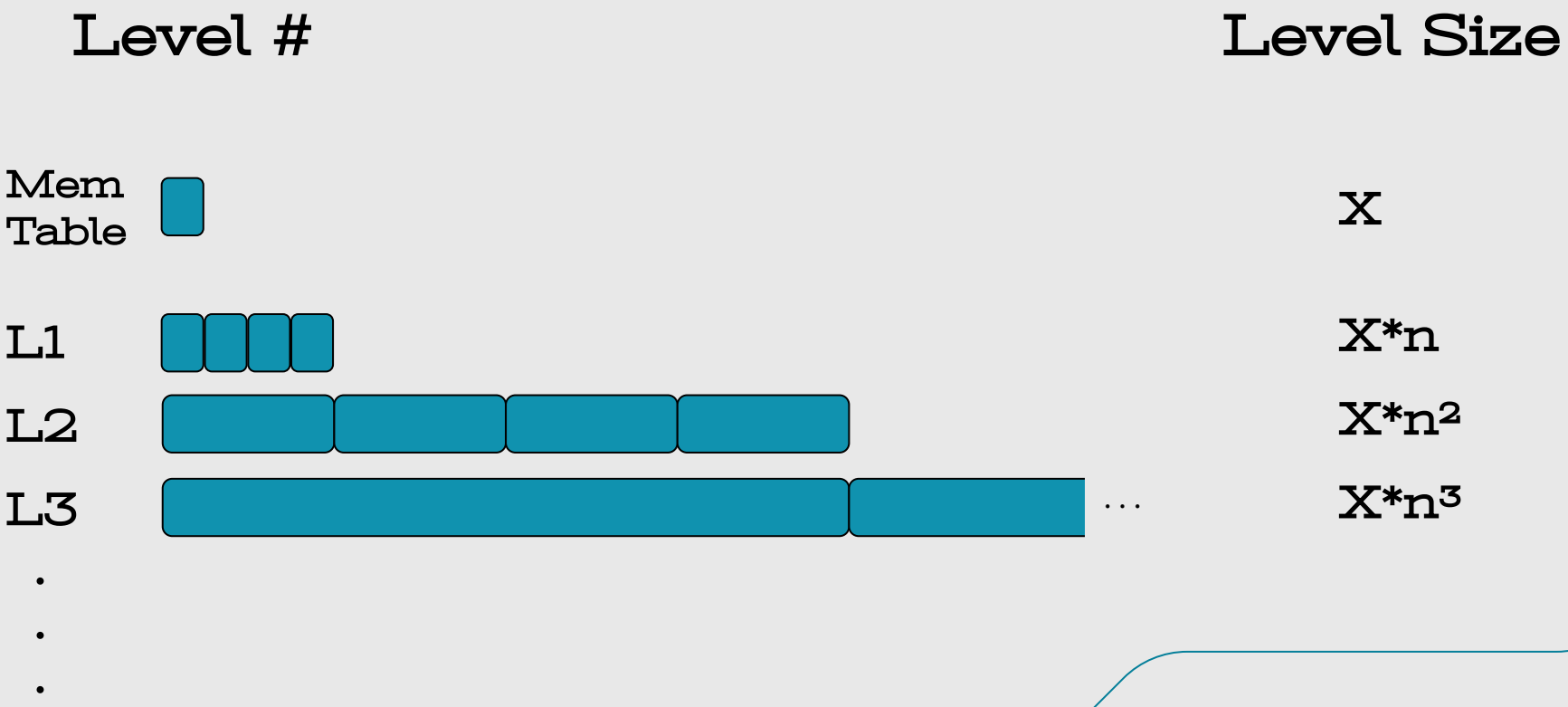
Implementation of a Log-Structured Merge (LSM) Tree

