Optimizing GPU-Accelerated Similarity Joins: Addressing Data-Dependent Workloads

Benoît Gallet, Michael Gowanlock
benoit.gallet@nau.edu, michael.gowanlock@nau.edu
School of Informatics, Computing and Cyber Systems, Northern Arizona University

Introduction

Given D a dataset in n dimensions

- Distance similarity self-join → Find pairs of objects in D whose distance is within ε
- Use a grid indexing to prune the search space
- $Thread_i$ = $Query_i$
- Depending on data characteristics → Workload between threads varies a lot
- GPU's architecture (SIMT) → Idle periods for threads with less workload
  - Results in higher execution time

Solution

- Balance the workload between threads to reduce the response time
- Sort the points by workload, from most to least
  - Consecutive threads should have similar workload (Fig. 1)
- However → GPU's hardware scheduler controls the execution order of the threads
- Use a work queue to force this execution's order
  - Atomic operation to retrieve the first element of the queue, i.e., the one with most workload (Fig. 2)

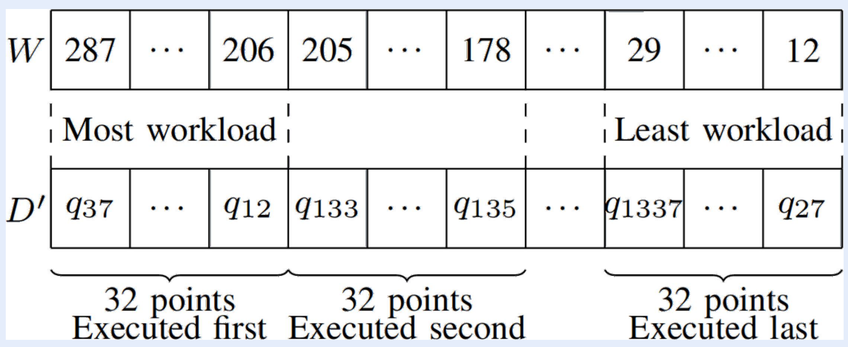


Figure 2

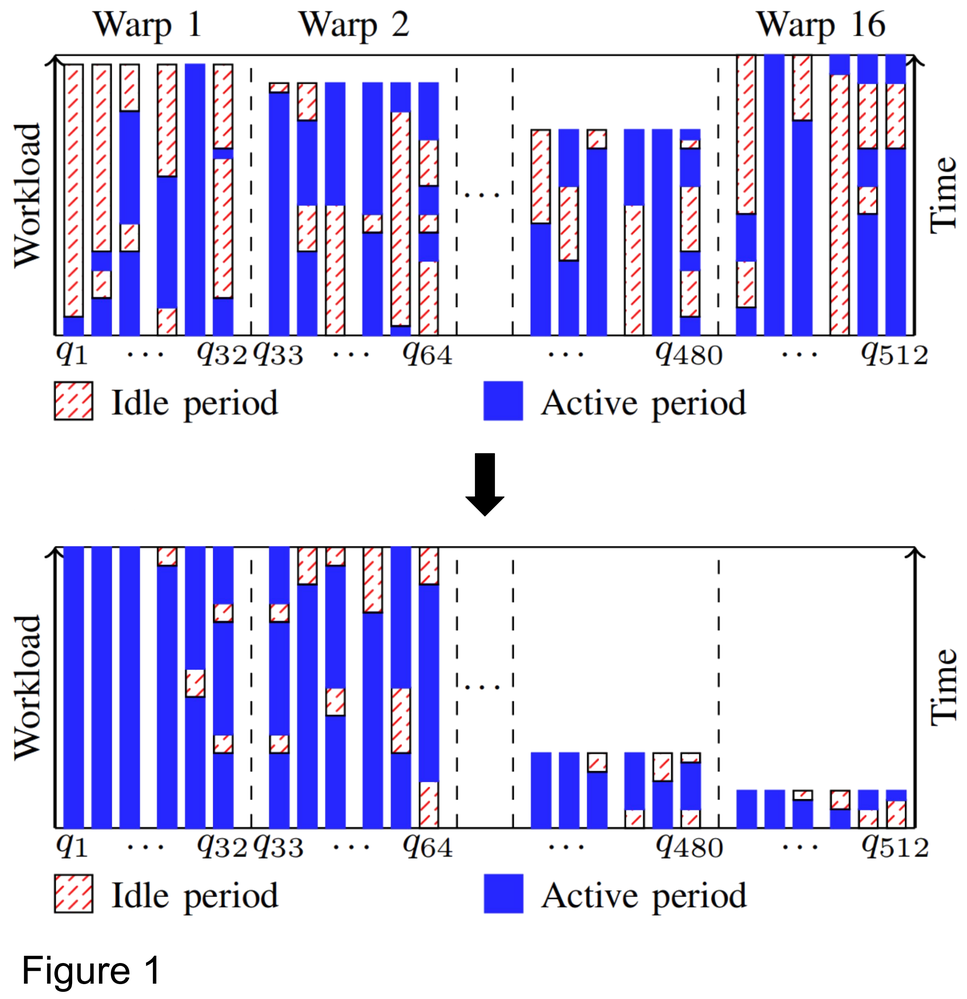


Figure 1

Results

- Compare GPUCalcGlobal [1], sorting by workload (SortByWL) and our work queue (WorkQueue) [2]
- Focus on the execution time and warp execution efficiency (WEE)
  - Percentage of active threads within a warp → higher is better
- Uniformly (Fig. 3) and exponentially (Fig. 4) distributed synthetic datasets, n = 6, 2M points

