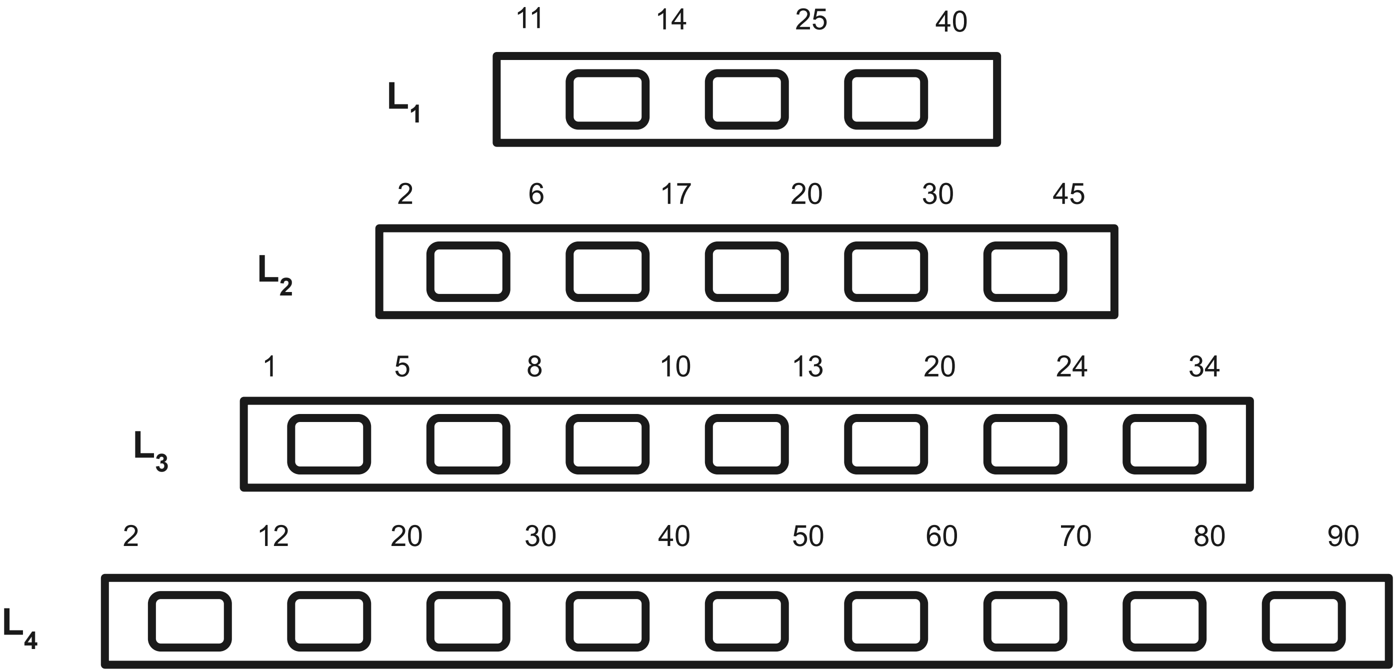
Query-driven compaction in LSM-trees

Kaushik Shubham . Agrawal Nishil . Karatsenidis Kostas

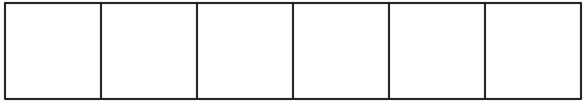
Background

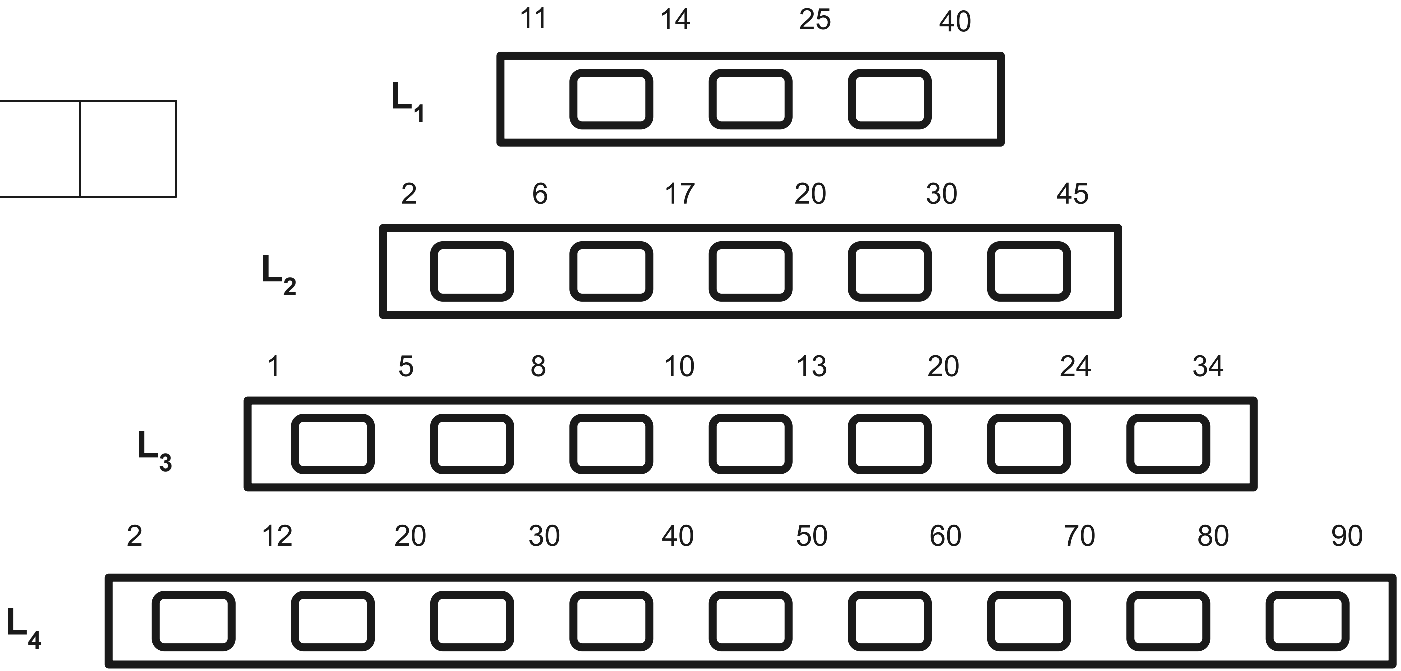
Range Queries in RocksDB



Range Queries in RocksDB

Query: (12, 20)

