
Peer to Peer Hashing Group Project

Compare and contrast costs and benefits of contrasting chaining and open addressing collision resolution strategies.

The chaining operation is a very redundant and accident-proof hashing technique. Chaining can scale indefinitely with overlapping values and keys being chained together. Open addressing, on the other hand, is much faster and more efficient. By using open addressing physical space is reduced a lot. Unfortunately, open addressing is a set size meaning you can not add more values than there are keys. These two techniques both have their tradeoffs. Chaining is much slower but is stable and can scale up. Open Addressing is quick, small, and efficient but is a fixed size.

Analyze performance tradeoffs for bucket sizes used in hashing.

Chaining operation:

In chaining, the hash table is indexed into an array of pointers. The bigger the list, the more pointers there are. When searching, the computer must follow these pointers. As the program gets bigger, the more pointers the computer must go through meaning slower speeds. fortunately, this increase will always be linear.

Open Addressing:

Open addressing puts the pointers in pairs. One key and one value. Having this linked pair, the computer can instantly find the value given the key. Unfortunately, as the list gets more filled it will exponentially get slower as the pointer is hopping around trying to find the designated slot.

Use Big-O notation to describe algorithm run time.

Chaining operation:

Depending on how big the table and its values are, the chaining operation will have an average of $O(1)$ with slight variation due to table size. The worst-case scenario (which is usually the last value) will be $O(N)$.

Open Addressing:

Open Addressing is usually faster then changing. Its average run time will be $O(1)$ and its worst case scenario will be $O(N)$.