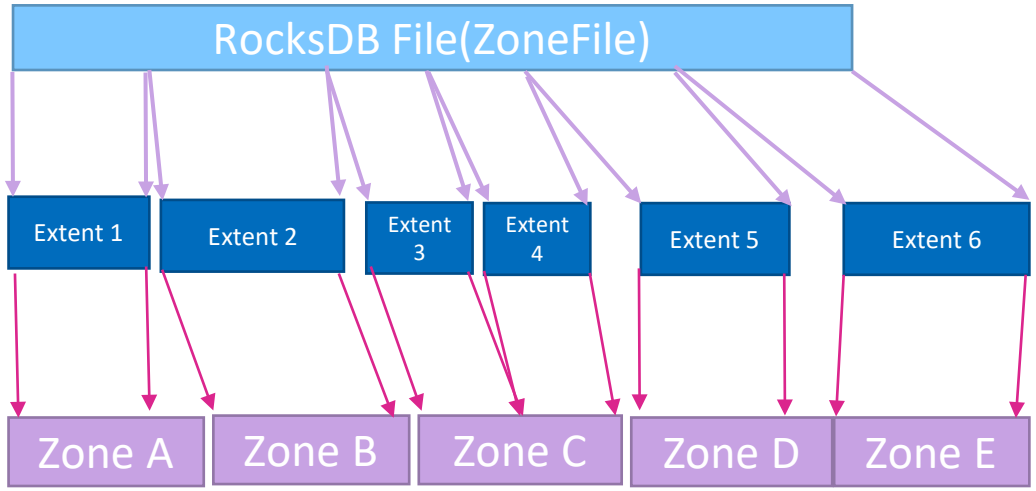
Introduction to ZenFS

- ZenFS is an open source software developed and maintained by Hans Holmberg.
- ZenFS is available in RocksDB and TerarkDB. It plugs into their file-system interface on the front end and places data on to zones of a zoned block device (ZNS SSD) in back end.
- ZenFS utilizes any hints from the database system (e.g., write hints) and separates files into different zones and fill up any remaining space with data that has less expected life time.
- ZenFS ensures that there is no background garbage collection in the file system, improving performance in terms of throughput, tail latencies and disk endurance.



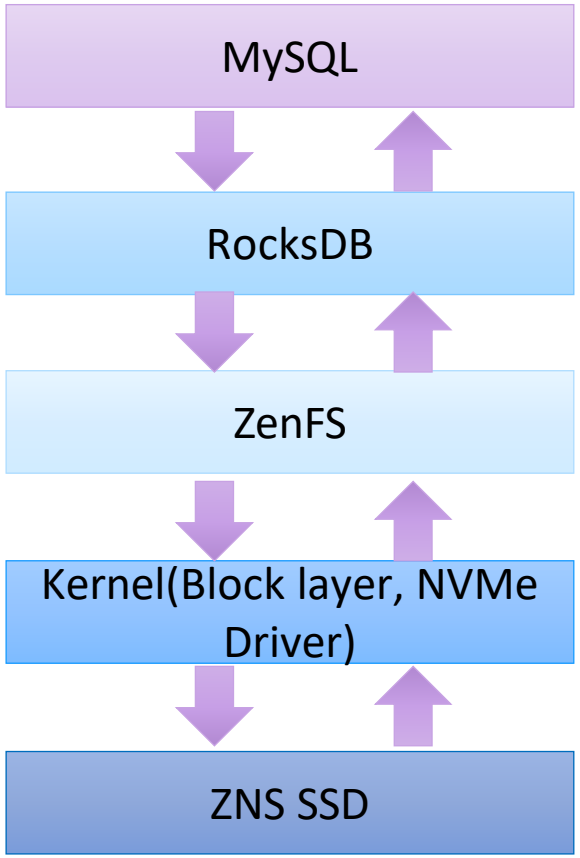
Introduction to ZenFS (continued)

- ZenFS stores data in terms of Zonefiles, each Zonefile is a set of extents. Zonefiles can be placed across one or more zones. Individual extents do not span across zones.
- When all zonefiles in a zone are invalidated, the zone is reset by ZenFS, reclaiming the capacity of the zone.
- RocksDB does the garbage collection, there is no garbage collection being done by either ZenFS or within the ZNS SSD.
- Some zones in the device are marked as meta zones and the filesystem metadata is stored on theses zones. Meta data is also written as a Log into zones and is rolled over to a new meta zone when the current zone fills up.



