Pulsar Storage on BookKeeper

Seamless Evolution

June 17, 2020

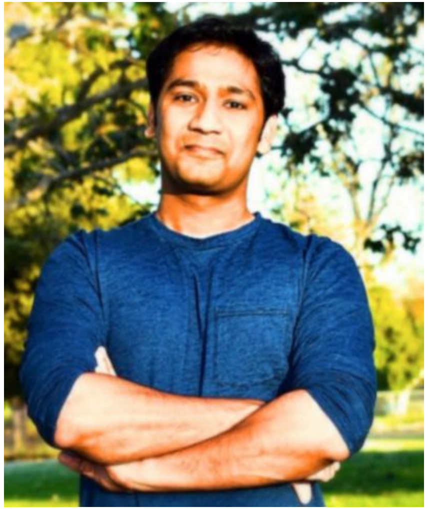
Joe Francis joef@verizonmedia.com
Rajan Dhabalia rdhabalia@verizonmedia.com

verizon media

Speakers

Joe Francis

Director, Verizon Media



Rajan Dhabalia

Principal Software Engineer, Verizon Media

Agenda

- Pulsar in Verizon Media
- Benchmarking for production use
- Pulsar IO Isolation
- BookKeeper with different storage devices
- Case-study: Kafka use case on Pulsar
- Future

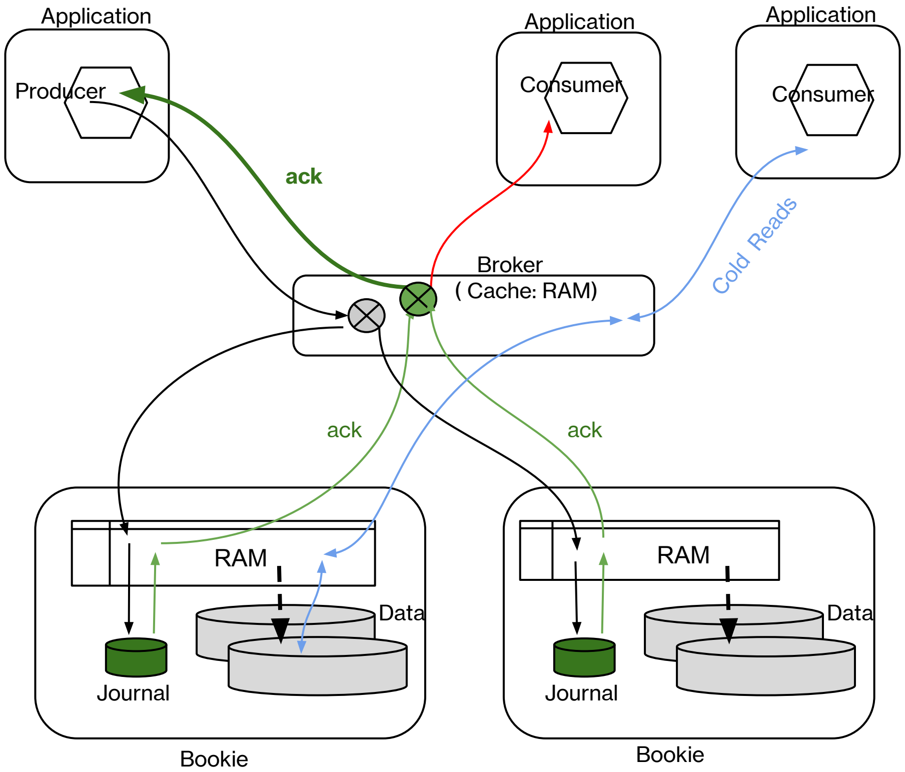
Verizon Media & Pulsar

- Developed as a hosted pub-sub service within Yahoo/VMG
  - open-sourced in 2016
- Global deployment
  - 6 DC (Asia, Europe, US)
  - full mesh replication
- Mission critical use cases
  - Serving applications
  - Lower latency bus for use by other low latency services
  - Write availability

