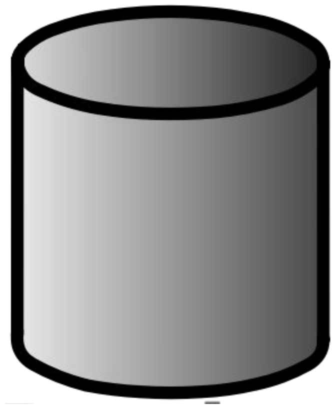
Range Deletes in LSM-tree

Benchmark and Analysis

Xiao Zhang, Wangyi Chen, Xingjian Zhang

Introduction

- LSM-trees in modern databases
- Problem of range deletes
- Exploring the algorithm for efficient range deletions

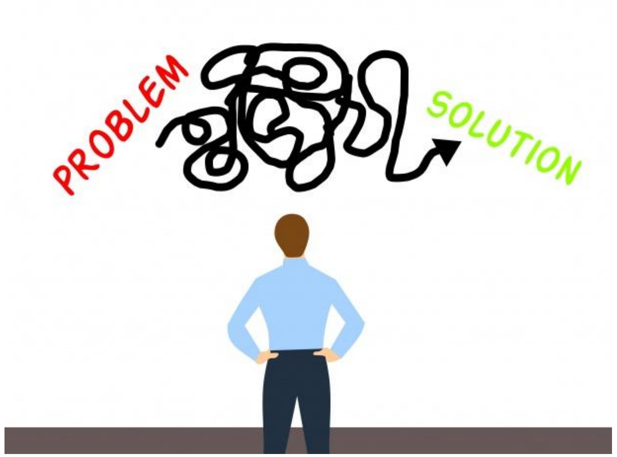


LSM-Trees: Origins and Challenges

- History of LSM-trees
- Role in database management systems
- Advantages and Challenges

Problem Statement

- Need for efficient range deletion in LSM-trees
- Performance challenges in range deletion
- Goal: minimize I/O and CPU overhead while maintaining consistency



