RocksDB Festival

Personal Research

Supported by IITP, StarLab.

July 5, 2021

Hoijn Shin, Jongmoo Choi
choijm@dankook.ac.kr
http://embedded.dankook.ac.kr/~choijm

RocksDB Festival: Members

- 일정: 7월 5일 (월) 오후 1시 첫번째 미팅
- 장소: 미디어센터 509호
- 참여자
  - ✓ Student
    - 송인호, 한예진, 허진, 이정원, 김산, 강정현, 최민국, 조광훈, 박경미, 김정민, 황예진, 고산하, 김민준, 김한얼, 이빈, 이규열, 이성준
  - ✓ Assistant
    - 신호진, 이성현
  - ✓ Professor
    - 최종무 교수님, 유시환 교수님
  - ✓ Reference Site
    - https://github.com/DKU-StarLab/RocksDB_Festival.git
    - https://github.com/DKU-StarLab/RocksDB_Explorer.git

RocksDB Festival: Calendar

- Upcoming Events
  - ✓ 7.5 : Install RocksDB and explanation, Team research
  - ✓ 7.12 : Explain RocksDB architecture and operation, Team activation
  - ✓ 7.19 : Online class and discussion with Team
  - ✓ 7.26 ~ 8.2, 9, 16, 23 : Team discussion and share progress

  - ✓ Final Goal : New idea and Paper submission (KSC 2021 …)
  - ✓ Each team announces at least 3 times

RocksDB Festival: Personal Research

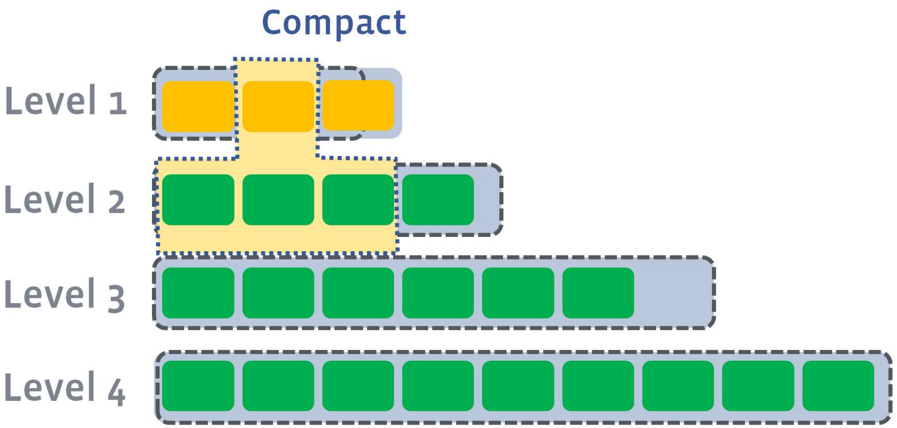
- Main Topic
  - ✓ 1) Compaction Related
  - ✓ 2) WAL (Write-Ahead-Log) Related
  - ✓ 3) Read Optimization Related
  - ✓ 4) Memtable/SSTable Related
  - ✓ 5) Key/Value Related
  - ✓ 6) Parallelism Related
  - ✓ 7) Interface Related
  - ✓ 8) Hybrid Storage Related
  - ✓ 9) Others

RocksDB Festival: Personal Research

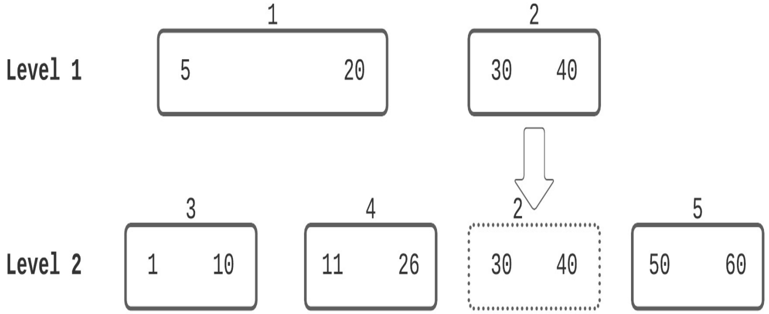
- Common Research Area
  - ✓ RocksDB: Establishment of experimental environment
  - ✓ RocksDB db_bench practice and interpret results
  - ✓ RocksDB Wiki