Benchmarking for production

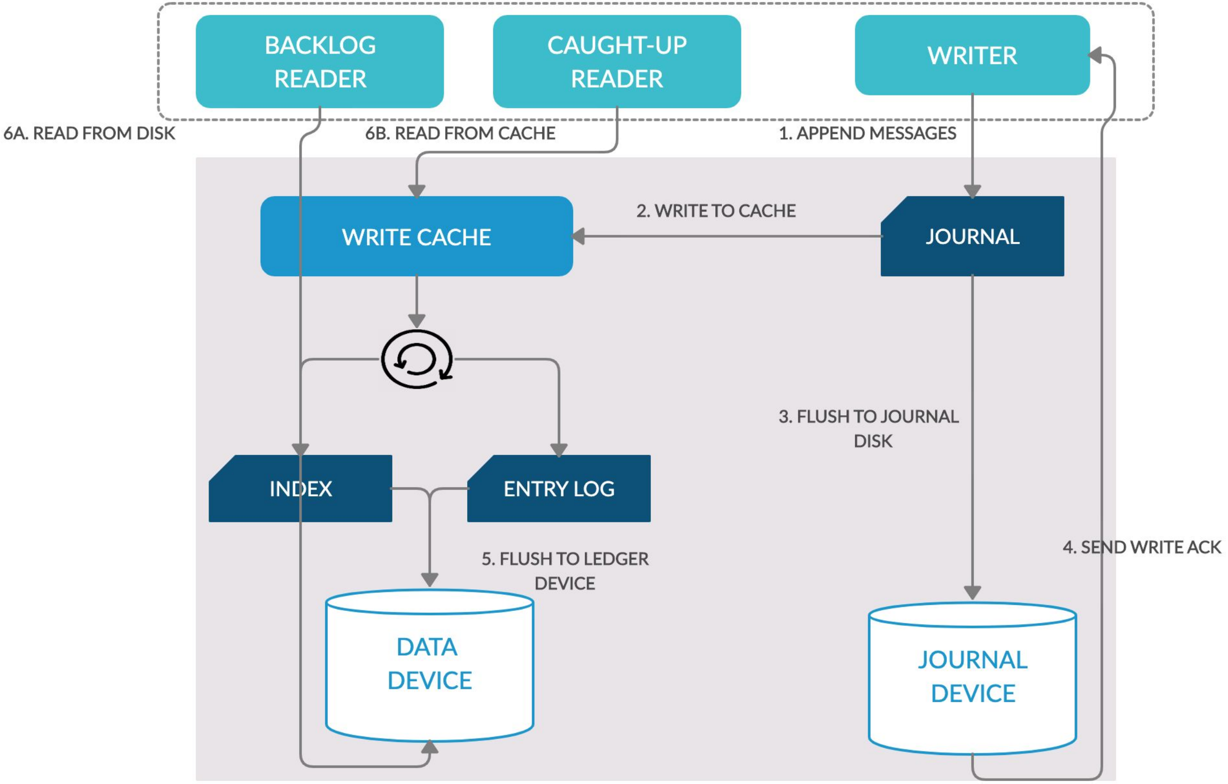
- Most benchmark numbers do not test production scenarios
    - Messaging systems work well when
        - data fits in memory
        - no disk I/O in critical path (write or read)
- Pulsar was designed to work well under real world work load..
    - Lagging consumers, replay
        - Backlog read from disks will occur.
    - Disks and brokers crash/fail
        - Pulsar ack **guarantee:** data is synced to disk on 2+ hosts
    - Latencies remain unaffected by load variations
        - backlog reads (I/O isolation)
        - failures (instantaneous recovery)
- Cost matters
    - Compute ($) vs Storage ($$)
- Benchmark for production use !!!