Goals

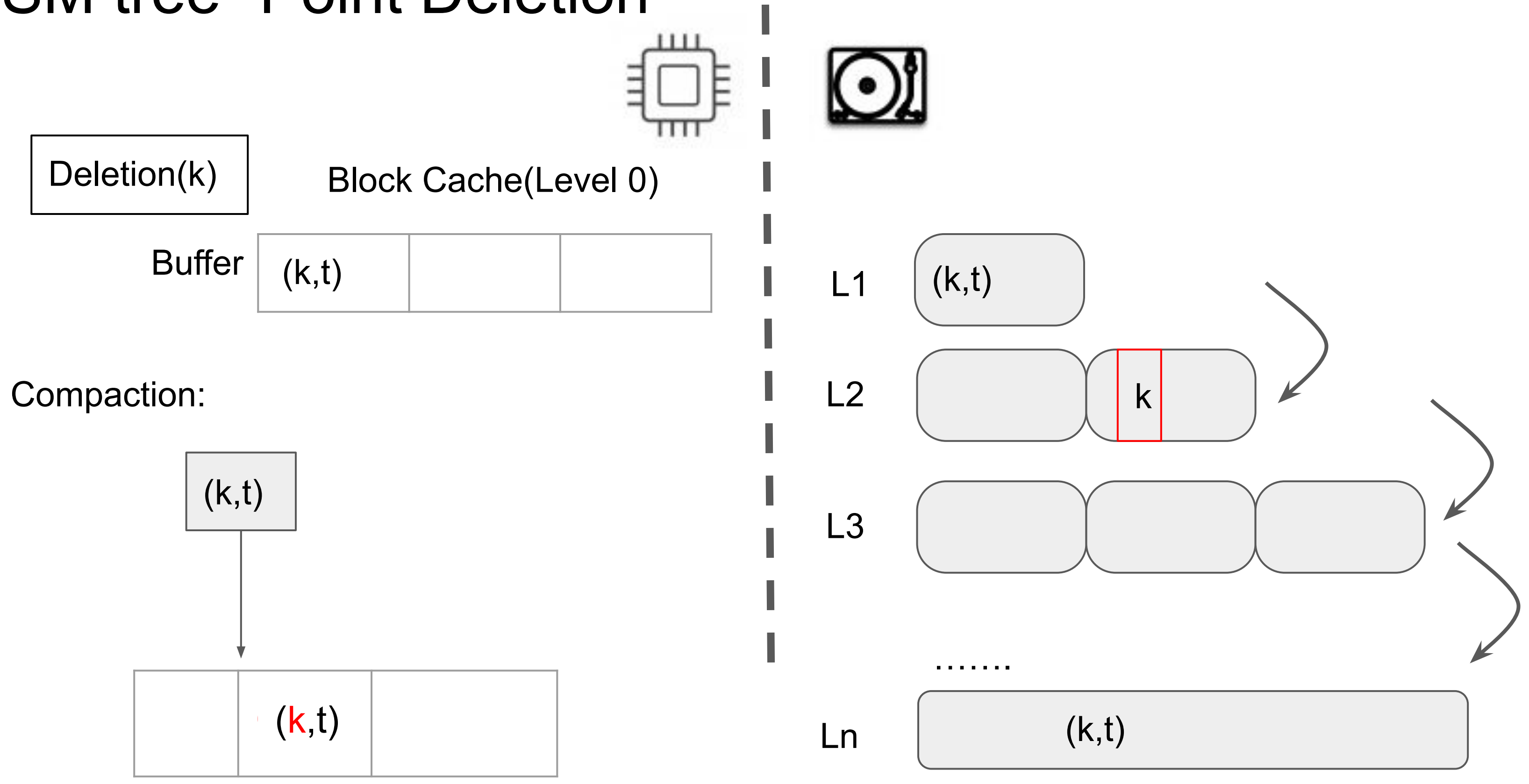
Benchmarking on RocksDB

Give intuitions on performance bound and potential improvements

Focus:

- Skyline Performance
- Memory Footprint
- Cost of Consistency

LSM tree- Point Deletion

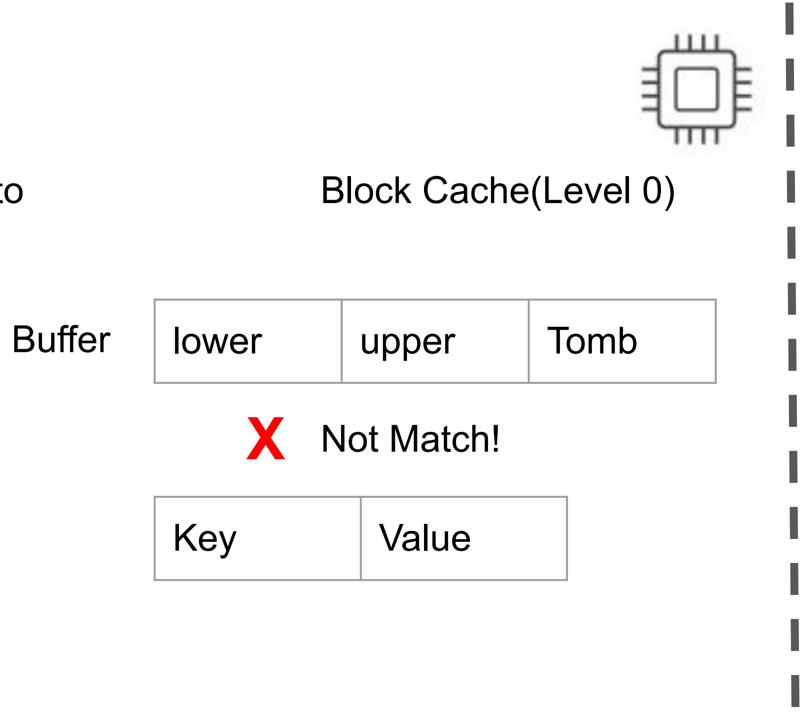


Range Deletion

Are we going to add point_delete * N? NO!