RocksDB Festival : Personal Research

- Personal Research
  - ✓ 1) Compaction Related
    - Classic Leveled
    - Leveled-N
    - Tiered
    - Tiered+Leveled
    - FIFO



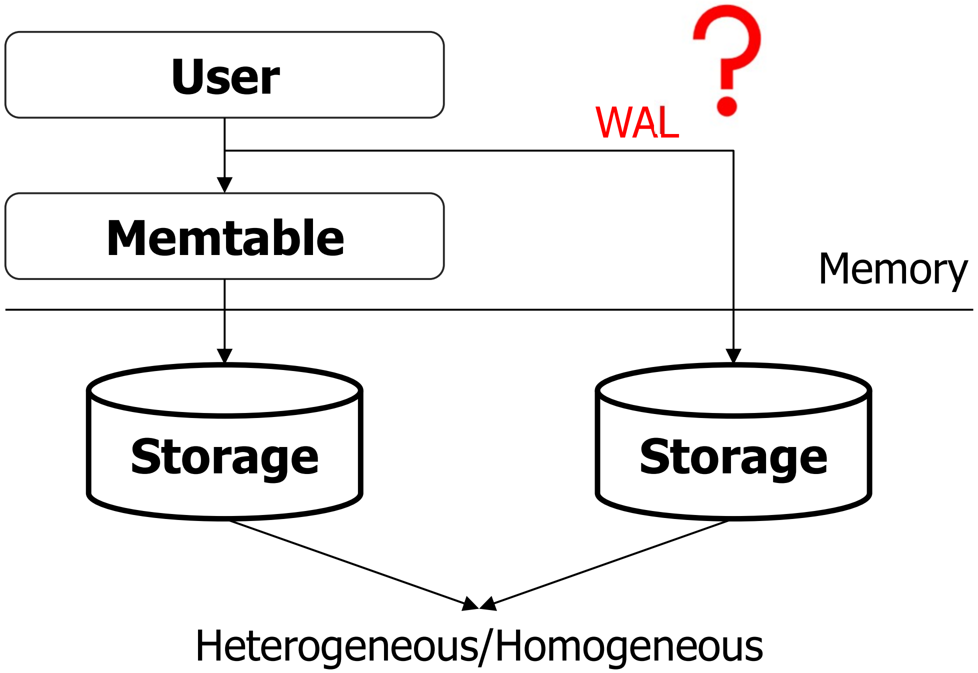
Leveled Compaction



Trivial Move

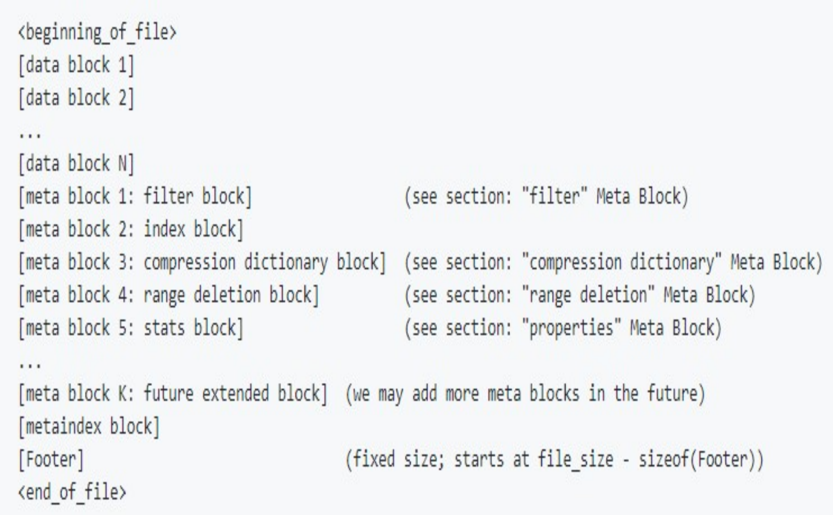
RocksDB Festival : Personal Research

- Personal Research
  - ✓ 2) WAL Related
    - With/Without WAL
    - WAL in separated SSDs
    - With/Without buffering (fsync, fdatasync)



