

---

---

# Implementation of LSM Tree

Richard Andreas  
Su Jingyu  
Yin Xingkun

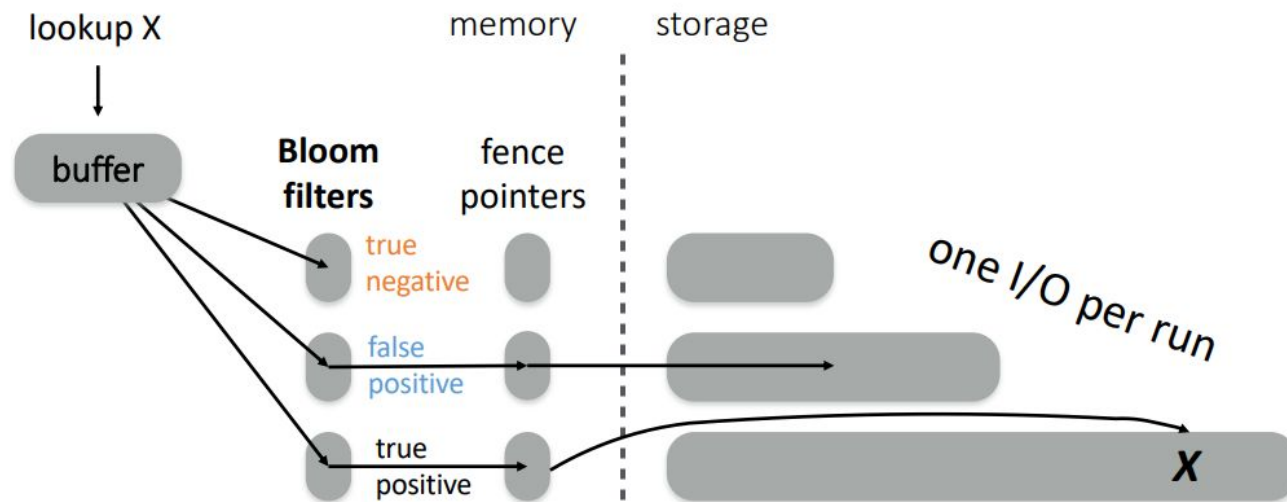
---

---

# Project Implementation

Components:

- Tuple
- FileMeta
- Level
- Run
- Buffer
- Bloom Filter
- Fence Pointer



# Project Implementation

## Data File Organization

# of tuples	Tuple 0 start offset	Tuple 1 start offset	Tuple 2 start offset	...
Tuple 0 key	Tuple 0 value	Tuple 1 key	...	

```
A4 5F 91 00 08 1A 06 00 98 1A 06 00 A8 1A 06 00
B8 1A 06 00 C8 1A 06 00 D8 1A 06 00 E8 1A 06 00
F8 1A 06 00 08 1B 06 00 18 1B 06 00 28 1B 06 00
38 1B 06 00 48 1B 06 00 58 1B 06 00 68 1B 06 00
78 1B 06 00 88 1B 06 00 98 1B 06 00 A8 1B 06 00
B8 1B 06 00 C8 1B 06 00 D8 1B 06 00 E8 1B 06 00
F8 1B 06 00 08 1C 06 00 18 1C 06 00 28 1C 06 00
38 1C 06 00 48 1C 06 00 58 1C 06 00 68 1C 06 00
78 1C 06 00 88 1C 06 00 98 1C 06 00 A8 1C 06 00
B8 1C 06 00 C8 1C 06 00 D8 1C 06 00 E8 1C 06 00
F8 1C 06 00 08 1D 06 00 18 1D 06 00 28 1D 06 00
38 1D 06 00 48 1D 06 00 58 1D 06 00 68 1D 06 00
78 1D 06 00 88 1D 06 00 98 1D 06 00 A8 1D 06 00
B8 1D 06 00 C8 1D 06 00 D8 1D 06 00 E8 1D 06 00
F8 1D 06 00 08 1E 06 00 18 1E 06 00 28 1E 06 00
38 1E 06 00 48 1E 06 00 58 1E 06 00 68 1E 06 00
78 1E 06 00 88 1E 06 00 98 1E 06 00 A8 1E 06 00
B8 1E 06 00 C8 1E 06 00 D8 1E 06 00 E8 1E 06 00
F8 1E 06 00 08 1F 06 00 18 1F 06 00 28 1F 06 00
38 1F 06 00 48 1F 06 00 58 1F 06 00 68 1F 06 00
78 1F 06 00 88 1F 06 00 98 1F 06 00 A8 1F 06 00
B8 1F 06 00 C8 1F 06 00 D8 1F 06 00 E8 1F 06 00
```

# Project Implementation

Supported Operations:

- Get
- Put
- Delete
- Range Scan
- Range Delete

# Project Implementation

Other Features:

- Support persistence:

Persist buffer (Memtable) & metadata: Size of buffer, # of levels, size ratio, etc.