To avoid the memory buffer explode, we only need to insert the (lower, upper, tombstone)

To keep this range delete information, RocksDB provides an implementation of RangeDeleteBlock

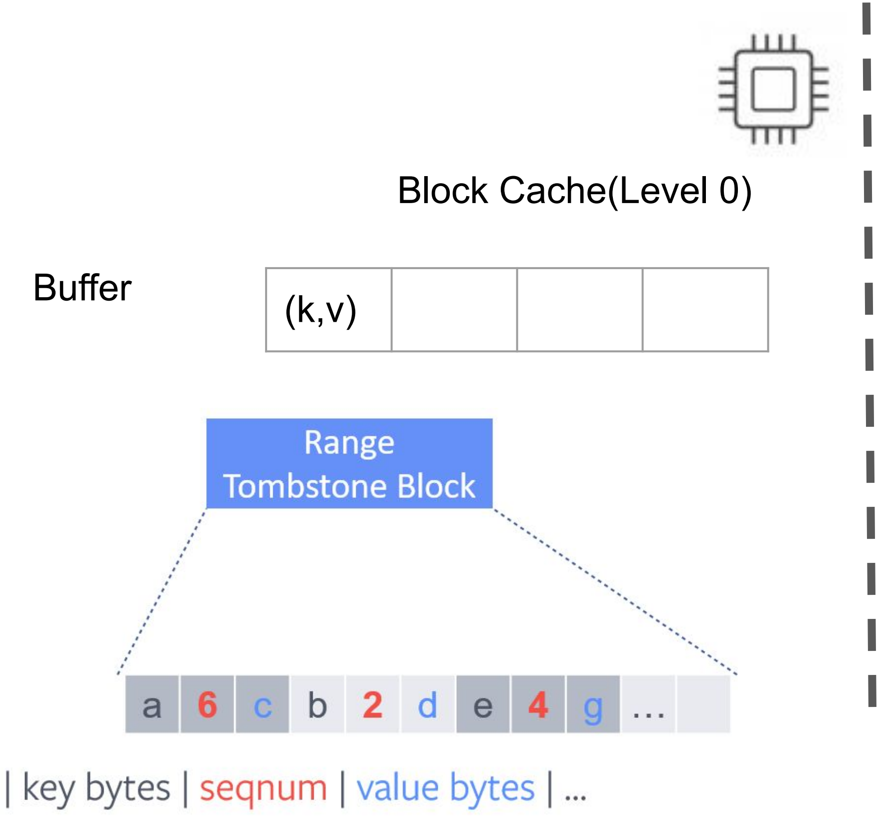


Range Deletion

Are we going to add point_delete * N? NO!

Range Delete Block

Begin key	End key	Seqnum
“a”	“c”	6
“b”	“d”	2
“e”	“g”	4
…	…	



Range Deletion

Compaction

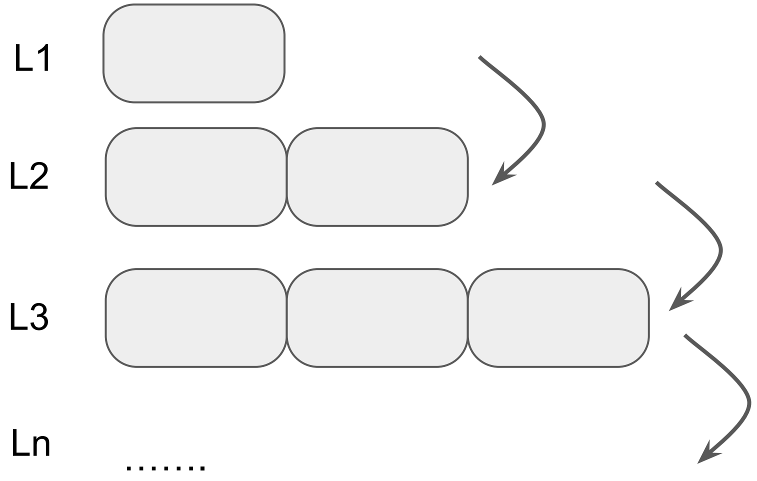
Pages in L1:
[1,8],[11,18],[20,30]

