CAS CS 561: Data Systems Architectures
Data-intensive Systems and Computing Lab
Department of Computer Science
College of Arts and Sciences, Boston University
http://bu-disc.github.io/CS561/

Short Research Project Description

Title: Bitmap Indexing for Massive Joins
Responsible mentor: Teona Bagashvili and Junchang Wang

Background: Unlike the textbook joins between fact and dimension tables, where the fact table is typically orders of magnitude larger, practical scenarios often involve joining two large fact tables (e.g., TPC-DS Q29 [1]), which we refer to as "massive joins." When employing classic hash-based join algorithms, such joins generate prohibitively huge intermediate data, potentially leading to disk spills [2].

Recent advances in bitmap indexing [3,4] offer a new, bitmap-based approach for massive joins. The core idea involves transforming join operations into predicates on the join columns [5], and evaluating the predicates using the maintained bitmap indexes on these columns. This design transforms expensive hash table maintenance to efficient logical bitwise operations, accelerating query performance.

Objective: This project challenges students to address massive joins, a critical performance bottleneck in analytical DBMSs. Using a customized DuckDB [6] with state-of-the-art bitmap indexes, students will investigate the native join procedure, evaluate its performance bottlenecks, and hopefully address this challenge using bitmap indexes.

(a) Download the custom DuckDB (using branch *Experiments*), study its native join executor, and analyze the execution of a join-dominated query (TPC-DS Q29).
(b) Review the implementations of bitmap-based join operations (TPC-H Q5).
(c) Implement a bitmap-based massive join executor (potentially tailored for TPC-DS Q29 to reduce workloads), evaluate its performance, and summarize its strengths and weaknesses.

References

[1] https://www.tpc.org/tpcds/
[2] L. Shrinivas, S. Bodagala, R. Varadarajan, A. Cary, V. Bharathan, and C. Bear. 2013. Materialization strategies in the Vertica analytic database: Lessons learned. In Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE), pp. 1196–1207.
[3] J. Wang and M. Athanassoulis. 2024. CUBIT: Concurrent Updatable Bitmap Indexing. Proc. VLDB Endow. 18(2), pp. 399–412.
[4] J. Wang, F. Xiao, and M. Athanassoulis. 2025. RABIT: Efficient Range Queries with Bitmap Indexing. Proc. ACM Manag. Data. 3(6), pp. 354.
[5] D. J. Abadi, S. Madden, and N. Hachem. 2008. Column-stores vs. row-stores: how different are they really?. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 967–980.
[6] https://github.com/junchangwang/CS561-Bitmap-Indexing-Massive-Joins.git