Clip slide

RocksDB Compaction

Embedded Key-Value Store for Flash and RAM

안미진

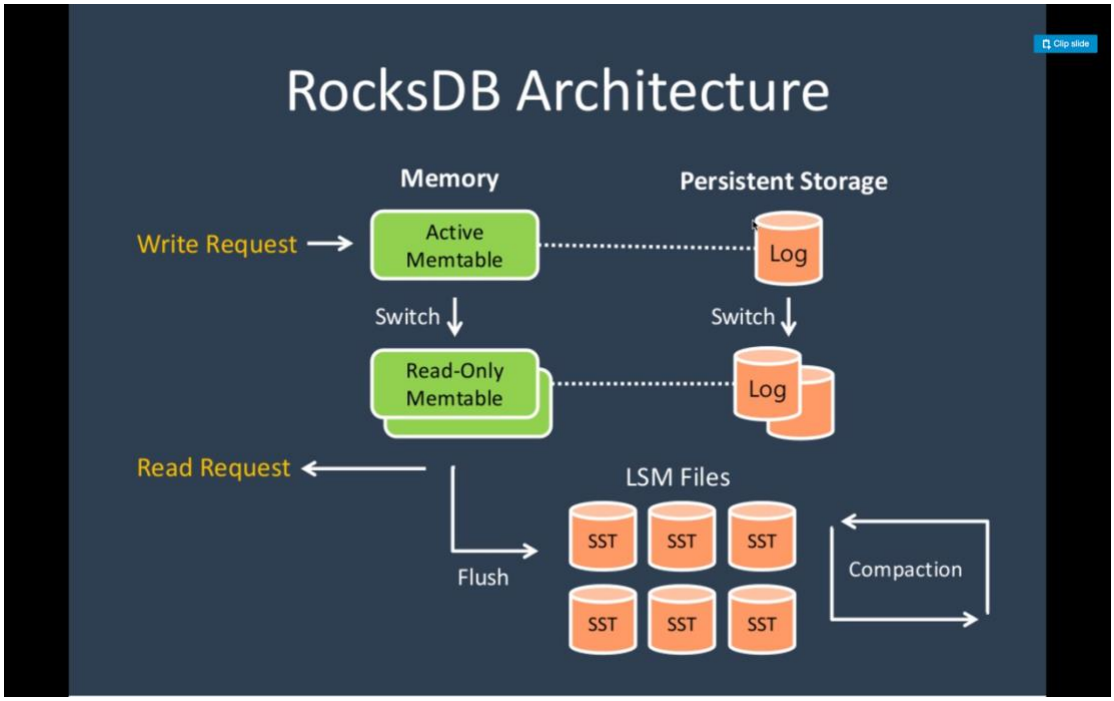
RocksDB

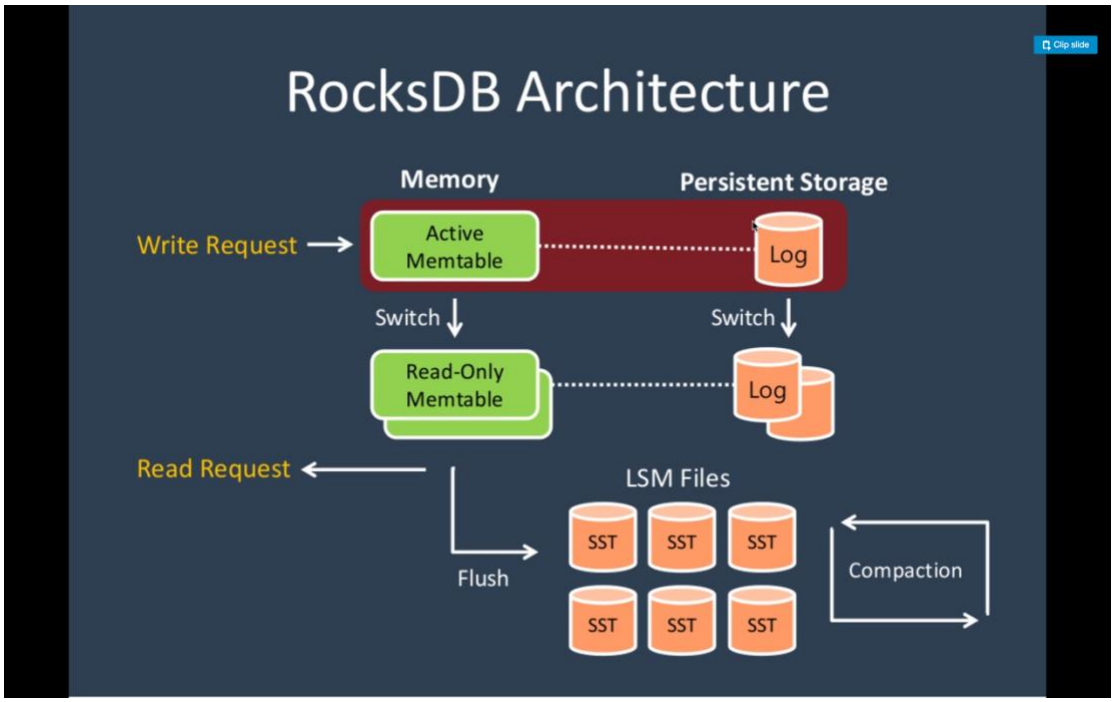