Pages in L2: [20,28],[30,48]

>>[10,30] from RDBlock

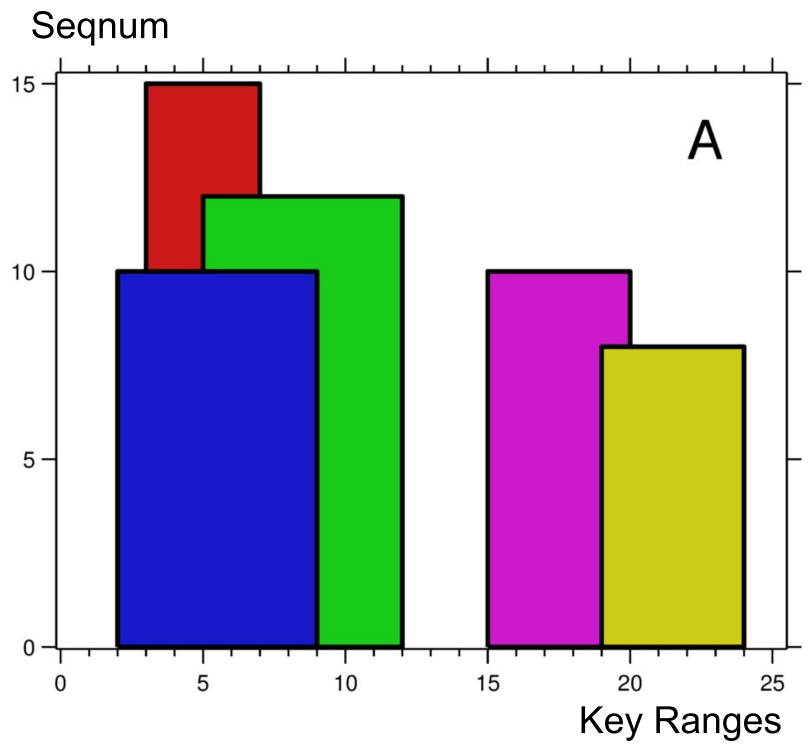
[11,18],[20,30],[20,28]

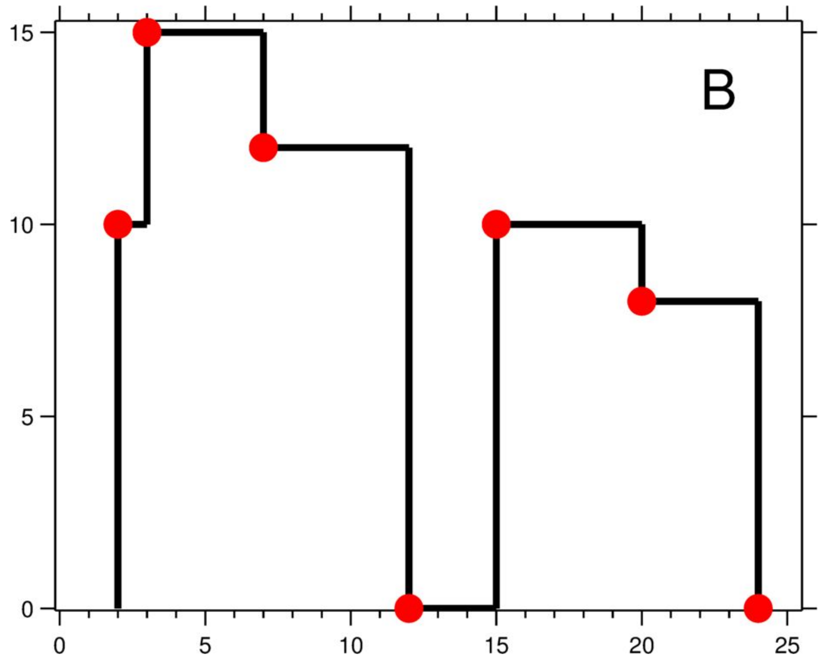
Pages in L2: [1,8],[30,48]



