Project Workflow

Implementation Ideas

1. Slab Lists with universal hash function; Extend it to try different hash functions and evaluate performance [1]
2. Multivalue hash tables (with radix sort/binary search) [2]
3. Bucketed hash table with different probing schemes : Iceberg hashing, Cuckoo Hashing and Multiple Hashing Techniques [3, 4]
4. Robin Hood hashing (Rejection of queries that are not in table) [5]
5. Initializing hash tables with different capacities [6]
6. Cuckoo hashing scheme with d sub tables, where each sub table is configured by a hash function, and introduces a resizing policy to maintain the filled factor within a bounded range [7]

Evaluating Performance

Evaluating performance by varying parameters with different workloads (each workload can have different ratios of insert, search, delete, update queries).
We will evaluate the following:

- Build performance
- Query performance
- Memory efficiency

We can have the following plots:

- Build/Search/Insert throughput (operations/sec) vs number of elements
- Build/Search/Insert throughput (operations/sec) vs load factor (we can find a "sweetspot" here)
- Average number of probes vs. load factor
- Operation throughput vs Memory utilisation

More informative ones:

- Build time vs number of duplicates
- DRAM sectors per key vs load factor
- Cache lines per key vs load factor

- DRAM throughput vs load factor
- Build/Search throughput vs number of buckets

Comparisons can be made with existing GPU implementations.

Existing Implementations

We found the following implementations. We can use one of them to make comparisons with ours.
Ref 1, Ref 2