Clip slide

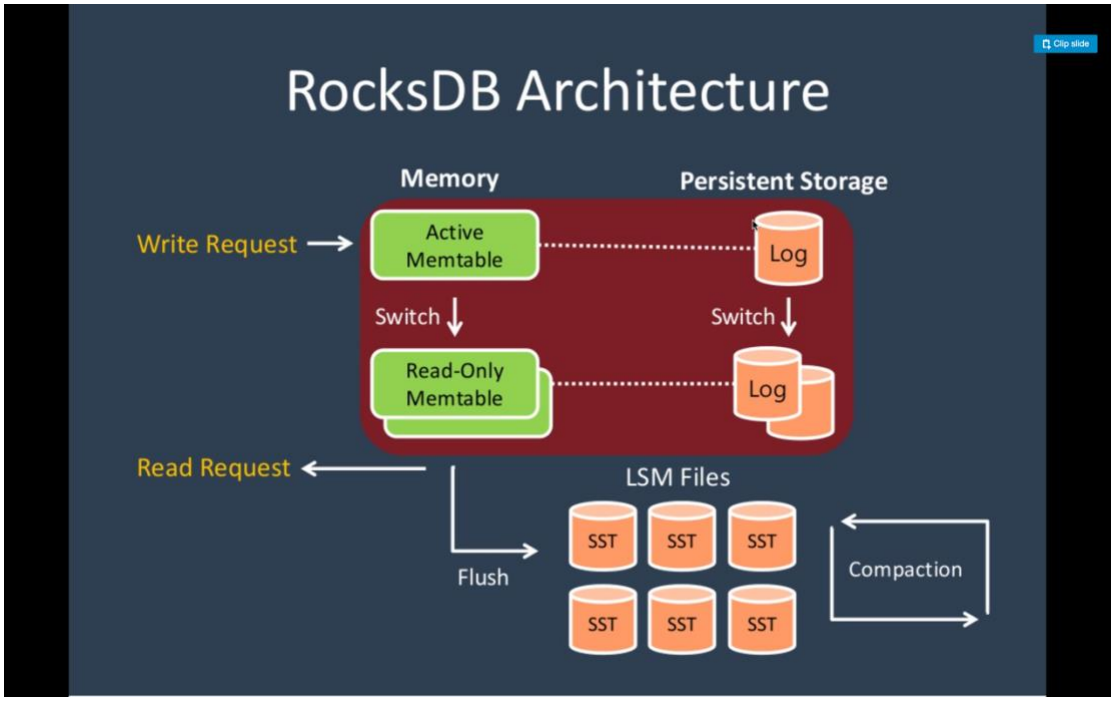
Contents

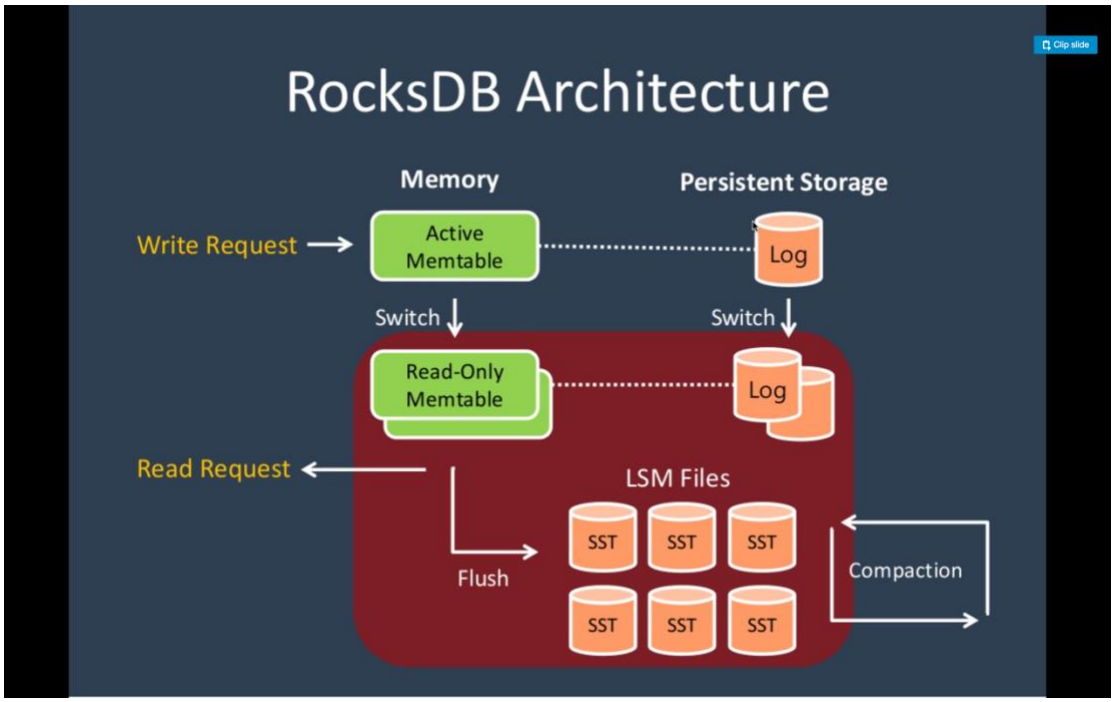
Overview

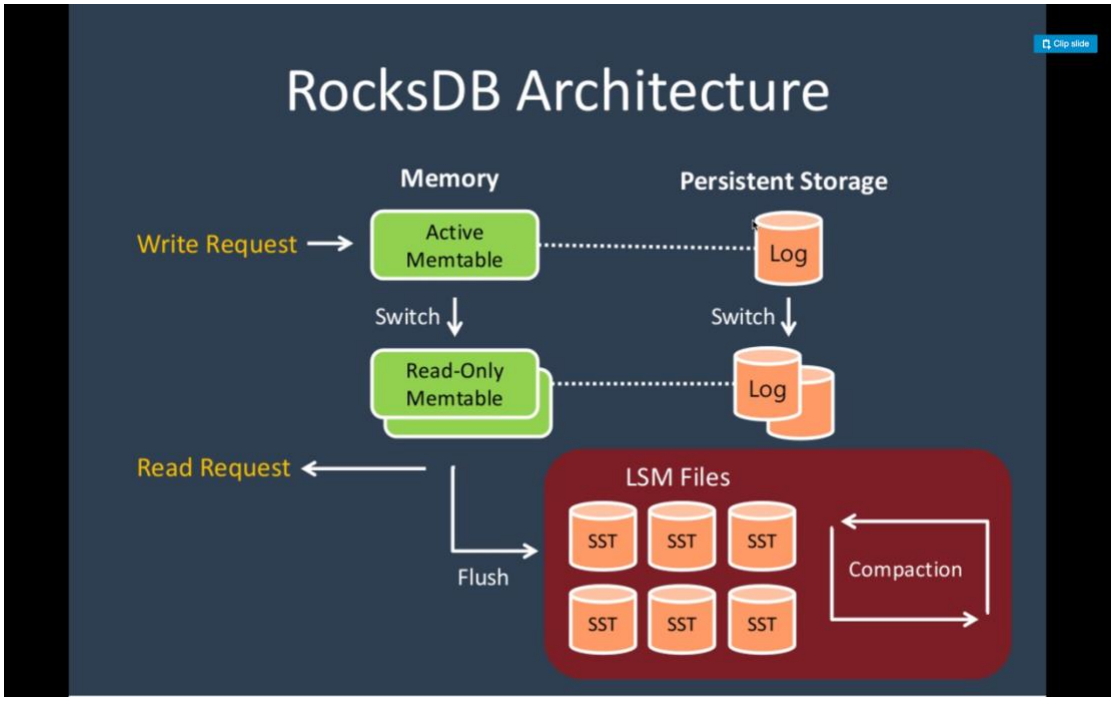