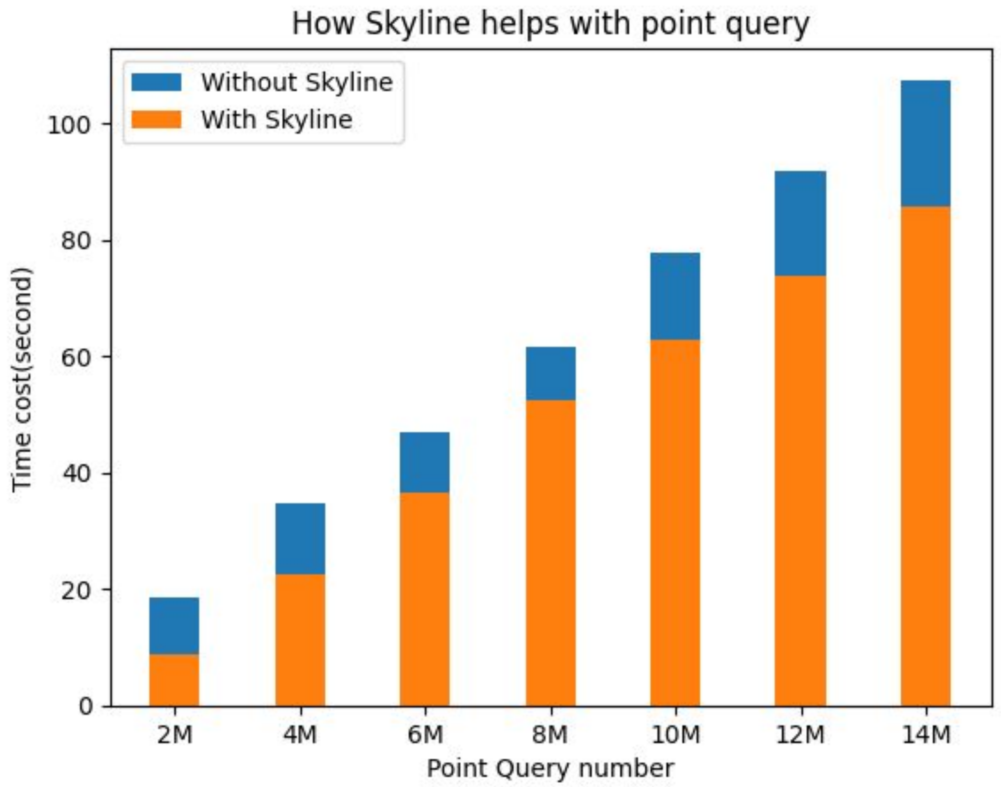
Skyline Model

Skyline Model





Skyline Performance Tests



20M values in Database.
2M inserts of Deletion

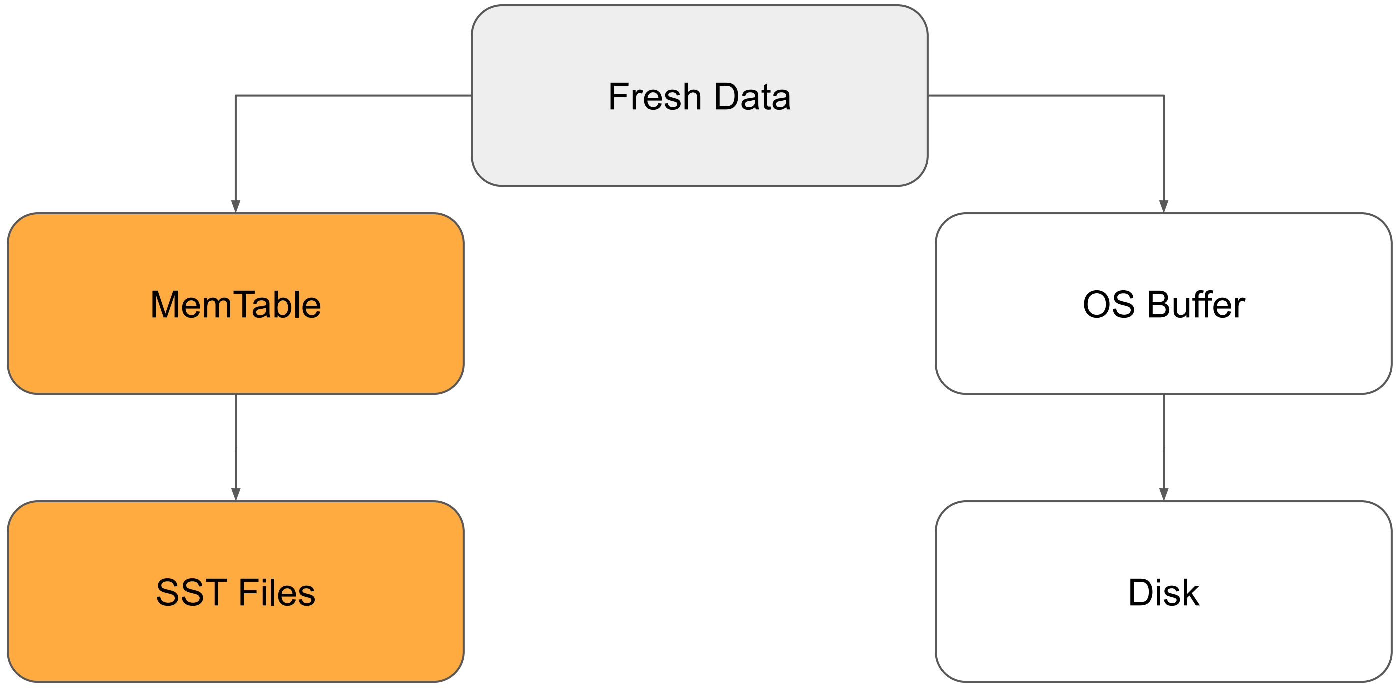
Reduces the time cost of point query by 20%

Cost of Consistency

Consistency

- Write Append Log
- Checksum

