Lock Free Extendible Hash Table In Database Management systems: Compare and Benchmark

PRESENTER: **Ruijie Zhai, Yicheng Zhang**

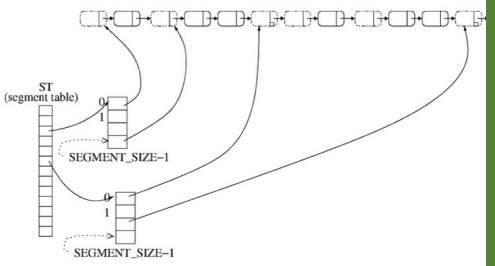
INTRODUCTION

Extendible hash table is a type of dynamic hashing method that allows for the growth and shrink of the hash table periodically. It's particularly useful in database management systems (**DBMS**) for efficiently handling searches, insertions, and deletions.

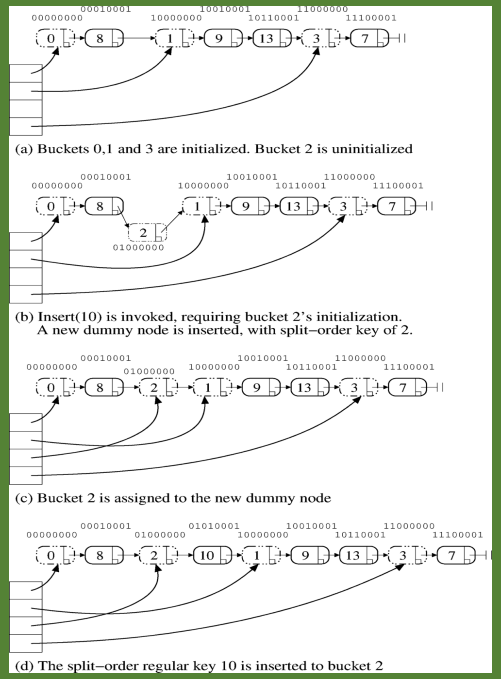
However, there was few implementation specific to lock-free extendible hash tables in **DBMS**. Lock-free extendible hashing in DBMS would enhance concurrent access performance and eliminate the overhead and complexities associated with traditional locking mechanisms.

ALGORITHM

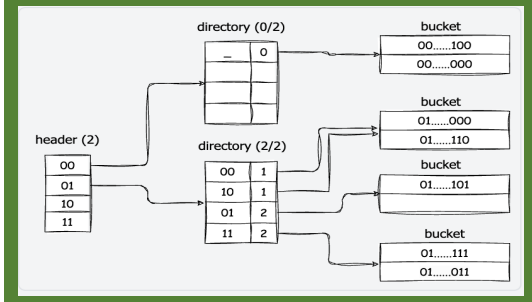
- **Recursive split-ordering:** ordering on keys that keeps items in a given bucket adjacent in the list, using **binary reversal**.
- **Lock-free linked list** + an **expanding array of pointers (buckets)** to achieve efficient data storage and access.
- Initially, the array has two buckets, doubling in size as the number of items surpasses a threshold defined by a load factor.
- **Dummy node**: marks the start of each bucket. These nodes simplify the deletion process as they're never deleted.
  - MSB set to 0, while regular nodes set to 1.
- **Segments**. A main array points to segments of buckets, each of which is a bucket array.
- **Complexity**. Expected constant time given a constant extendibility rate.



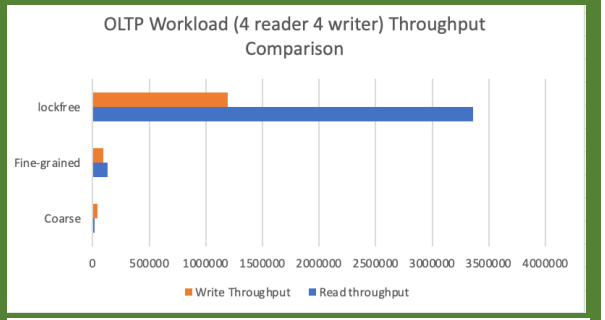
Lock-free Extendible Hash Table - *Much Faster!*