1. RocksDB Architecture
2. Level Style Compaction
3. Universal Style Compaction
4. RocksDB Compaction

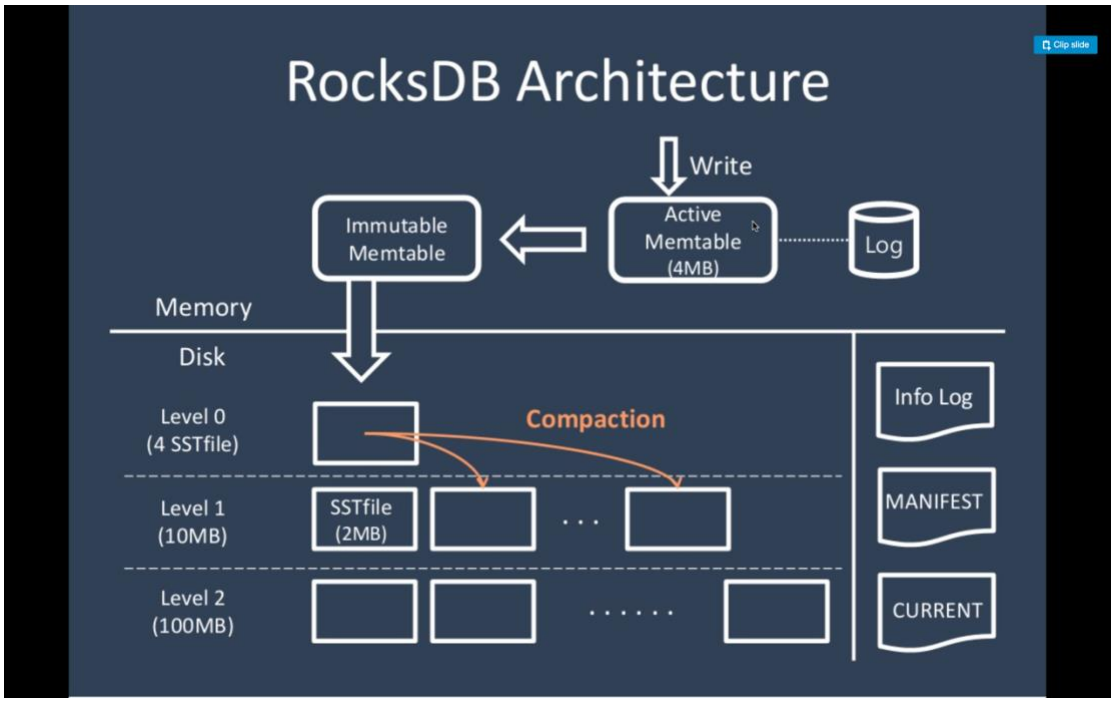

Clip slide
RocksDB Architecture
Memory
Persistent Storage
Write Request
Active Memtable
Log
Switch
Switch
Read-Only Memtable
Log
Read Request
LSM Files
SST
SST
SST
SST
SST
SST
Flush
Compaction


