CuckooQ: Short Insertion Delay in Cuckoo Hashing

Kritka Sahni kxs4926@rit.edu  Advisor: Dr. Minseok Kwon mxkvcs@rit.edu

Introduction
Hashing is widely used for fast data retrieval especially in large-scale distributed systems. Cuckoo hashing is an efficient hashing scheme that guarantees worst case constant lookup time. But the insertion delay is significant in cuckoo hashing when the table is highly occupied. This project explores a modification to cuckoo hashing, called CuckooQ that exploits “the power of two choices paradigm” during insertion to reduce the insertion delay in cuckoo hashing, without affecting the lookup time of the existing scheme.

Cuckoo Hashing

1. Uses 2 hash functions that map an item to 2 buckets – b1 & b2. Each bucket has 4 slots.
2. Item is inserted into b1 or b2, and will always be found in b1 or b2.
3. Collision may result in a chain of evictions to search for an empty slot in the table. The evicted items are reallocated to the alternative buckets. This process is called a cuckoo move.

Pro: Worst-case constant lookup time.
Con: Long cuckoo moves when the table is around 95 percent, thus, causing high insertion delays.

CuckooQ

1. When both the buckets have empty slots, insert into the least loaded bucket. This is according to “the power of two choices paradigm”.

2. Insert item into the leftmost empty slot. Items are present in the buckets in the order of insertion. Oldest item is at first slot.

3. In the case of collisions, evict the oldest item.

- Each item is inserted into a bucket because its alternative bucket is relatively more loaded.
- This is true for both the old and the new items in any bucket.
- In a given bucket, the newer items are inserted recently so there is a higher probability that their alternative buckets are more loaded.
- There is more uncertainty about the load of alternative buckets of oldest item because more time has elapsed since its insertion.
- Hence, the oldest item is evicted.

Results
All the experiments are conducted on a table which is prefilled to 96 percent occupancy.

Conclusion
1. Insertion: CuckooQ shows speedup of 28 percent on average during. But this comes at the cost of additional rehashes that result in larger tables.
2. Deletion: Slower on average in CuckooQ, by 24 percent.
3. Lookup: CuckooQ guarantees worst case constant lookup time, same as cuckoo hashing.
4. Runtime Memory Needs: Less than cuckoo hashing.