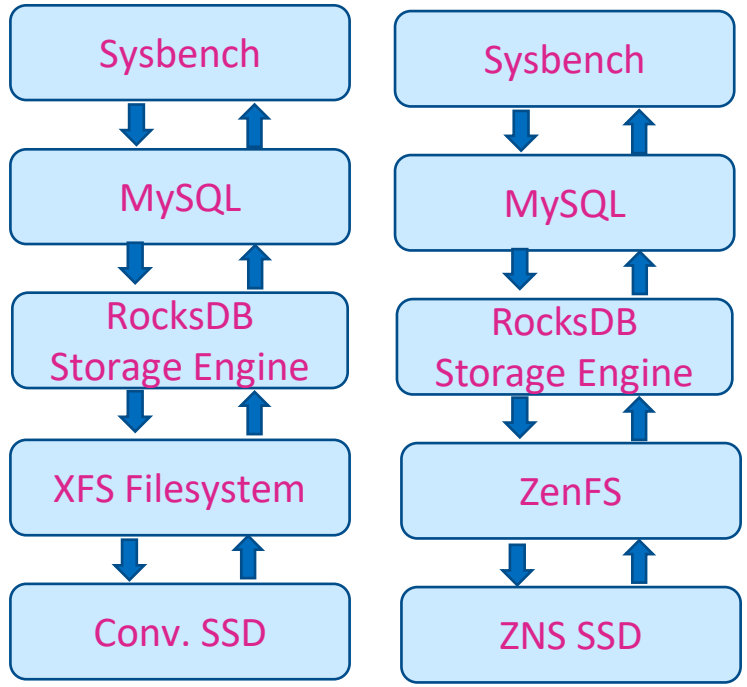
MySQL™ Software Stack

- We use RocksDB as the MySQL storage engine.
- RocksDB requires ZenFS as a plugin to support ZNS devices.
- ZenFS uses libzbd to do zone management operations.
- ZenFS directly writes/reads to ZNS SSD zones, using pwrite/pread().
- Linux® kernel (5.9+) supports ZNS SSDs.
- ZNS SSDs that implements the NVMe™ ZNS Specification.



