Techniques in Accelerating Query Processing on GPU (Lightning Talk)

Jimmy Lu[1,†]

[1]Meta Platforms, Inc. Corporate, 1 Hacker Way, Menlo Park, CA 94025, USA

Abstract

Velox is a pioneer effort aimed at unifying query processing engines. This unification provides a compelling framework where hardware accelerators can be leveraged, and made available to any engines integrated with Velox. In this talk we present Velox Wave, a new framework for hardware accelerator built in Velox, and show early results illustrating the potential benefits of combining such a composable engine with GPU accelerators.

Keywords

Query Processing, Hardware Accelerator, GPU, CUDA

1. Introduction

Composability of the execution layer in data management systems allows accelerators to be integrated in a single library, like Velox, and leveraged by many engines. The Velox Wave hardware acceleration framework proposes a common interface for composing operators customized to accelerators. GPU is one of the most promising and universally available accelerators that can be used for query processing. For this reason, we have conducted experiments to investigate the potential benefits of GPU accelerators in query processing. These experiments can also serve as a proof-of-concept of how the Velox Wave framework could look like in the future.

2. Experiments

Previous work by A. Shanbhag et al. [1] has shown the advantage as well as limitations of using GPU for query processing. In this talk, we focus on components and techniques that have not been covered in that work.

The first experiment is a file reader to read Meta's new format Alpha, which caters to machine learning use cases. The implementation includes GPU decoders for 8 different encodings and composition of them. Running the benchmark on NVIDIA A100 cards shows the throughput is very promising, ranging from 300 GB/s to 1000 GB/s for most of the encodings, showing a very large advantage over CPU implementations.

The second experiment tests various hash table optimization options. We tested hash tables with and without tags, with and without partitioning, and ran it under different table sizes, loading factors, and matching rates. The results shows that for a fairly large table under heavy workload, a partitioned GPU hash table is able to probe 6.8 billion rows per second, which is also very promising.

Something we have not done but will try soon is to experiment shuffling on NVLink or other HPC connections. Some early experiment using OpenUCX shows that throughput of more than 80 GB/s can be achieved between two GPU devices on A100. The number itself is very positive, also we are amazed by the work done by OpenUCX to abstract out the hardware details at connection level, simplifying the implementation of exchange operators.

3. Conclusion

These initial experiments suggest that GPU acceleration can be a viable architecture to accelerate query processing, in a composable, general, and unified manner.

References

[1] A. Shanbhag, S. Madden, X. Yu, A study of the fundamental performance characteristics of gpus and cpus for database analytics, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, SIGMOD '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1617–1632. URL: https://doi.org/10.1145/3318464.3380595. doi:10.1145/3318464.3380595.

Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23) — Second International Workshop on Composable Data Management Systems (CDMS'23), August 28 - September 1, 2023, Vancouver, Canada

[†] Also known as Gongchuo Lu.

jimmylu@meta.com (J. Lu)