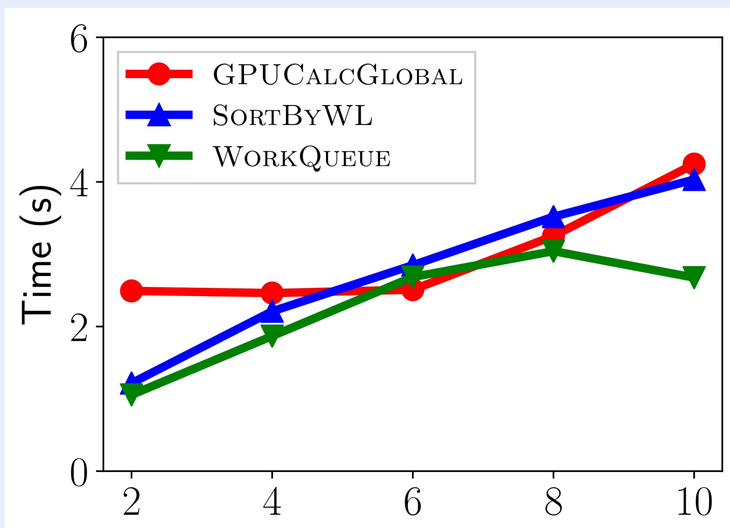


Figure 3: Uniform

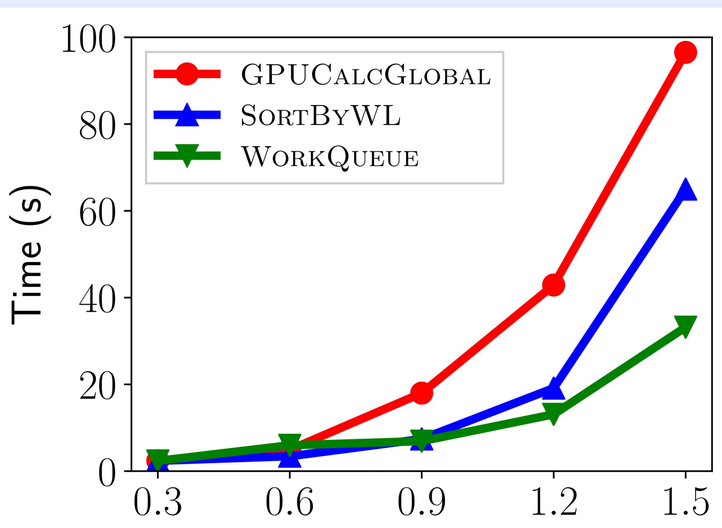
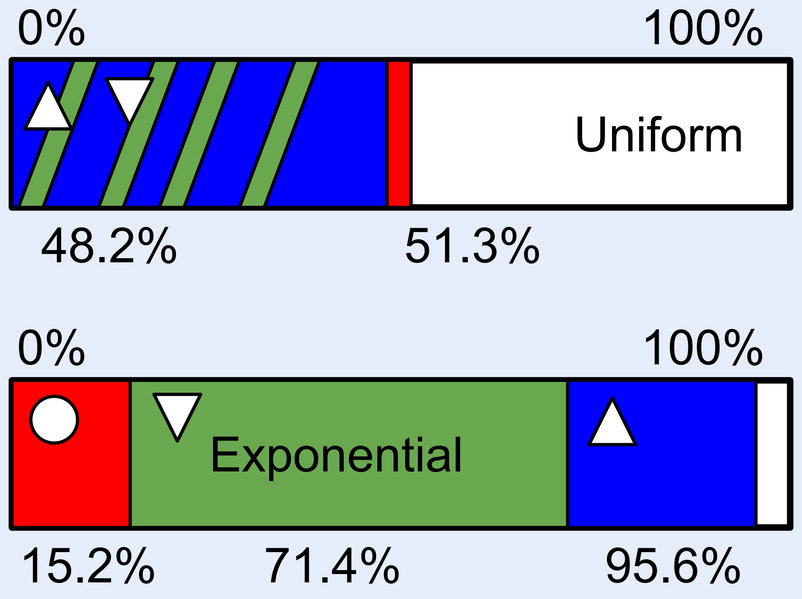


Figure 4: Exponential



- Uniformly distributed dataset → uniform workload → no need to balance, contrary to exponentially distributed

Conclusion

- Warp execution efficiency impacts response time
- 100% warp execution efficiency may indicate a computational bound
- Use the WorkQueue to improve other data dependent applications

References

[1] M. Gowanlock and B. Karsin, "GPU Accelerated Self-join for the Distance Similarity Metric," Proc. of the 2018 IEEE Intl. Parallel and Distributed Processing Symposium Workshops, pp. 477–486, 2018.

[2] B. Gallet and M. Gowanlock, "Load Imbalance Mitigation Optimizations for GPU-Accelerated Similarity Joins", Proc. of the 2018 IEEE Intl. Parallel and Distributed Processing Symposium Workshops, 2019

Poster abstract: Optimizing GPU-Accelerated Similarity Joins: Addressing Data-Dependent Workloads

Authors: Benoît Gallet and Michael Gowanlock

Advisor: Michael Gowanlock, *School of Informatics, Computing and Cyber Systems, Northern Arizona University*, michael.gowanlock@nau.edu

Description of Work: As the similarity join is an irregular application, the GPU's threads may be assigned query points with dissimilar workloads. The SIMT architecture of the GPU and the groups of 32 threads being executed concurrently within warps, induces idle periods for the threads with lower workloads. Overall, this results in a longer execution time. By reducing the intra-warp workload imbalance, achieved by grouping threads with similar workloads within the same warp, we reduce the potential idle periods of the threads within the same warp. Hence, the average ratio of active threads increases, while the execution time decreases. This work is based on the solution proposed in [1].

Abstract of the Results: Figure 1 plots the average speed-up of our developed solution against (a) the state-of-the-art CPU parallel algorithm SuperEGO [2] and (b), GPUCalcGlobal, a previous GPU solution [1]. We observe, especially when looking at the exponentially distributed datasets or the real-world datasets, that our solution greatly improves the performance of the distance similarity self-join, and therefore that reducing the load imbalance between the threads within the same warp is essential. On average, we achieve a speed-up against SuperEGO of 2.6×, and 1.6× against GPUCalcGlobal.