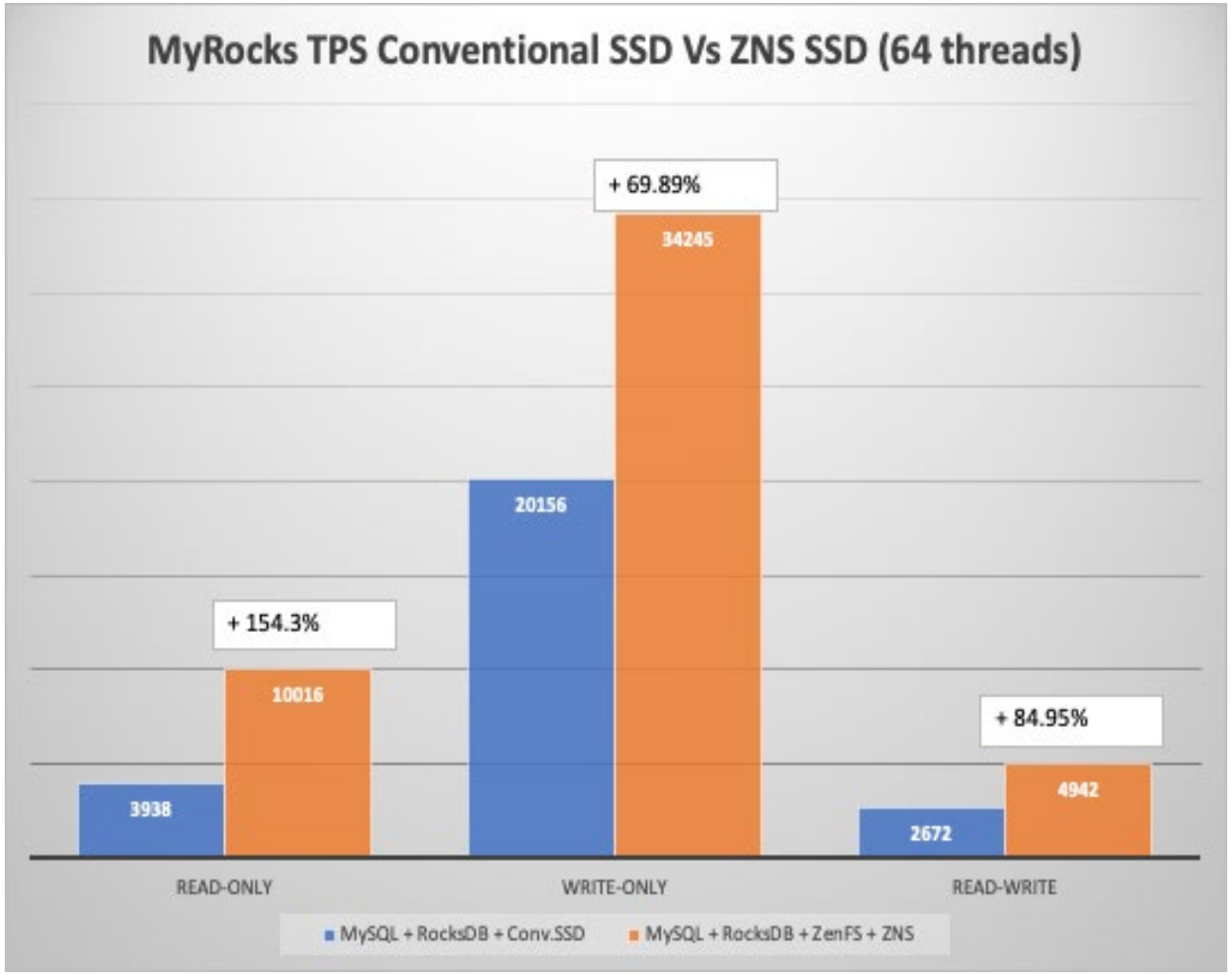
MySQL-RocksDB-XFS vs MySQL-RocksDB-ZenFS

- System Configuration
  - CPU: Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz, 112 Cores
  - RAM: 356GB
  - SSDs
    - ZNS compliant drive for Zenfs (3.7 TB).
    - Conventional SSD with XFS file system (3.7TB).
    - SSDs share the same hardware platform and has the same type of media.
  - Database size: ~2.5 TB, 500 million rows per table, 24 Tables.
  - Both SSDs have similar raw performances capabilities to ensure the comparisons are valid.

