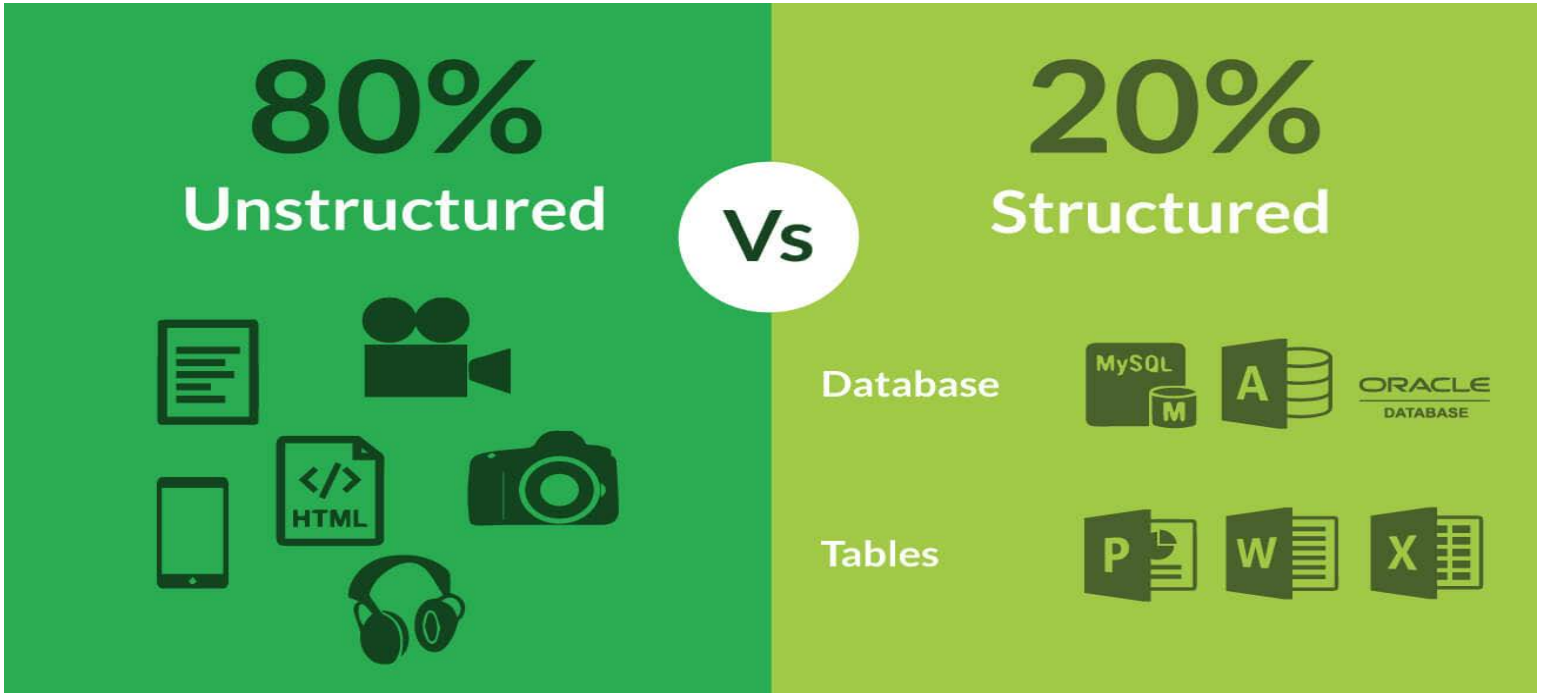
Key/Value SSD
Design Overview and Use Cases

Stanley Miao
Chief Engineer
Shannon Systems

Unstructured data dominates



Use Cases

Pika

ceph

CockroachDB

MyRocks

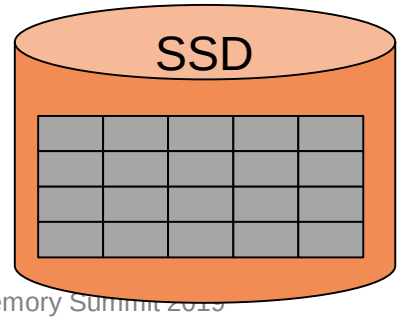
TiDB

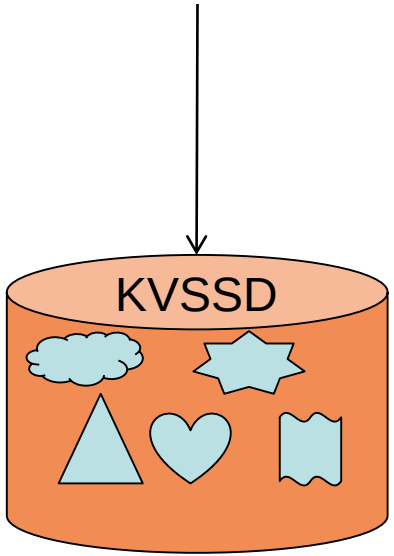
Apache Flink

mongoDB.

