L-Store
Milestone 2

ECS 165A

Natheenthorn Teachaurangchit
Michael Shaw
Stuart Feng
Henry Chou
Eric Wang

Durability & Bufferpool Extension

Save & Load Pages from Persistent Storage

Pinning & Unpinning Pages

Data Reorganization

Merge

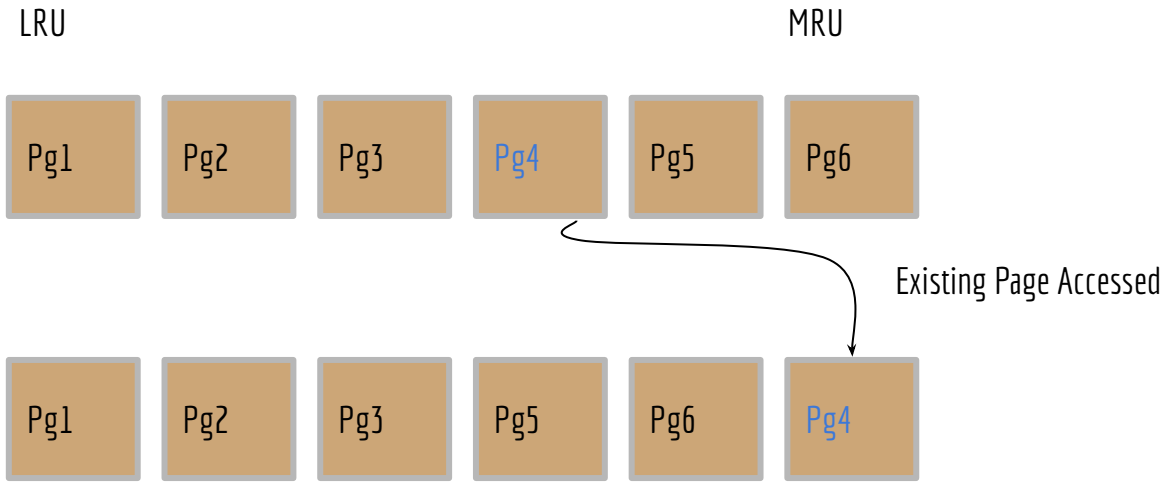
Indexing

B-Tree Index

Durability & Bufferpool Extension

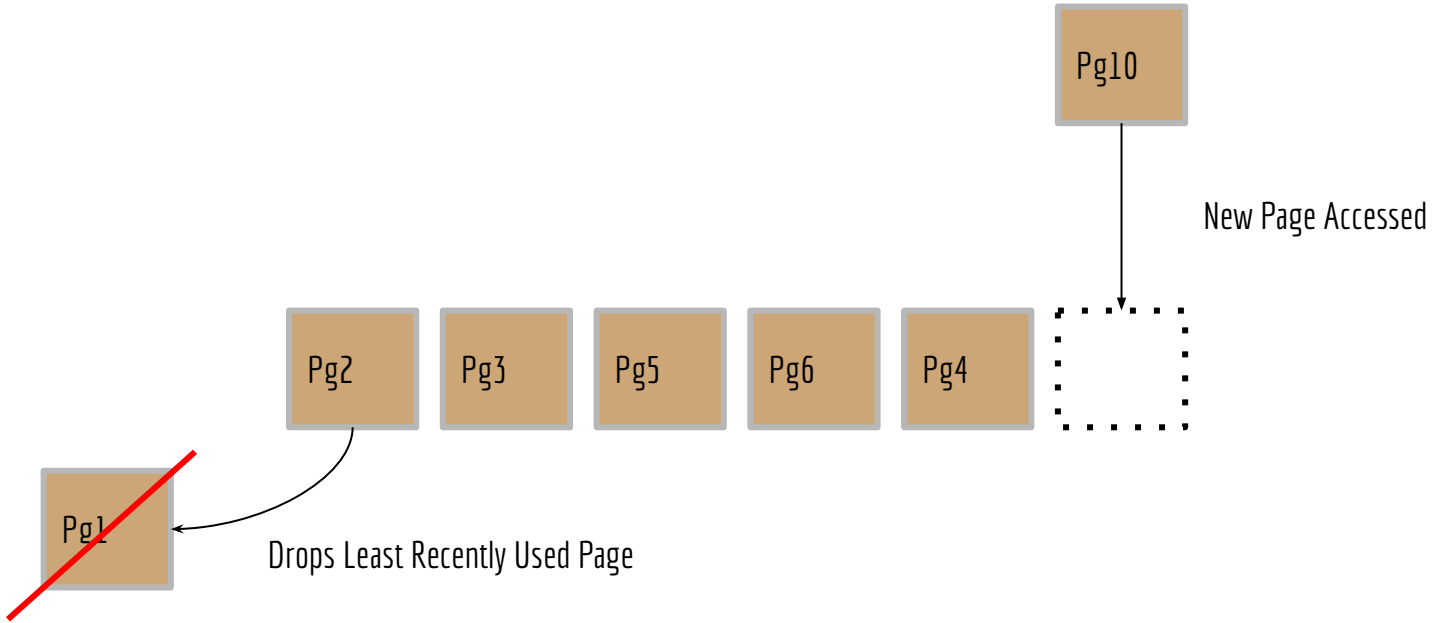
Persistent Storage - Accessing within Bufferpool