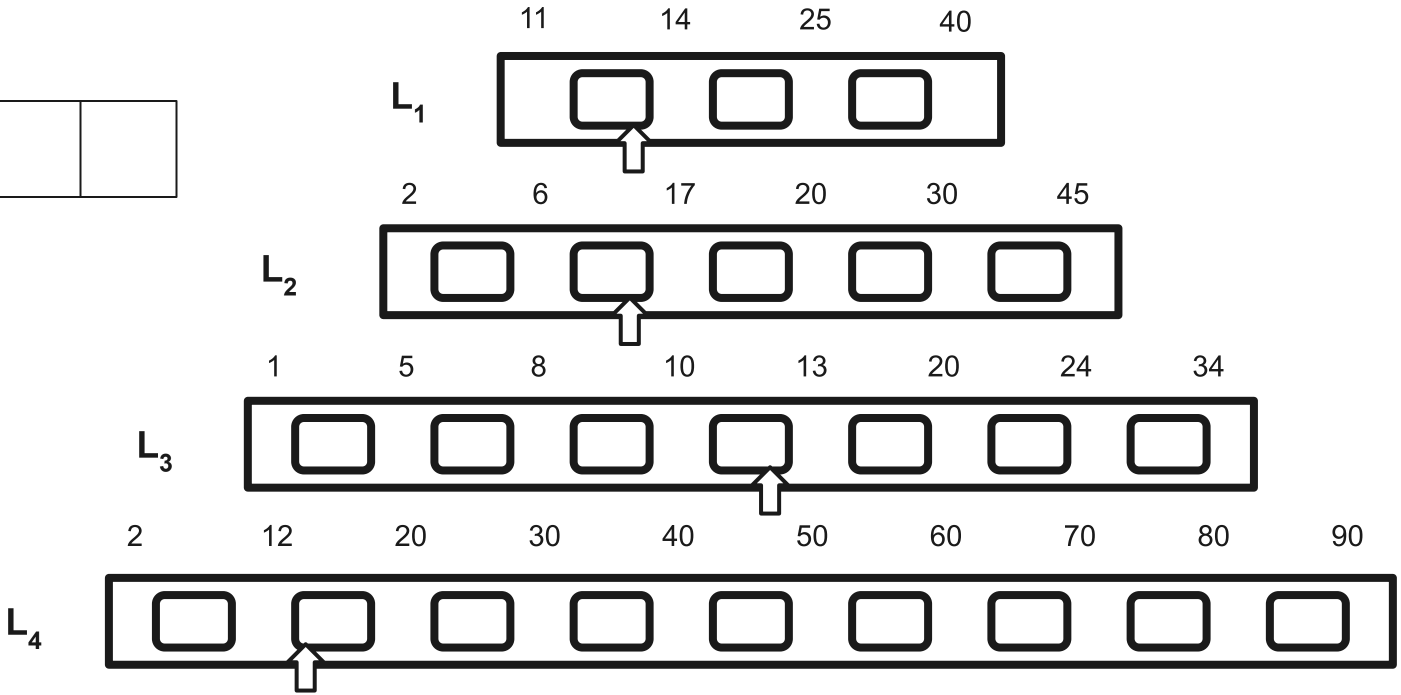




Range Queries in RocksDB

Query: (12, 20)

$iter_1$	13
$iter_2$	16
$iter_3$	12
$iter_4$	12

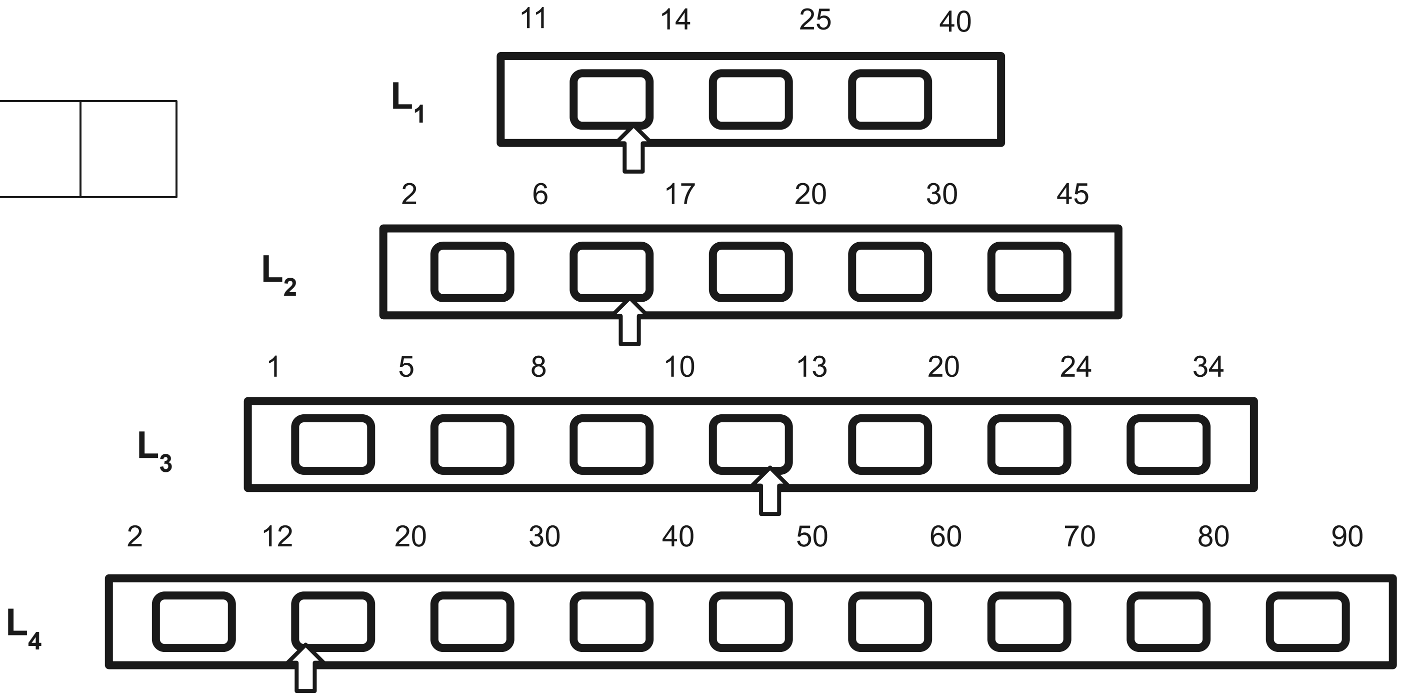


Range Queries in RocksDB

Query: (12, 20)

12					

iter$_1$	13
iter$_2$	16
iter$_3$	
iter$_4$	

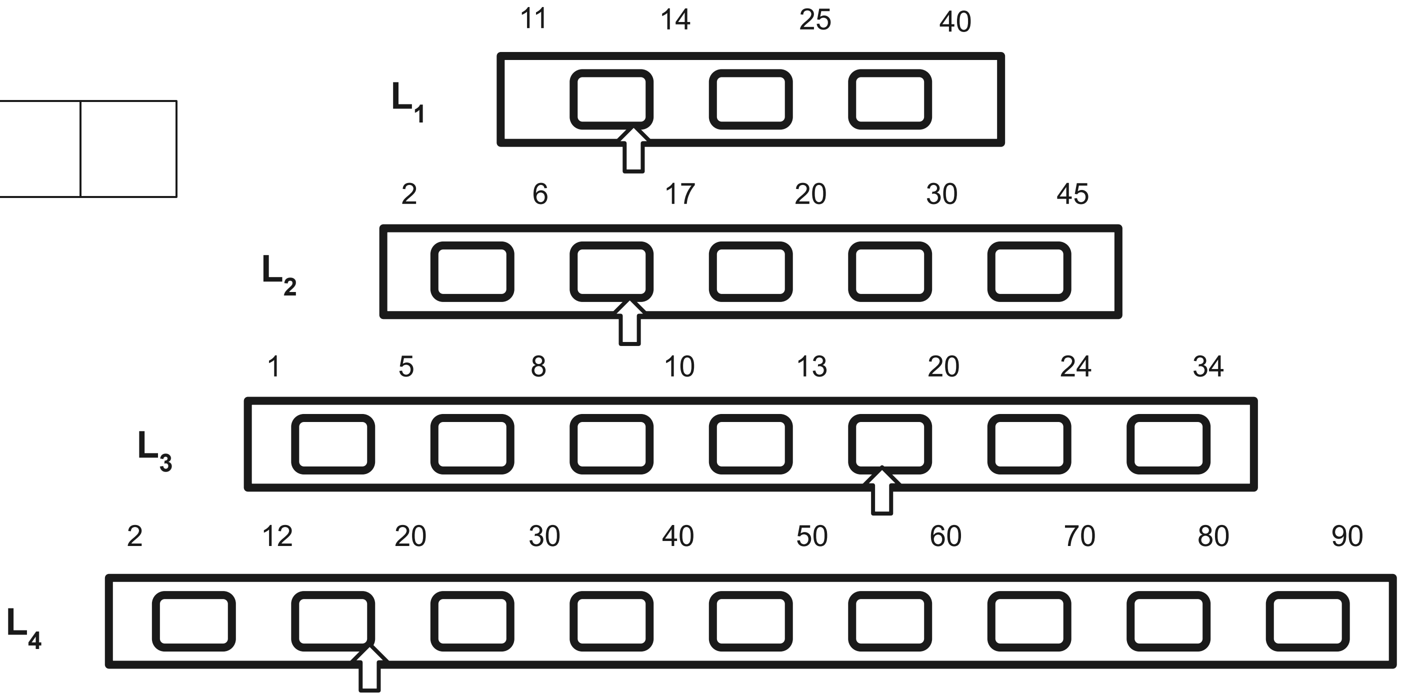


Range Queries in RocksDB

Query: (12, 20)