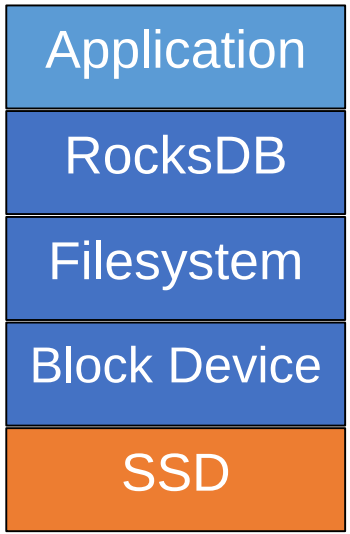
levelDB

RocksDB

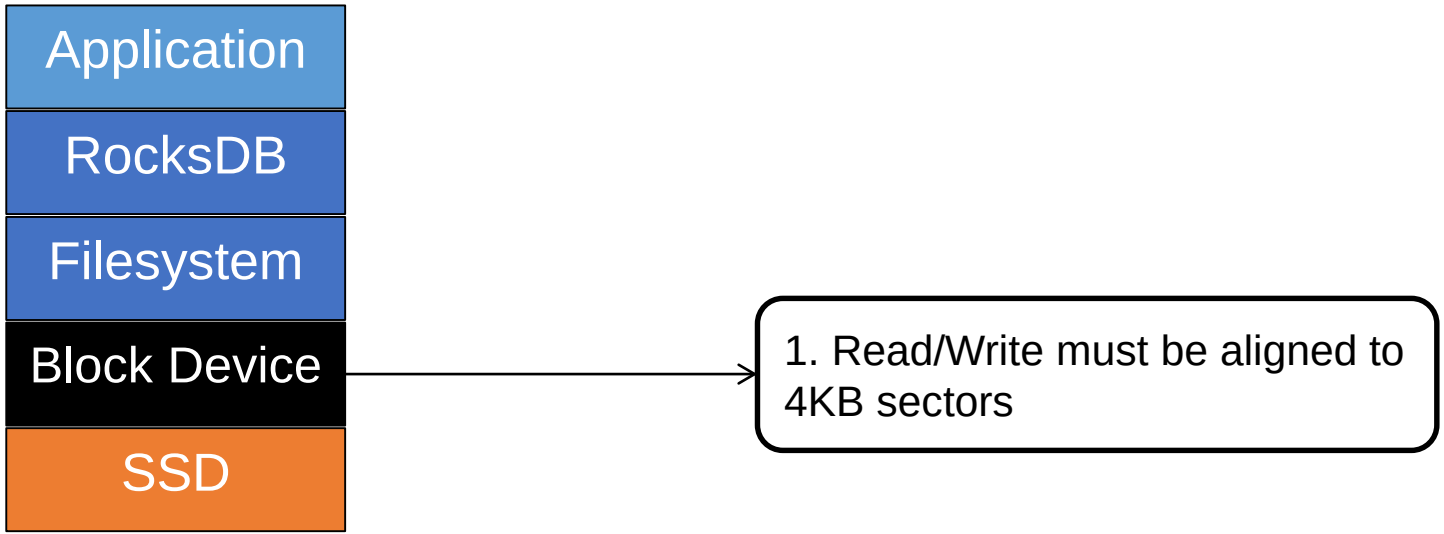




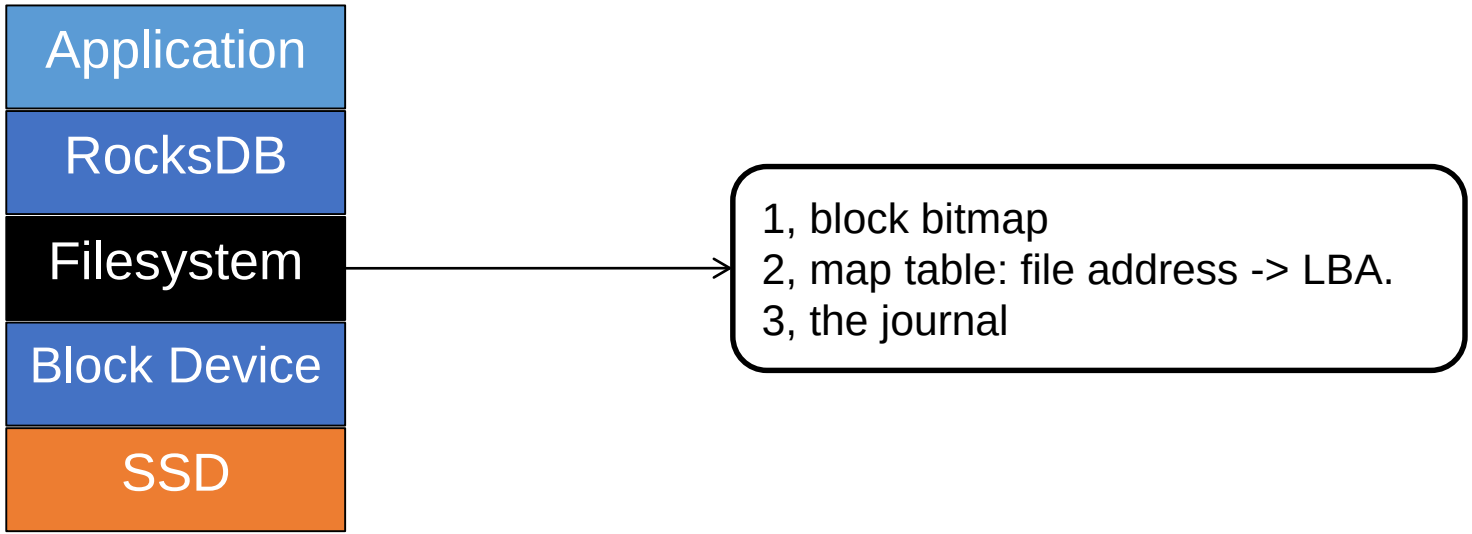
The overhead cost of RocksDB



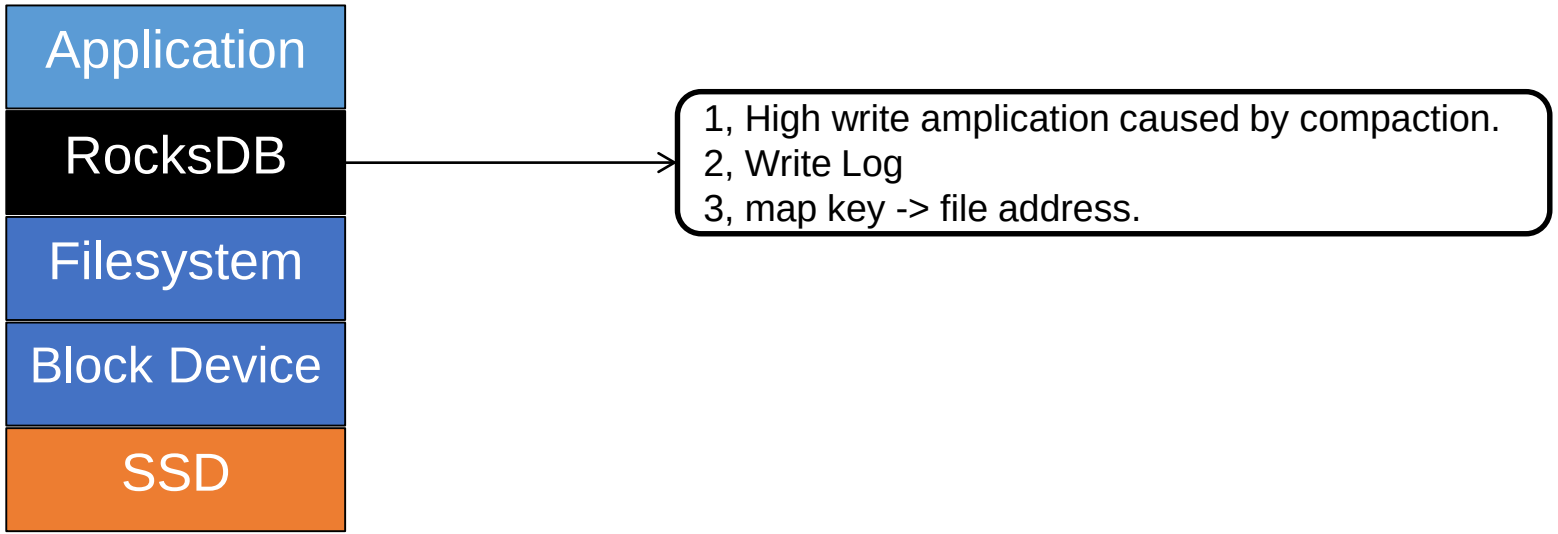