Ordered Dictionary



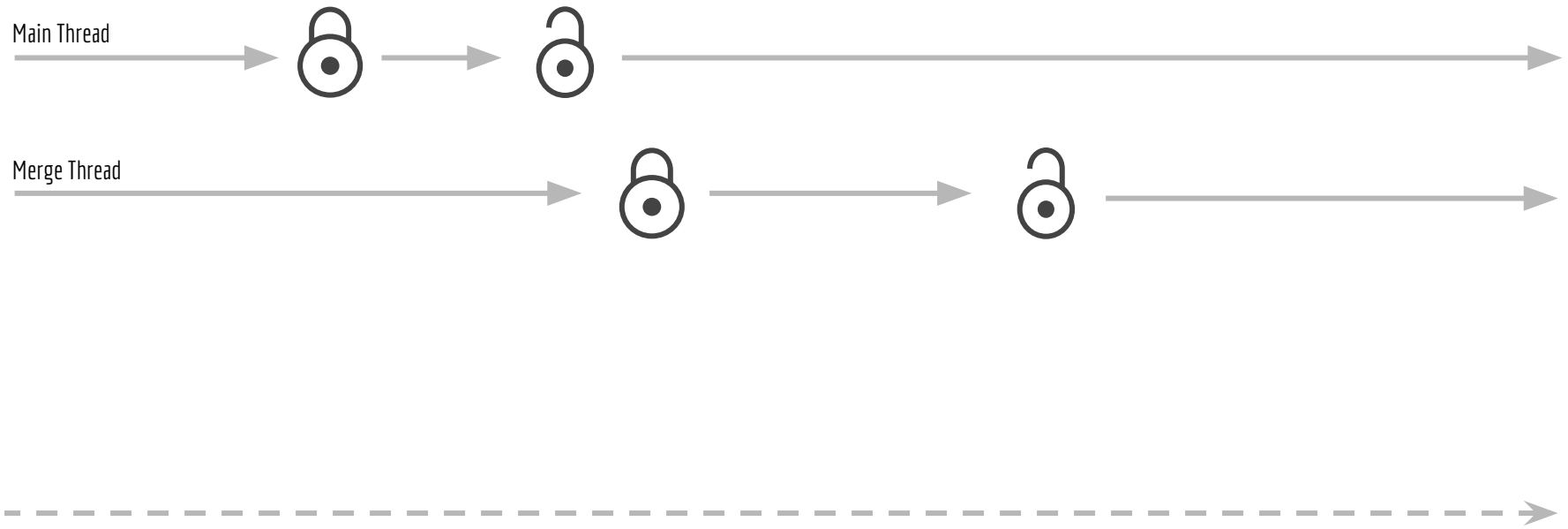
Persistent Storage - Accessing at Capacity