Clip slide
RocksDB Architecture
Memory
Persistent Storage
Write Request
Active Memtable
Log
Switch
Switch
Read-Only Memtable
Log
Read Request
LSM Files
SST
SST
SST
SST
SST
SST
Flush
Compaction


Clip slide
RocksDB Architecture
Memory
Persistent Storage
Write Request
Active Memtable
Log
Switch
Switch
Read-Only Memtable
Log
Read Request
LSM Files
SST
SST
SST
SST
SST
SST
Flush
Compaction


Clip slide
RocksDB Architecture
Memory
Persistent Storage
Write Request
Active Memtable
Log
Switch
Switch
Read-Only Memtable
Log
Read Request
LSM Files
SST
SST
SST
SST
SST
SST
Flush
Compaction


Clip slide
RocksDB Architecture
Memory
Persistent Storage
Write Request
Active Memtable
Log
Switch
Switch
Read-Only Memtable
Log
Read Request
LSM Files
SST
SST
SST
SST
SST
SST
Compaction
Flush


Clip slide
RocksDB Architecture
Write
Immutable Memtable
Active Memtable (4MB)
Log
Memory
Disk
Level 0 (4 SSTfile)
Compaction
Level 1 (10MB)
SSTfile (2MB)
Level 2 (100MB)
Info Log
MANIFEST
CURRENT

RocksDB Compaction

Multi-threaded compactions

- Background Multi-thread
  → *periodically* do the "compaction"
  → parallel compactions on different parts of the database can occur simultaneously
- **Merge** SSTfiles to a bigger SSTfile
- **Remove** multiple copies of the same key
  – Duplicate or overwritten keys
- Process **deletions** of keys
- Supports two different styles of compaction
  – Tunable compaction to trade-off

Level Style Compaction

Compaction options

- level0_file_num_compaction_trigger
  - Number of files to trigger level0 compaction
  - Default : 1
    Ex) candidate files size < the next file's size (1% smaller)
    → include next file into this candidate set
- Level0_file_
  - The minimum number of files in a single compaction
  - Default : 2
- max_merge_width
  - The maximum number of files in a single compaction
  - Default : UINT_MAX

1. Level Style Compaction

Inherited from LevelDB

- RocksDB default compaction style
- Stores data in **multiple levels** in the database
- More **recent** data → L0
  The **oldest** data → Lmax
- Files in *L0*
  - overlapping keys, sorted by flush time

  Files in *L1 and higher*
  - non-overlapping keys, sorted by key
- Each level is 10 times larger than the previous one

Level Style Compaction

Compaction process

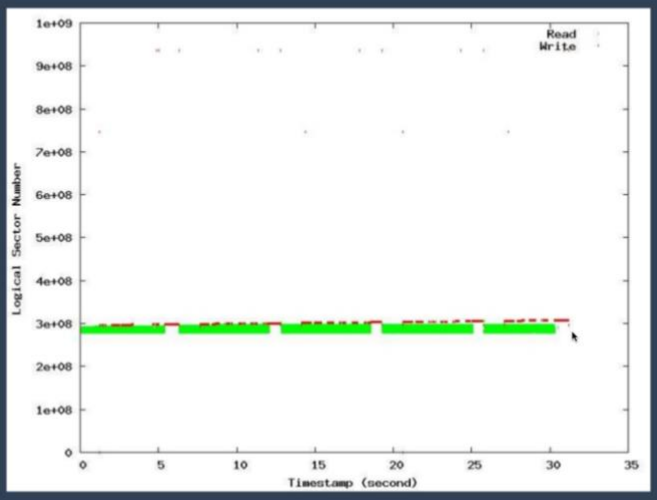
① Pick one file from level N
② Compact it with all its overlapping files from level N+1
③ Replace them with new files in level N+1