- Support Tiering and Leveling: decide when first build

# Running Prototype

- `./main <outputFilePath>`
- `./main <outputFilePath> <data file path>`

```
Start reading command from terminal
'exit' for exit
'help' for display this information again
'status' for display current LSM Tree status
'I K V...' for inserting key value pair, number of V should be the same as the length specified above
'Q K' for querying the value of the specified key
'S minK maxK' for querying the value of the keys between minK and maxK
'D K [max_K]' for deleting the specified key, 2 parameters for range delete
*** invalid input will cause error
[command input]>>> Q 1 10
Q with incorrect size
[command input]>>> S 1 10
Found rangeScan [
    key: 4 value: Value: 54063 34237 5153
    key: 8 value: Value: 44846 4195 49526
    key: 10 value: Value: 25144 58822 348
]
[command input]>>> Q 15
query result : key: 15 not in the lsm tree, not entered or deleted
[command input]>>> Q 4
query result : key: 4 value: Value: 54063 34237 5153
[command input]>>> D 8
Deleted 8
[command input]>>> I 10 10 10 1
Insert key: 10 values: 10 10 1
[command input]>>> █
```

# Experiment Results

- Correctness in Operations
  - Generated Python and C++ Output files

# Experiment Results

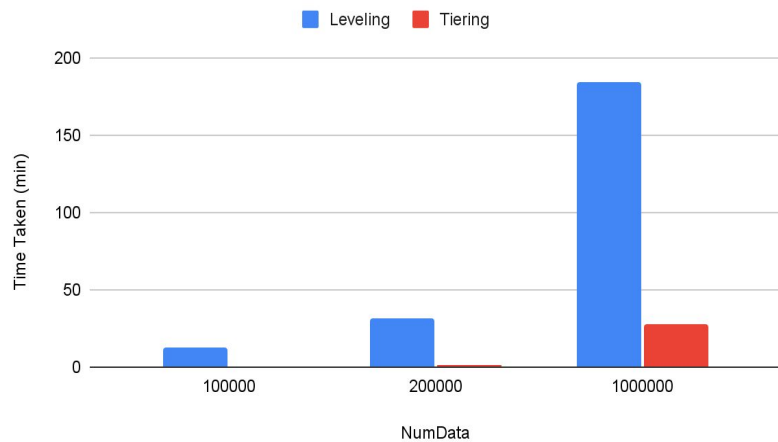
- Correctness in Operations
  - Generated Python and C++ Output files
- Latency



# Experiment Results

- Correctness in Operations
  - Generated Python and C++ Output files
- Latency

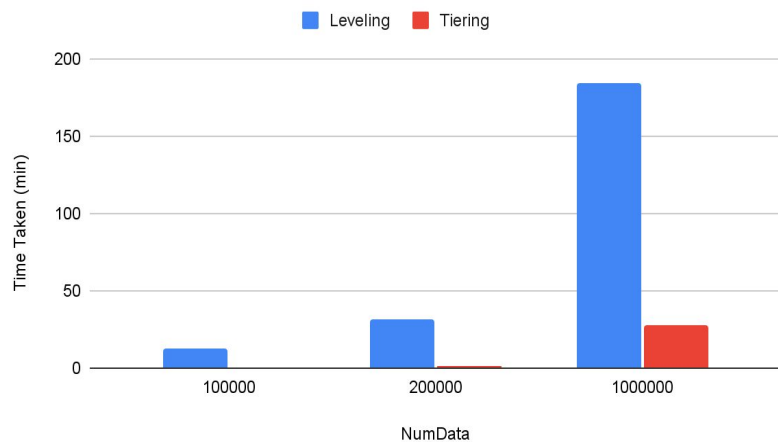
Insert Data



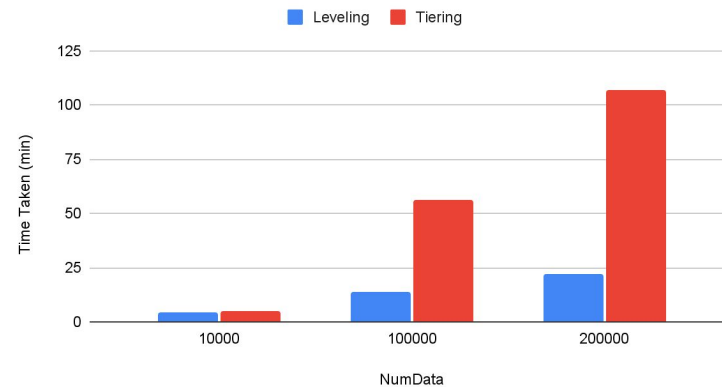
# Experiment Results

- Correctness in Operations
  - Generated Python and C++ Output files
- Latency

Insert Data



Querying Data



# Challenges, Interesting thing

Locating bottle neck

- bloom filter

Merging code together

- plan ahead of time
- create APIs together

Timeline issue

- start working early

## Further Works to Consider

- Improving Optimizations for Merging and Range Query Operations
- Adjusting Bloom Filter Optimizations
- Comparing with RocksDB and other industry level LSM Tree implementation