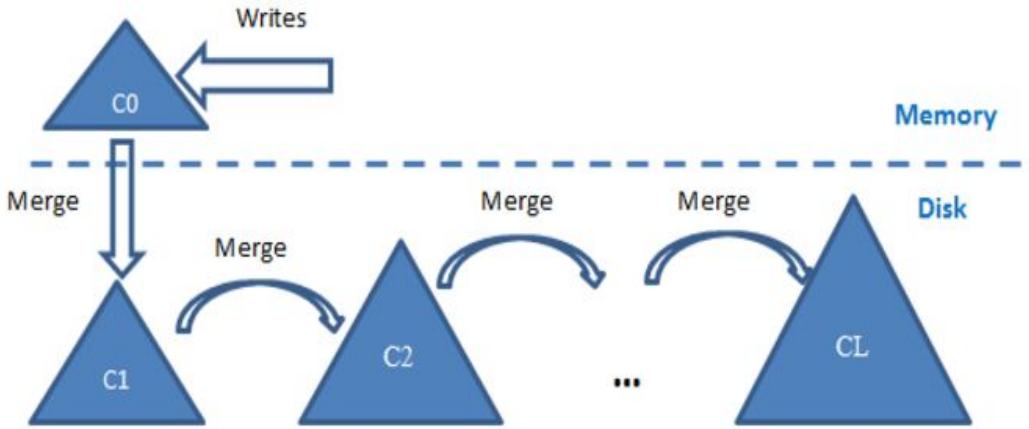
LSM Tree Implementation

By Matthew Cote

What's the Goal?

LSM Tree Goals:

- Tunability for read/write
- Durable storage on disk
- Flexible storage engine



What's the Goal?

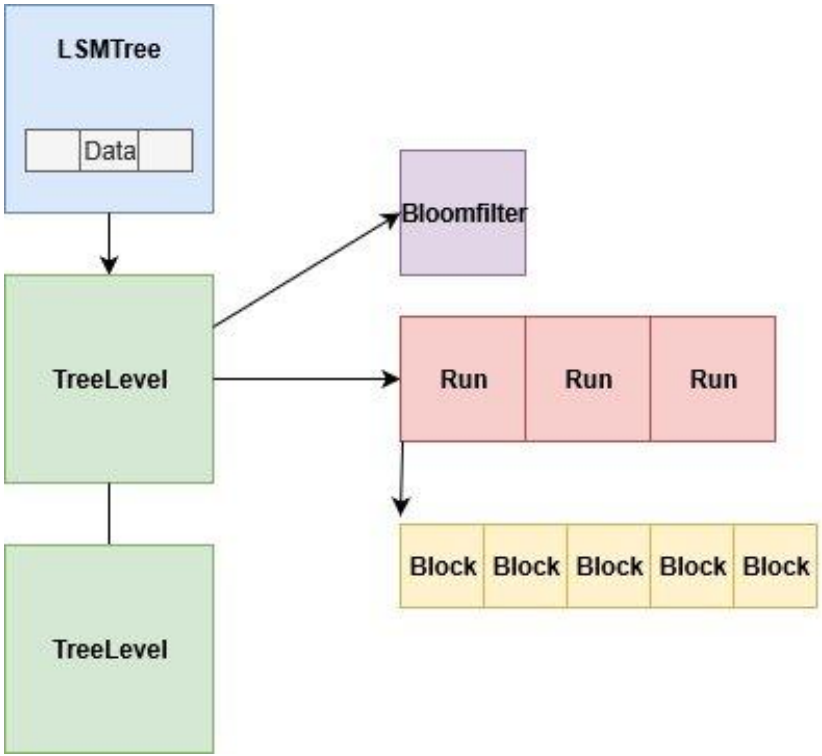
Project Goals:

- Contribute to LSM Tree design knowledge
- Improve low-level system skills
- Better understand LSM Trees