12					

iter_1	13
iter_2	16
iter_3	15
iter_4	20

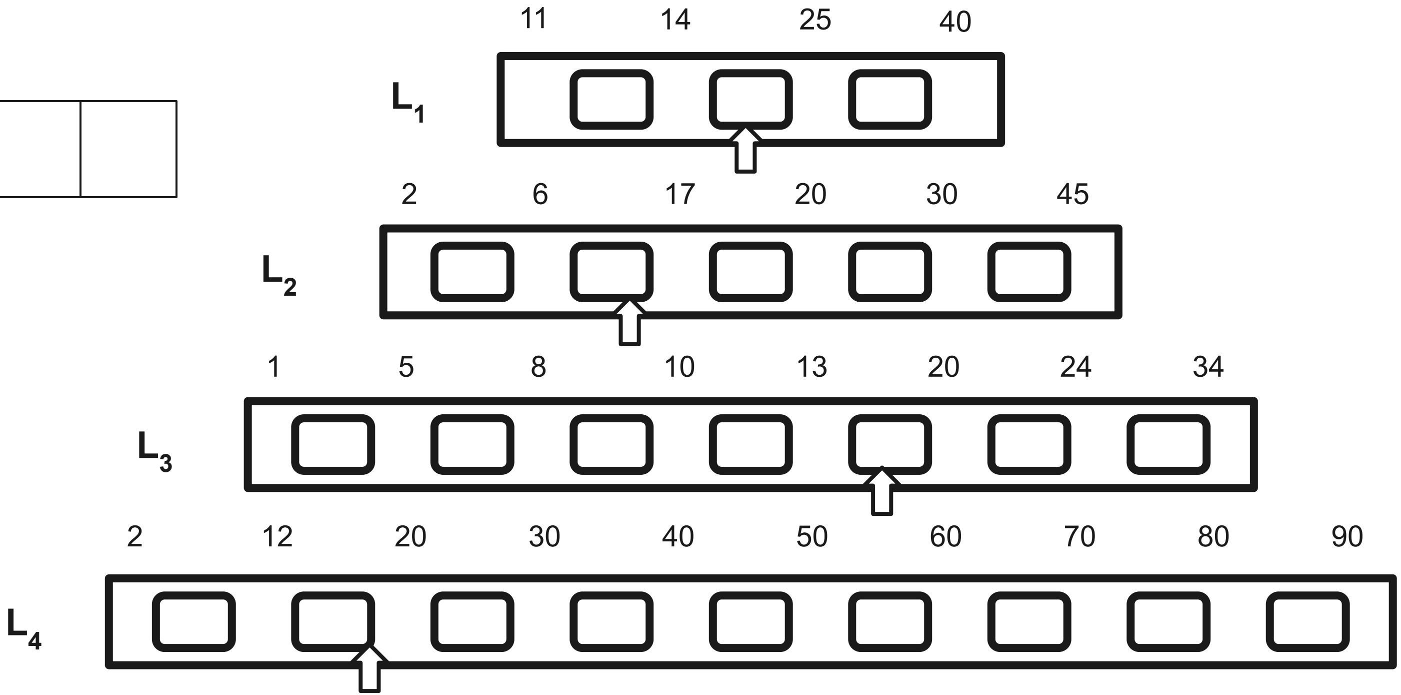


Range Queries in RocksDB

Query: (12, 20)

12	13				

iter_1	15
iter_2	16
iter_3	15
iter_4	20

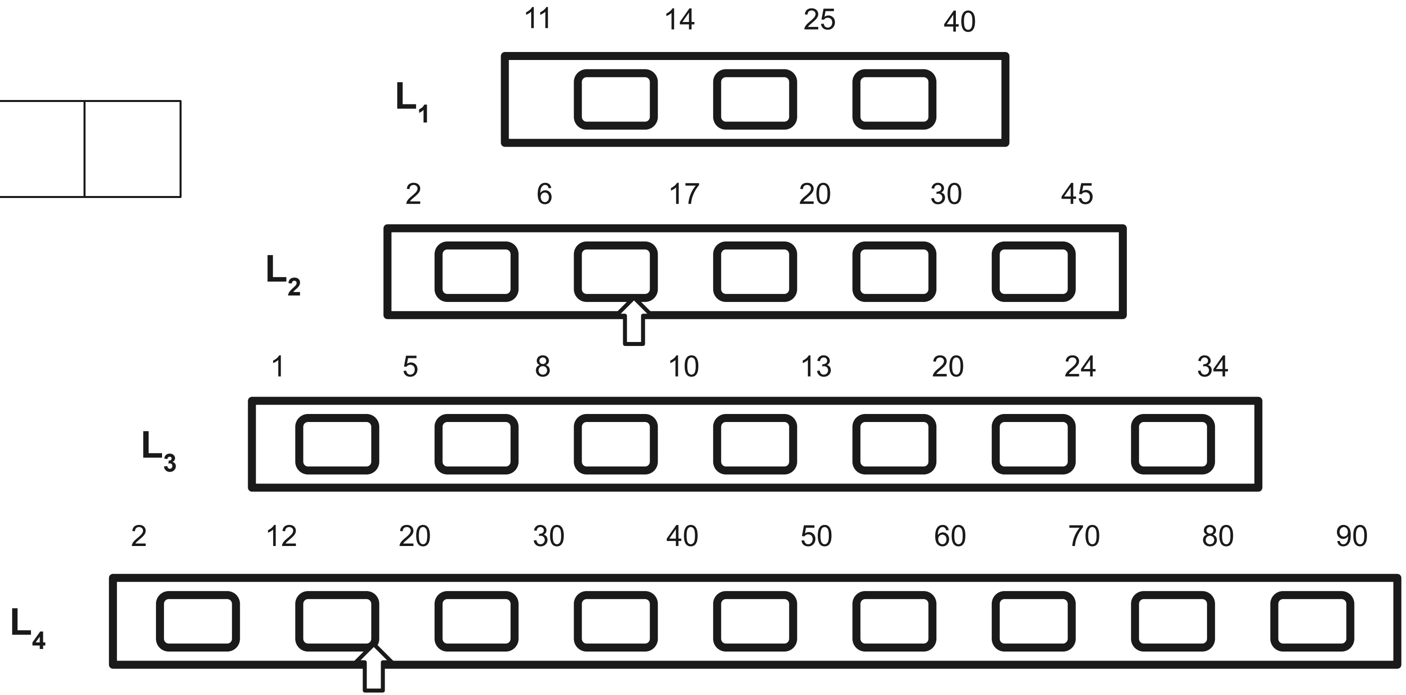


Range Queries in RocksDB

Query: (12, 20)