Write Ahead Log (WAL)



Experiment Setup

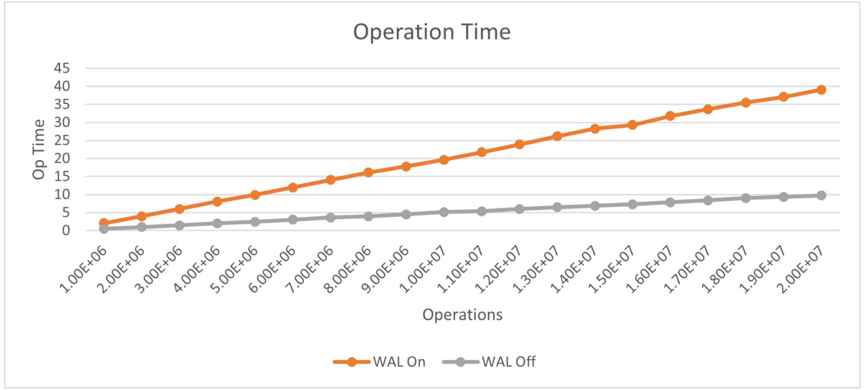
Number of keys: [1:20] * 1,000,000

Number of levels: 7

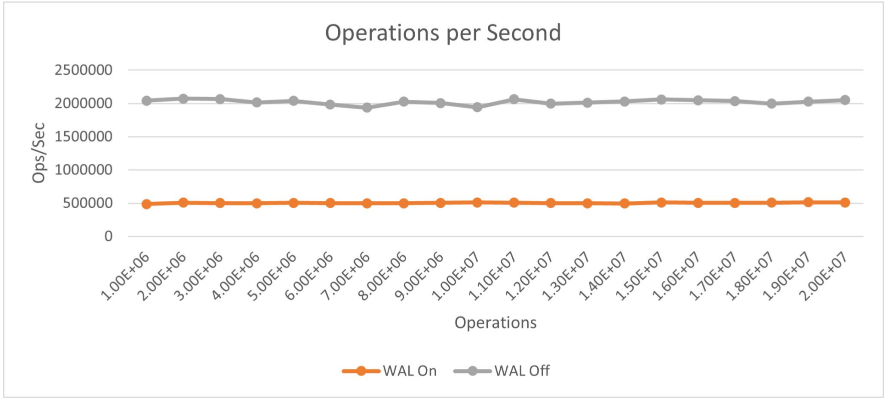
Platform: db_bench

- fillseq: Fill the database with sequential data
- deleteseq: Delete a range of data
- readrandom: Read random data

