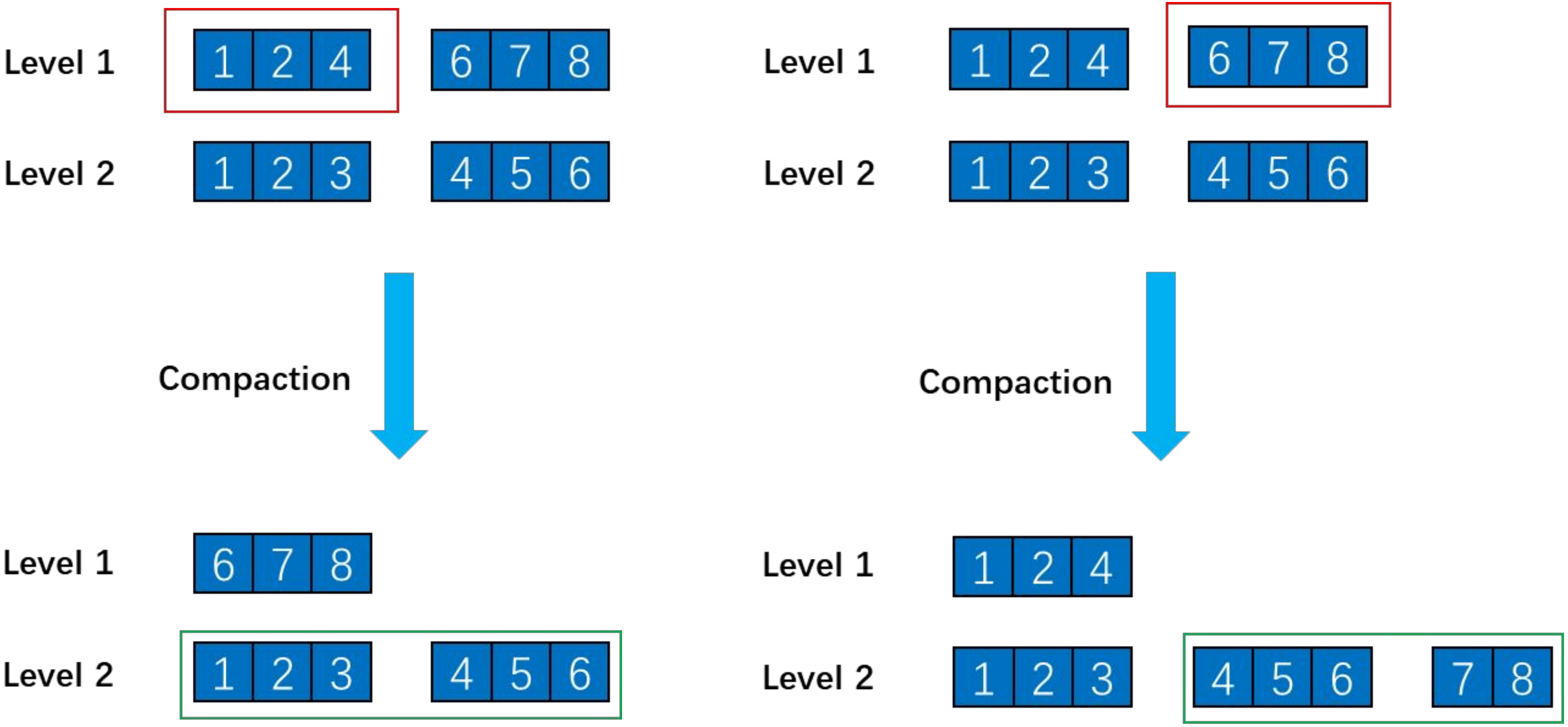
Exploring the Optimal Compaction Strategy for A Given Workload

