ACM SIGMOD 2011 Programming Contest

A Durable Main-Memory Index Using Flash

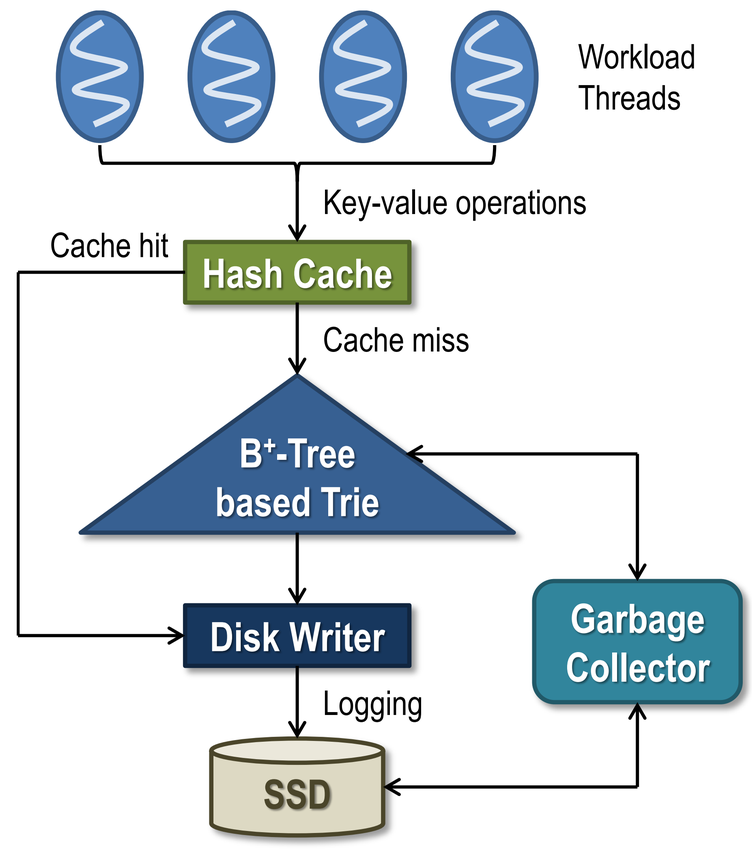
Team *greensky*

Jung-Sang Ahn, KAIST (Korea Advanced Institute of Science and Technology)

The Task

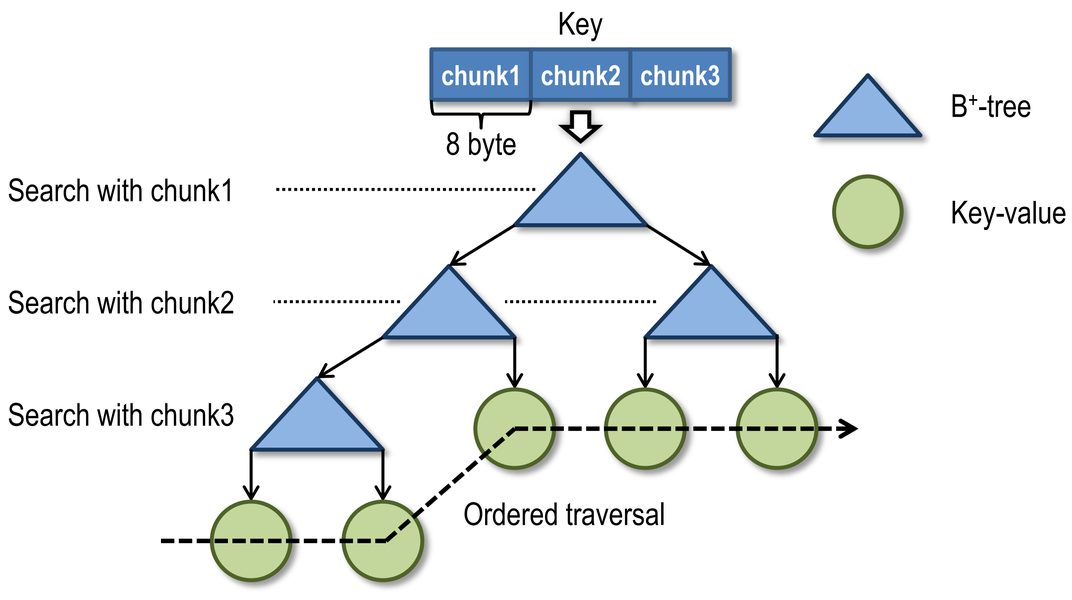
- To implement high-throughput main-memory index
- All completed updates must be durable using SSD
  - Recoverable after system crash
- Stores key-value pairs
  - Key (unique): up to 1024 byte
  - Value: up to 4096 byte
- Offers 5 key-value operations:
  - Insert, find, delete, compare & swap, and iterate (lexicographical ordered traversal)
- Should support high-concurrent accesses
  - All single-key operations must be atomic
- SSD is formatted using ext4 filesystem

