FEAST: A Lightweight Lock-free Concurrent Binary Search Tree

Apurva Patel, Harvey Ko
CMU 15618 Parallel Computer Architecture and Programming

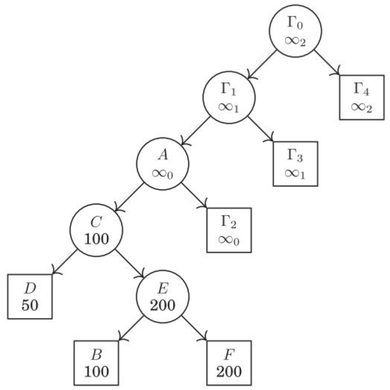
Implementation Summary

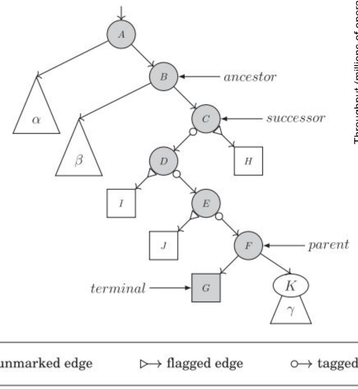
Coarse-grained
- Uses 1 global lock for entire operation
- External representation

Fine-grained
- Uses a lock per node and 1 global lock for root
- Internal representation

Lock-free
- Uses __sync_compare_and_swap & __sync_fetch_and_or
- External representation

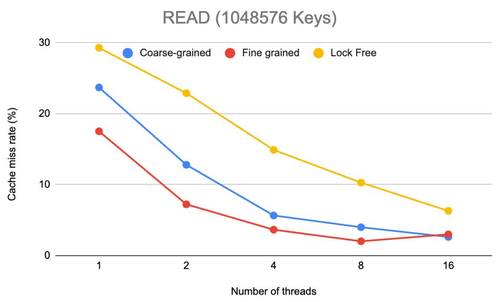
Key Operations: Insert, Delete, Search

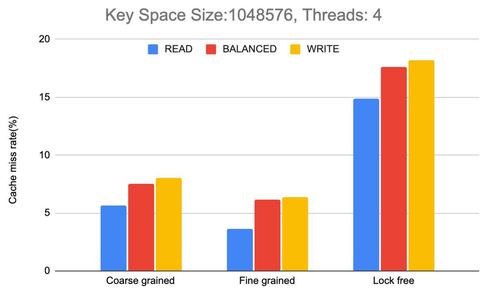




Results

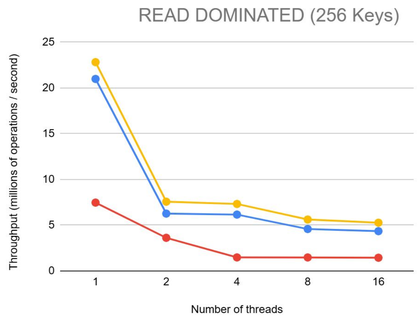
- Fine grained with internal representation has lowest cache miss rate. Lock free has highest cache miss rate.
- Cache miss rate decreases as number of thread increases due to sharing.
- Write-dominated workload has the highest miss rate compared to other workloads due to higher probability of eviction and invalidation.