Ran Wei, Chen Zhu, Peixu Xin

Problem: Minimizing Write Amplification



Level 1 | 1 2 4 | 6 7 8

Level 2 | 1 2 3 | 4 5 6

Compaction

Level 1 | 6 7 8

Level 2 | 1 2 3 | 4 5 6

Level 1 | 1 2 4 | 6 7 8

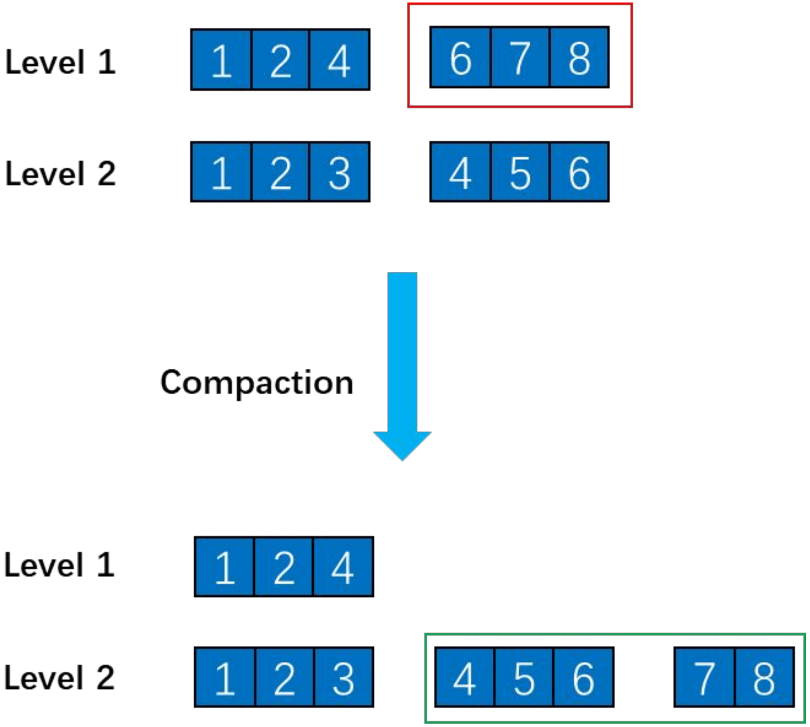
Level 2 | 1 2 3 | 4 5 6

Compaction

Level 1 | 1 2 4

Level 2 | 1 2 3 | 4 5 6 | 7 8

Baseline: Min Overlapping Ratio

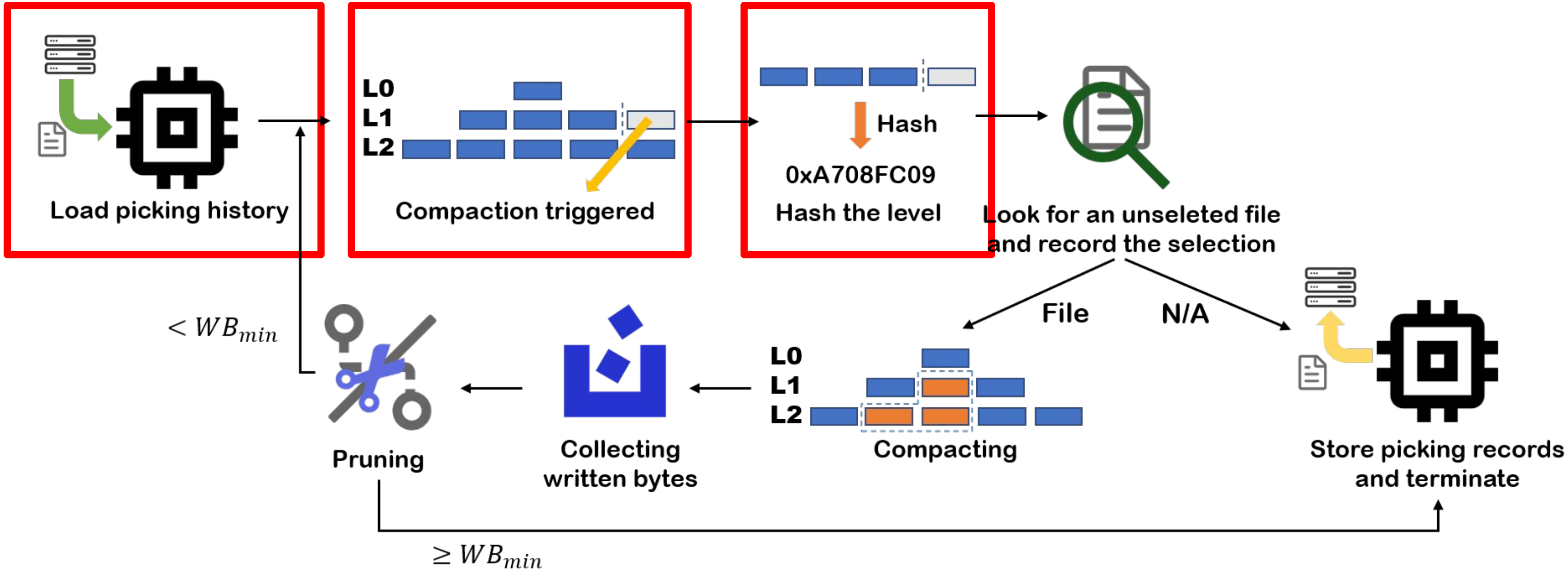


Local Minimum!