Data paths



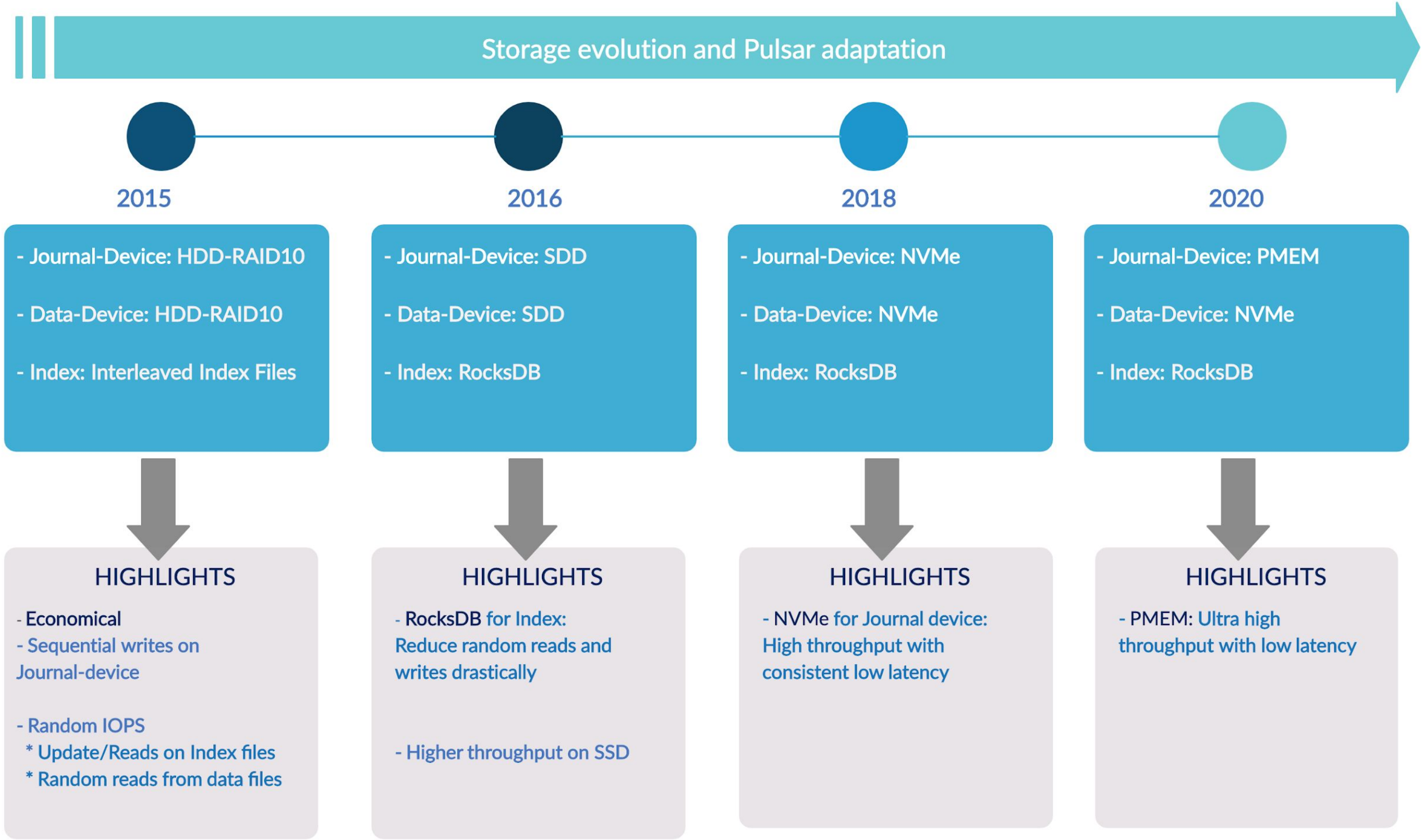
verizon media

BookKeeper IO Isolation



verizon media

Pulsar Journey



Storage evolution and Pulsar adaptation

2015

- Journal-Device: HDD-RAID10
- Data-Device: HDD-RAID10
- Index: Interleaved Index Files

HIGHLIGHTS

- Economical
- Sequential writes on Journal-device

- Random IOPS
  * Update/Reads on Index files
  * Random reads from data files

2016

- Journal-Device: SDD
- Data-Device: SDD
- Index: RocksDB

HIGHLIGHTS

- RocksDB for Index: Reduce random reads and writes drastically

- Higher throughput on SSD

2018

- Journal-Device: NVMe
- Data-Device: NVMe
- Index: RocksDB

HIGHLIGHTS

- NVMe for Journal device: High throughput with consistent low latency

2020

- Journal-Device: PMEM
- Data-Device: NVMe
- Index: RocksDB

HIGHLIGHTS

- PMEM: Ultra high throughput with low latency

First Generation Storage - HDD

- JOURNAL-Device HDD with RAID10

- DATA-Device HDD with RAID10

- Index: Interleaved index files

- **HDD**
  - Fast low latency sequential writes on HDD with battery backed RAID controller
  - Random seek time is much longer for HDD
  - Economical

- **Journal Device**
  - Fast sequential writes

- **Ledger Device**
  - Sequential writes on single entry-log data file for multiple streams
  - Most of the IOPs is utilized for
    - Backlog draining (cold reads)
    - Reads and writes on Index files

Optimizing random IOs for Indexing

- **Index on interleaved file**
  - One index file for each topic
  - Random IO while updating index
  - Scaling number of topics increases random IOs and file handles

- **Index on Rocks DB**
  - LSM based **embedded** key-value store
  - Used as a library within bookie process; no additional operational efforts
  - Less write-amplification and better compression
  - Drastically reduces random IOPs for indexing
  - Small footprint (< 10 GB); mostly in RAM

Second Generation: SSD/NVMe

- JOURNAL-Device NVMe/SSD

- DATA-Device NVMe/SSD

- Index: RocksDB

SSD/NVMe

- SSD provides better performance for sequential and random I/O
- NVMe supports large command queue (64K) with parallel IO

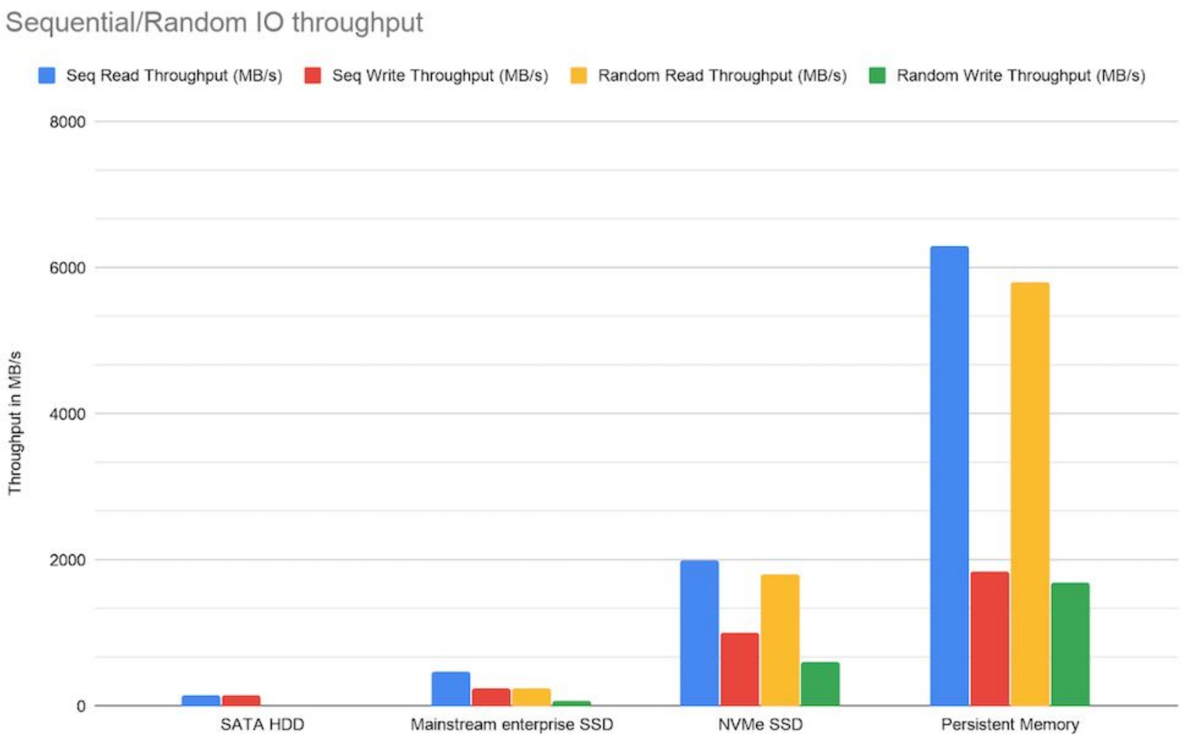
Journal Device

- Bookie can use multiple journal directories to utilize parallel write on NVMe
- Achieve 3x Pulsar throughput with low latency, compared to HDD