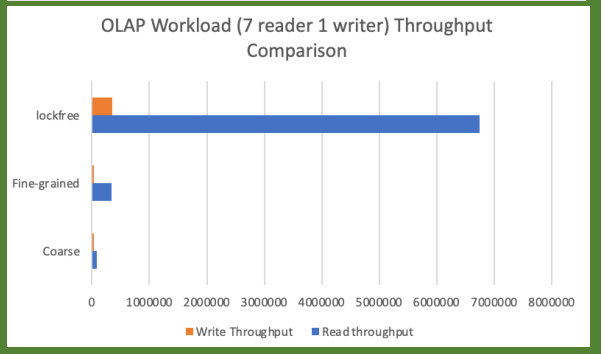


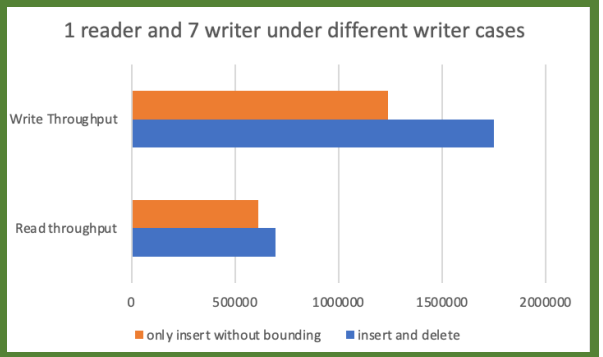
Insert at a lock free split-ordering EHT

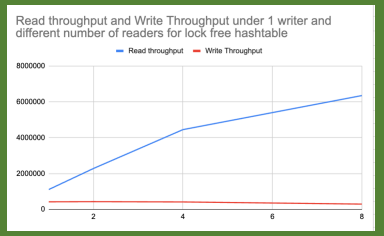


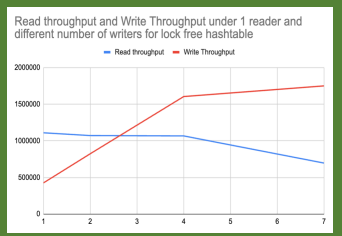
Fine-grained Extendible hash table structure

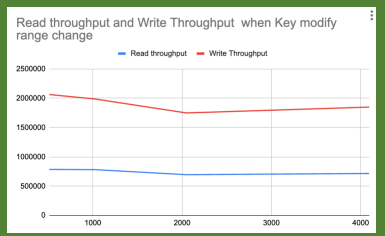












Benchmark result using Mac M1 Max with 8 performance cores and 2 efficiency cores.

THE THREE IMPLEMENTATIONS

- Implemented three versions of extendible hash tables: **coarse-locked, fine-grained-locked, and lock-free.**
- Benchmarked using **BusTub**, an educational DBMS system.
- Simulated two types of DBMS workloads: **OLTP** and **OLAP**.
- Aimed to assess performance and robustness under concurrent conditions. Focused on key performance metrics: **throughput, scalability, and data integrity**.
- Tested the system's ability to maintain consistent and accurate data access amidst concurrent modifications.

RESULTS

- Lock-free extendible hash table **significantly outperforms** coarse-locked and fine-grained-locked implementations.
- In OLTP workloads:
  - Achieved approximately **25x** speedup in read throughput compared to fine-grained.
  - Write throughput speedup was about **13x** over fine-grained.
- In OLAP workloads:
  - Lock-free reads were about **20x** faster than fine-grained.
  - Write throughput speedup was about **10x** over fine-grained.
- Benchmarks showed a **linear increase** in read throughput with more readers.
- A **bottleneck** in write throughput with increasing number of writers.

CONCLUSION

- Faster, faster, faster
- Thanks to reduced system calls and elimination of list-moving operations via linked lists.
- "Bending" shape in write throughput, **suggesting frequent resizing remains a potential performance bottleneck** for write operations.