DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: https://www.epfl.ch/labs/dias

Faster Join: When Learned Meets GPU

Keywords: Learned Index, Learned Joins, Learned Partitioning, Compression, GPUs

Project: Join operations are a critical component in relational databases and data analytics systems. With ever-growing datasets and the need for real-time analytics, leveraging the massive parallelism of GPUs is promising to accelerating join operations [3]. However, GPU join algorithms often suffer from two key issues:

- **Random Memory Accesses**: Many current GPU join implementations (e.g., hash joins and sort-merge joins) incur significant overhead due to uncoalesced or random accesses, especially during the materialization phase.
- **Limited GPU Memory**: On systems where GPU memory is scarce relative to the dataset size, data transfers and memory allocation become bottlenecks.

On the other hand, recent developments in learned index structures (e.g., RMI, ALEX) have shown that using lightweight machine learning models can guide data access patterns effectively [1,4 - 6]. These approaches have been used to improve lookup performance in CPU-based joins and can potentially ***reduce random access patterns*** by reorganizing data for sequential memory access [2]. Moreover, learned techniques can be used to “summarize” or compress data, thereby ***reducing the size of data*** that must be loaded into GPU memory.

This proposal explores the ***integration of learned approaches with GPU join processing***. Our goal is to design a novel join operator that leverages learned models to both (a) reorganize data to reduce random memory accesses during join execution, and (b) reduce the volume of data transferred to and stored on the GPU, thereby mitigating memory constraints.

Background: ***GPU-Accelerated Joins***

Recent research demonstrates that mapping join operations to GPUs can lead to significant performance improvements due to GPUs’ high parallelism and memory bandwidth [7]. In particular, techniques such as radix-sort-based partitioning and the innovative GFTR (gather-from-transformed-relations) pattern have been shown to reduce costly random memory accesses which is a major bottleneck when materializing join results [3]. These methods optimize the workflow by transforming not only the join keys but also the payload data, thereby converting random access patterns into sequential ones that better exploit the GPU memory hierarchy.

Learned In-Memory Joins

Learned join techniques leverage statistical models (like CDF-based models) to predict data positions, guide partitioning strategies, and even replace or optimize components of traditional join algorithms [2]. The core idea is to use a model trained on data distribution to better organize the join processing, for example, by predicting the correct partition or index location, which can substantially reduce lookup times and overhead compared to traditional CPU-based indexes. Although there have been efforts to integrate learned indexes on GPUs [9], no existing work has yet combined these approaches into a fully integrated framework for learned joins on GPUs.

Aim: This proposal aims to fill these gaps by:

- Designing a novel GPU join operator that incorporates a learned index model (or learned partitioning model) to re-order join keys and payloads.
- Leveraging GPU parallelism to perform real-time model inference to predict data layout, thereby reducing random accesses.
- Exploring data reduction techniques based on learned models to limit the amount of data transferred into GPU memory, making it feasible to process larger datasets even when GPU memory is scarce.

Plan:

1. Literature and Feasibility Study: study the GPU join techniques and learned join approaches, focusing on the GFTR optimization and the learned join models .
2. Design and Algorithm Development: develop a unified join operator that integrates learned model predictions with GPU-based partitioning [and GFTR transformation].
3. Evaluation and Optimization: experiment and test on serval benchmark datasets [8, 10 - 11].

Supervisor:	Prof. Anastasia Ailamaki, anastasia.ailamaki@epfl.ch
Responsible collaborator(s):	Liang Liang, liang.liang@epfl.ch Antonio Boffa, antonio.boffa@epfl.ch
Duration:	6 months

Reference:

1. Kraska, T., Beutel, A., Chi, E. H., Dean, J., & Polyzotis, N. (2018, May). The case for learned index structures. In Proceedings of the 2018 international conference on management of data (pp. 489-504).

2. Sabek, I., & Kraska, T. (2021). The case for learned in-memory joins. arXiv preprint arXiv:2111.08824.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: https://www.epfl.ch/labs/dias

3. Wu, B., Koutsoukos, D., & Alonso, G. (2025). Efficiently Processing Joins and Grouped Aggregations on GPUs. Proceedings of the ACM on Management of Data, 3(1), 1-27.

4. Ding, J., Minhas, U. F., Yu, J., Wang, C., Do, J., Li, Y., ... & Kraska, T. (2020, June). ALEX: an updatable adaptive learned index. In Proceedings of the 2020 ACM SIGMOD international conference on management of data (pp. 969-984).

5. Wu, J., Zhang, Y., Chen, S., Wang, J., Chen, Y., & Xing, C. (2021). Updatable learned index with precise positions. arXiv preprint arXiv:2104.05520.

6. Ferragina, P., & Vinciguerra, G. (2020). The PGM-index: a fully-dynamic compressed learned index with provable worst-case bounds. Proceedings of the VLDB Endowment, 13(8), 1162-1175.

7. Paul, J., Lu, S., He, B., & Lau, C. T. (2021, June). MG-Join: A scalable join for massively parallel multi-GPU architectures. In Proceedings of the 2021 International Conference on Management of Data (pp. 1413-1425).

8. Kipf, A., Marcus, R., van Renen, A., Stoian, M., Kemper, A., Kraska, T., & Neumann, T. (2019). SOSD: A benchmark for learned indexes. arXiv preprint arXiv:1911.13014.

9. Liu, J., Zhang, F., Lu, L., Qi, C., Guo, X., Deng, D., ... & Du, X. (2024). G-learned index: Enabling efficient learned index on GPU. IEEE Transactions on Parallel and Distributed Systems.

10. Transaction Processing Performance Council. (n.d.). TPC-H Benchmark. Retrieved from http://www.tpc.org/tpch/

11. Transaction Processing Performance Council. (n.d.). Star Schema Benchmark (SSB). Retrieved from http://www.tpc.org/ssb/