Systems Project: Implementation of an LSM Tree

What if I told you we built…

Read-Optimized

Log(n)!

Write-Optimized

Instant Writes

Small Memory Footprint!

Only a block in memory at a time!

We would be Lying!

But now we have your attention...

Presentation Overview

- Introduction to LSM trees
- Design Goals
- System Specifications
- Experimentation

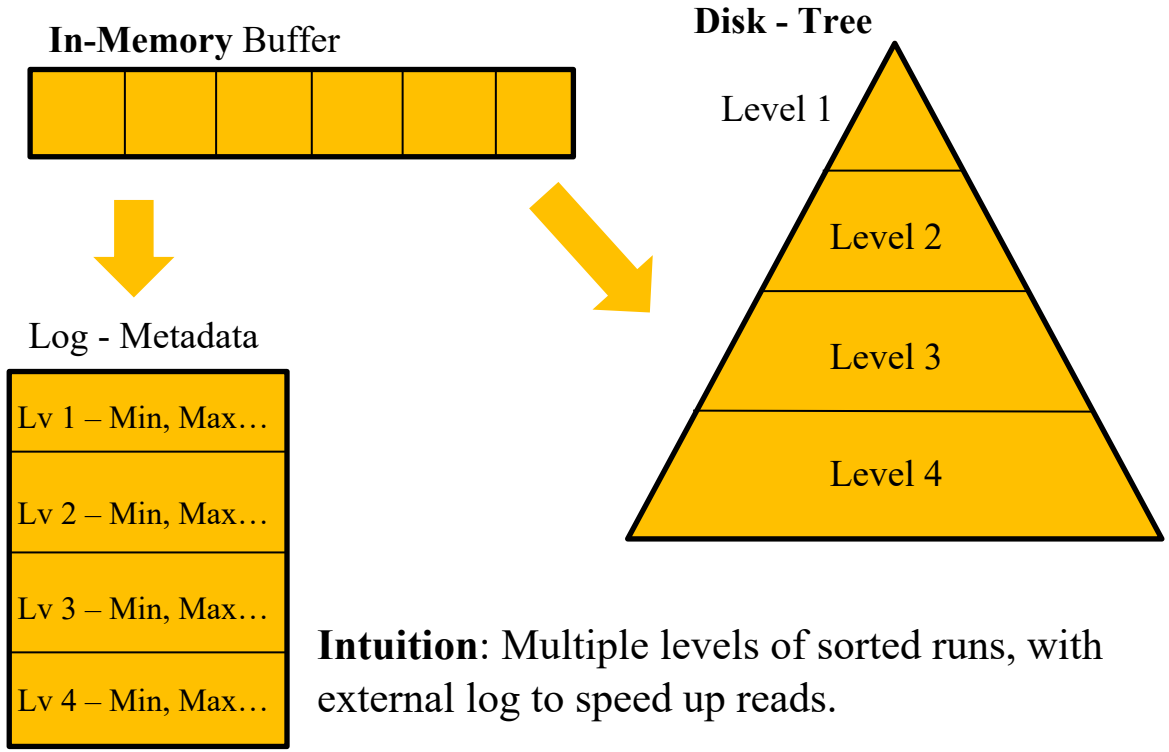
1 Introduction

What we actually built

LSM - Tree:

Data Structure that performs well in reads and vastly outperforms B-trees in writing.

Why? Because avoids constant dispersed update operations



Intuition: Multiple levels of sorted runs, with external log to speed up reads.

Design Goals

Full Functionality

Support operations:

- Write
- Update
- Delete
- Point Read
- Range Read

Merging Policies

- Tiering:
  - Flush Now, Merge tomorrow
- Leveling:
  - Flush now, Merge now.

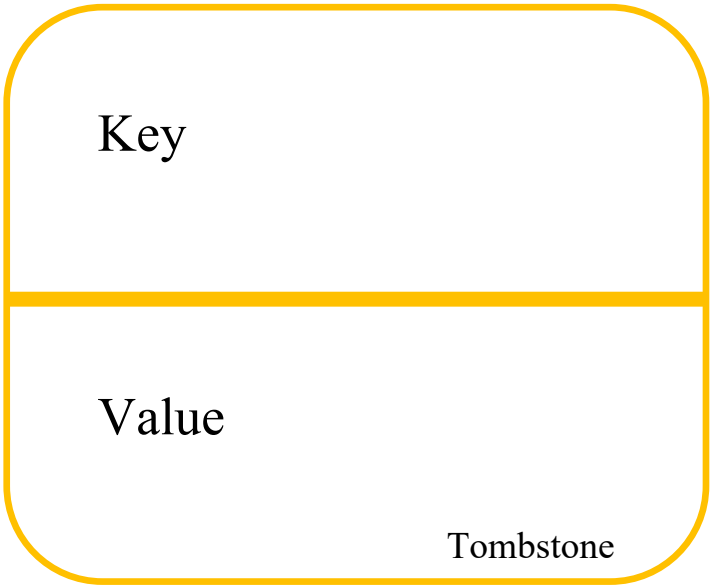
Performance

The trinity of Data:

- Fast Writes (the whole reason behind LSM)
- Decent Reads (specially using leveling)
- Low Memory Footprint

2 The System

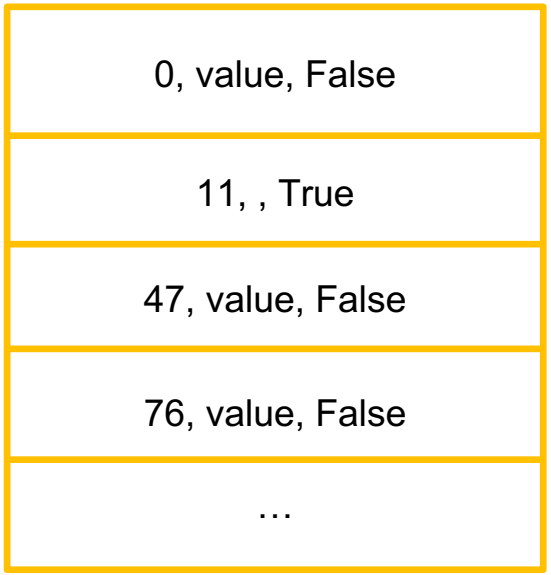
System Description



The Smallest unit

- Unique key: Duplicates are interpreted as updates
- Value: String
- Tombstone: Supports deletion

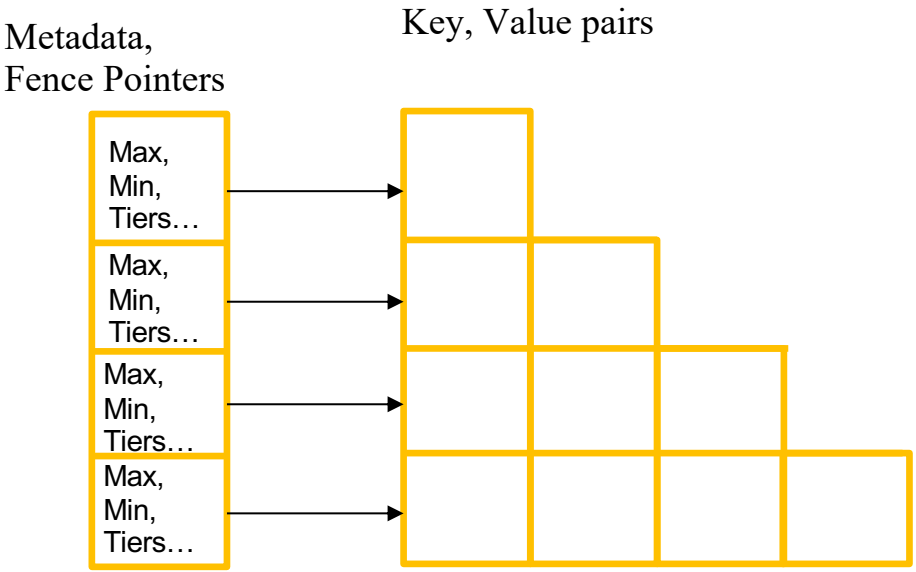
System Description



The Buffer Class

- Tunable size by user
- Sorted inserts
- No duplicates: On time Deletions and Updates
- Merging-policy specific flushing

System Description



The LSM Class

- Policy Specific Drivers
- Reads Support – Fence Pointers
- Writes – Overview
  - Flushing Buffers
  - Flushing Levels
  - Merging Levels/Tiers

3 Experimentation

What we asked:

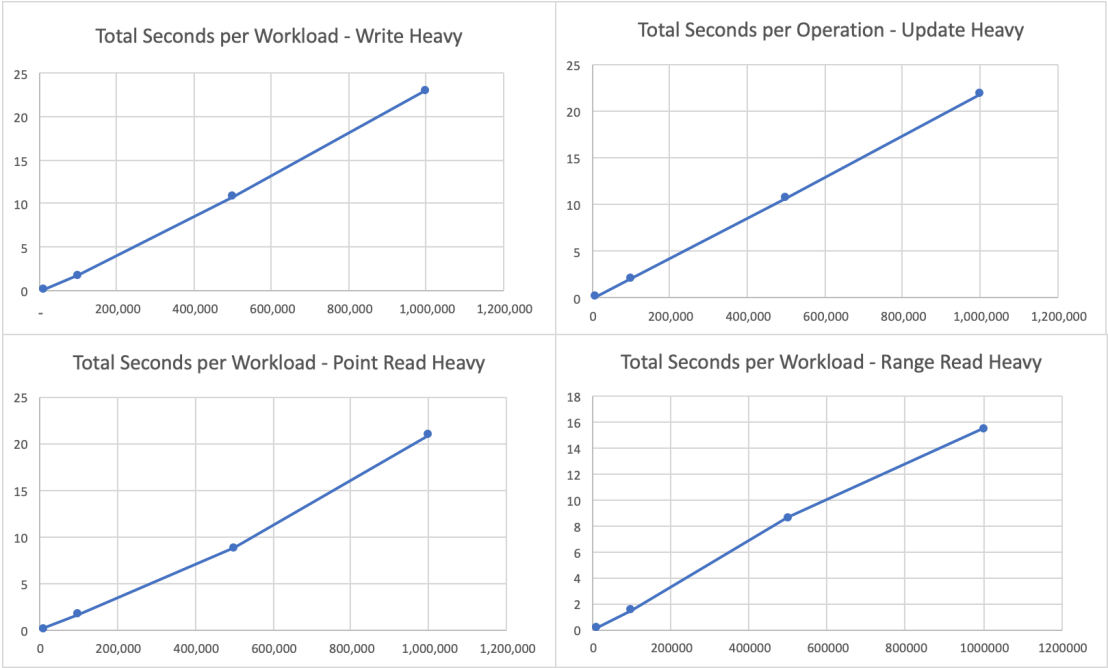
Scalability: System performance

- How does performance scale with the number of operations?
- How does the workload impact the scalability?

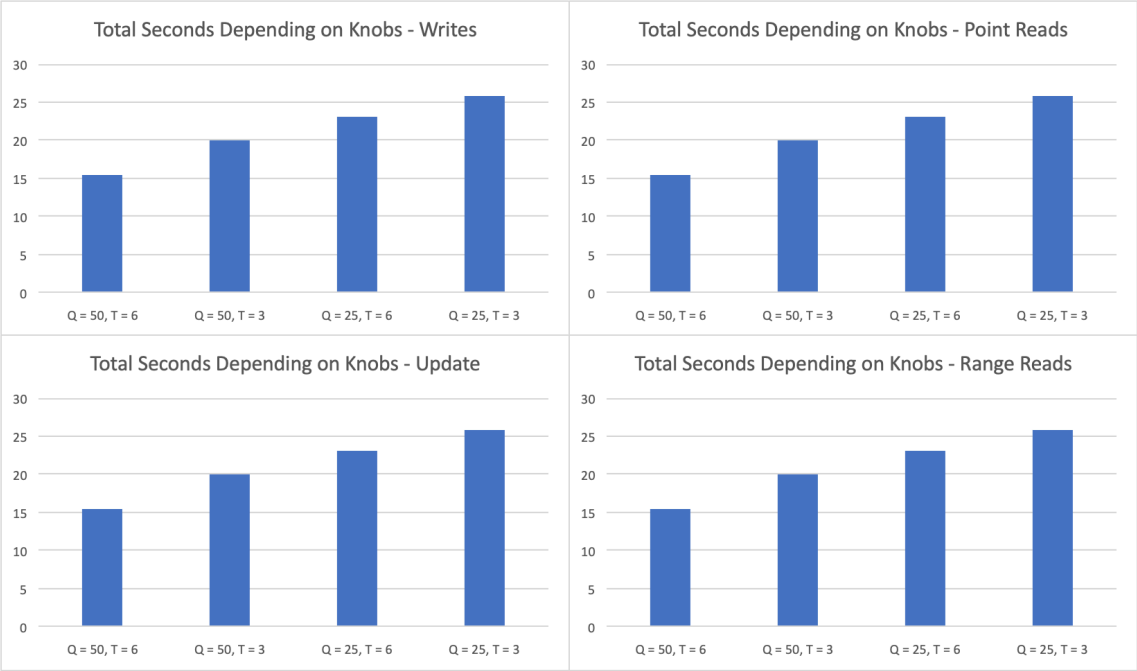
Tuning: Buffer Size and Ratio

- What is the impact of changin this values on our system?
- What values are best for read intensive workloads?
- What values are best for write intensive ones?

Results - Scalability



Results - Knobs



Thanks!

Any questions ?

Appendix

		Tiering			
Operation	Number of Operations	Q = 50, T = 6	Q = 50, T = 3	Q = 25, T = 6	Q = 25, T = 3
Write	10,000	0.137487	0.217731	0.204025	0.275852
	100,000	1.723890	2.118570	2.450830	2.943550
	500,000	10.784900	13.165600	14.901100	18.458500
	1,000,000	23.005700	28.546000	28.834700	35.180900
tes - Update/Delete He	10,000	0.148235	0.180082	0.219480	0.268841
	100,000	2.042160	2.279290	2.646050	3.083380
	500,000	10.719600	13.261000	14.004200	17.611300
	1,000,000	21.879200	29.099300	31.239100	37.327200
Point Reads	10,000	0.147790	0.188584	0.169152	0.226547
	100,000	1.716690	2.134470	1.999680	2.515850
	500,000	8.860930	11.631600	11.857000	13.502600
	1,000,000	20.908600	24.368100	25.018700	30.041700
Range Reads	10,000	0.147096	0.166553	0.168165	0.199130
	100,000	1.485910	2.692720	1.764910	2.865070
	500,000	8.652140	10.179600	10.640000	11.784300
	1,000,000	15.519000	20.028000	23.126100	25.806000