Supported Operations

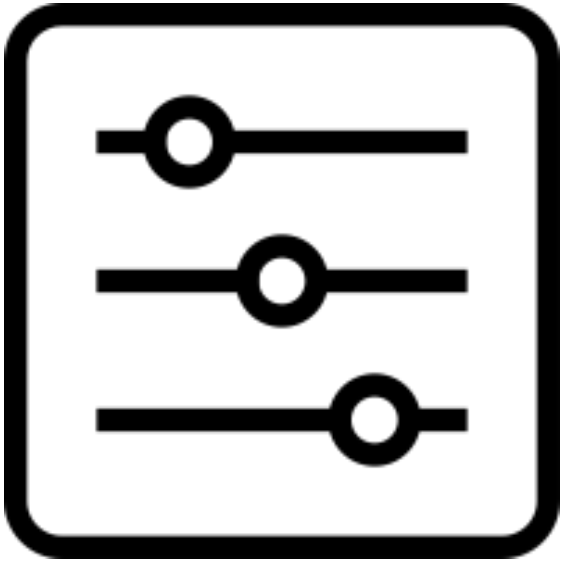
- Put
- Get
- Get Range
- Delete
- Update

Class Design



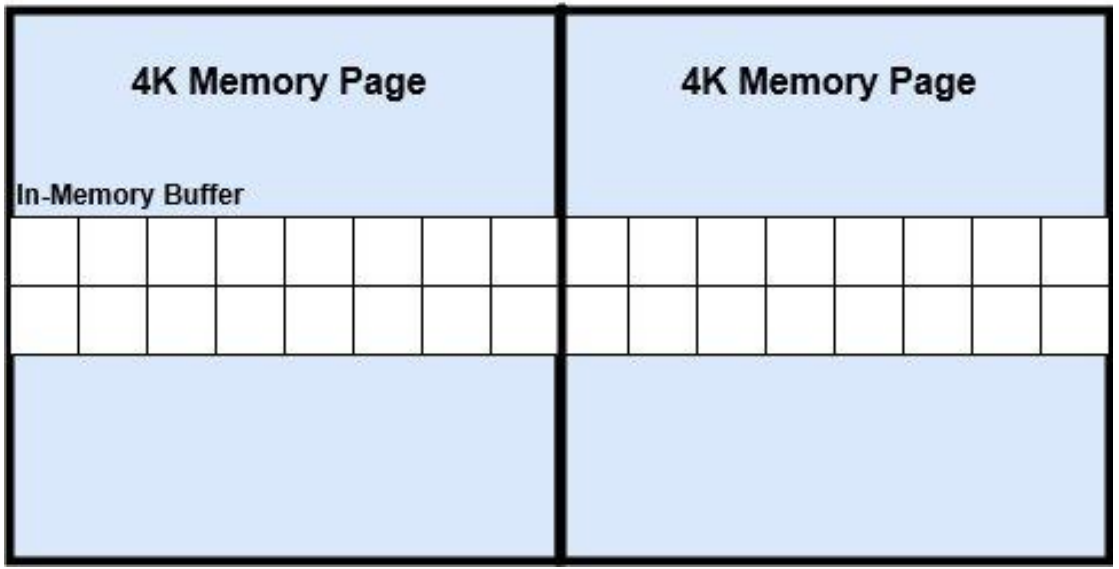
Design Decisions

- Templatized
- Fixed length data
- Tuning Parameters:
  - Memory Buffer Size
  - Bloom Filter Size
  - File Size
  - Merge Policy
  - Tiering Factor