The overhead of RocksDB



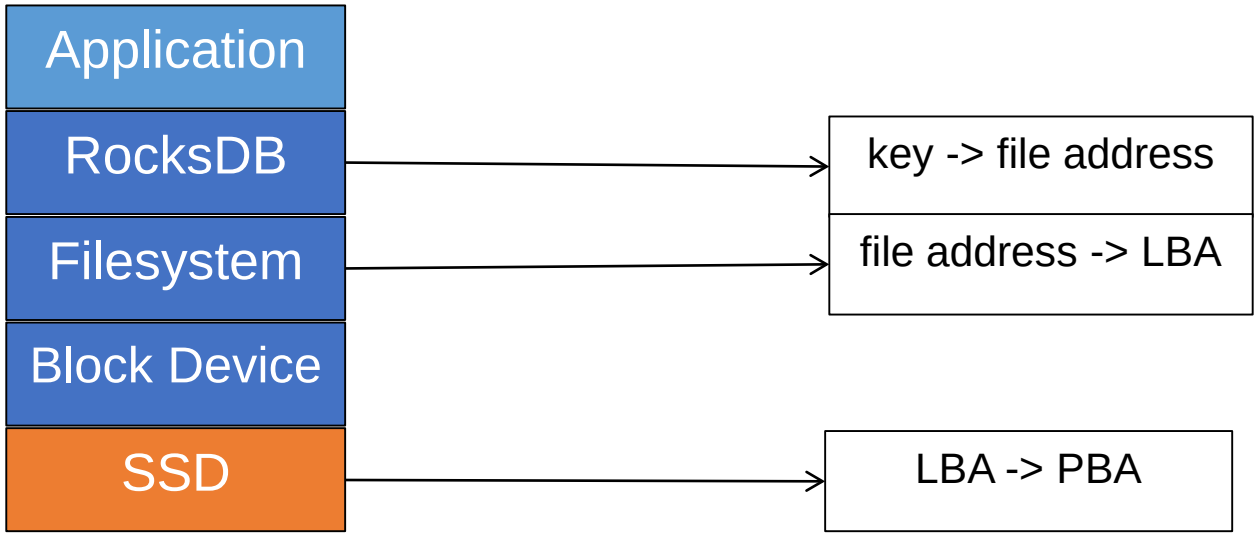
The overhead of RocksDB



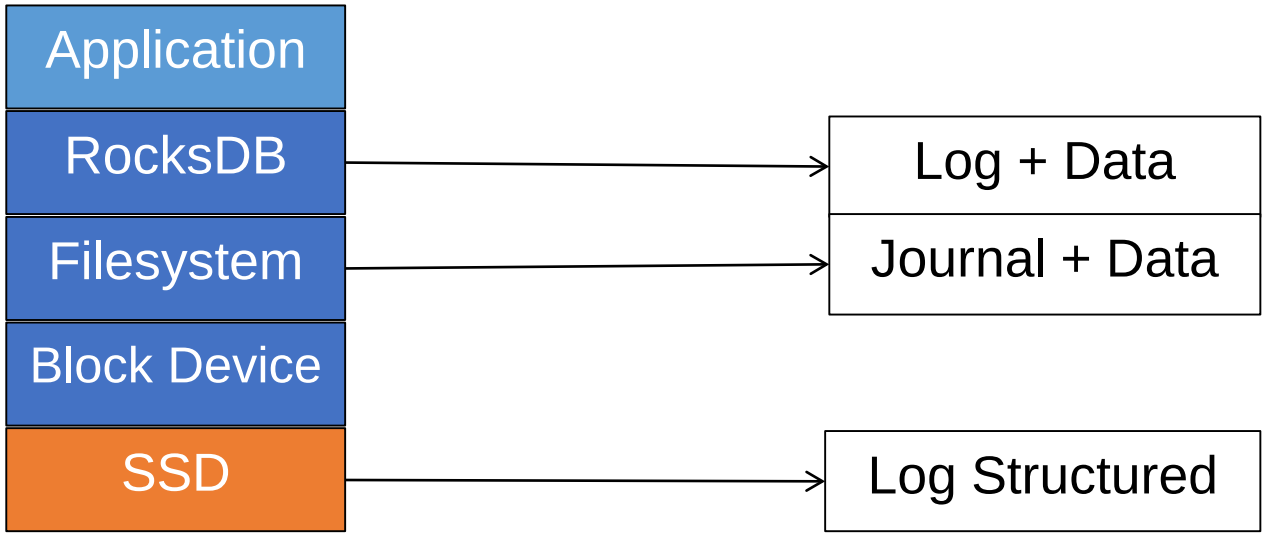
The overhead of RocksDB



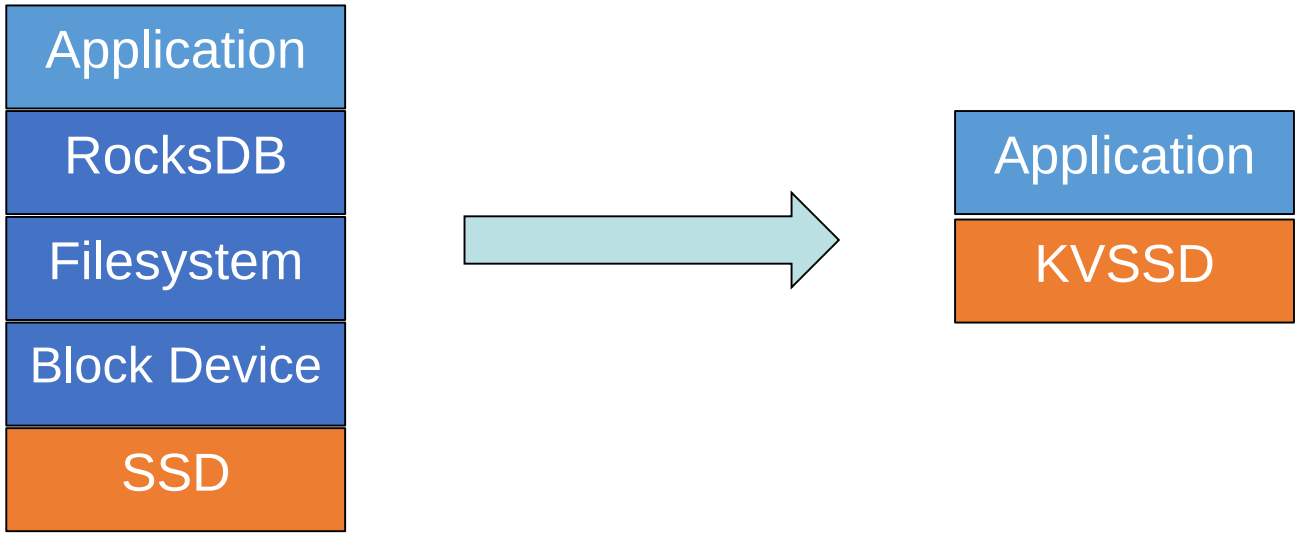