A Systems Project by:
Adit Mehta
Amara Nwigwe
Huda Irshad
Satha Kitirattragarn

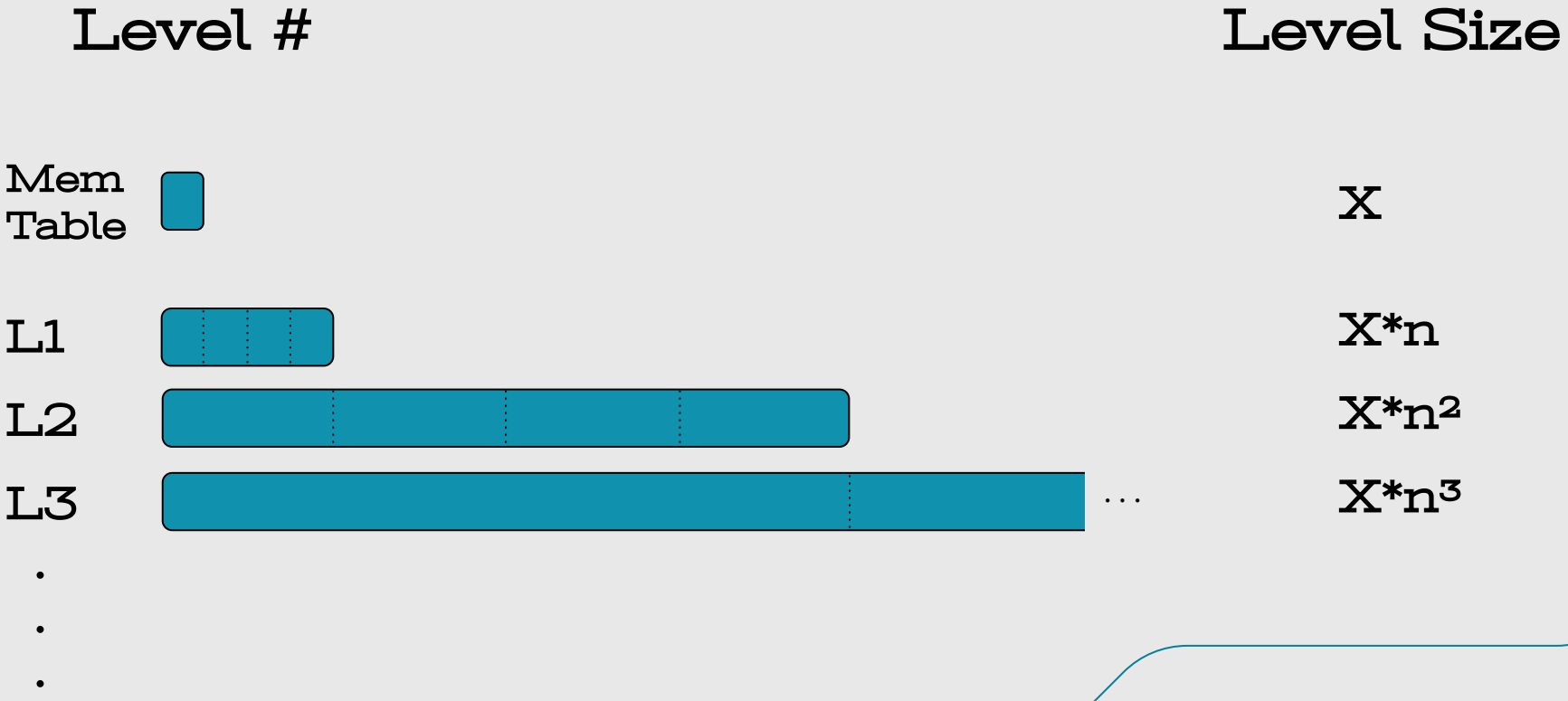
01/

Overview

Tiering



Leveling



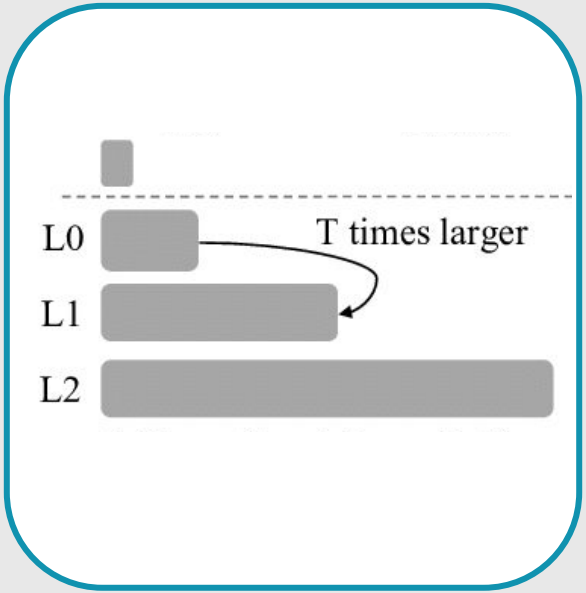
Operational Policies

- Put → Point-Insert
- Write → Bulk-Insert
- Get → Point-Query
- Scan → Range-Query, Bulk-Query
- Delete → Point-Delete, Range-Delete