Architecture Overview

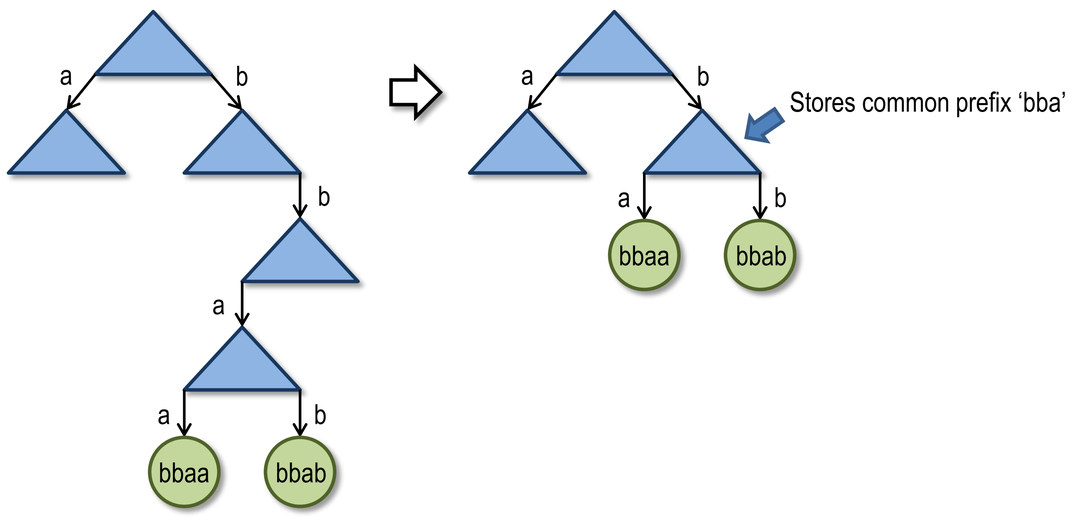


Main Structure: B+-Tree based Trie

- Trie uses B+-tree as a node
- Key is split into 8-byte chunks
  - Each chunk is used as a key for each level of B+-tree



- Skewed trees are merged into one tree
  - To speed up traversing common prefix

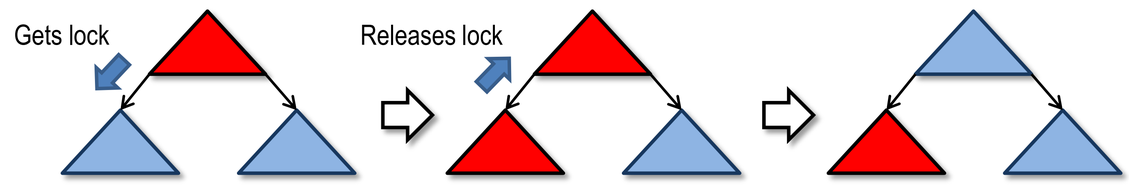


Hash Cache

- Maps from key to key-value object for fast lookup

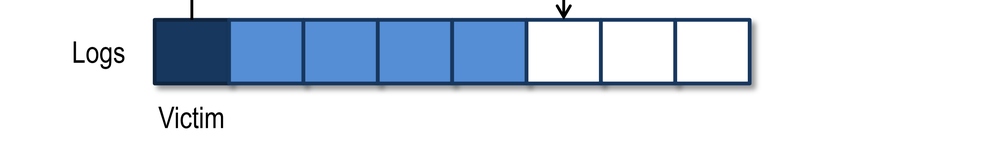
Concurrency Control

- Tree level readers-writer lock
- Only downward propagation
  - To avoid deadlock



Disk Writer

- All update operations send redo log to Disk Writer
  - Delete does not include value-related fields



- Logs are queued to be written in bulk
  - Associated operations are completed after writing is done
- Recovery: redoing all written logs

Garbage Collector

- Reclaims invalid logs to gather free space
  - Victim selection: round-robin policy