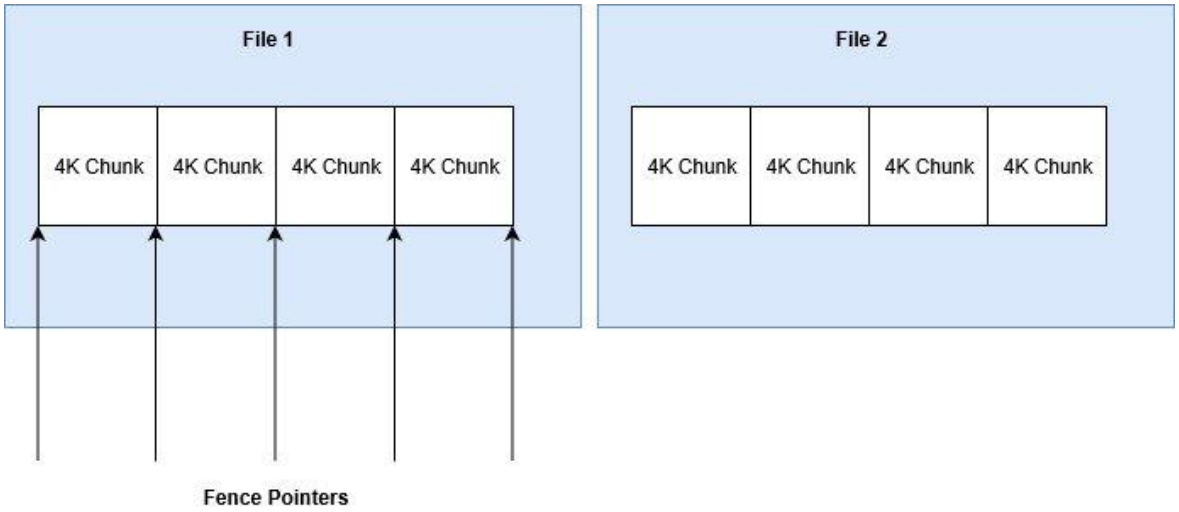
In-Memory Storage

- Array of std::pairs
- Page Aligned
- Locked into memory



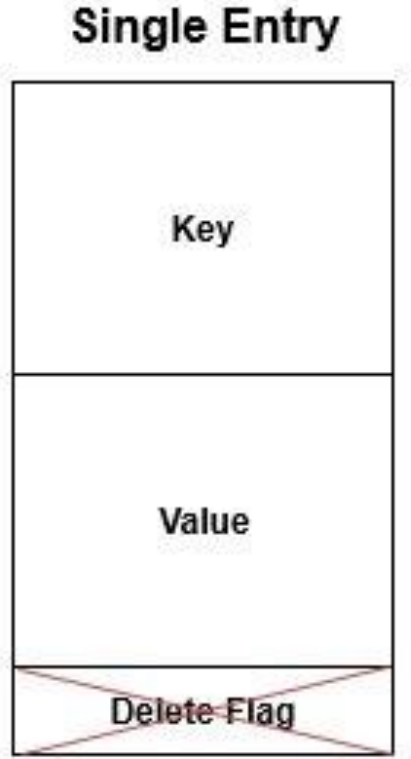
On-Disk Storage

