01

ECS 165A Milestone 2

Group 4: Zhengyu Wu, Leran Chen, Tedder Lao, Xingyu Pan, Jiaming Mai

What was accomplished and how?

Goal 1: Build an extension to ensure all data is stored on disk and can be recovered from it and to assume that data cannot fit entirely in memory.

- Bufferpool
- Trash Bin

Goal 2: Make the tail pages are periodically merged with their corresponding base pages on it and to assume that data cannot fit entirely in memory.

- Merge

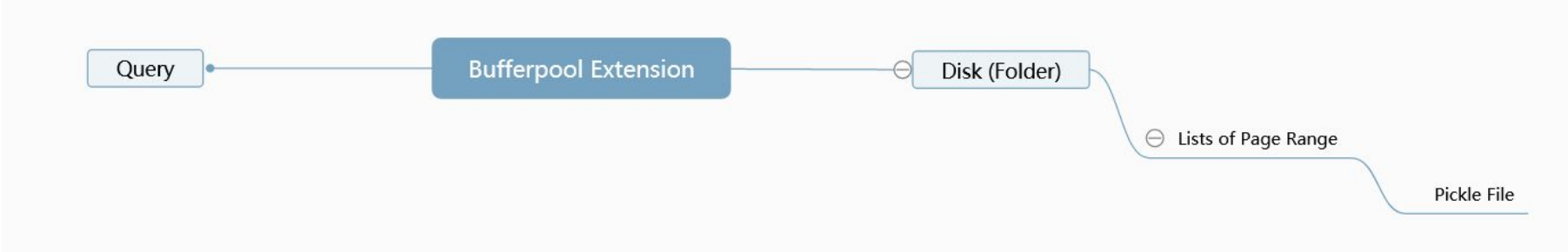
Goal 3: Achieving a faster L-Store Implementations.