The overhead of RocksDB



The overhead of RocksDB



Software Architecture Evolves

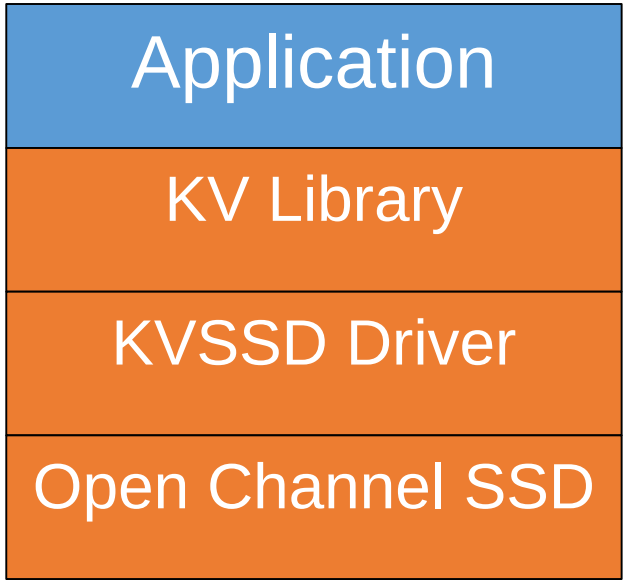


KVSSD Software Stack

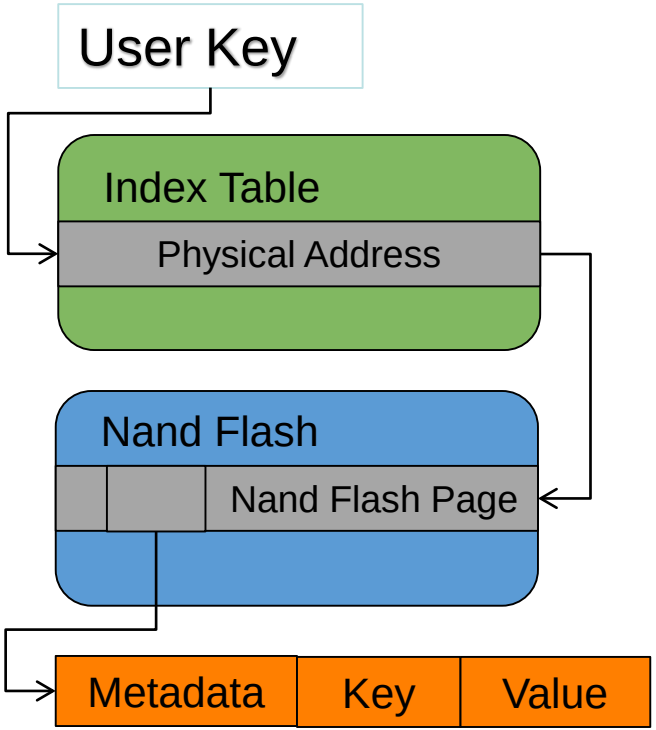
