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/

CS561 Spring 2024 - Research Project

Title: *Joins with Near-Sorted Data*

Background: Modern day applications often generate data that appear *almost-sorted* or *nearly-sorted* with respect to some attribute. For example, near-sorted data is frequently collected by stock market applications, event-based applications like sensor failures, and data that have correlated columns. In fact, near-sorted data can also occur as a result of a previous join operation, or data that was recently sorted but received a few updates that occurred out-of-order.

Join algorithms are fundamental to database systems to answer complex queries with real-world data. Sort-merge join (SMJ) is a prominent algorithm known for its high efficiency when the input relations are sorted on the join attribute. In such a scenario, the sorting phase of the algorithm is eliminated, and the algorithm only merges the two relations.

Objective: This project aims to explore the design space of join algorithms with near-sorted data. Particularly, the project targets to answer the following questions: *What if the input releations are instead, **near-sorted**? Does sort-merge join offer a close-to-ideal performance? Can indexes that exploit data sortedness help SMJ become sortedness-aware?*

Technical: This project requires C++ programming skills (particularly working with pointers). The following steps outline the high-level milestones:

1. Conduct brief background study on data sortedness and indexes that exploit near-sorted data.
2. Develop a simple API that will use SMJ to sort two input data streams of integers.
3. Generate differently sorted data using the workload generator from the Benchmark on Data Sortedness (BoDS).
4. Measure the runtime of SMJ with differently sorted data.
5. Integrate the Quick Insertion Tree (QuIT) into the API.
6. Use QuIT to replace the ***sorting*** phase of SMJ and measure performance.
7. Write a report/short paper explaining the results.

Responsible Mentor: *Aneesh Raman*

References: