hosted by Alibaba Group 阿里巴巴集团 APACHE HBASE

HBaseConAsia2018

Gehua New Century Hotel Beijing, China

August 17,2018

HBase On Persistent Memory

Anoop Sam John, Ramkrishna S Vasudevan

intel

Content

hosted by Alibaba Group 阿里巴巴集团 APACHE HBASE

01 HBase Present Model

02 Region Replica

03 Persistent Memory Technology

04 HBase On Persistent Memory

05 Performance Numbers

Apache HBase Present Model

❖ Accumulate data in Memory

- ○ Sorted Map
- ○ Cell data bytes in Local Allocation Buffers (LABs) in DRAM
- ○ LABs with size 2 MB in RAM.

❖ Also write to Write Ahead Log (WAL)

- ○ Cell data in Volatile RAM.
- ○ To recover from server crash
- ○ HDFS interaction adding more latency

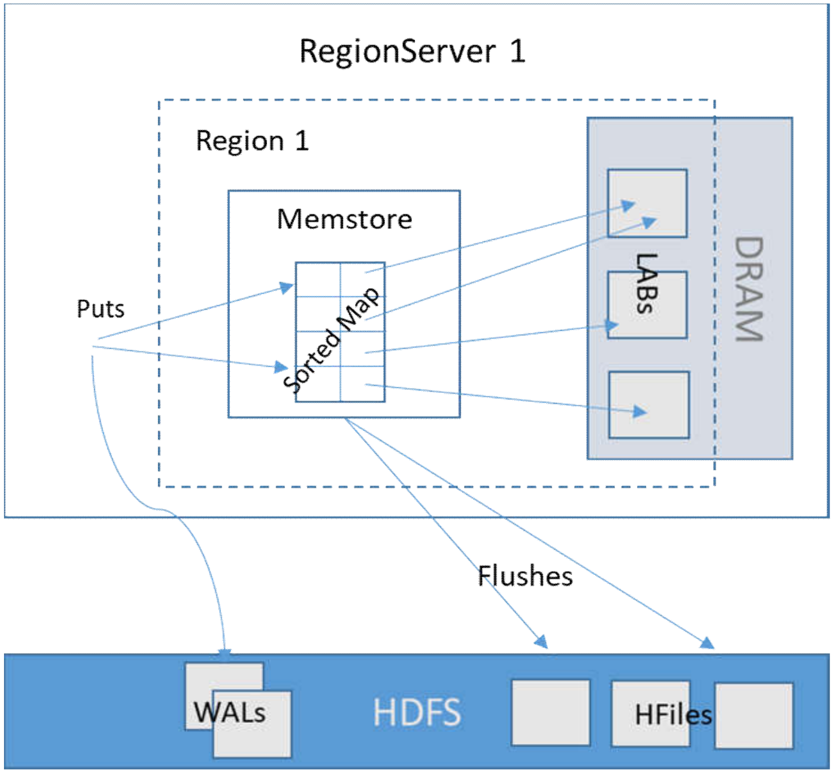
  hsync vs hflush - HBASE-19024

❖ Flushes as files to HDFS on reaching memstores size

- ○ 128 MB default flush size

❖ Replay WAL on server crash

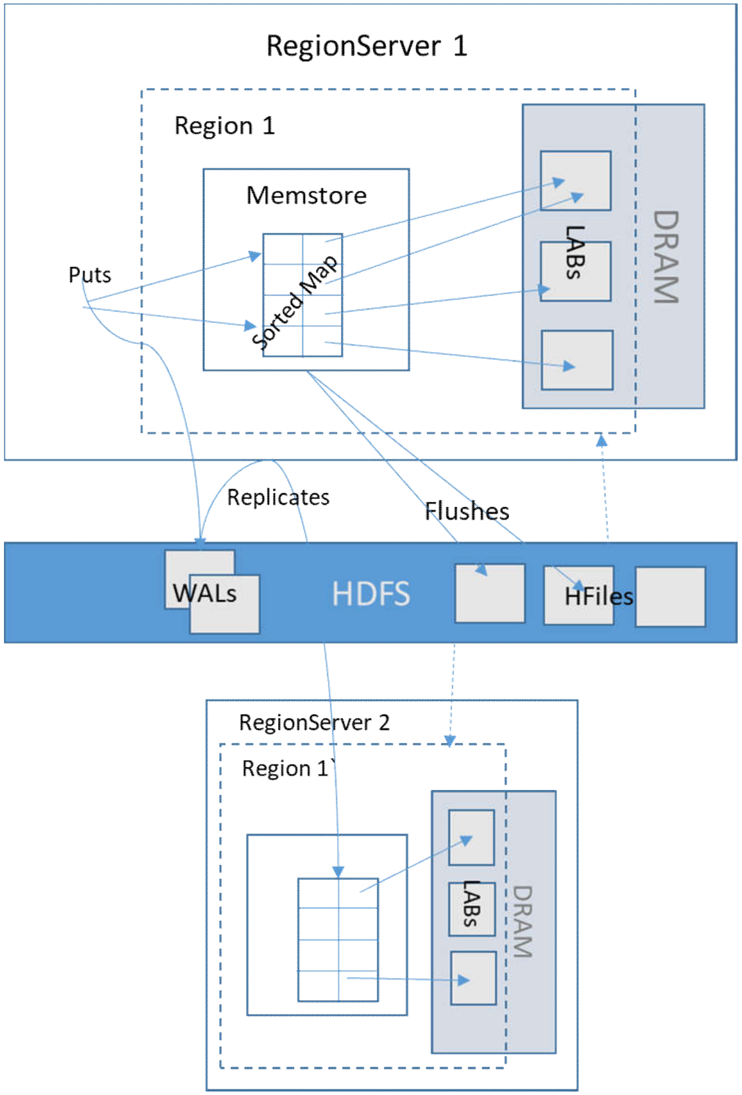
- ○ Data unavailable till replay completes
- ○ Large Mean Time To Recover (MTTR)
- ○ Takes several minutes on large cluster *(Complaint from many users like Alibaba – HBaseConAsia , Huawei)*



Apache HBase Present Model

Region Replica for better availability (?)

- ❖ Read only replica regions on other RSs
  - ○ Refer to same HFiles in HDFS
  - ○ Memstore data also can be replicated
    - ○ Using WAL read replication path
    - ○ Eventual consistency
  - ○ Can read from replica regions when primary is down
    - ○ No strong consistency guarantees.
    - ○ Only when Scan/Get says TIMELINE Consistency, replica read happens (not by default)
  - ○ Better availability but only for selected use cases !!! – No strong Consistency



Persistent Memory Technology

Persistent Memory

- Get back data even after power cycles
- Accessed using memory APIs
- Processor load and store instructions

Intel Apache Pass (AEP)

- Persistent Memory (pmem)
- Big, affordable and persistent
- Accessed like volatile memory, using processor load and store instructions

3DXPoint

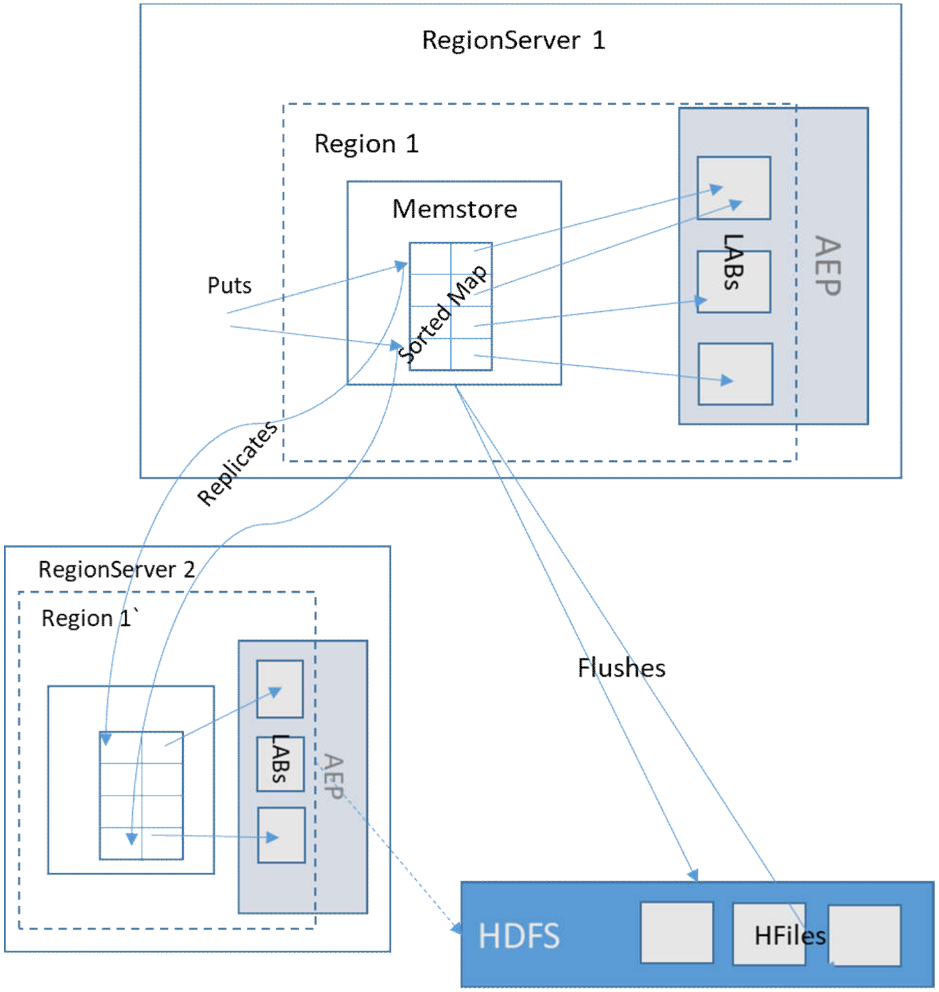
- New NVM technology
- Stackable cross-gridded data access array

Library

- NVML (Now called PMDK)
- Java wrappers around it
    - Apache Mnemonic – (used for PoC)
    - *https://github.com/pmem/pcj*

Apache HBase On Persistent Memory

- ❖ Accumulate data in Memory
  - ○ Sorted Map
  - ○ Cell data bytes in Local Allocation Buffers with size 2 MB
- ❖ Region Replica in other servers
  - ○ Replica regions feature already in place.
  - ○ Synchronous replication to replica regions
- ❖ No need to write to Write Ahead Log (WAL)
  - ○ Cell data in non volatile AEP.
- ❖ Server crash
  - ○ Fast switch to replica regions
  - ○ Consistent data in replica regions
  - ○ Full cluster down – Data in non volatile area. Fast way to recreate in memory Map.
- ❖ *HBASE-20003*



Apache HBase On Persistent Memory

❖ More memory size available - DRAM 100s of GBs. AEP even more

❖ More and More Data in Memory

○ Large memstore size and global memstore size

- More data in memory
- Less flushes and compactions = Less IO
- Subsequent reads can get data from memory mostly (?)
- More memstores size => More Java heap size => Larger GC pause issues.
- More cell entries to CSLM. More compares for ordering => Lower Throughput
- Server down - more data in live WAL files for replay => Higher MTTR

○ More Java heap size –> Off heap writes using Off heap memstores

○ More cell entries to CSLM –> Compacting Memstore work by Yahoo , New faster CSLM implementation by Alibaba

○ More data in live WALs –> HBase on AEP with no WALs. Persistent memstores LABs. Instant rebuild of memstores CSLM.