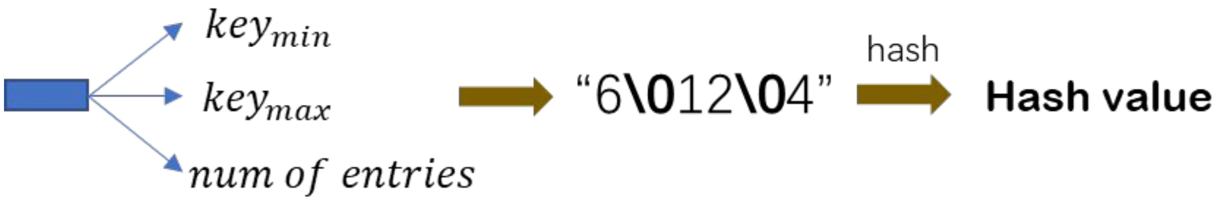
Find Optimal Compaction Strategy

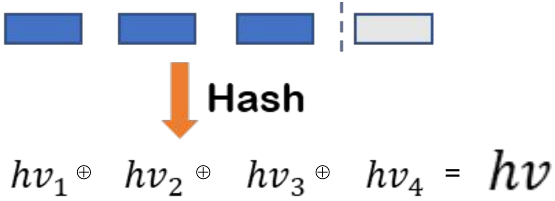
- Finding the minimum WA for a given workload. ★ **We are here!**
- Generalizing the file picking pattern.
- Building the strategy based on the generalized pattern.

Pipeline


Load picking history
L0
L1
L2
Compaction triggered
Hash
0xA708FC09
Hash the level
Look for an unseleted file
and record the selection
File
N/A
Store picking records
and terminate
L0
L1
L2
Compacting
Collecting
written bytes
Pruning
$< WB_{min}$
$\geq WB_{min}$

