Accelerated Hashing

A Case Study on High Throughput Hashtables for Highly Parallel Architectures

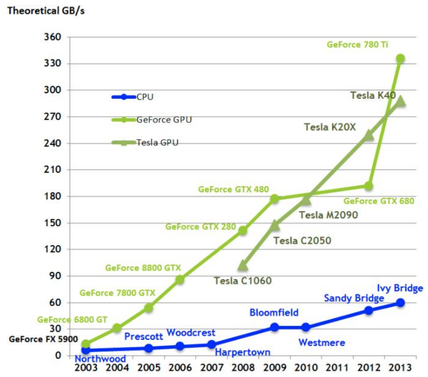
Brody Tingle, UNC Charlotte
Dr. Tyler Allen, College of Computing and Informatics

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Background

GPUs

- GPU = Graphics Processing Unit
- Problems are becoming more complex and computation heavy.
- GPUs can be used to accelerate problems through parallelism.
- Popular in machine learning, climate modeling, solar system modeling.



Hashtables

- Hashtables or hashgraphs are a commonly used data structure.
- Use key-value pairs to organize information.
- Keys are created using a hash function, values are stored in buckets.

Process

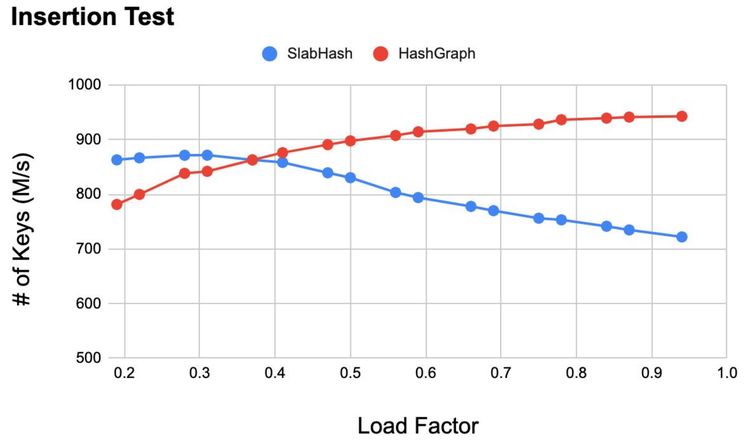
Method

- Researched and found hashtable algorithms from recent years.
- Compared them based on functionality and usability.
- Picked the best algorithms and did a deeper analysis on their performance and capabilities.

Challenges

- There are no standardized benchmarks or interfaces for comparing these data structures
- Databases and metagenomics are common examples of GPU hashtables, but finding "real-world" use cases is difficult.

Results



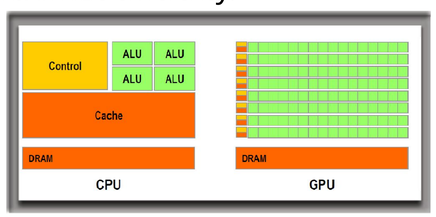
Future Work

- Measure performance of more algorithms.
- Find clear weaknesses in current hashtables.
- Make dynamic hashtable with speed comparable to current static implementations.

Motivations

Problem

- GPU data structures not given the special attention they need.
- The memory structure and parallelism of the GPU must be handled with precision to maximize efficiency.



Objectives

- Explore existing GPU hash table algorithms.
- Compare and contrast implementations.
- Run tests to collect data on our machines, evaluate performance.
- Find weaknesses in implementations.

Algorithm Comparison

Algorithm	Collision Handling	Static/Dynamic	Supports: Insertions, Deletions	Keys	Other
HashGraph by Oded Green (NVIDIA)	Combination of Open Addressing and Separate Chaining.	Static	Deletions	Integer Only	Uses a sparse graph data structure.
SlabHash by The Owens Group	Separate Chaining	Dynamic	Insertions, Deletions	Integer Only	One of the first well-developed dynamic hash tables. Table consists of an array of linked lists.
Warpcore by NVIDIA	Open Addressing	Dynamic	Insertions, Deletions	Integer Only	Novel "Bucket List" hash table allows for dynamic usability with great performance.
Better GPU Hash Tables by the Owens Group	Cuckoo hashing	Static	Insertions	Integer Only	Compared a Bucketed Cuckoo hash table, Bucketed Power-of-two hash table, and an Iceberg hash table. Determined Bucketed Cuckoo hashing was superior.

References

- Shah, Shachi. "Do We Really Need GPU for Deep Learning? - CPU vs GPU." *Medium*, Medium, 7 Dec. 2018, https://medium.com/@shachishah.ce/do-we-really-need-gpu-for-deep-learning-47042c02efe2.
- Green, Oded. "HashGraph–Scalable Hash Tables Using a Sparse Graph Data Structure" ACM Transactions on Parallel Computing, 15 July 2021, Volume 8, Issue 2, Article 11, Pages 1-17.
- D. Jünger *et al.*, "WarpCore: A Library for fast Hash Tables on GPUs," *2020 IEEE 27th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, Pune, India, 2020, pp. 11-20, doi: 10.1109/HiPC50609.2020.00015.
- Ashkiani, Saman, Farach-Colton, Martin, Owens, John D. "A Dynamic Hash Table for the GPU" arXiv:1710.11246
- Awad, Muhammed A., Ashkiani, Saman, Porumbescu, Serban D., Farach-Colton, Martin, Owens, John D. "Better GPU Hash Tables." arXiv:2108.07232