Results - WAL

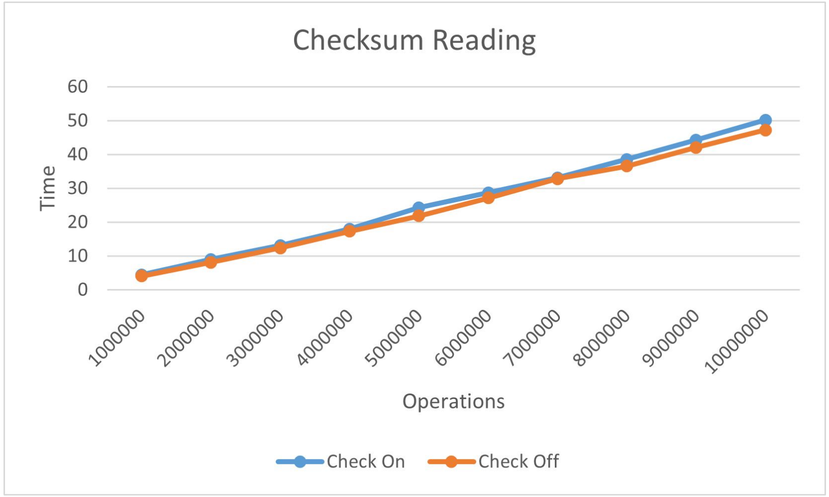


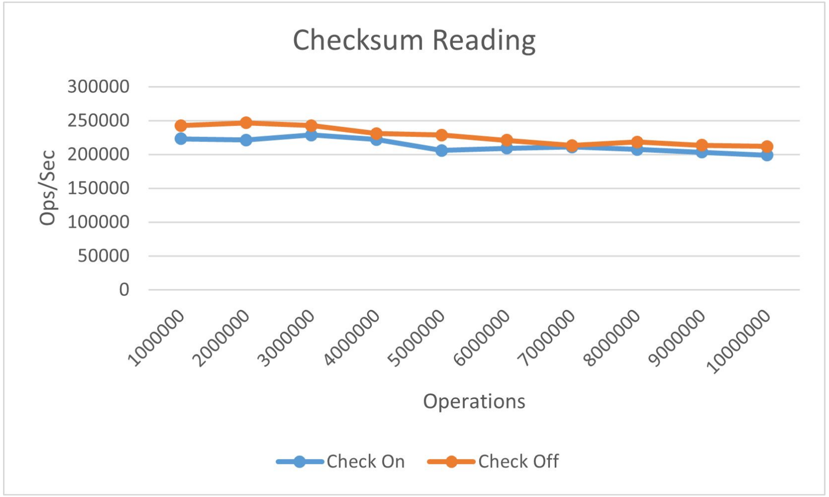
Operation Time
Op Time
45
40
35
30
25
20
15
10
5
0
1.00E+06
2.00E+06
3.00E+06
4.00E+06
5.00E+06
6.00E+06
7.00E+06
8.00E+06
9.00E+06
1.00E+07
1.10E+07
1.20E+07
1.30E+07
1.40E+07
1.50E+07
1.60E+07
1.70E+07
1.80E+07
1.90E+07
2.00E+07
Operations
WAL On
WAL Off



Operations per Second
Ops/Sec
2500000
2000000
1500000
1000000
500000
0
1.00E+06
2.00E+06
3.00E+06
4.00E+06
5.00E+06
6.00E+06
7.00E+06
8.00E+06
9.00E+06
1.00E+07
1.10E+07
1.20E+07
1.30E+07
1.40E+07
1.50E+07
1.60E+07
1.70E+07
1.80E+07
1.90E+07
2.00E+07
Operations
WAL On
WAL Off

Results - Checksum





Thank you

Q & A