User Space

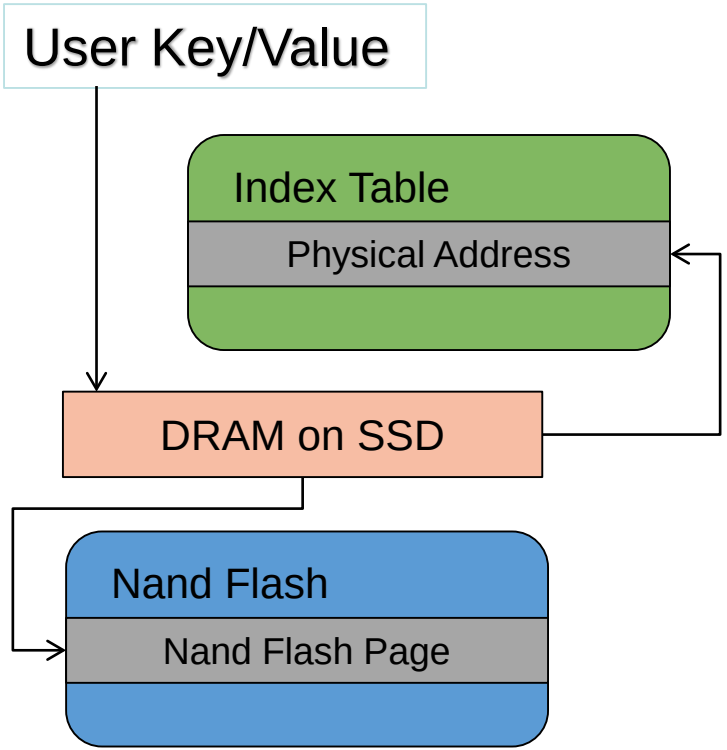
Kernel Space

Hardware



KVSSD Driver Read/Write Flow

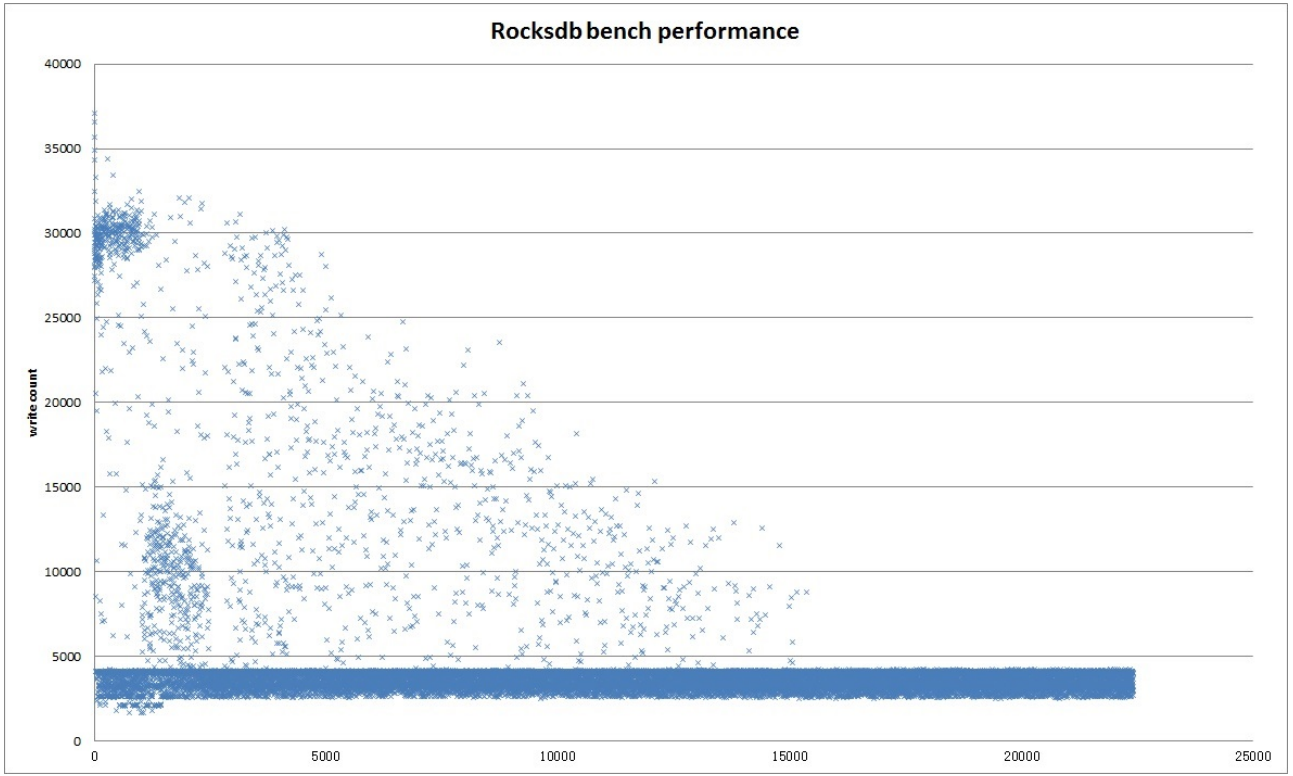




KVSSD Overview

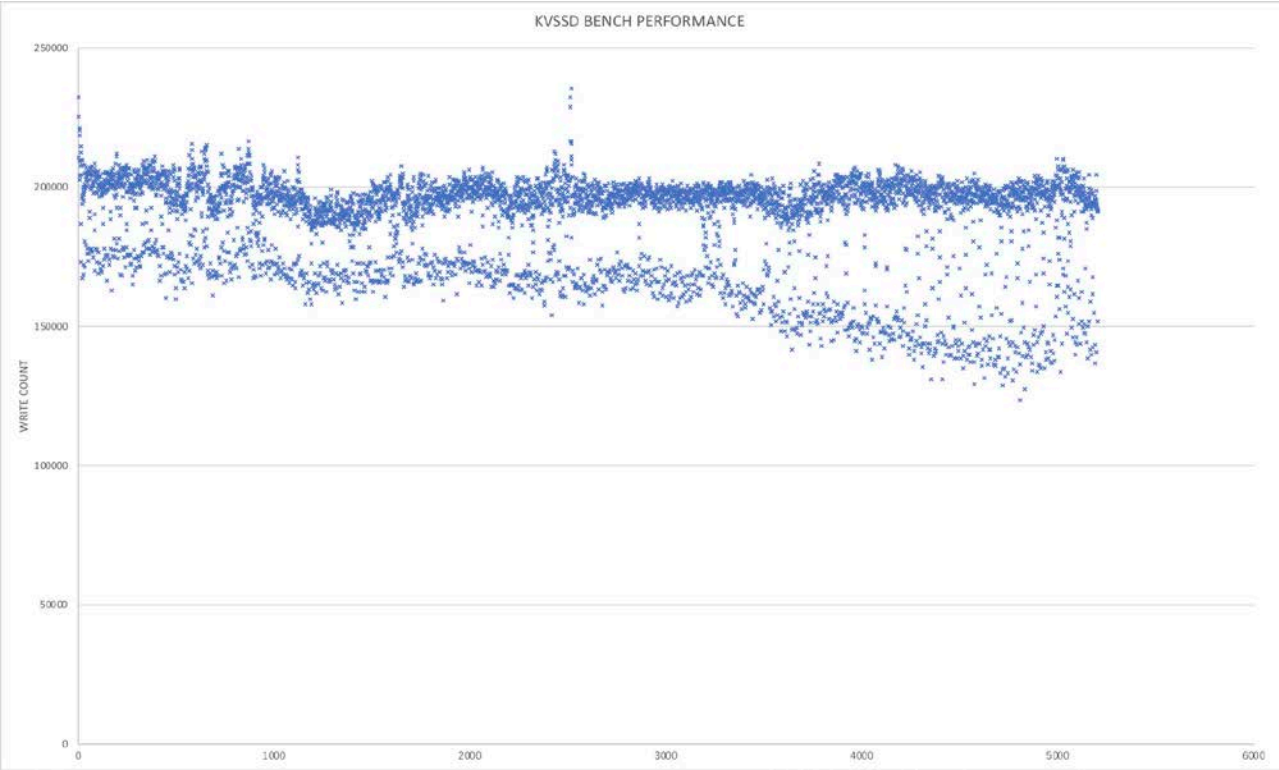
- KEY_SIZE: 1B ~127B
- VALUE_SIZE: 1B ~ 16MB
- Persistency
- Atomic Write

RocksDB Random Write IOPS



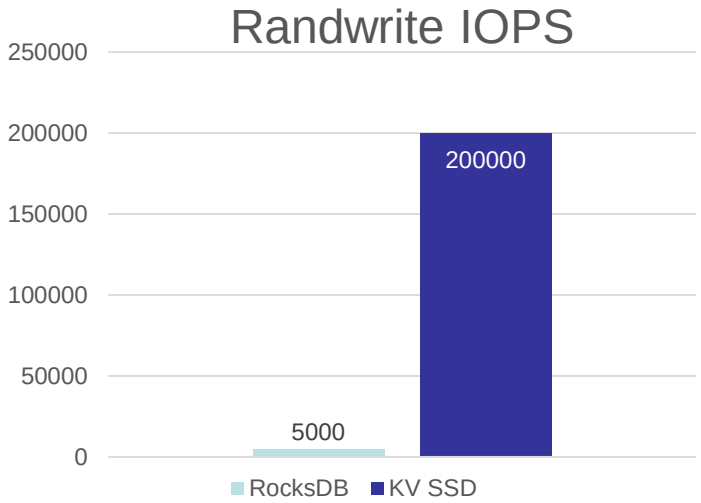
Test Conditons：key_size = 16bytes, value_size = 4KB, threads = 10, WriteOption.sync=true

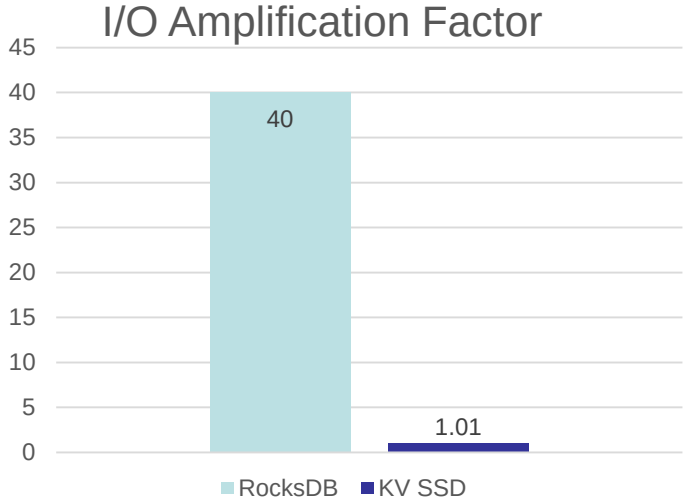
KVSSD Random Write IOPS



Test Conditons: key_size = 16bytes, value_size = 4KB, threads = 10, WriteOption.sync=true