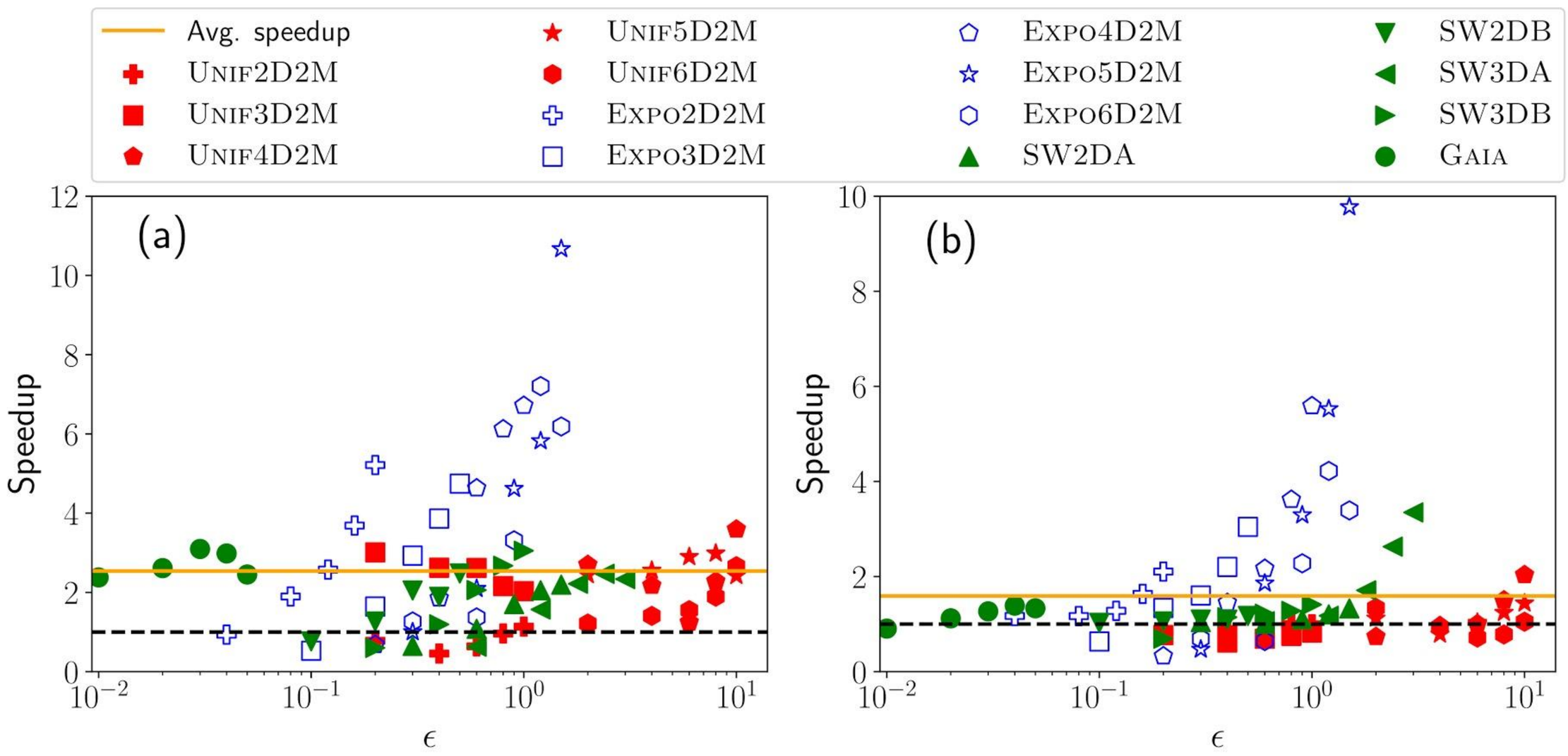


Figure 1: Average speed-up of our solution compared to (a) the state-of-the-art CPU parallel algorithm SuperEGO, and (b) against a previous GPU version GPUCalcGlobal on different datasets. *Expo-* and *Unif-* are synthetic datasets respectively exponentially and uniformly distributed, while *SW-* and *Gaia* are real-world datasets. We use datasets ranging from 2 to 6 dimensions, and represent the distance threshold ϵ on a log-scale to observe all data points.

References

[1] Michael Gowanlock and Ben Karsin. GPU Accelerated Self-join for the Distance Similarity Metric. *Proc. of the 2018 IEEE Intl. Parallel and Distributed Processing Symposium Workshops*, pages 477–486, 2018.

[2] Dmitri V. Kalashnikov. Super-EGO: fast multi-dimensional similarity join. *The VLDB Journal*, 22(4):561–585, 2013.