- Index

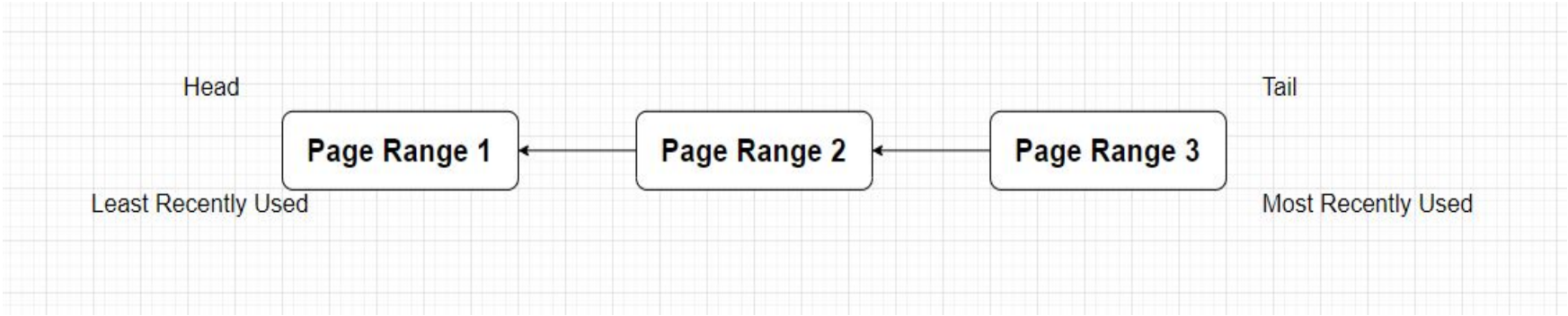
Presentation Highlights

Our Design & Solution

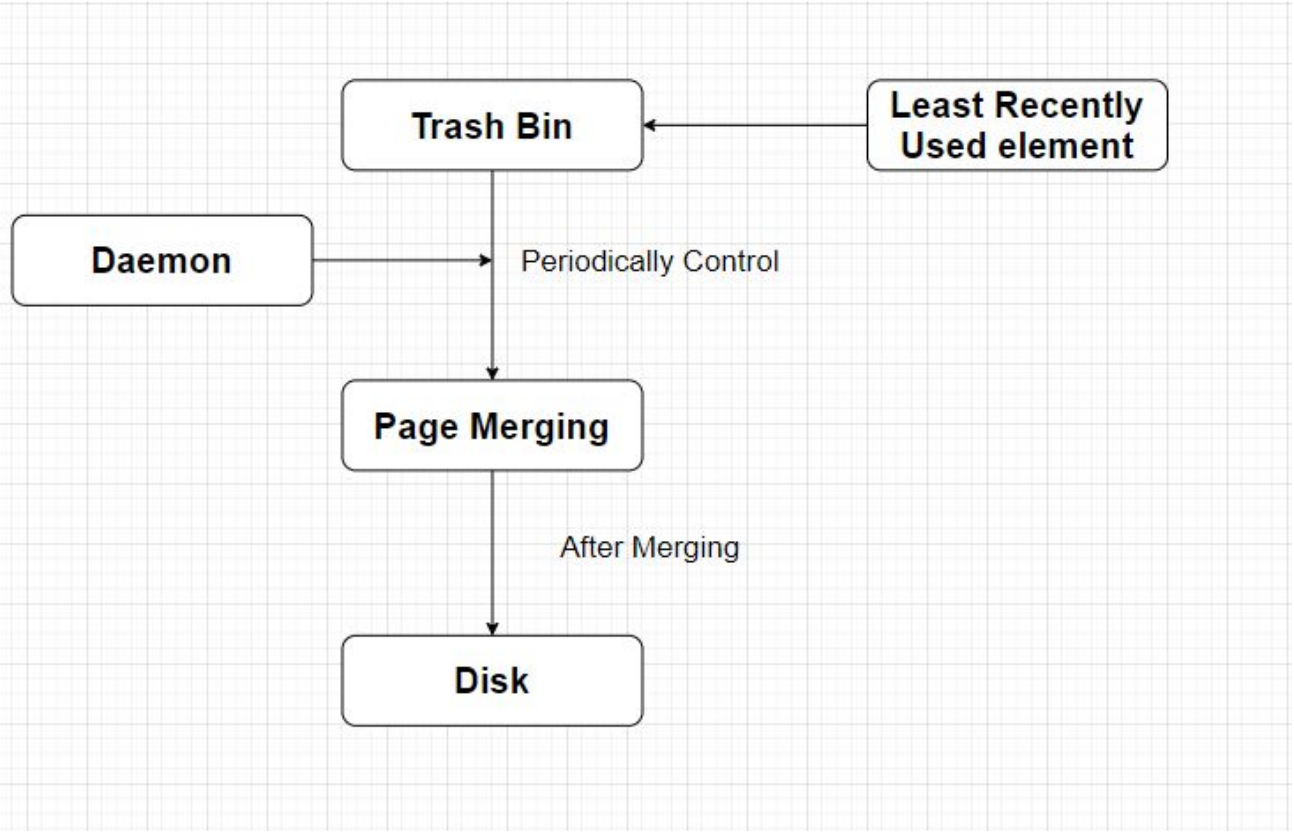
Bufferpool



Inside Bufferpool



Inside Trash bin

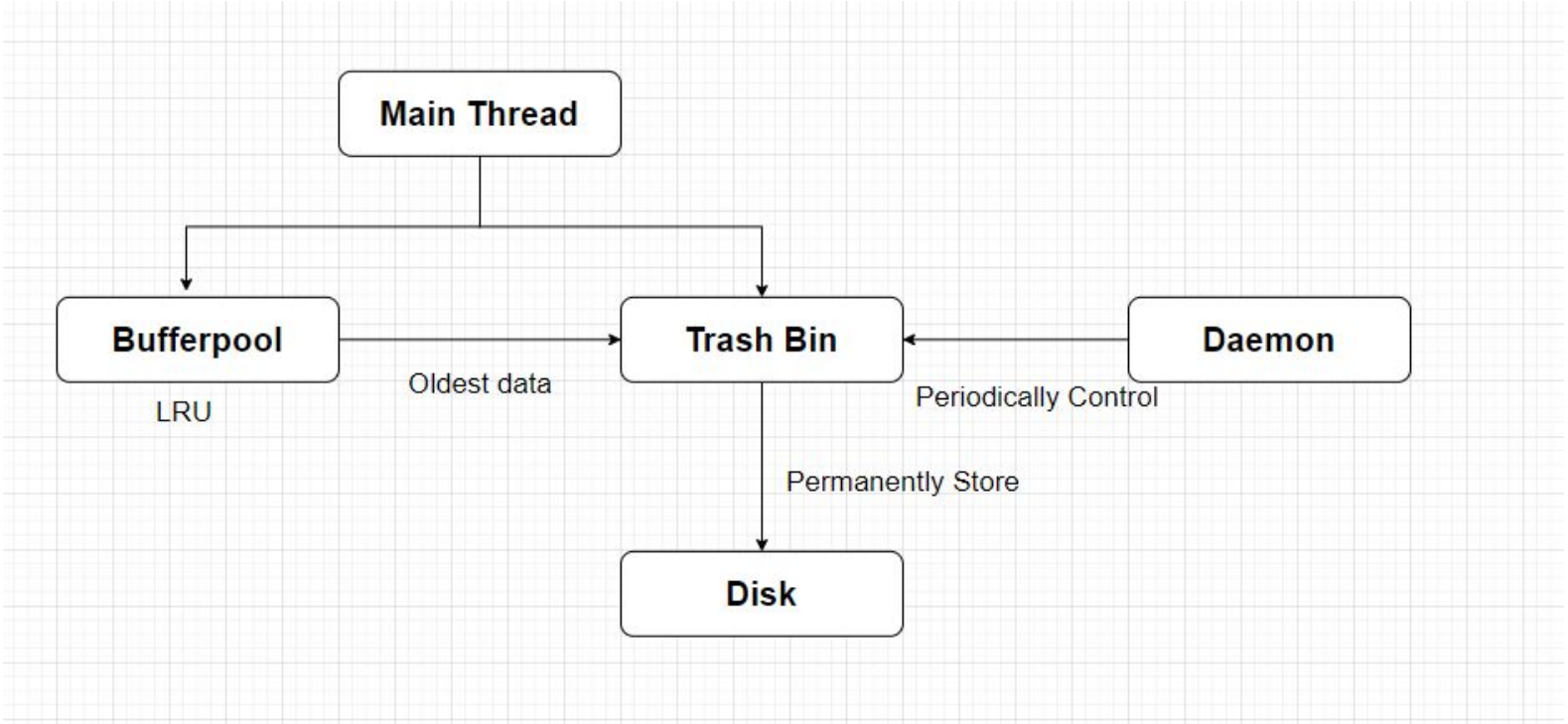


Use of Daemon

Advantages

- Operate missions periodically
- Merge the page in background
- Improve update efficient
- Save system resource
- Provide flexibility