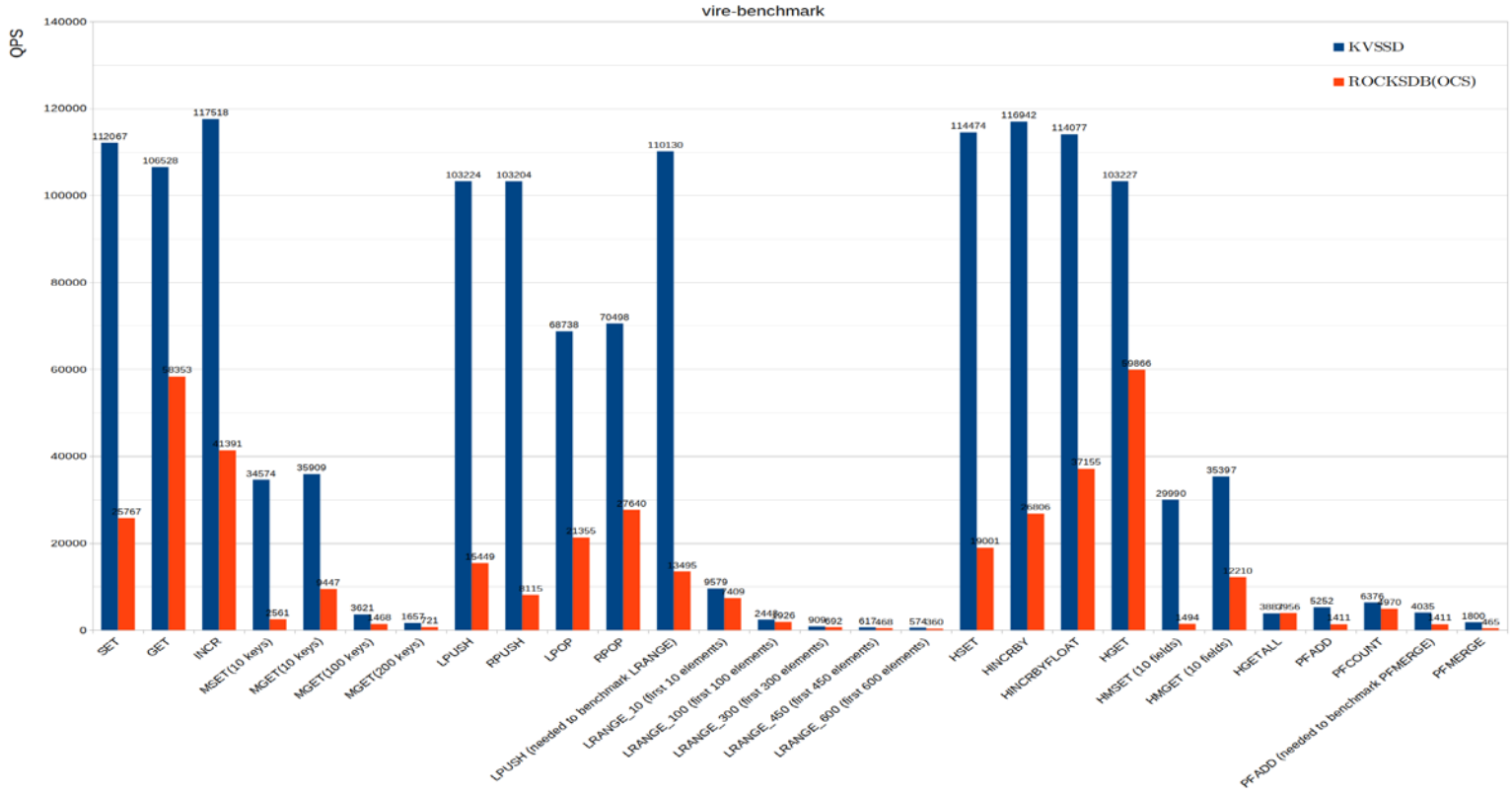
Random Write Performance Comparison





Test Conditons: key_size = 16bytes, value_size = 4KB, threads = 10, WriteOption.sync=true

Use Cases: Pika's vire-benchmark result



github projects

https://www.github.com/shannon-sys

- pika
- mongodb
- kv_library
- benchmark tools

KV Library API

- Provide both C and C++ API
- Compatible with RocksDB API
- kv_put/kv_get/kv_delete/kv_exist
- Column Family
- Snapshot
- Iterator
- Write_batch/Read_batch

Shannon KVSSD products

- Interface: PCIe/u.2
- Capacity: 1TB/2TB/4TB

Questions?

Contact me:
Booth: 413
Email: stanley@shannon-sys.com
Wechat: stanleymiao