Ordered Dictionary



Pinning & Unpinning

Utilizing Binary Semaphores

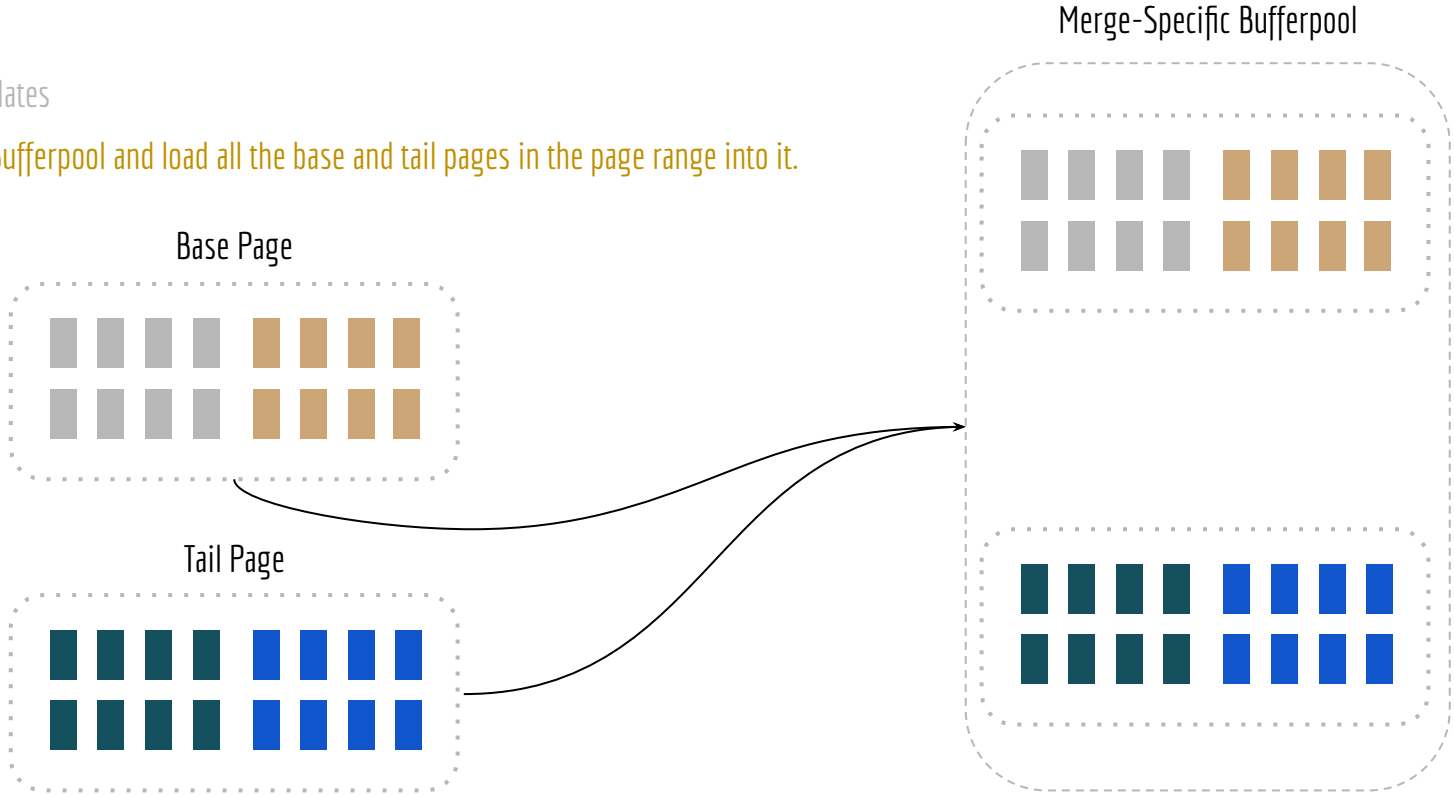


Data Reorganization

Merge

Default Threshold: 2048 updates

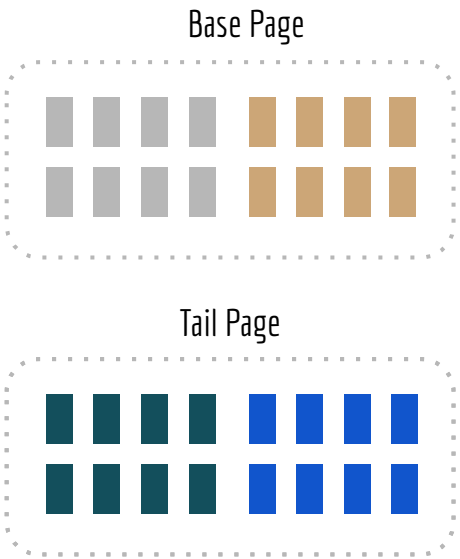
1. Create a Merge-Specific Bufferpool and load all the base and tail pages in the page range into it.