- Multiple files
- Page size chunks
- 2 Pointers Per Chunk



Other Details

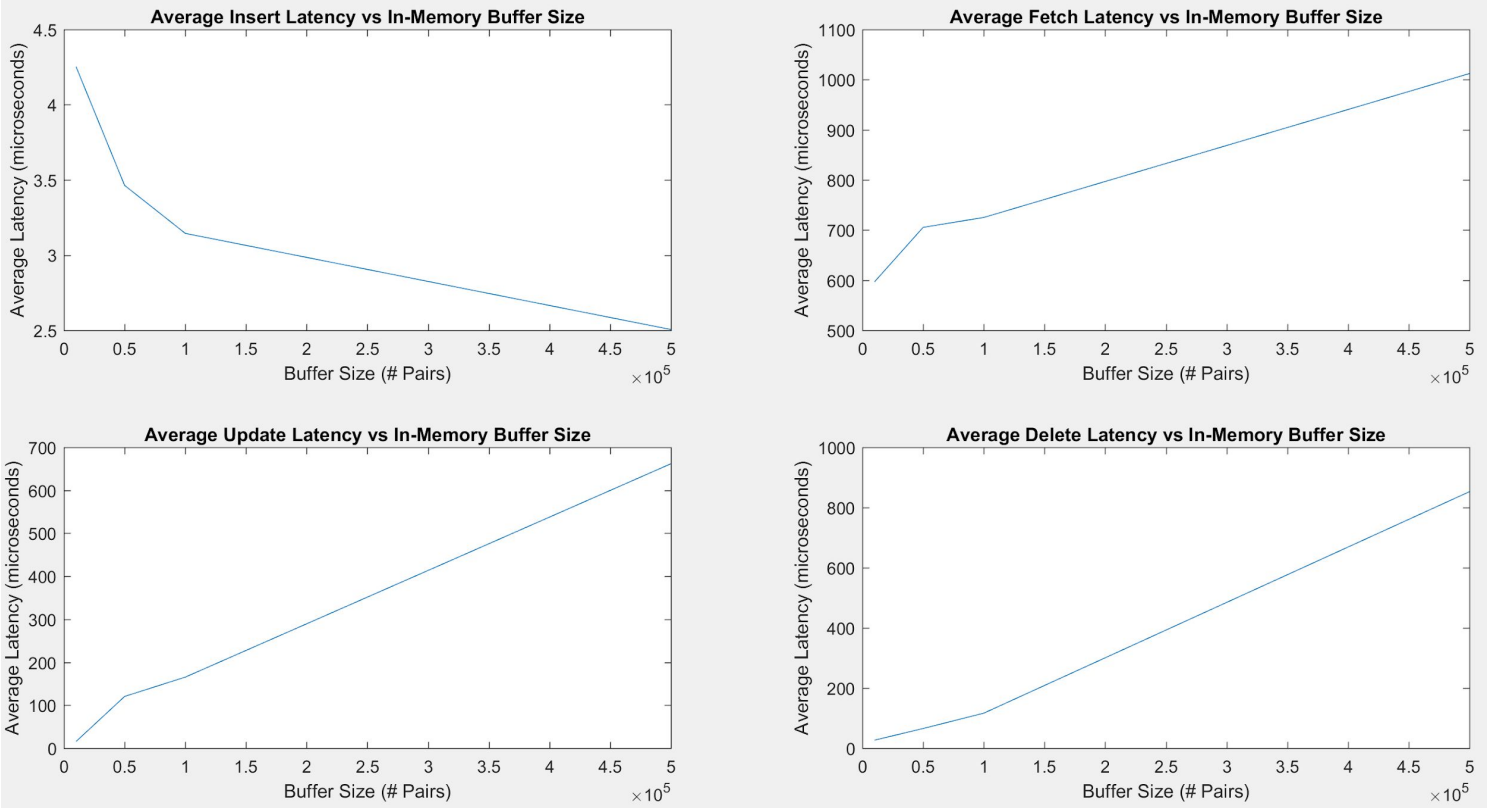
- Delete sentinel values
- Constant Bloom Filter sizes