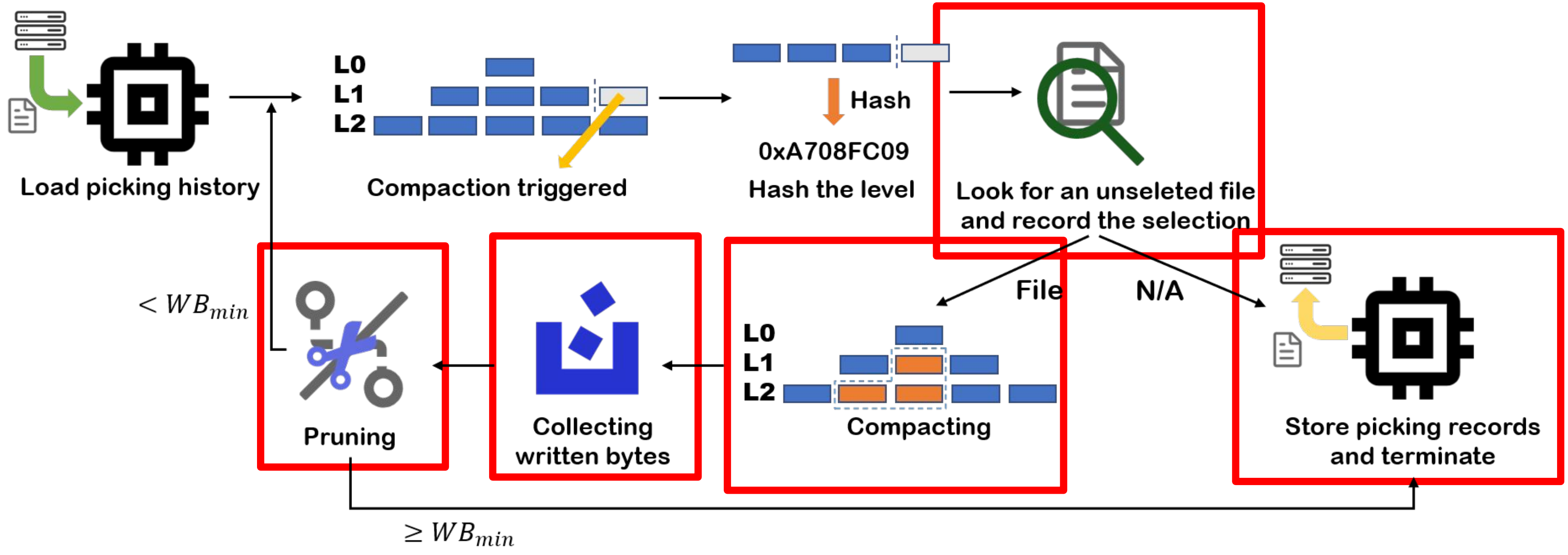
Hash a Level





$$hv_1 \oplus hv_2 \oplus hv_3 \oplus hv_4 = hv$$

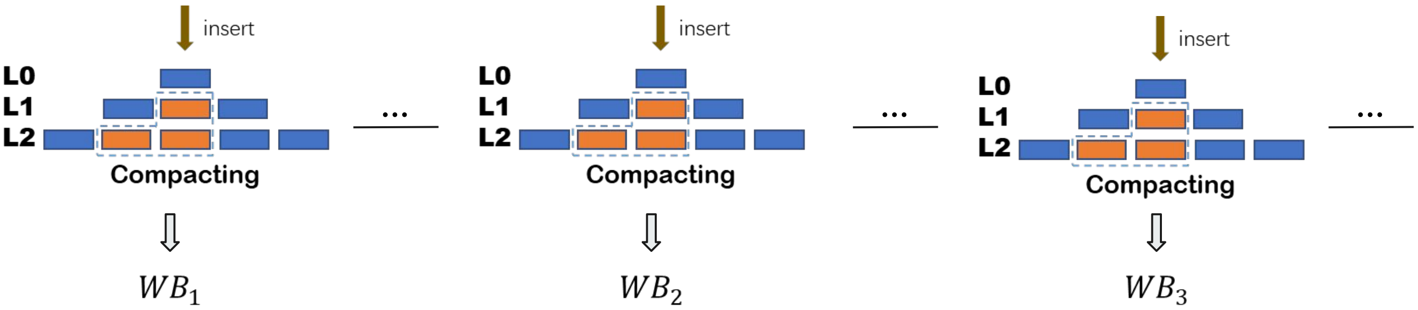
Pipeline



Load picking history
L0
L1
L2
Compaction triggered
Hash
0xA708FC09
Hash the level
Look for an unseleted file and record the selection
File
N/A
L0
L1
L2
Compacting
Collecting written bytes
Pruning
$< WB_{min}$
$\geq WB_{min}$
Store picking records and terminate