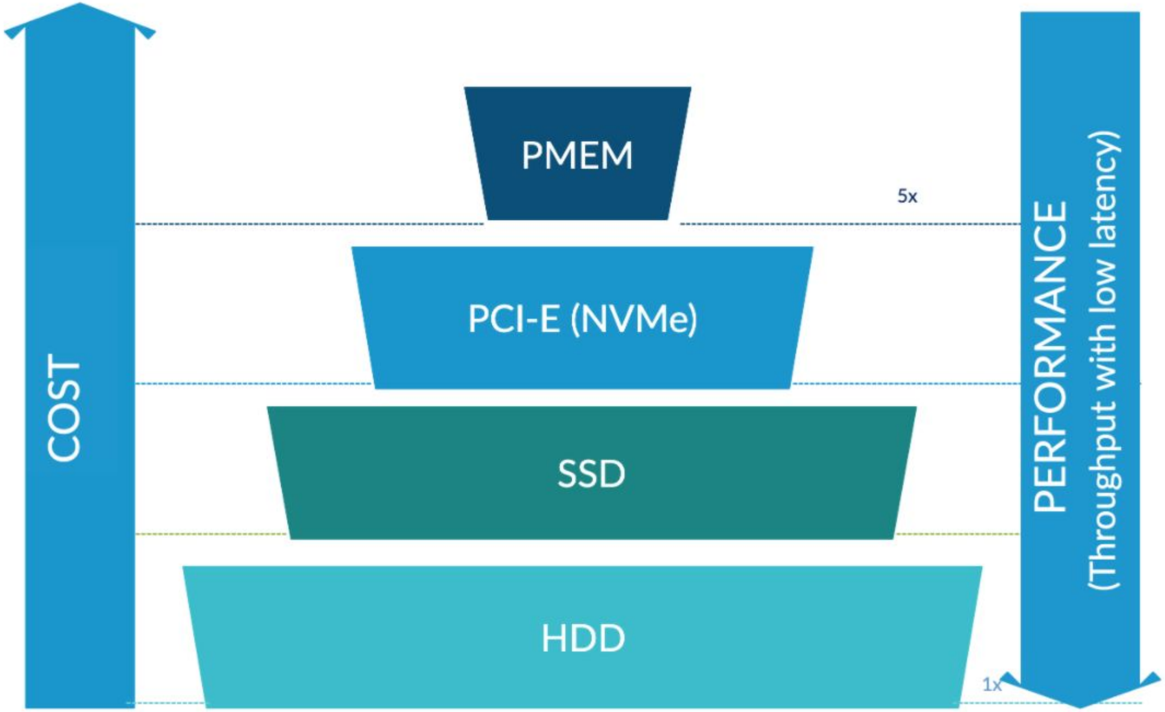
Ledger Device

- Significantly faster random reads than HDD
- Faster backlog draining while doing cold reads for multiple topics

Storage Device: Sequential Vs Random IO



Storage Device: Performance Vs Cost



verizon✓
media

Storage Evolution & Pulsar Adaptation: PMEM

PMEM

- Highest performing block storage device
- Ultra fast, super high throughput with consistent low latency
- Expensive; well suited as small device for WRITE intensive use cases