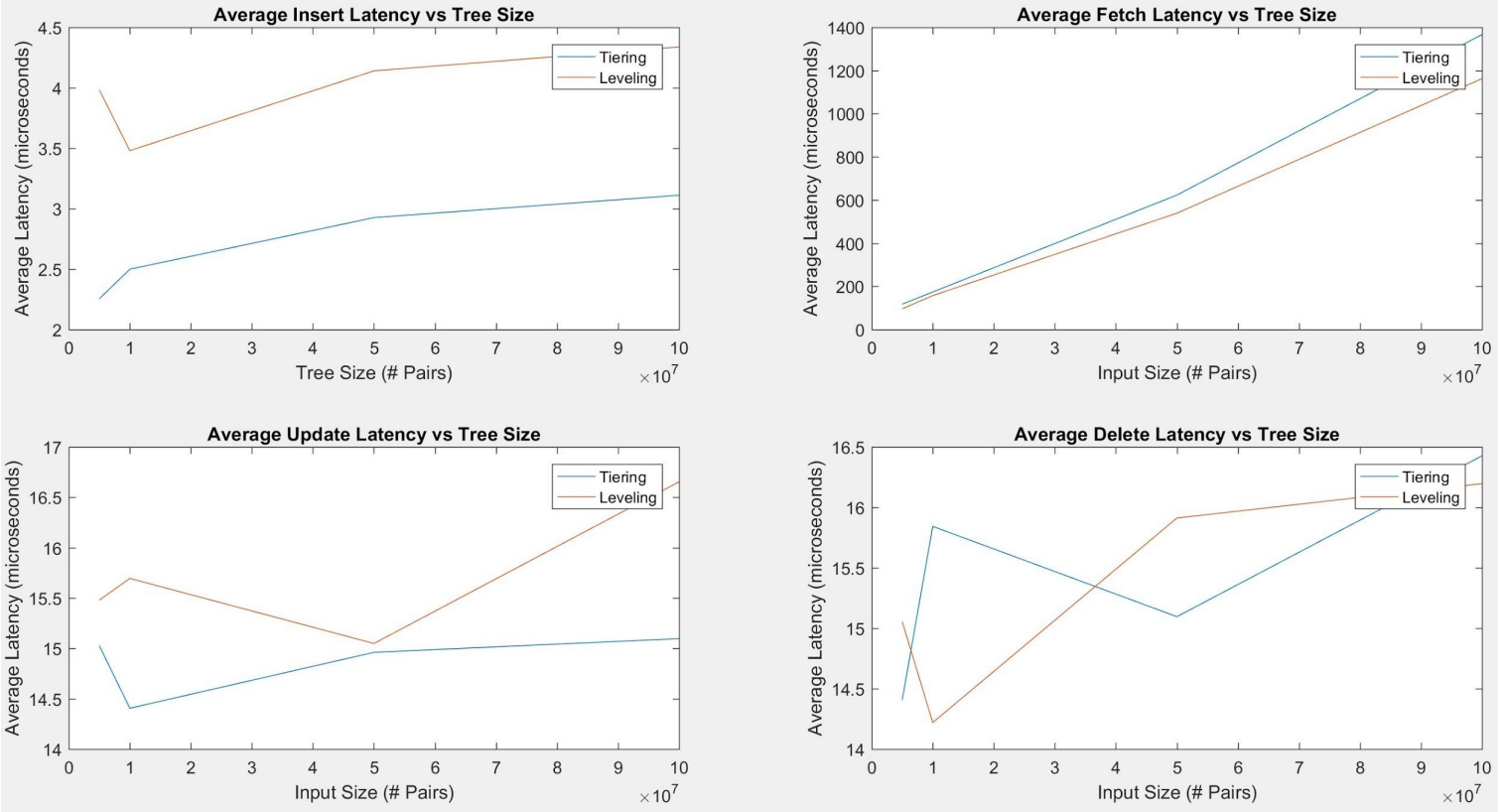
Experimental Approach

- Vary each tunable parameter
- Build tree with uniformly random data
- Time randomly ordered operations
  - 100,000 gets
  - 500,000 puts
  - 500,000 updates
  - 500,000 deletes

Experiment Example 1: Buffer Size



Experiment Example 2: Merge Policy



Experimental Conclusions

- Issues Faced:
  - Cache warming effects
  - Randomization impacts
  - Potential process interrupts
- Generally show expected trends

Image Sources:

http://www.cs.ucr.edu/~vagelis/publications/LSM-secondary-indexing-sigmod2018.pdf

http://sites.ieee.org/futuredirections/2018/10/21/x-2/

https://thenounproject.com/term/parameters/972183/