Bf-Tree: A Modern Read-Write-Optimized Concurrent Larger-Than-Memory Range Index

Xiangpeng Hao*
University of Wisconsin-Madison

Badrish Chandramouli
Microsoft Research

*Research done during internship at MSR.

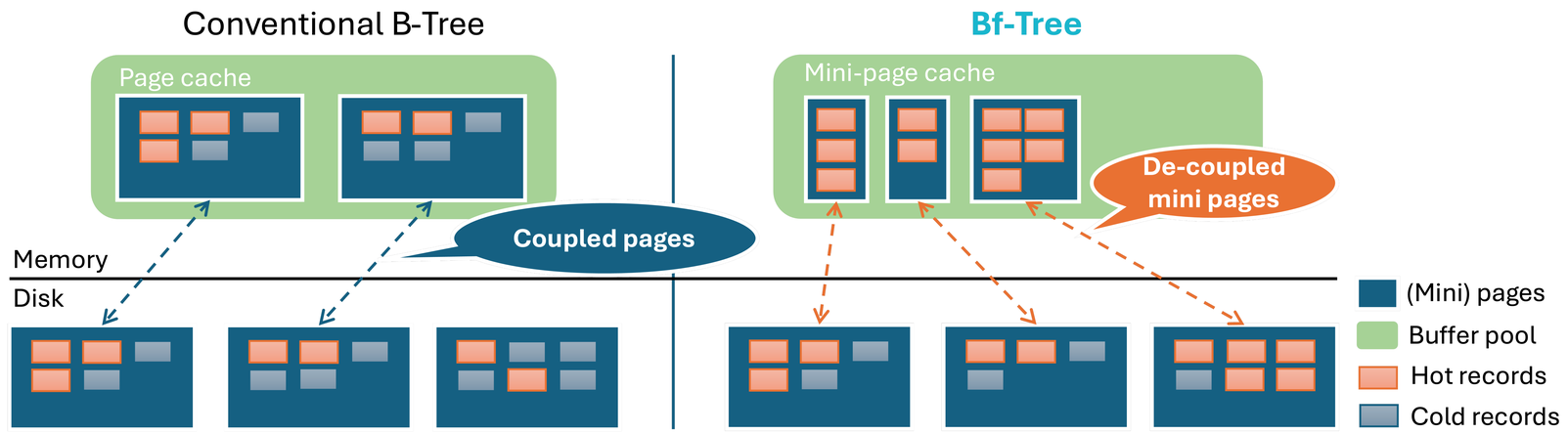
B-Tree Problems

Coarse-grained caching granularity.
Hot records are cached along with cold records.

Incurs write amplification:
small modification → full page write.
Problems exacerbated for secondary indexes (small keys and values)

Root problem: fixed sized pages (disk).
B-Trees organize data into pages.
Pages are much larger than records.

Key Idea: Decouple Cache Pages from Disk Pages



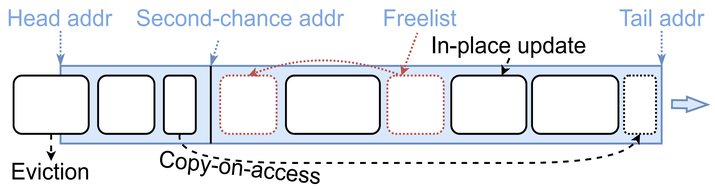
The Mini-Page Abstraction

Cache individual records - read & write
Cache records at small multiples of cache-line granularity, instead of full-page granularity.

Buffer recent writes, in-place updates.
Grow to buffer more writes.
Shrink to flush records to disk.

Cache range gaps or logical key ranges.
Reduce unnecessary disk lookup.
Improve negative lookup.

Novel Mini-Page Buffer Pool



Variable-length buffer pool duties
- Grow/shrink mini-pages
- Disk interactions
- Identify hot/cold records of a mini-page
- Identify hot/cold mini-pages
- Memory management

Design inspired by FASTER hybrid log.

Outperforms B-Trees & LSM-Trees