12	13	15			

$iter_1$	
$iter_2$	16
$iter_3$	
$iter_4$	20

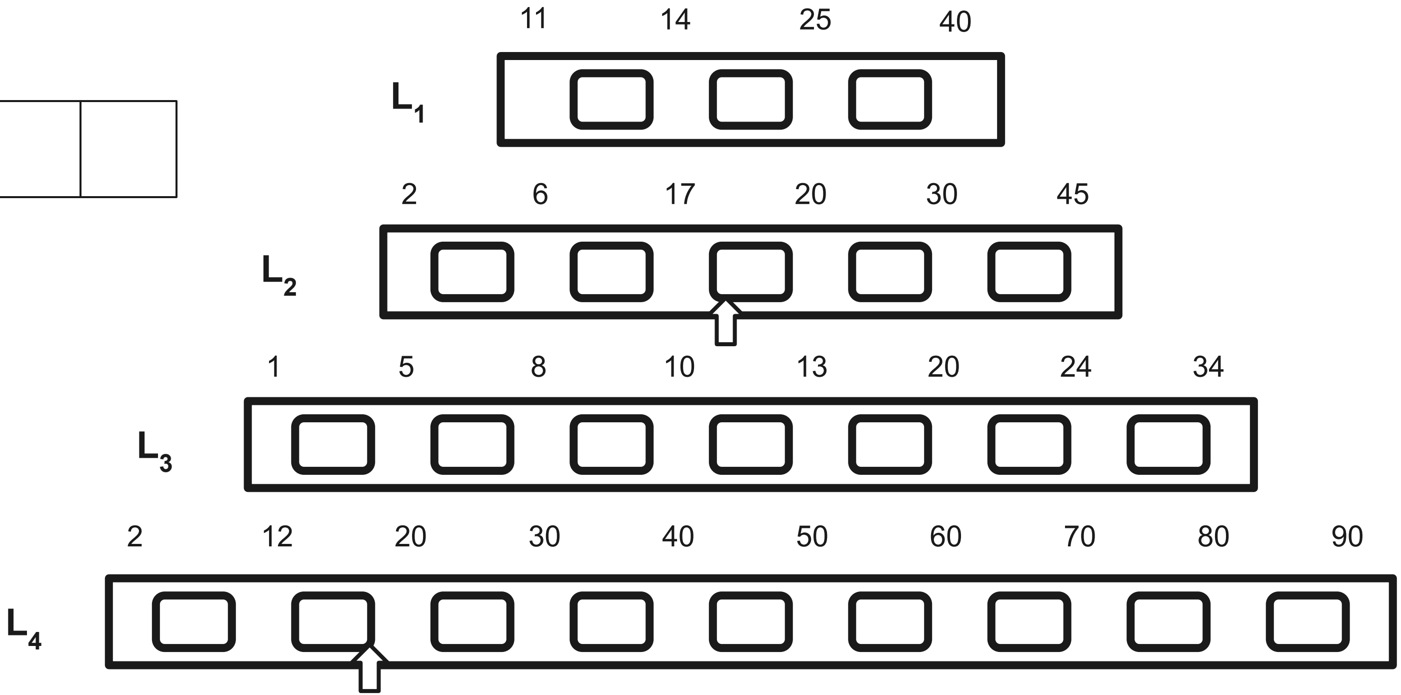


Range Queries in RocksDB

Query: (12, 20)

12	13	15	16		

iter_1	
iter_2	17
iter_3	
iter_4	20

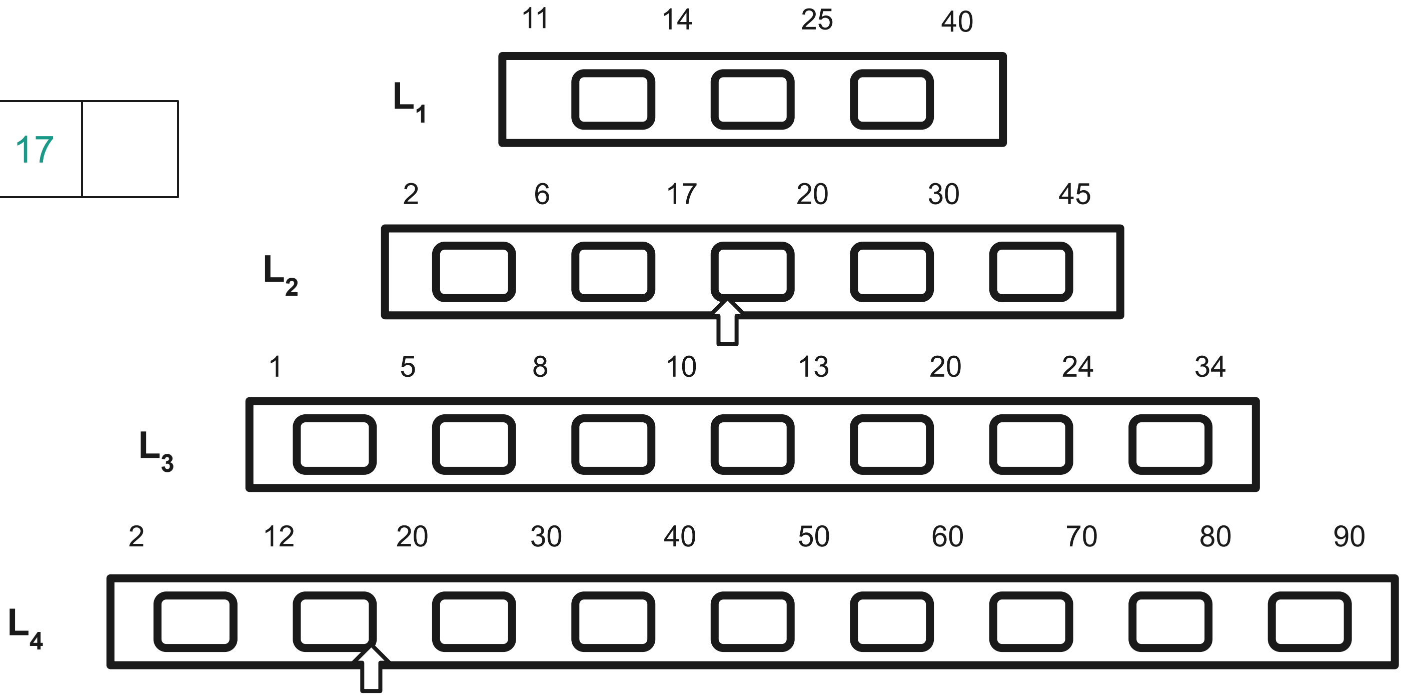


Range Queries in RocksDB

Query: (12, 20)