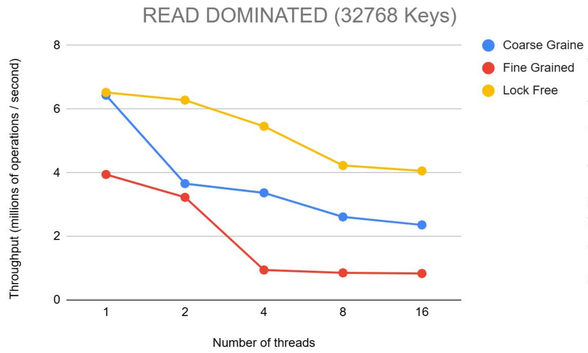


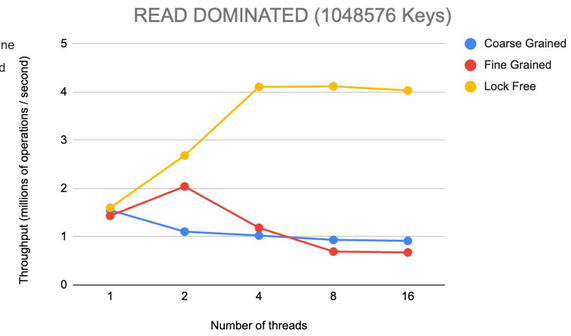


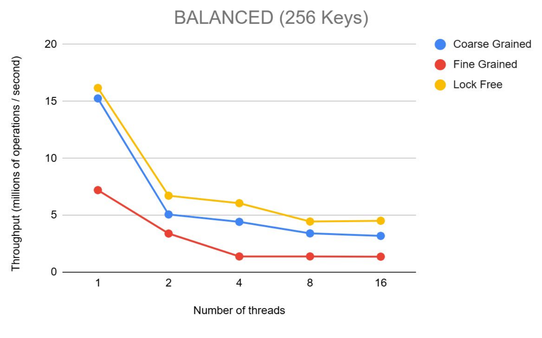
Results

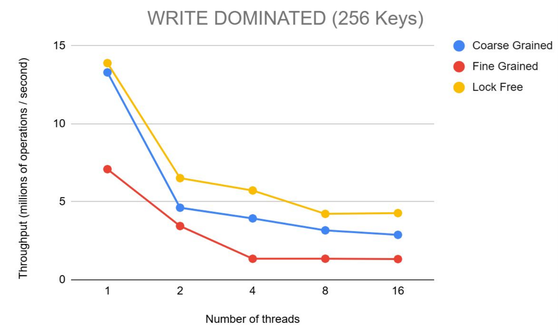
- Different Key Space Sizes
  - For smaller key space, throughput decreases with number of threads due to contention. For larger key space, throughput increases as number of thread increases as working set of threads less likely to overlap, reducing retries.
  - No improvement in throughput from 8 to 16 threads due to context switch overhead.
- Lock free have highest throughput for READ dominated workload.

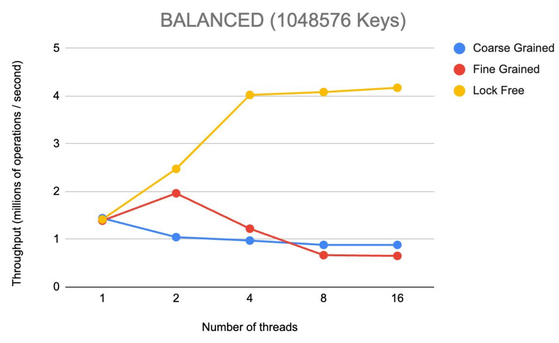


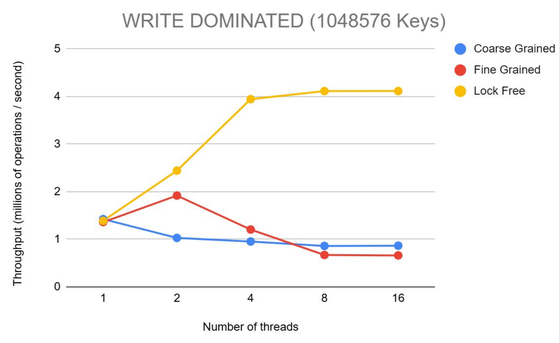












Results

- Similar trends for balanced and write dominated workloads for all key space sizes.
- Read-dominated has higher throughput because mostly lookups that hold locks for much shorter duration (coarse/fine grained), do not require tagging/flagging nodes (lock-free) & do not modify nodes.
- Type of workload has greater impact on throughput at smaller key space size due to greater contention.
- Lock-free BST more suitable for larger key space size with higher thread count.
- Coarse-grained BST sufficient for single threaded application for all key space sizes.

References

[1] Aravind Natarajan, Arunmoezhi Ramachandran, and Neeraj Mittal. 2020. FEAST: A Lightweight Lock-free Concurrent Binary Search Tree. ACM Trans. Parallel Comput. 7, 2, Article 10 (June 2020), 64 pages. https://doi.org/10.1145/3391438
[2] Poster template:
https://docs.google.com/presentation/d/1J6rQtA95_g-rEl-WLKhItFrUferanWS7vdaJzTB45rM/edit#slide=id.p3