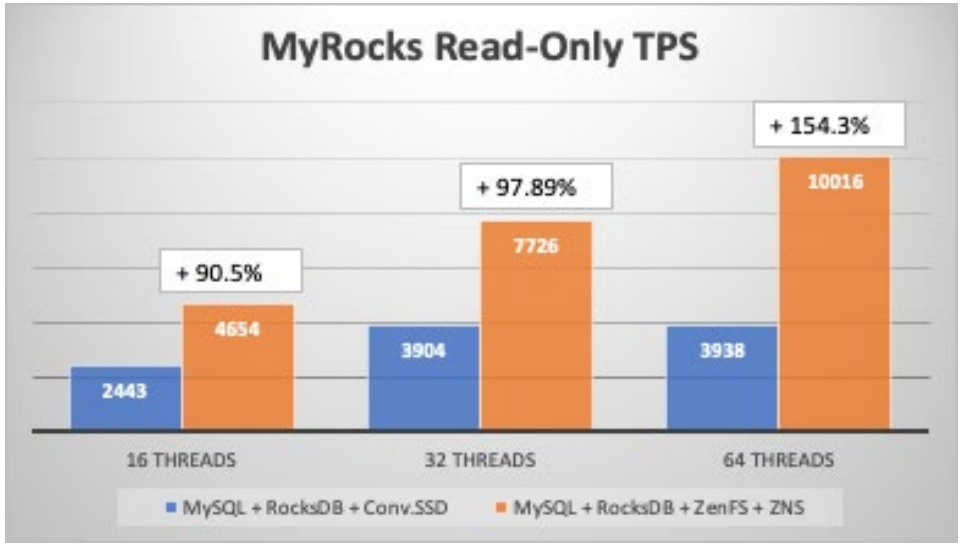


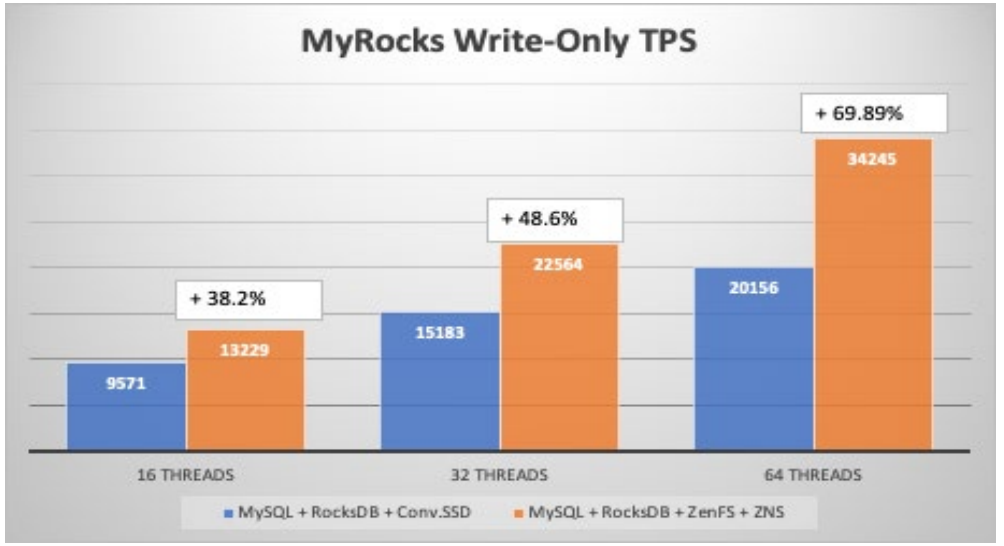
MySQL Benchmarks –

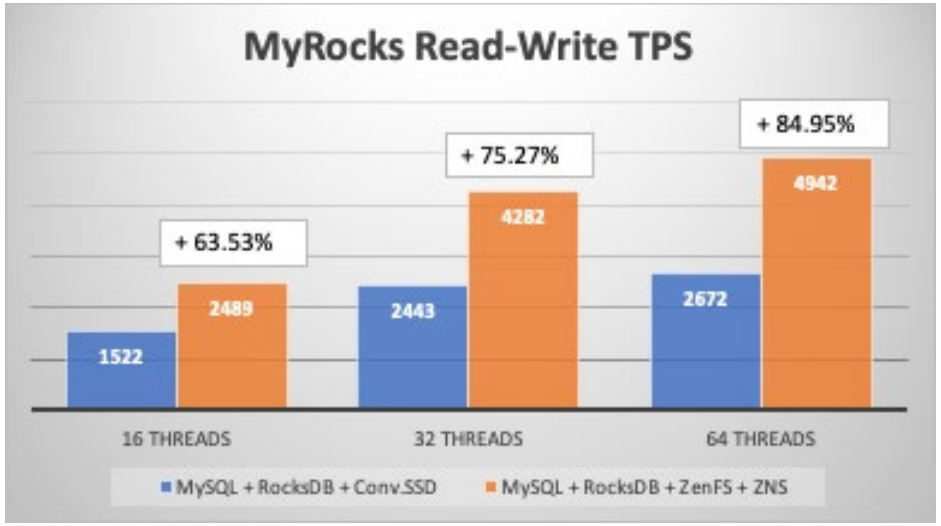
- Sysbench Workloads:
  - Read Only
  - Write Only
  - Read Write
- Workloads were run with 16 threads, 32 threads, 64 threads.
- Shows transactions per second.
  - 69% - 154% more transactions compared to the conventional SSD!