12	13	15	16	17	

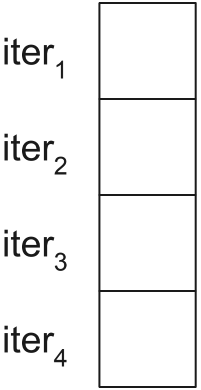
iter_1	
iter_2	20
iter_3	
iter_4	20

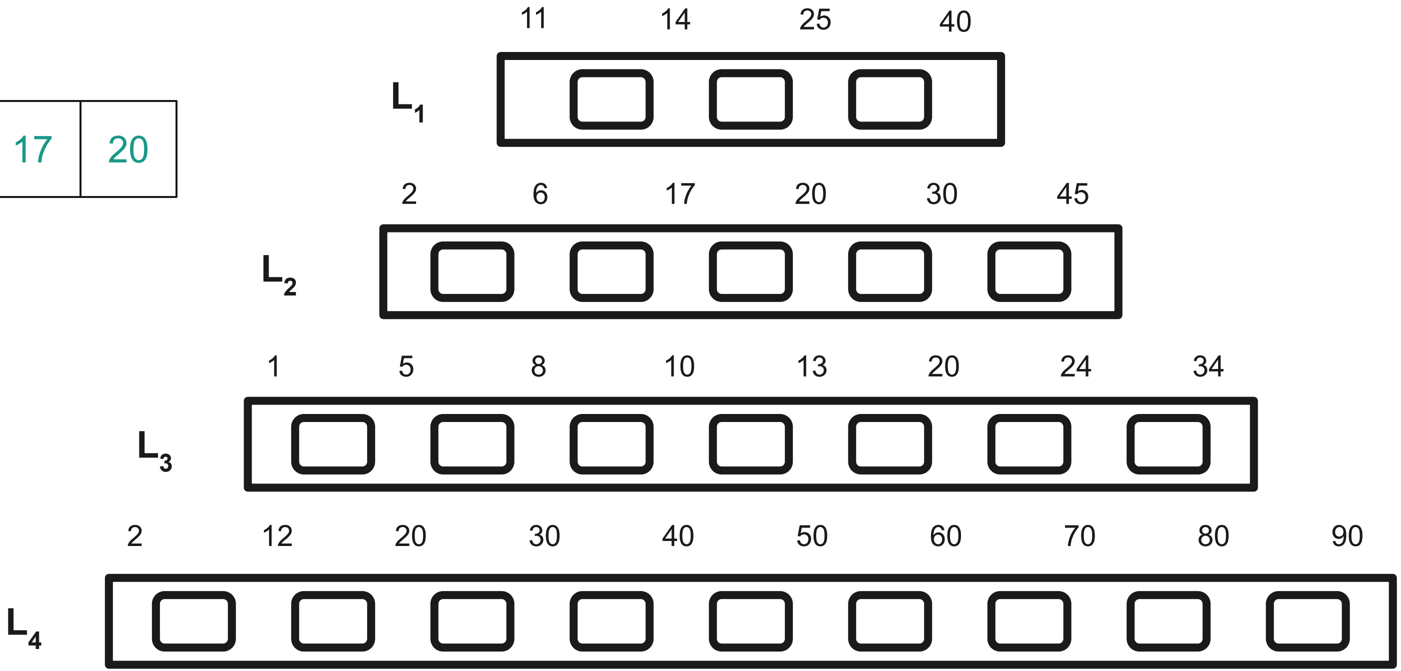


Range Queries in RocksDB

Query: (12, 20)

12	13	15	16	17	20

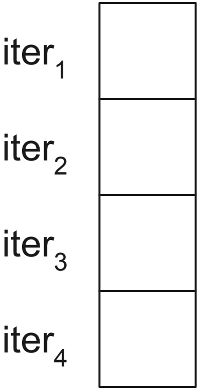


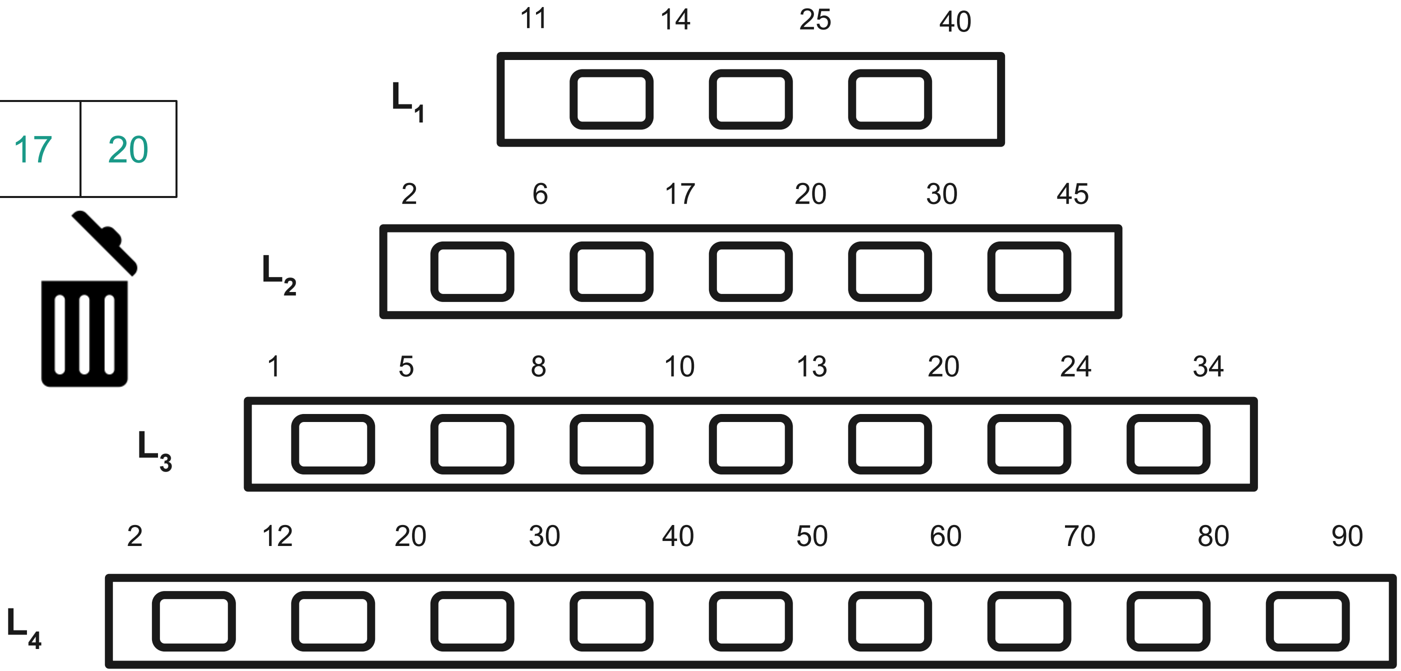


Range Queries in RocksDB

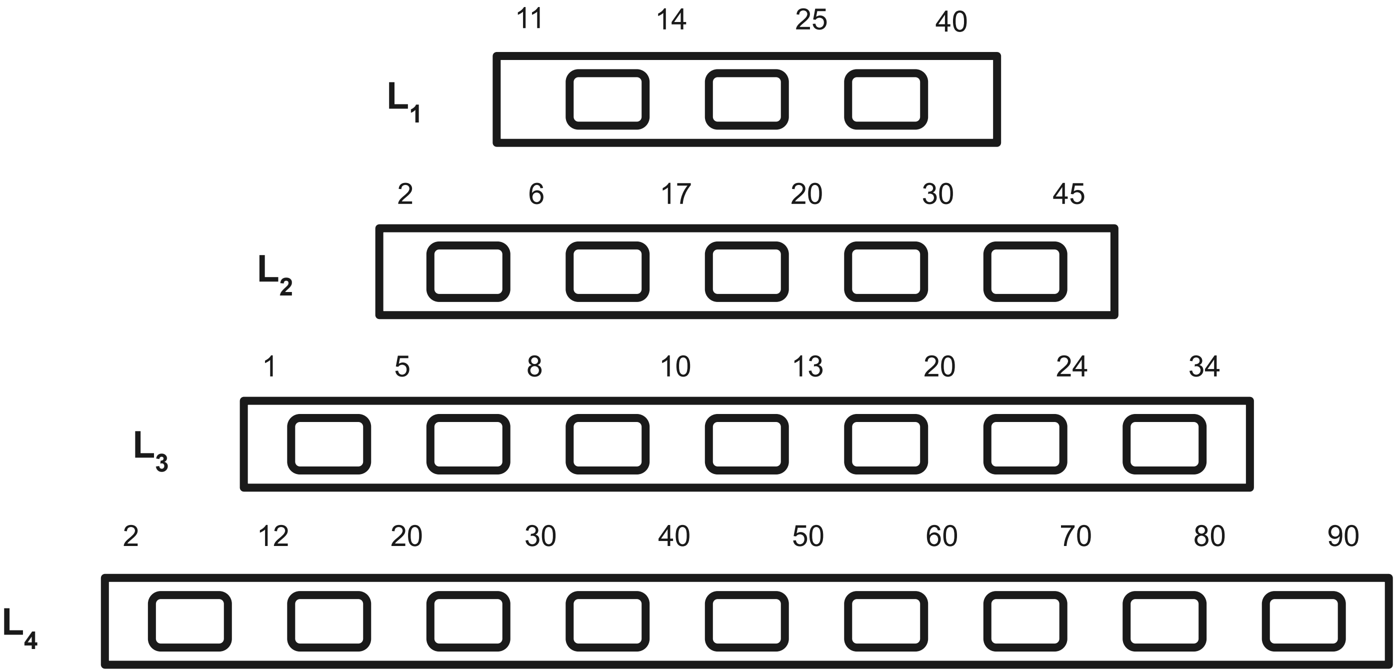
Query: (12, 20)