Pruning

Example: 1M inserts (8M bytes), WB_{min} = 20M

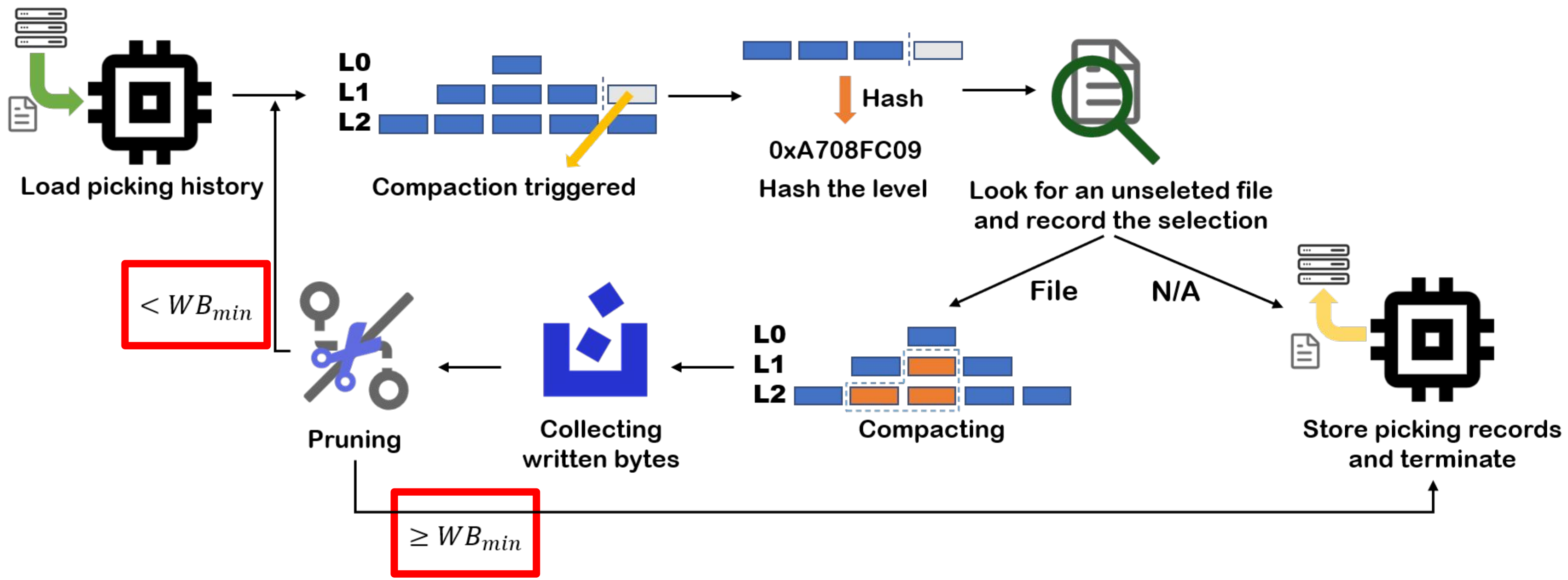


$WB_1 + WB_2 + WB_3$ = 18M

remaining bytes = 3M

18M + 3M = 21M > 20M

Pipeline


Load picking history
L0
L1
L2
Compaction triggered
Hash
0xA708FC09
Hash the level
Look for an unseleted file and record the selection
File
N/A
Store picking records and terminate
L0
L1
L2
Compacting
Collecting written bytes
Pruning
$< WB_{min}$
$\geq WB_{min}$

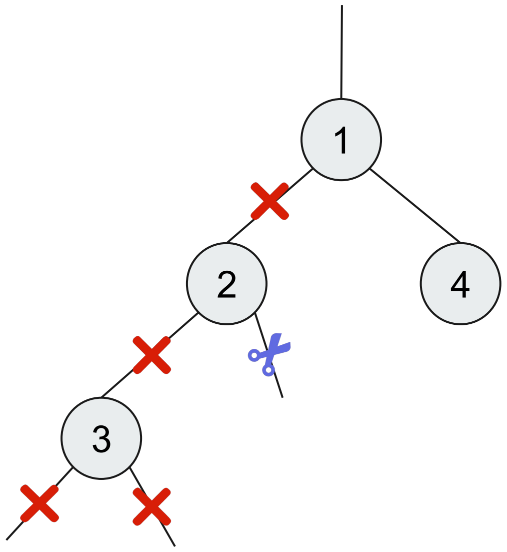
Find an unselected file

- Regard the enumeration as DFS
- Build a node for each version
  - version ID
  - parent version ID (node in the last compaction)
  - children version ID (node in the next compaction)
  - fully enumerated

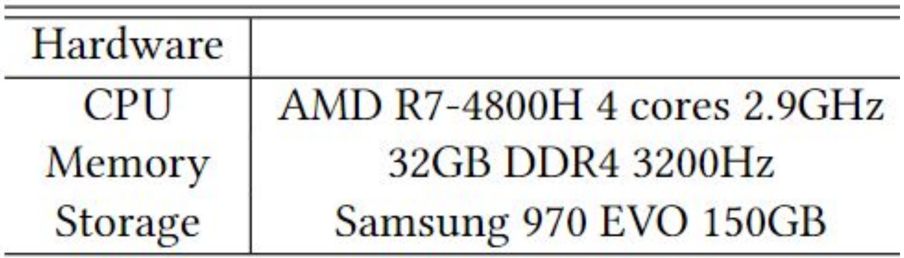
Find an unselected file

Example:

- Focus on compaction from level 1 to level 2
- 2 files in level 1
- 3 compactions