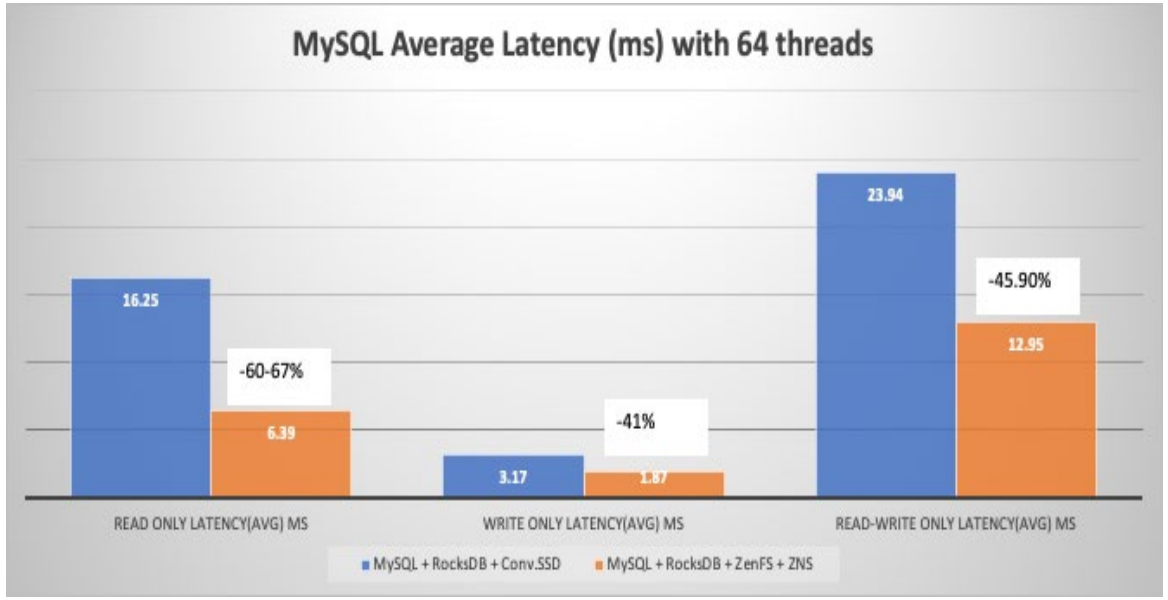
Read-Only, Write-Only, Read-Write Benchmarks