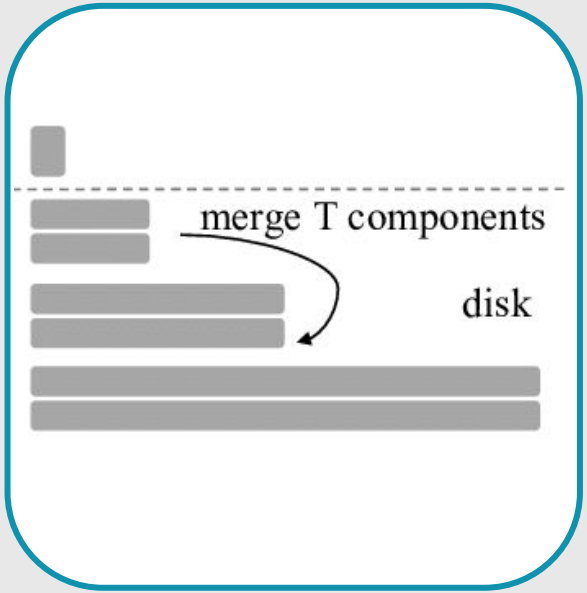
02/

Interests/ Challenges

Compaction Implementation



Leveling



Tiering

Design vs Code

Code:

- Variables
- Functions – modularity
- Accessibility
- Code Paths

Design:

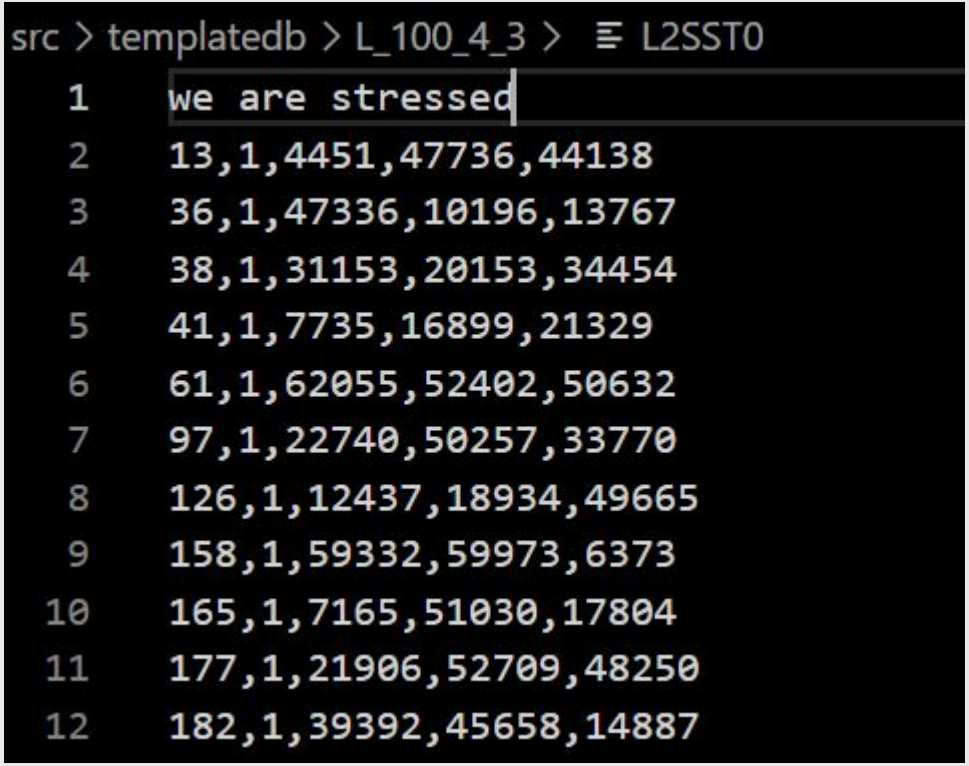
- Physical Layout
- Theoretical transaction policies