Level 0 → Level 1 Compaction

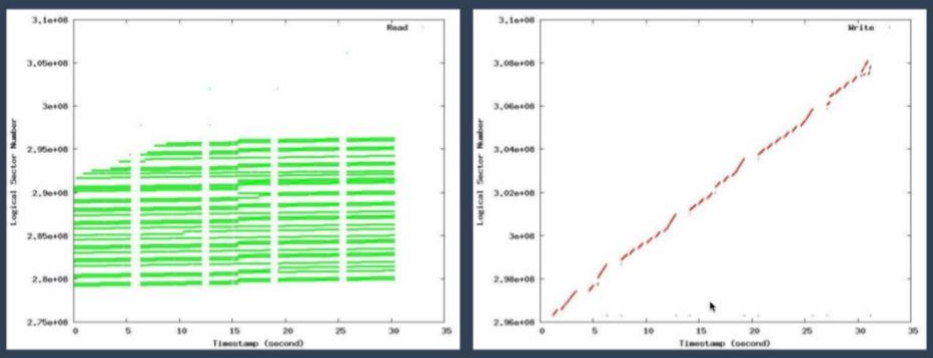
Tricky Compaction

- Level 0 → overlapping keys
- Compaction includes all files from L1
- All files from L1 are compacted with L0
- L0 → L1 compaction completion
  - ➡ L1 → L2 compaction start
- **Single thread** compaction → not good throughput
- Solution : Making the size of L0 similar to size of L1

Level Style Compaction



Level Style Compaction



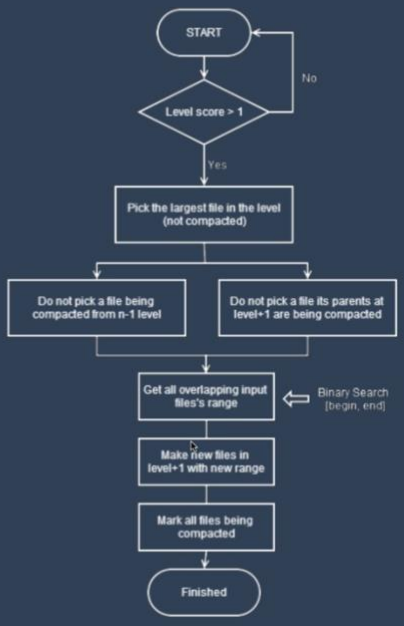
Level Style

Flowchart

· Level score = $\frac{current\ level\ size}{\text{max level size}}$

· max file size
= target_file_size_base * target_file_size_multiplier
(Default=2MB) (Default=1)

· Overlapping range search
: Binary Search



Clip slide

Level Style Compaction

- Read : **128**KB / Write : **512**KB

Clip slide

2. Universal Style Compaction

- For write-heavy workloads

  → Level Style Compaction may be bottlenecked on disk throughput
- Stores all files in L0
- All files are arranged in **time order**
- Temporarily **increase size** amplification by a factor of two
- Intended to **decrease write amplification**
- But, **increase space amplification**

Clip slide

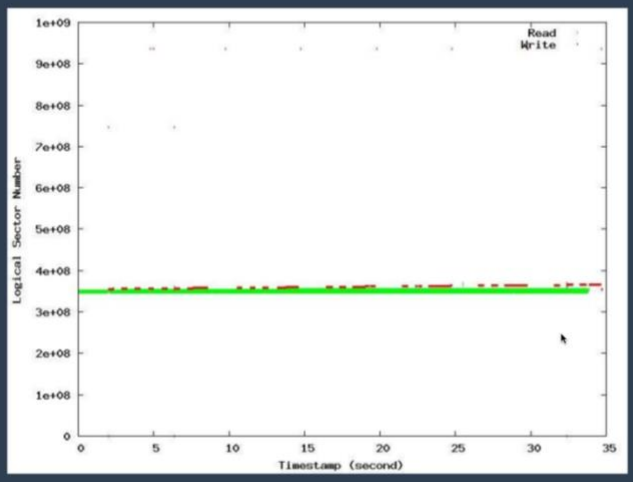
Universal Style Compaction

Compaction process

① Pick up a few files that are chronologically adjacent to one another
② Merge them
③ Replace them with a new file in level 0