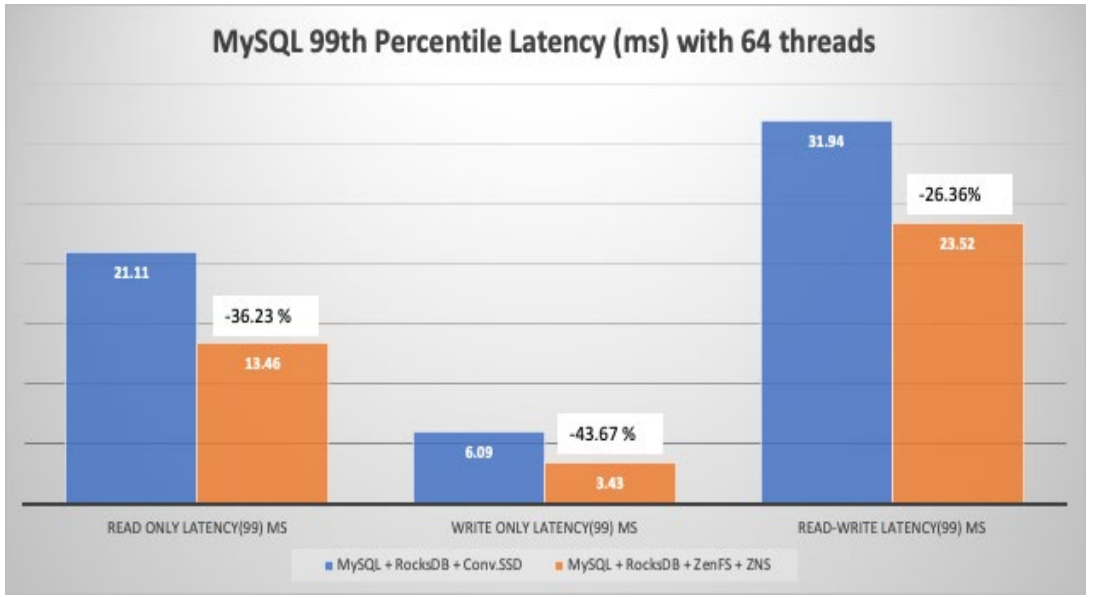


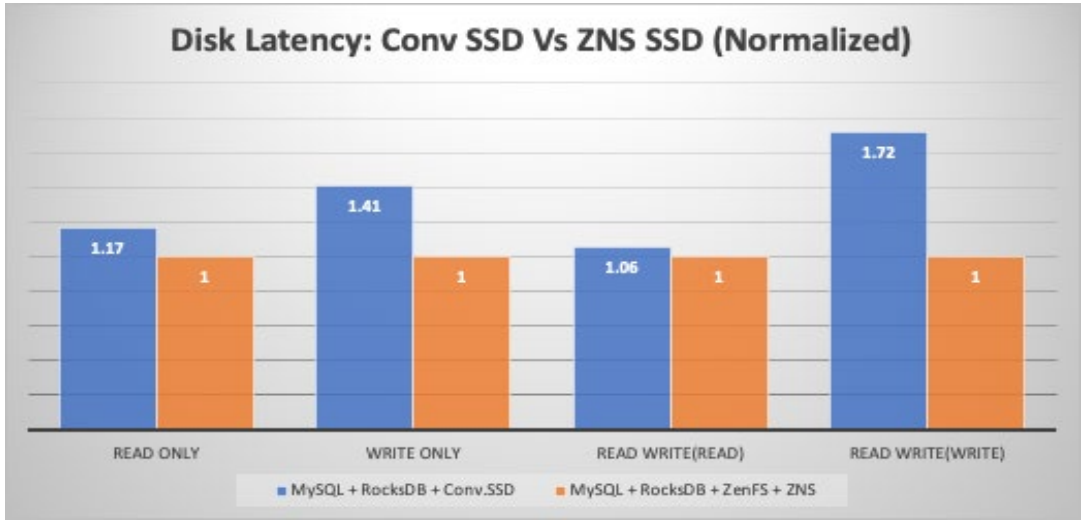




Latency Improvements (Lower is better)







The Road to Enabling MySQL

- Developed, and open sourced ZenFS.
- Enabled ZenFS as a first-class citizen:
  - MySQL to support RocksDB with ZenFS.
  - RocksDB to support and add ZenFS as a first-class storage backend.
- Enabling Linux® kernel support and libraries to support ZNS SSDs.

References

- https://zonedstorage.io
- https://github.com/westerndigitalcorporation/zenfs
- https://github.com/facebook/rocksdb
- https://github.com/facebook/mysql-5.6
- https://nvmexpress.org/new-nvmetm-specification-defines-zoned-namespaces-zns-as-go-to-industry-technology/
- https://nvmexpress.org/wp-content/uploads/NVM-Express-Zoned-Namespace-Command-Set-Specification-1.1-2021.06.02-Ratified-1.pdf

Thank You.
Please take a moment to rate this session.

Your feedback is important to us.