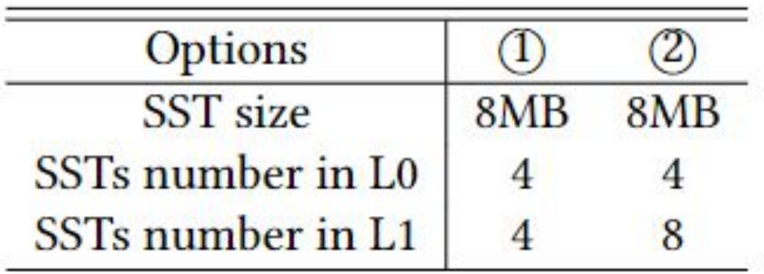


Experiments



Hardware	
CPU	AMD R7-4800H 4 cores 2.9GHz
Memory	32GB DDR4 3200Hz
Storage	Samsung 970 EVO 150GB

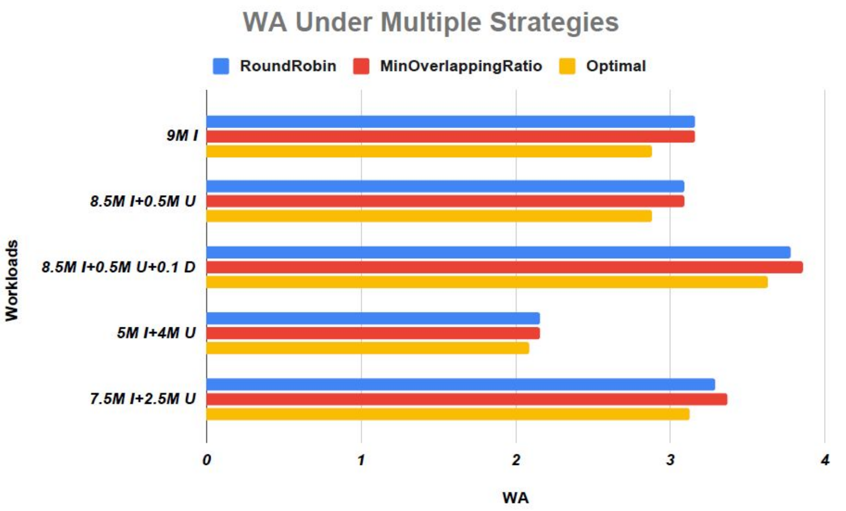
Hardware Settings



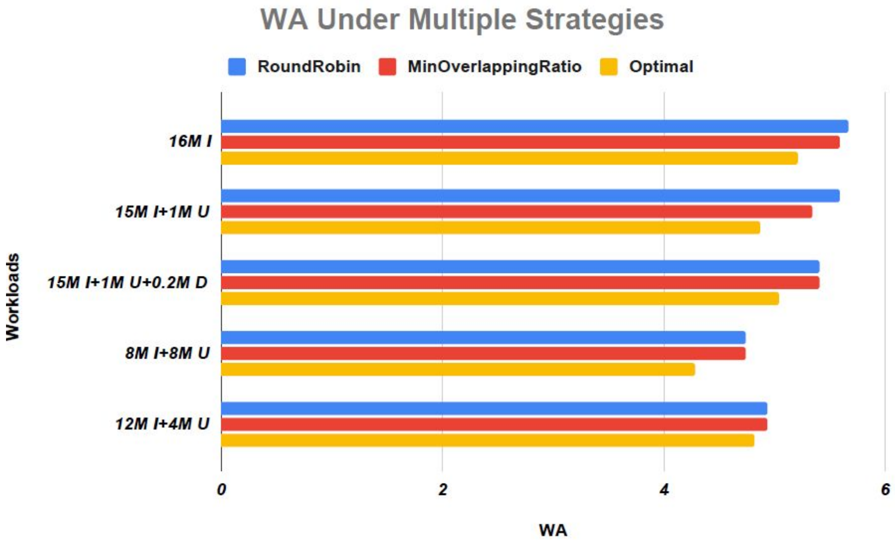
Options	①	②
SST size	8MB	8MB
SSTs number in L0	4	4
SSTs number in L1	4	8

RocksDB Settings

Result



4 SSTs in L1



8 SSTs in L1

2%~8% Optimizing Space

Conclusion and Future Works

- We prove the space of optimizing RocksDB's compaction strategy exists.
- The optimal file selecting pattern is still waiting to be found.
- The larger workload and deeper level can be applied in the future experiment.

Thank you