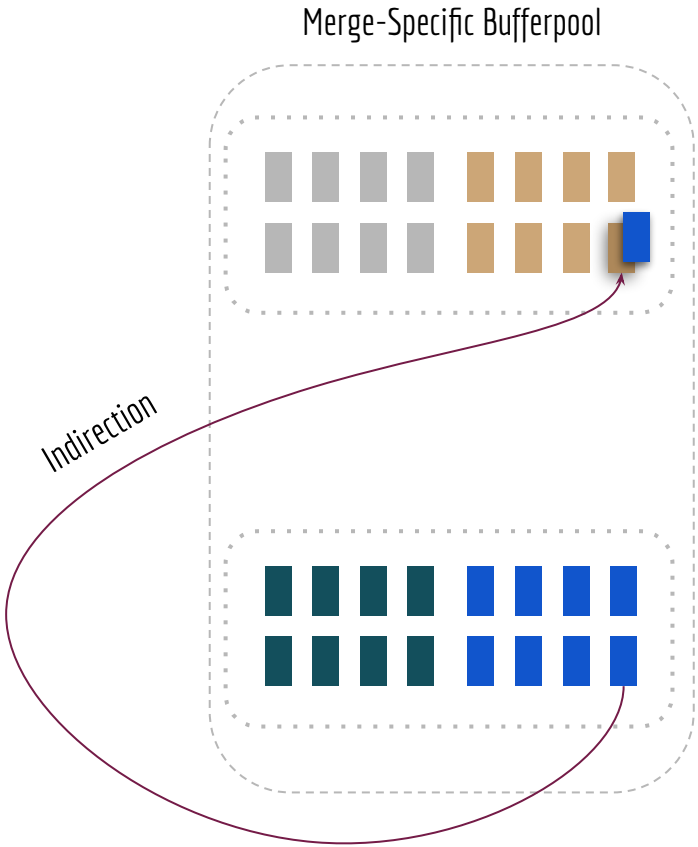


Merge

Default Threshold: 2048 updates

2. Consolidate all the records from bottom up

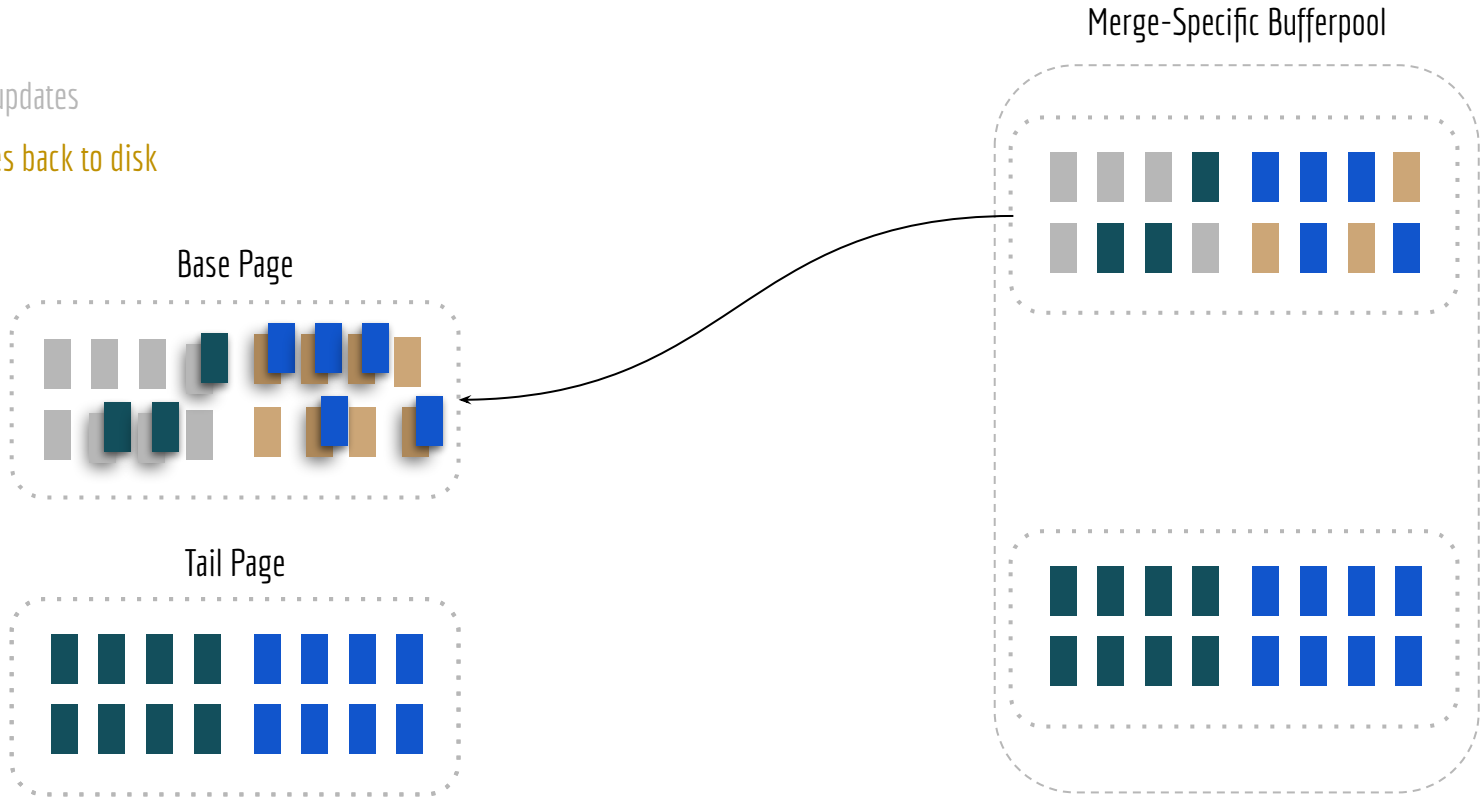




Merge

Default Threshold: 2048 updates

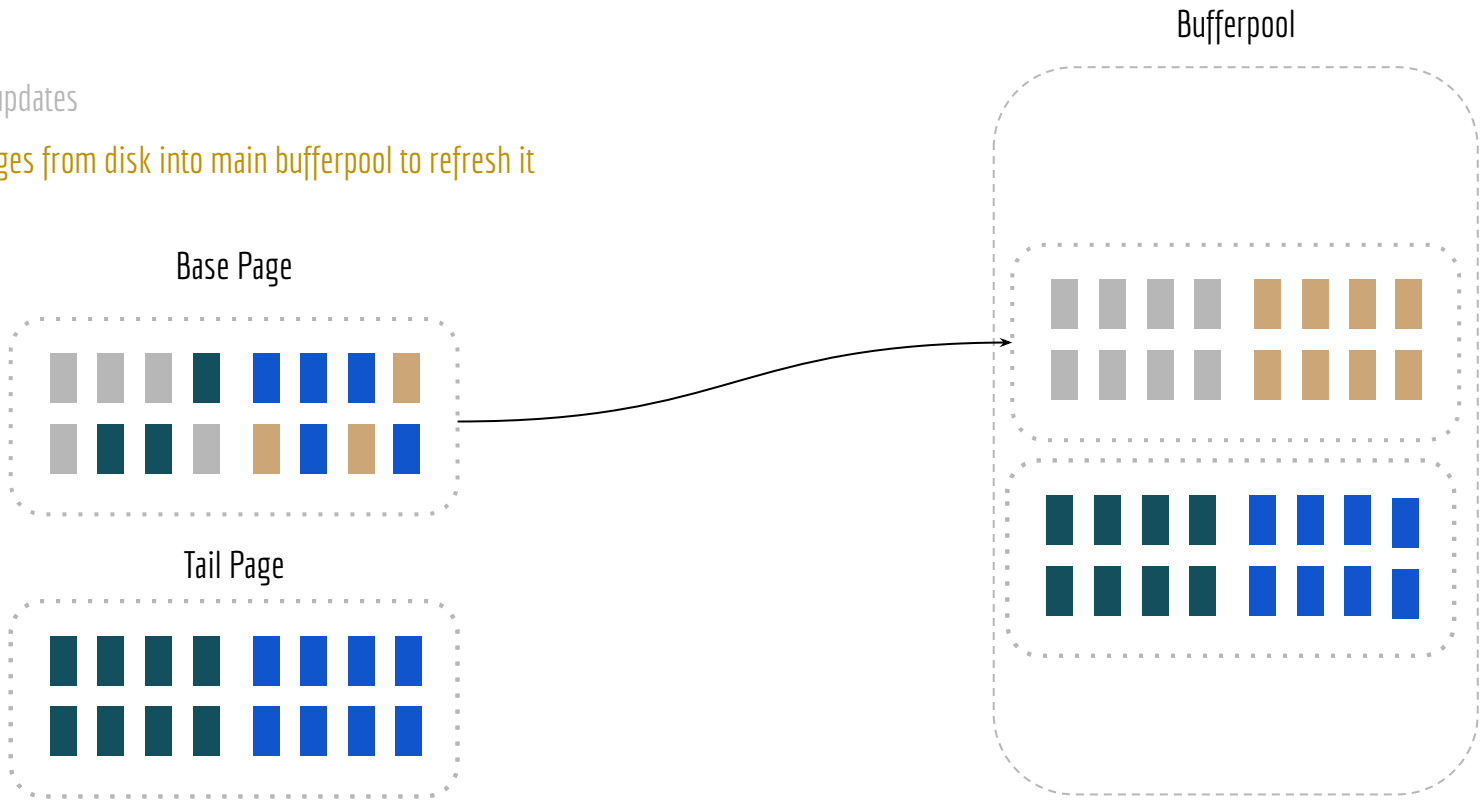
3. Write the updated pages back to disk



Merge

Default Threshold: 2048 updates