12	13	15	16	17	20



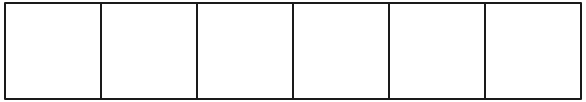


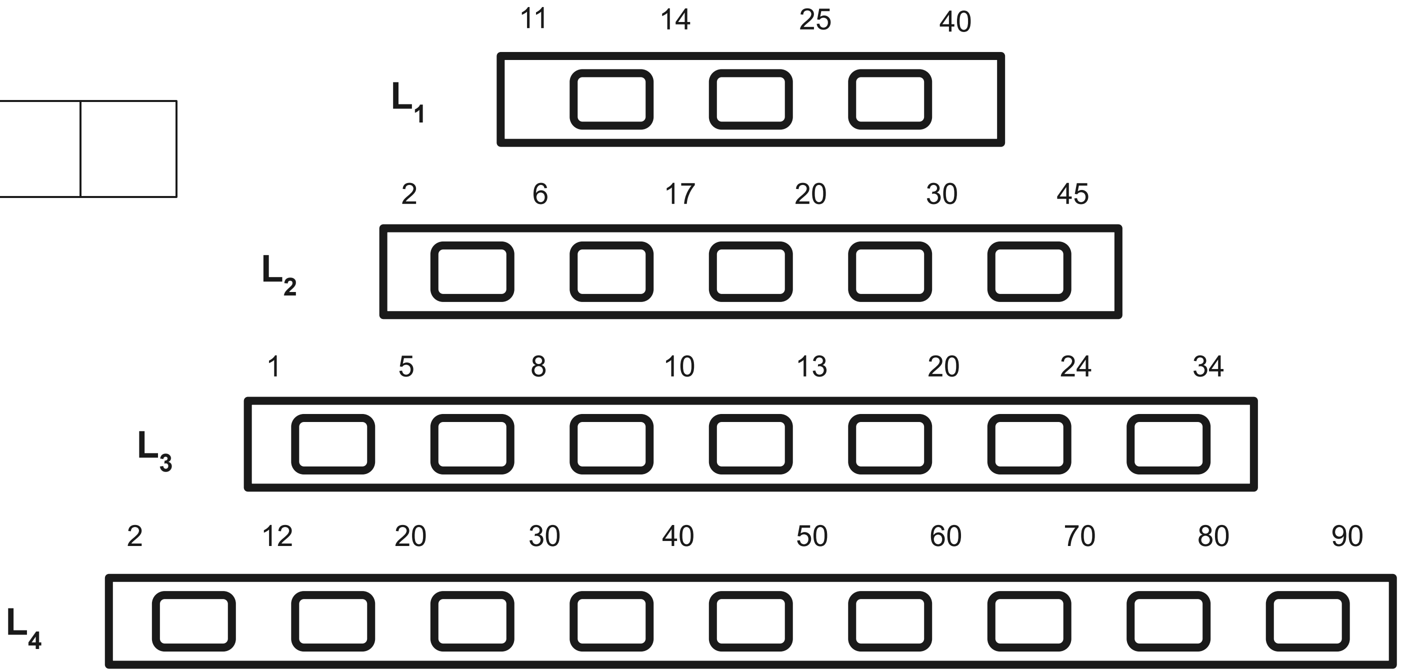
Range Queries in RocksDB



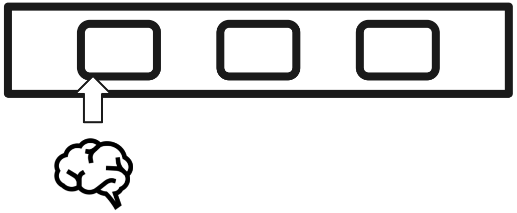
Range Queries in RocksDB

Query: (12, 20)