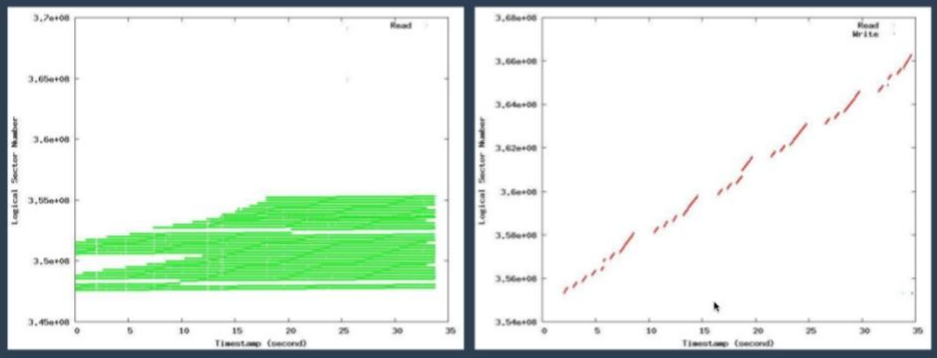
Clip slide

Universal Style Compaction



Clip slide

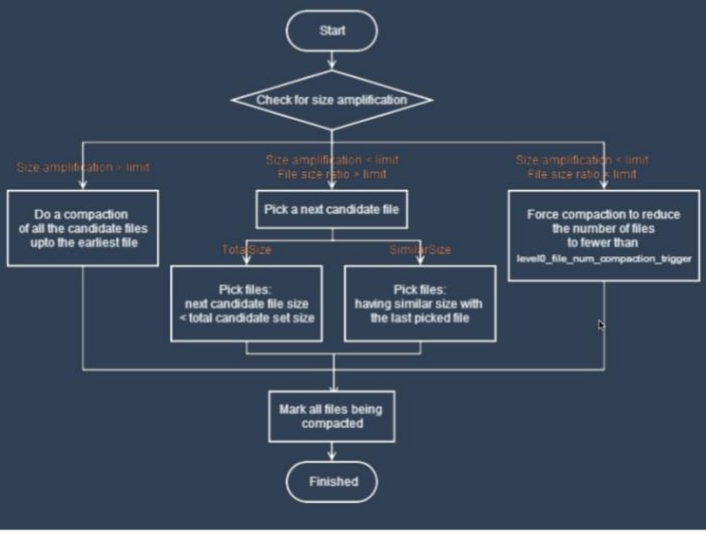
Universal Style Compaction



Clip slide

Universal Style Compaction

Flowchart



Universal Style Compaction

- Read : **128**KB / Write : **512**KB

Universal Style Compaction

Compaction options

- size_ratio
  - Percentage flexibility while comparing file size
  - Default : 1

    Ex) candidate set size < size of next file (1% smaller)

    → include next file in candidate set
- min_merge_width
  - The minimum number of files in a single compaction
  - Default : 2
- max_merge_width
  - The maximum number of files in a single compaction
  - Default : UINT_MAX

Universal Style Compaction

Compaction options

- max_size_amplification_percent
  - The amount of additional storage needed to store a single byte of data in the database
  - Controls the amount of space amplification in the database
  - Does not determine when calls to Put & Delete are stalled
  - Determines when compaction is done
  - Default : 200

Universal Style Compaction

Compaction options

- stop_style
  - The algorithm used to *stop picking files* into a single compaction run
  - kCompactionStopStyleSimilarSize
    → Pick files of similar size
  - kCompactionStopStyleTotalSize
    → total size of picked files > next files
  - Default : kCompactionStopStyleTotalSize