4, Load newly updated pages from disk into main bufferpool to refresh it



Indexing

Indexing

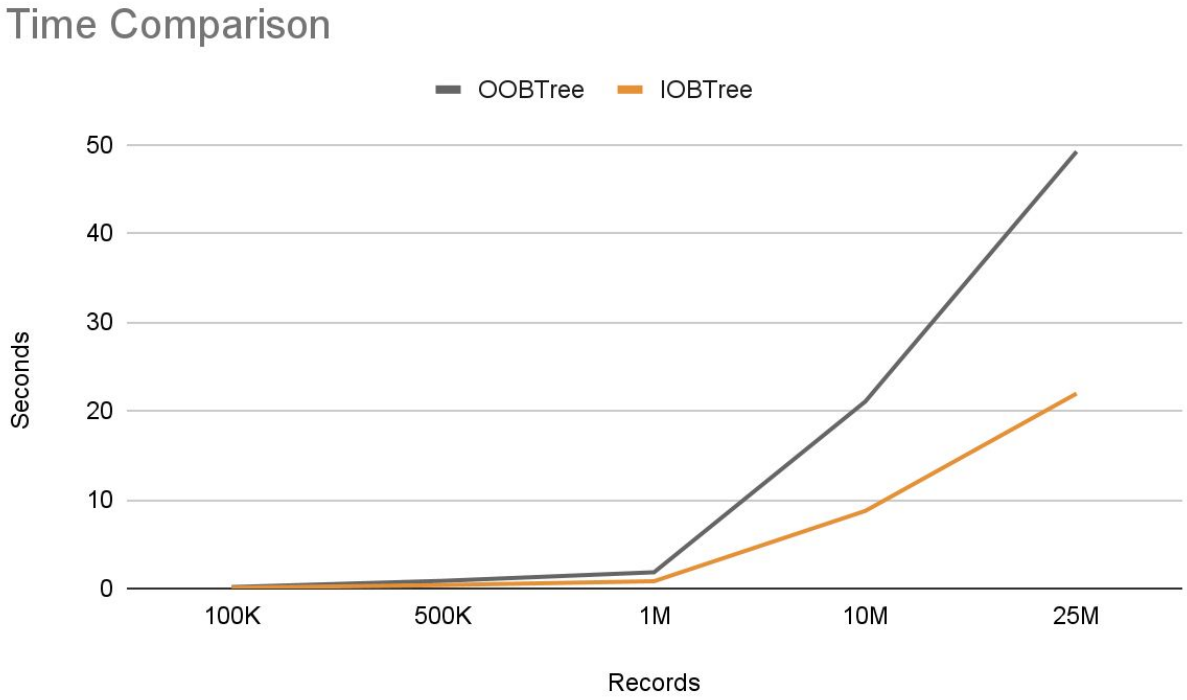
Our BTree previously utilized OOBTree, where both the keys & values are Objects. We switched it to IOBTree, where the key is an Int and value is an Object.

IOBTree is superior to OOBTree both in terms of **memory usage** and **time**.

