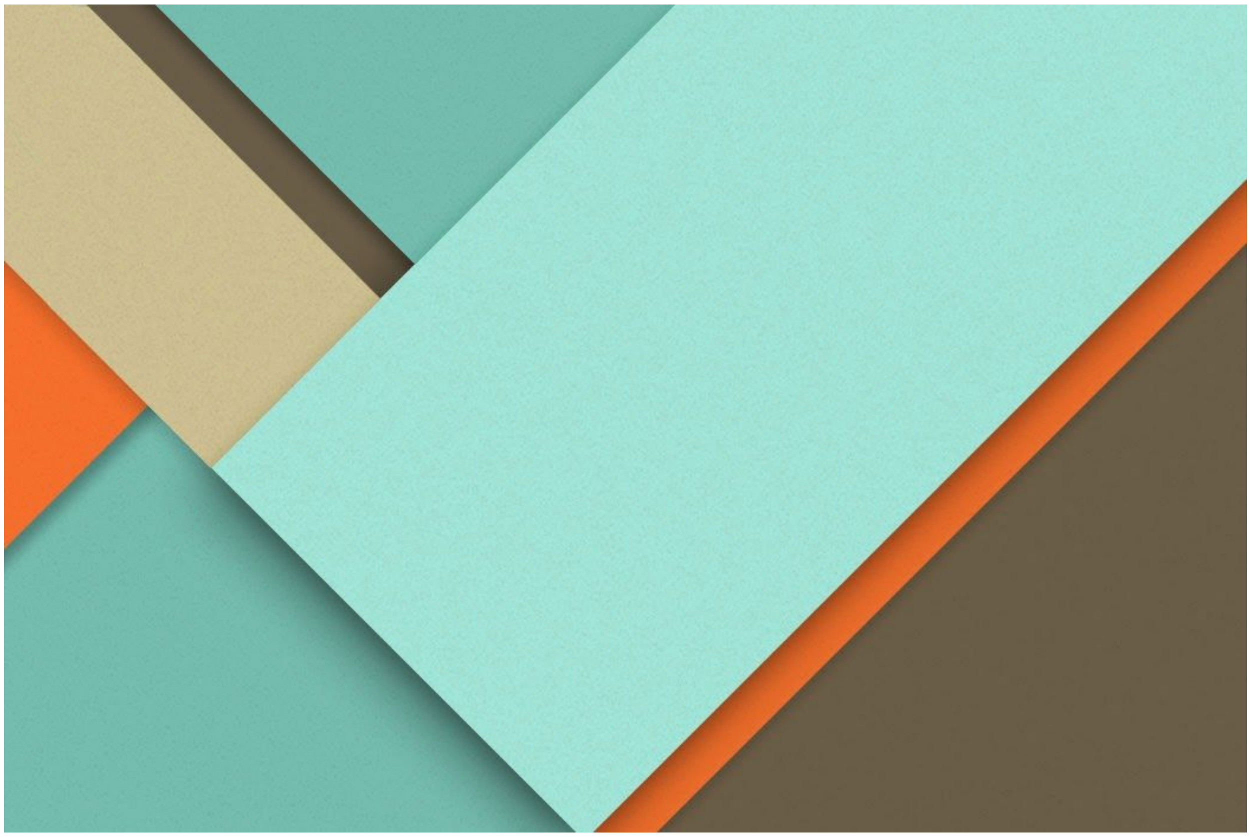

Key-Value Storage System

SPP Project

—

Amey Kudari (2018101046)

Raghav Saboo (2018101065)

Divanshi Gupta (2018101050)

Apoorva (2019121012)

Sonu Guru (2018101004)

Overview

Build a In-Memory Key-Value Storage Software in C++ supporting the following APIs

- get(key): returns value for the key
- put(key, value): add key-value, overwrite existing value
- delete(key)
- get(int N): returns Nth key-value pair
- delete(int N): delete Nth key-value pair

Goals

- To optimize the software to get maximum possible performance and reduce memory usage as much as possible.
- Optimize the TPS (transaction process system) to make sure multiple operations can run simultaneously to use all the processors available.

Implementation details:

Key encoding:

- As Key can only have ‘A’ to ‘Z’ and ‘a’ to ‘z’, store an array of long unsigned integers (64 bit) that have 11 letters encoded in each.
- Encoding done using relevant bithacks.
- Encoding takes the order of letters into consideration to make sure the encoded values when sorted result in lexicographically sorted of the values.

Data Structure Used:

Main Hash-Table:

- The first 5 values of the key are encoded to form the key for the hash table.
- The first 5 are encoded as (52^i)*(ith letter).
- The main hash table will also store the number of keys in each cell. to allow Nth value operations.
- Each node in the hash table has a linked list attached to it.

Fenwick Tree:

- This will hold the number of keys in each hash table.
- When we get an operation to get Nth value, we find the Nth value in by binary search on the fenwick tree to find the first value where the sum is just greater than N. The value will lie in that particular cell.

Sorted Linked List:

- This will hold the key-value pairs on each hash, all operations are simple O(n) (getting nth key, particular key).
- Expected Number of items in each hash cell is 0.21 (calculated before)

Maths behind choosing chained hashing:

1. Number of values in hash table that can be hashed = (54**4)*9 = 76527504
    a. Each key, first 5 values are chosen and a number is given based on lexicographic order
2. N = 76527504 (number of values in hash table)
3. n = 10000000 (number of values inserted at a time
4. probability(any node getting value) = n/N = 0.13
5. E[number of values in one hashed cell] = average number of values in a hash cell = 0.13

    Hence, linked lists on an average will have 0.13 elements each.

6. Worst case: P(kv-pairs >=10) < P(kv-pairs =10)*2 (probability of number of key value pairs greater than or equal to 10 is less than 2 times the probability that it is 10 (trivial))
7. P(kv-pairs=10) = nc10 * (((1/N)**10)*(1-1/N)**(n-10)) * N < 0.001%.
    a. nc10 = choose 10 values
    b. ((1/N)**10)*(1-1/N)**(n-10) = probability 5 nodes go into the same hashed value.
    c. N = number of possible hashed values

Hence, as edge case is rare and expected number of nodes per cell is 0.13, it is safe to keep linked lists at each node

Multi-Threading

- Except for operations involving Nth values checks, and deletes, all operations can run simultaneously.
- The rest of the operations will go through checks deciding if they can run or should wait to maintain correctness of code.

References

[Fenwick Tree]

[Linear Probing]

"Thank You"