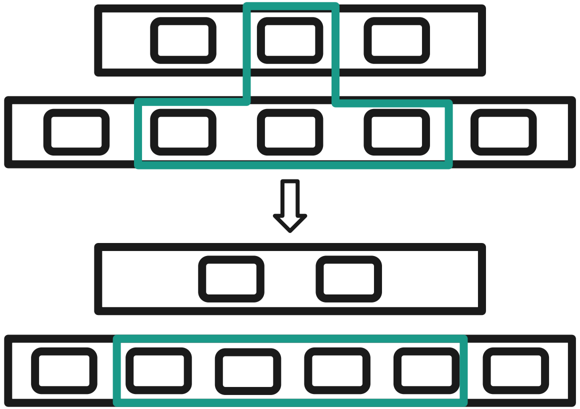
Possible Solutions



Smart Iterator

Optimized

High Complexity



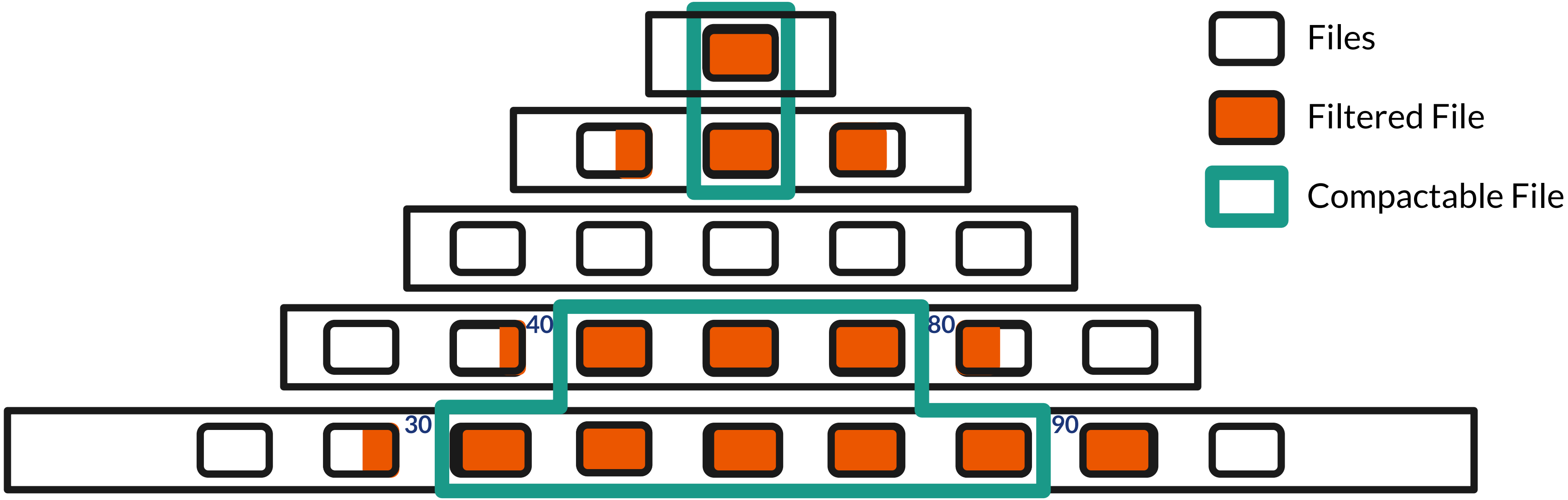
Eager Compaction

Slower

Low Complexity

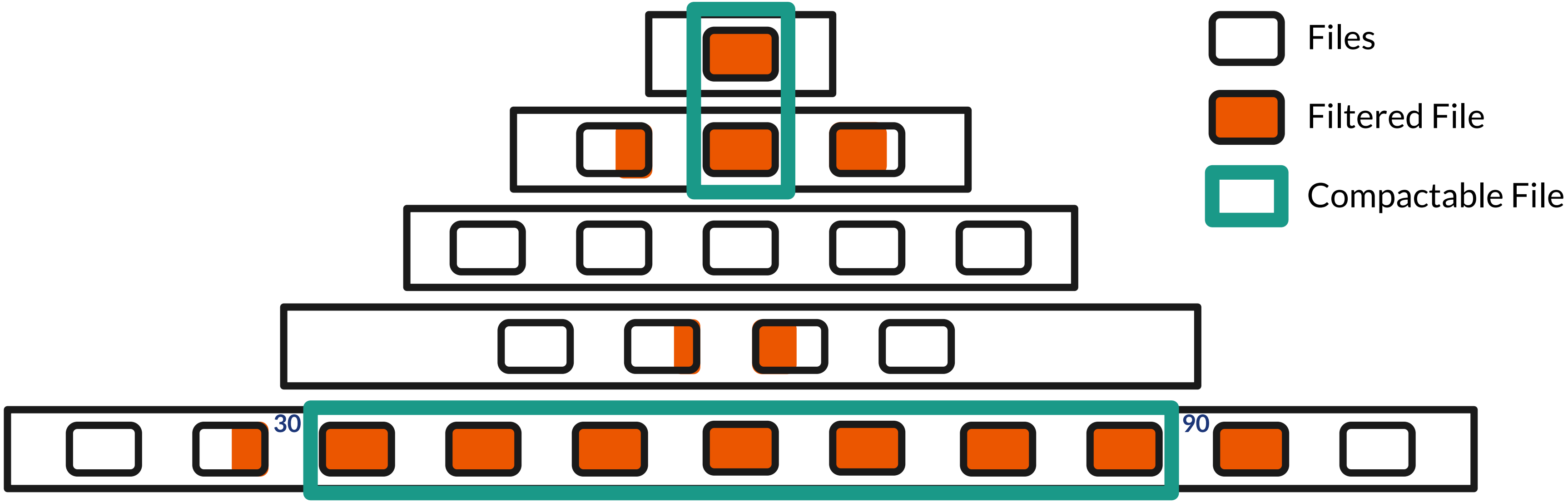
Implementation

Implementation



Range Query ⇨ *Filter(files, start, end)* ⇨ *SelectToCompact(files, start, end)*

Implementation



Range Query → *Filter(files, start, end)* → *SelectToCompact(files, start, end)* → *Compact(files)*

Challenges

Challenges

Exploring and Understanding RocksDB

Generating Optimal Workloads

Running Expensive Workloads

Results

Configuration

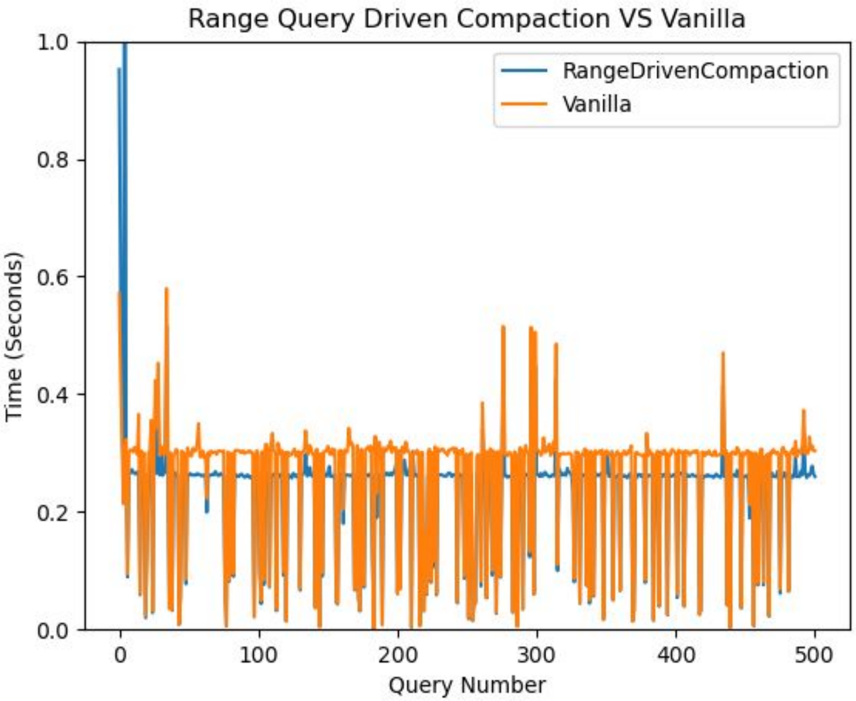
Intel Core i7-7700HQ CPU @ 2.8GHz x 4

16GB RAM

512GB SSD

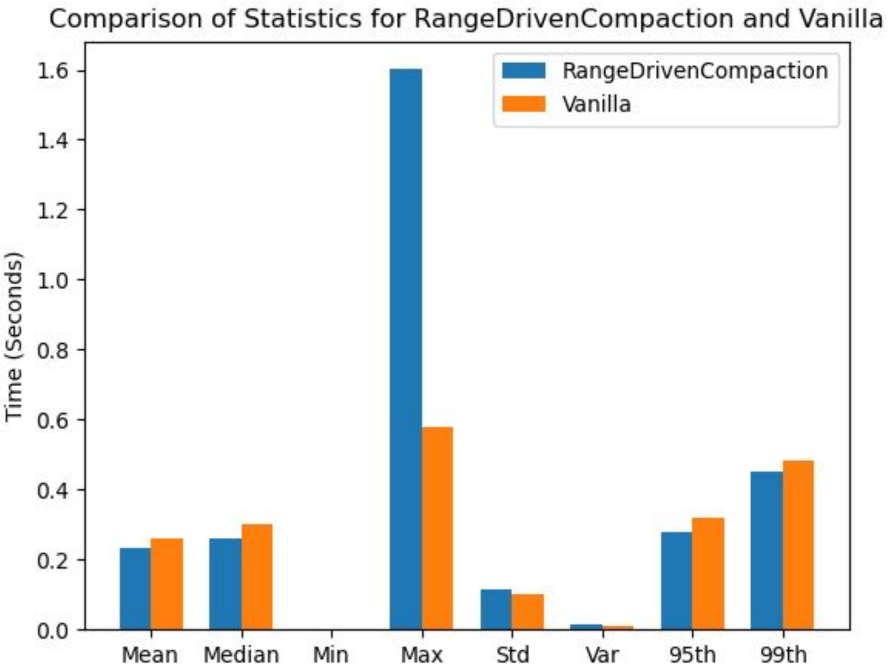
Compaction Style: Level
Base File Size: 512KB
File Size Multiplier: 1
Level Size Multiplier: 10

Results



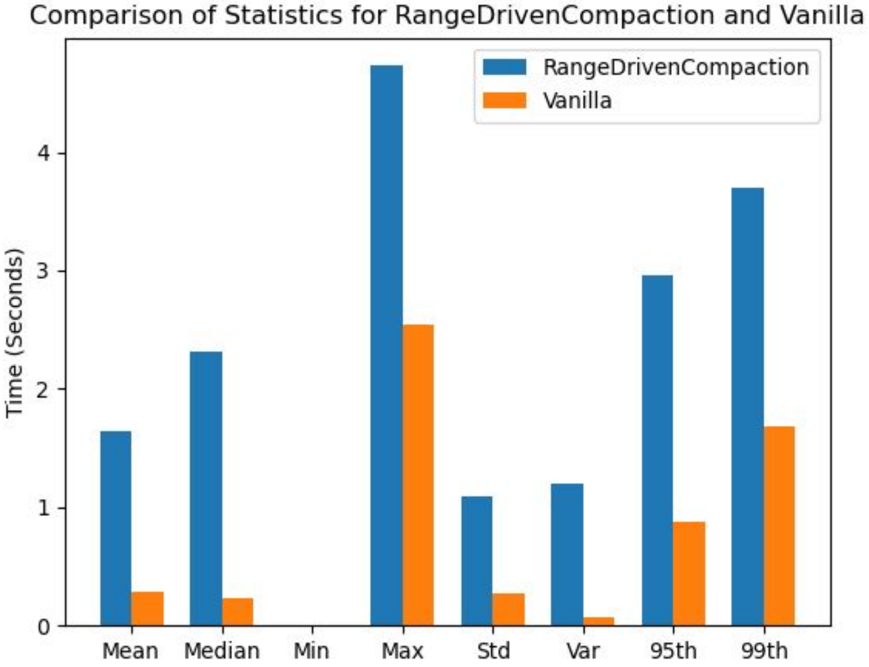
4M Inserts
6M Updates
10K Deletes
500 Range Queries
40% Selectivity