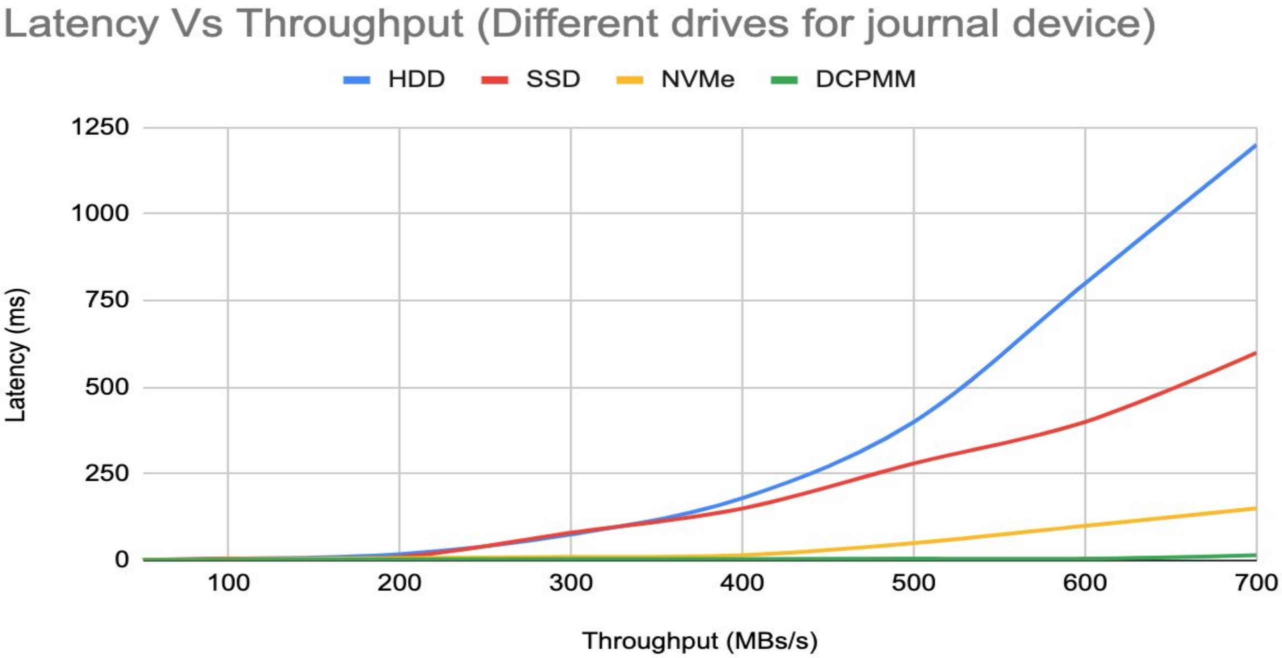
Journal Device

- WAL/journal is proven design in Databases
    - transactional storage and recovery
    - high throughput
- Write optimized append only structure
- Does not require much storage and keeps short lived transactional data
- Using PMEM for journal device
    - adds < 5% cost for each bookie
    - Increases Pulsar throughput 5x times, and with low publish latency

Pulsar Performance with Different BK-Journal Device

Performance configuration

- Enabled fsync on every published message
- Publish throughput with backlog draining
- SLA: 5ms (99%lie latency)
  - **HDD**: 120MB
  - **SSD**: 200MB
  - **NVMe**: 350MB
  - **PMEM**: 600MB



verizon media

Case-study: Migrate Kafka Use Case to Pulsar

- Cost and Throughput
    - Using PMEM for journal adds < 5% more cost per host but reduce overall cost and cluster footprints
    - Achieve 5x more throughput with 99%-ile @ <5ms write latency

- Cluster footprint
    - Kafka cluster : 33 Kafka Brokers
    - Pulsar cluster: 10 bookies and 16 brokers
        - Pulsar broker is a stateless component and costs 1/4x than bookie
    - Overall Pulsar cluster resources ½ of the Kafka cluster

Case-study: Migrate Kafka Use Case to Pulsar

USE CASES	APACHE PULSAR	APACHE KAFKA
Throughput with low latency	●	◐
Cost	◖	●
Geo-replication	●	◔
Queuing	●	◔
Committing messages	●	◔

Future

- Use PMDK API to access persistent memory
  - bypass the file system
  - better throughput
- Tiered Storage for historical data use cases
  - relaxed latency requirements
  - cheaper cost
  - Use cases
    - ML model training
    - audit, forensics

Thank you

joef@verizonmedia.com
rdhabalia@verizonmedia.com

verizon
media