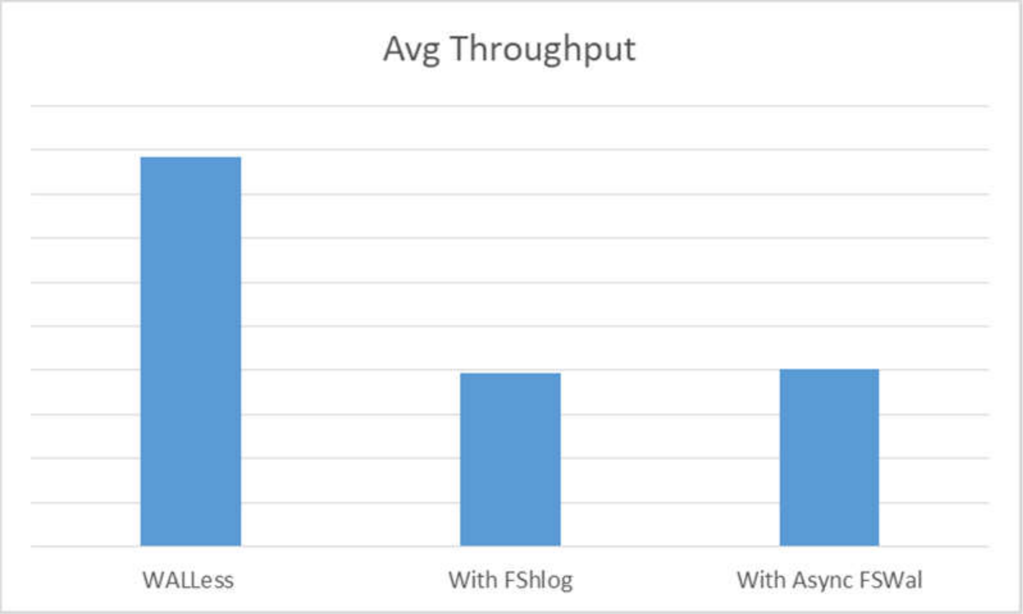
Performance Results

❖ PerformanceEvaluation Tool

- ○ Write only workload
- ○ 4 Node cluster
- ○ 100 client thread
- ○ 250 GB Total data
- ○ Single column per row
- ○ WAL – 3 Replicas (HDFS replicas)
- ○ WALLess – Primary and 2 replica regions

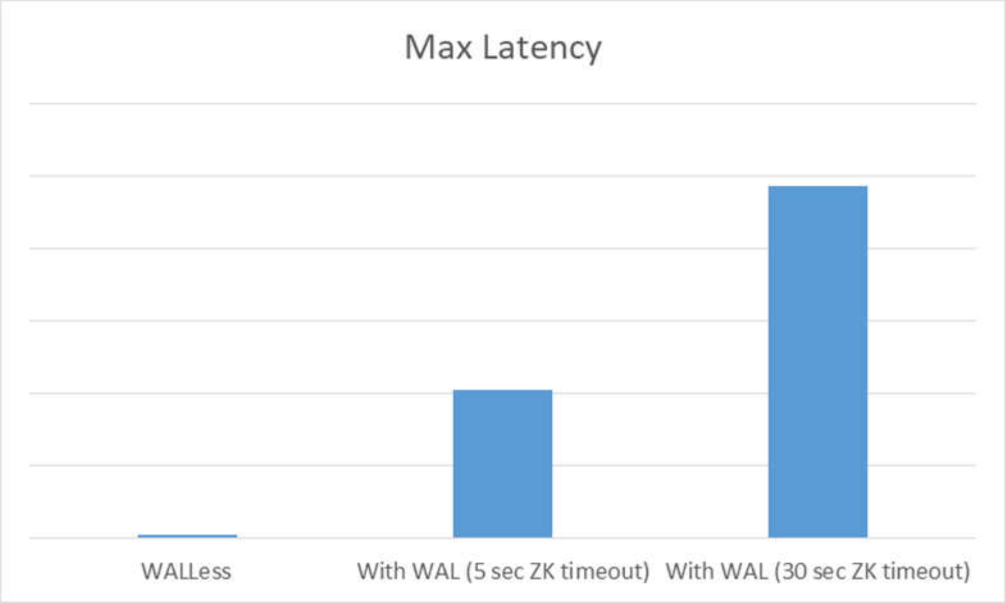
Average throughput is > 2x compared to With WAL cases.

Latency is consistent through out for the WALLess case. In WAL case the latency varies as the amount of data to ‘sync’ increases. (Here again with ‘fsync’ latency is more than ‘hflush’).



Performance Results

- ❖ PerformanceEvaluation Tool
  - ○ Random Reads
  - ○ 4 Node cluster
  - ○ 100 client thread
  - ○ 5 secs/ 30 sec ZK session time outs
  - ○ One RS node crash in between PE run
  - ○
- ❖ Max latency is 44x larger with WAL (ZK session time out = 5 sec)
- ❖ Max latency is 104x larger with WAL (ZK session time out = 30 sec)



Apache HBase On Persistent Memory

Project Status

- ❖ HBASE-20003
- ❖ *https://docs.google.com/document/d/1sYJS9lMZa_EMhTTOJ7y_KzVUjXdBgdKdPWmsN5p2lH0/*
- ❖ Write path, read path changes done in PoC
- ❖ Pending – WAL based features – Inter cluster replication, backup
  - ○ Similar issue as that in HBASE-20951 (Ratis LogService backed WALs)
  - ○ Work with this project - HBase improvements for Cloud
- ❖ Testing – Full cluster restart/ Rolling restart scenarios
- ❖ Load balancer , AM stabilization.
  - ○ Specially with Region replicas. Many bugs. Solving…
- ❖ http://pmem.io/

Thanks