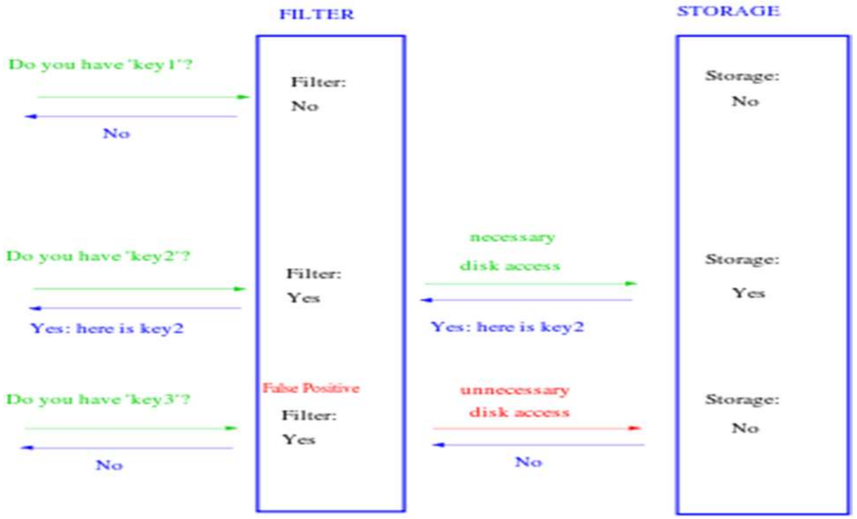
RocksDB Festival : Personal Research

- Personal Research
  - ✓ 3) Read optimization Related
    - Block-based Table format
    - PlainTable format
    - CuckooTable format
    - Index Block format
    - Bloom Filter



```
<beginning_of_file>
[data block 1]
[data block 2]
...
[data block N]
[meta block 1: filter block]                  (see section: "filter" Meta Block)
[meta block 2: index block]
[meta block 3: compression dictionary block]  (see section: "compression dictionary" Meta Block)
[meta block 4: range deletion block]          (see section: "range deletion" Meta Block)
[meta block 5: stats block]                   (see section: "properties" Meta Block)
...
[meta block K: future extended block]  (we may add more meta blocks in the future)
[metaindex block]
[Footer]                               (fixed size; starts at file_size - sizeof(Footer))
<end_of_file>
```

