KV-store

overview

- **Key-value storage: simple yet widely used(P7)**
- **Key-value storage is similar to file system(P8)**
- **Treating File System as Key-Value storage is inefficient(P10)**

A naïve Key-Value Storage (KVS)(P13)

- **Update(K,V) in the naïve KVS(P14-15)**
- **Insert(K,V) in the naïve KVS(P16)**
- **log file(P17)**
- **Get(K,V):terrible(P21)**
- **Accelerate Get with index(P22)**

Accelerate Get with index

- **Hash index & log-structured data file(P25)**
  - Issues of using hash index & log file(P49)
- Linked-list based hash index(P27)
- Cuckoo hashing(P31-47)
- Selecting a proper index for KV store is non-trivial(P48)
- how to prevent a file from growing forever?(P52-54)
  - Recall: how to handle delete()(P55)

B+Tree

- B+Tree indexes(P58)
- B+Tree indexes example(P65-66)
- B+Tree can support efficient range operations(P67)
- Note that Btree does not use log-structured file(P68)
- B-Tree indexes on the disk(P69)

LSM Trees

- SSTables (Sorted String Table)(P73)
- Benefits of SSTable(P74-76)
- MemTable(P77-
- Reads with MemTable + SSTable(P81)
- Organizing data sequentially has several drawbacks(P83)
- LSM Tree organizes LSM Tree as sorted Tree(P84)

- Example: Read in LSM-Tree(P85-90)
- What about crash(P91)
- LSM Tree Summary(P94)
- How to avoid write stall(P106)
- Slow lookup for non-existent key: solution(P108)