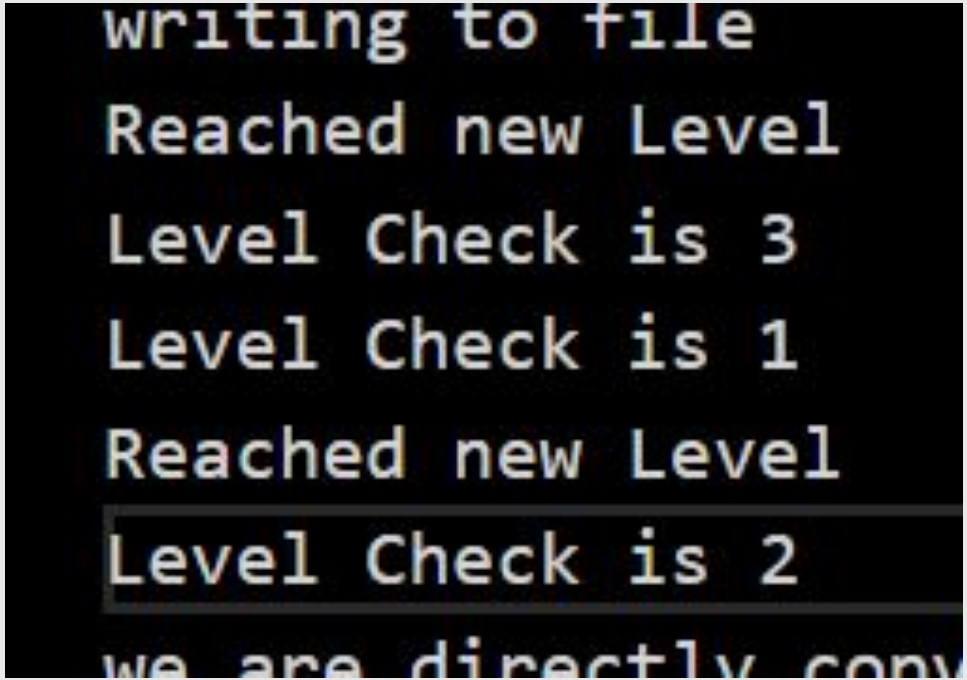
03/

Experimental Highlights

Dynamic Leveling

Does anyone recognize what seems odd?





Mystery Recursion

When it came to setting up tiering on our LSM tree, we were trying to create new files if levels were full and setup a recursive call. That caused us to see the strange level order you see on the right.

04/

Lessons Learned

Lessons Learned

01/

Makefiles are the best.

Saves a lot of time that would o/w be spent executing commands.

02/

Small progress, big difference.

Incremental progress boosts overall progress as much as group's morale.

03/

One door closes, another opens.

Lots of errors from not keeping track of open() and close().

04/

Taking breaks helps.

When unable to debug, taking the focus off work for a bit freshens you up.

05/

To squish a bug, think differently.

Bugs lie where you fail to notice. Think from a different perspective to catch them!

06/

When unsure, visualize.

Drawing things out when unsure of the logic helps.

Sincere Advice

- **Start work early**. Brush up on required programming knowledge before anything else.
- Before writing code, always **draw** the logical/conceptual design out first and foremost!
- Always **test out small parts of code** to ensure that it's working properly before moving on to other parts..
    - You'll be tempted to move on to work on subsequent parts even when the current part is still bugged. Resist the temptation and **focus on debugging the current part**.
- **Print statements** (printf's/cout's) are always handy for debugging.
- For each part done, **detailed** GitHub **commit description** will be vital when you need to check back on code that runs properly.

05/

Conclusion

Progress So Far

So far, our LSM tree works for with our:

- Put policy
- Write policy
- Delete policy
- Tunable parameters (implemented as command line arguments)
- Stored/organized data tables
- Tiering and Leveling, when it comes to inserts

Things to be fixed and brushed up:

- Read policy
- Get policy with tiering*
- Stored data tables with leveling

*a little issue with leveling and tiering when it comes to Queries, some items are found missing and we are working to fix that!

Thank You For Your Attention!