Benchmark Program:

1. Insert # key-value pairs
2. Select # key-value pairs
3. Delete # key-value pairs

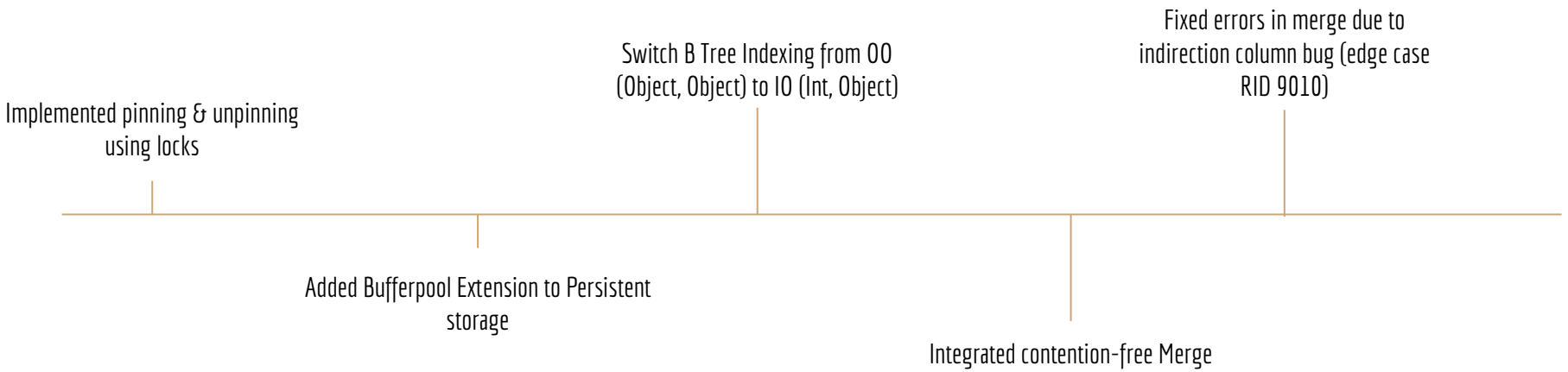
OOBTree vs IOBTree



Indexing

Additional implementation

Roadmap


Implemented pinning & unpinning using locks
Added Bufferpool Extension to Persistent storage
Switch B Tree Indexing from OO (Object, Object) to IO (Int, Object)
Integrated contention-free Merge
Fixed errors in merge due to indirection column bug (edge case RID 9010)

Improvements for Milestone 3

- Complete support for multithreading
- Organize code
- Test different values in configurations for better performance

Thank you!