Results



4M Inserts
6M Updates
10K Deletes
500 Range Queries
40% Selectivity

Results



10M Inserts
0 Updates
0 Deletes
1000 Range Queries

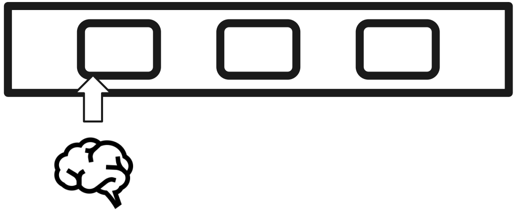
Conclusion and Future Work

Write Amplification

Space Amplification

Read Amplification

Optimize

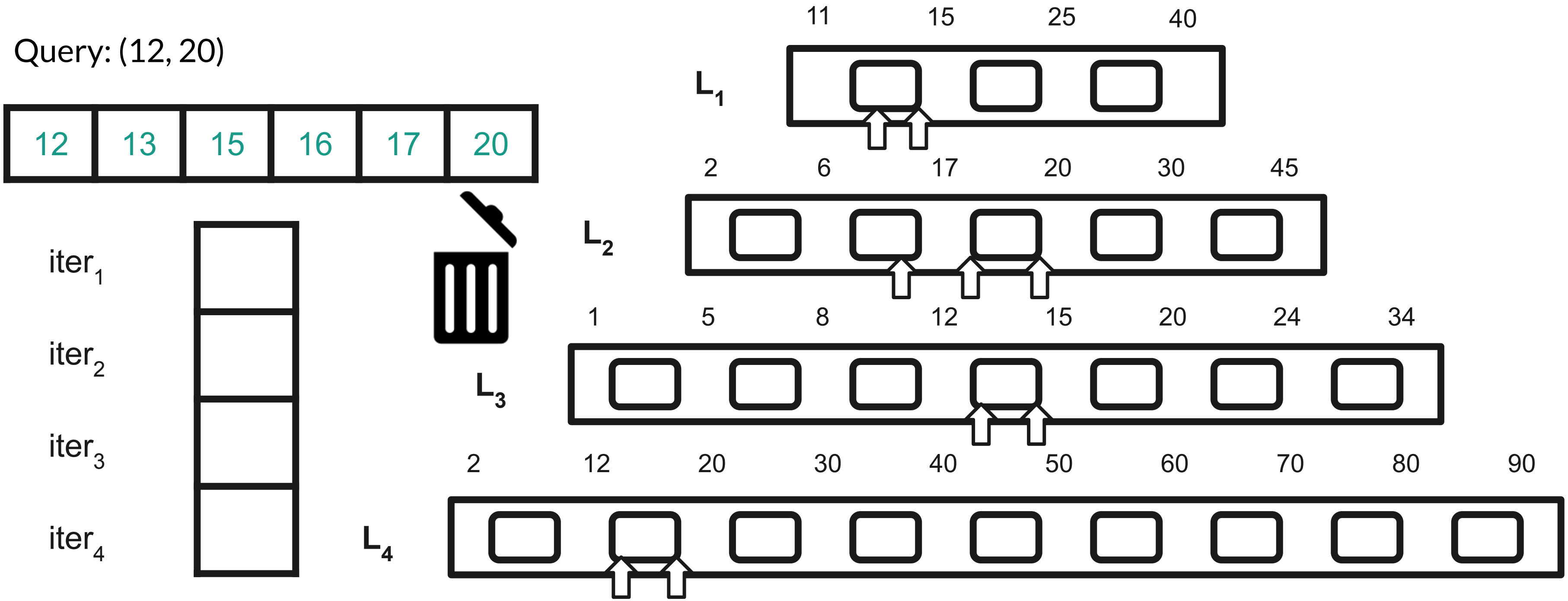


Implementing Smart Iterator

Thank You

Range Queries in RocksDB

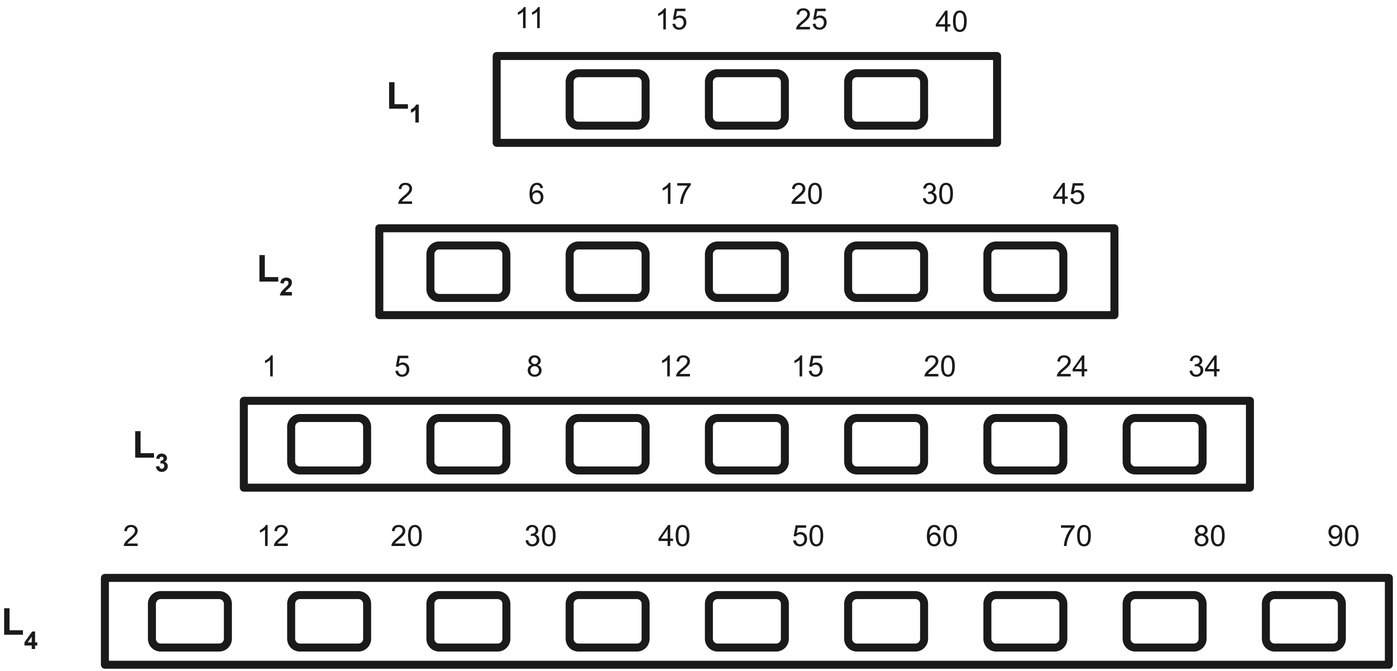
Query: (12, 20)



12	13	15	16	17	20

Range Queries in RocksDB

Query: (12, 20)


11 15 25 40
L1
2 6 17 20 30 45
L2
1 5 8 12 15 20 24 34
L3
2 12 20 30 40 50 60 70 80 90
L4