File format

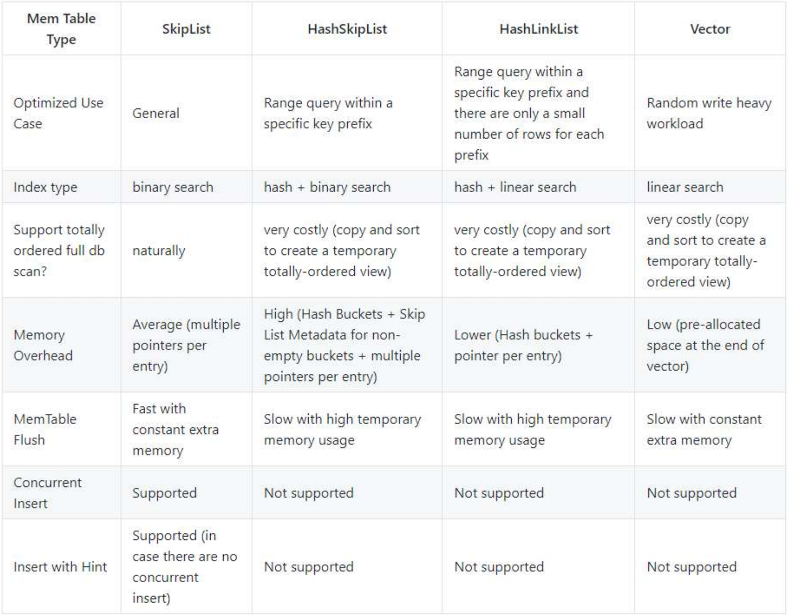


Bloom filter

RocksDB Festival : Personal Research

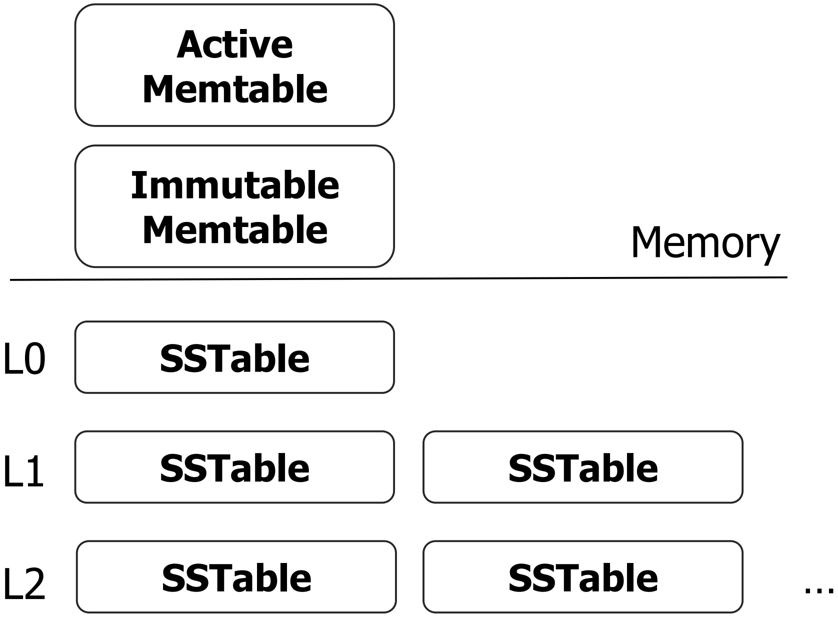
Personal Research

- ✓ 4) Memtable/SSTable Related
  - Control Memtable configuration
  - Control SSTable configuration



Mem Table Type	SkipList	HashSkipList	HashLinkList	Vector
Optimized Use Case	General	Range query within a specific key prefix	Range query within a specific key prefix and there are only a small number of rows for each prefix	Random write heavy workload
Index type	binary search	hash + binary search	hash + linear search	linear search
Support totally ordered full db scan?	naturally	very costly (copy and sort to create a temporary totally-ordered view)	very costly (copy and sort to create a temporary totally-ordered view)	very costly (copy and sort to create a temporary totally-ordered view)
Memory Overhead	Average (multiple pointers per entry)	High (Hash Buckets + Skip List Metadata for non-empty buckets + multiple pointers per entry)	Lower (Hash buckets + pointer per entry)	Low (pre-allocated space at the end of vector)
MemTable Flush	Fast with constant extra memory	Slow with high temporary memory usage	Slow with high temporary memory usage	Slow with constant extra memory
Concurrent Insert	Supported	Not supported	Not supported	Not supported
Insert with Hint	Supported (in case there are no concurrent insert)	Not supported	Not supported	Not supported