Permanent Store Operation



Future Direction

Bufferpool

- Redesign the transaction operation
- Implement flexible size
- Implement Undo log and Redo log

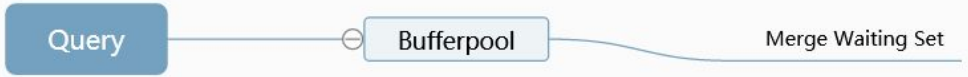
Future Direction

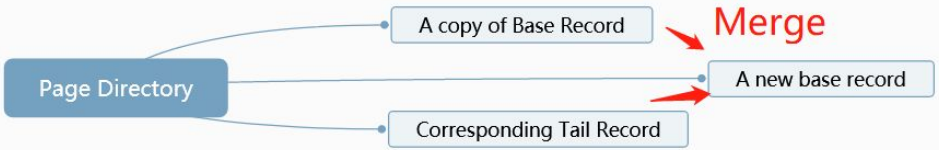
Multi-threads Concurrent Safety

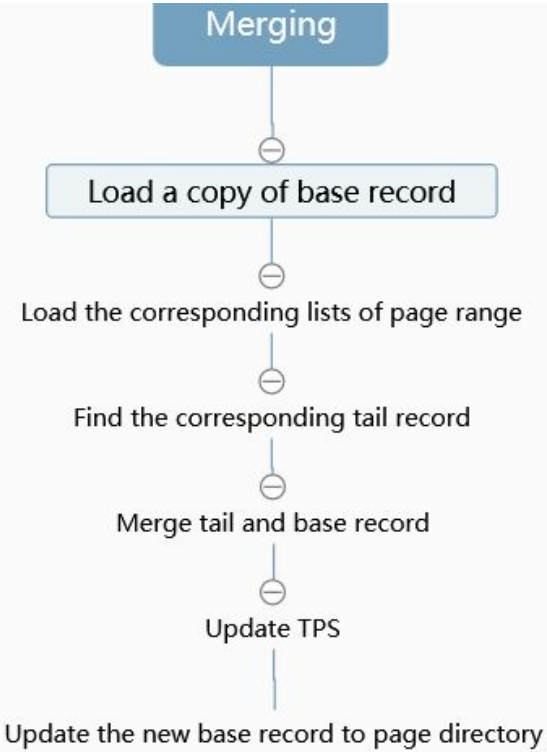
- Mutex Lock and Shared Lock
- Prevent Modifications Lossing
- Prevent Dirty Read
- Prevent Unrepeatable read
- Prevent Phantom read

Our Design & Solution

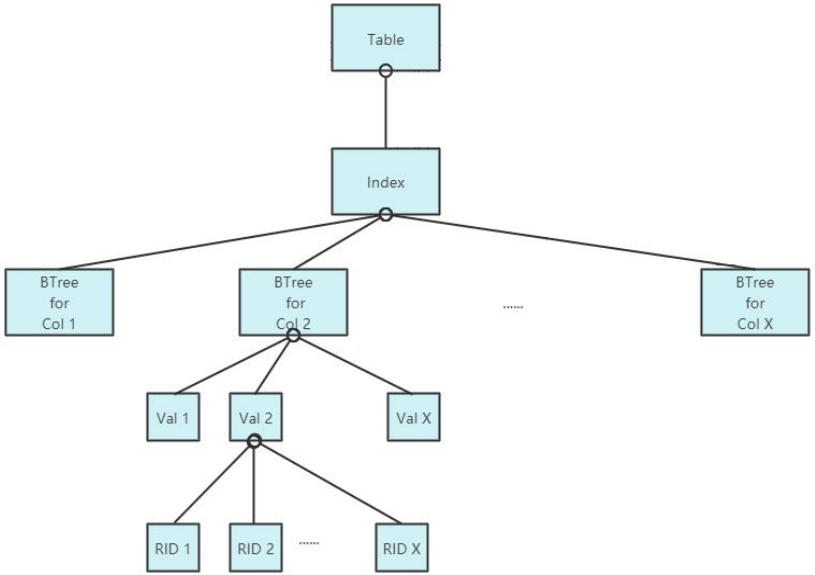
Merging







Data Model - Table -Index



Implementation & Solution:

Index.py

- def locate
- def locate_range

New:

- def insert - insert new index
- def delete - delete new index
- def update - update new index
- def create_index - create new BTree

Thank you