Compare with Memtable Type



Simple Architecture

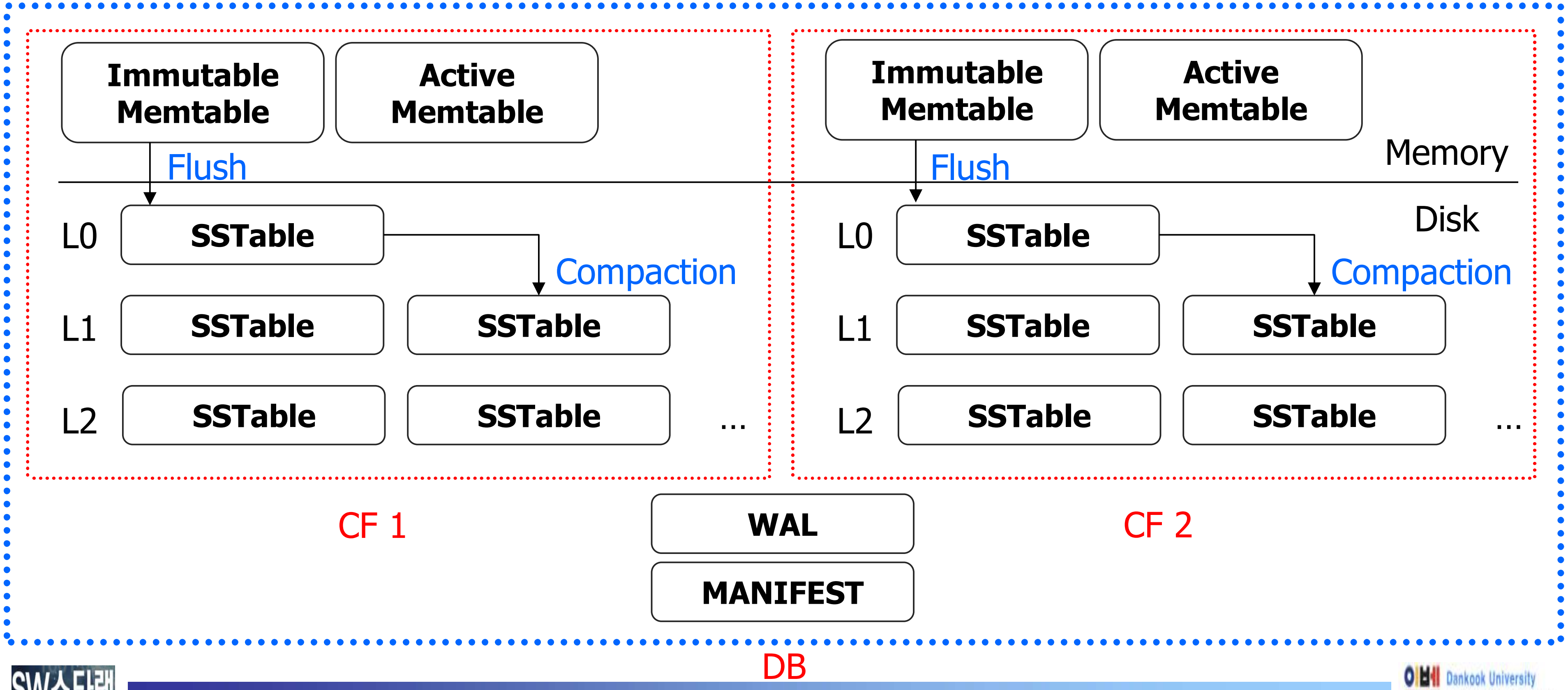
RocksDB Festival : Personal Research

- Personal Research
  - ✓ 5) Key/Value Related
    - Key distribution : Sequential, Random
    - Existing Key trace analysis

    - A large scale analysis of hundreds of in-memory cache clusters at Twitter, OSDI 20
    - https://github.com/twitter/cache-trace.git
    - From WiscKey to Bourbon: A Learned Index for Log-Structured Merge Trees, OSDI 20
    - https://registry.opendata.aws/

RocksDB Festival : Personal Research

- Personal Research
  - ✓ 6) Parallelism Related
    - Number of user threads
    - Number of background threads (flush/compaction)
    - How to make use of column family (or graph DB) of two instance



RocksDB Festival : Personal Research

- Personal Research
  - ✓ 7) Interface Related
    - Make Proxy application (ex. Smart factory, IoT log, SNS log …)
    - MyRocks(RocksDB+MySQL), MongoDB(Document DB)
  - ✓ 8) Hybird Storage Related
    - Optane + SSD
    - SSD + HDD
    - Heterogeneous
  - ✓ 9) Others
    - Adaptive Scheme
    - Layout (ex. Wisckey – BlobDB)

RocksDB Festival: Personal Research

- Main Topic
  - ✓ 1) Compaction Related
  - ✓ 2) WAL (Write-Ahead-Log) Related
  - ✓ 3) Read Optimization Related
  - ✓ 4) Memtable/SSTable Related
  - ✓ 5) Key/Value Related
  - ✓ 6) Parallelism Related
  - ✓ 7) Interface Related
  - ✓ 8) Hybrid Storage Related
  - ✓ 9) Others

Discussion