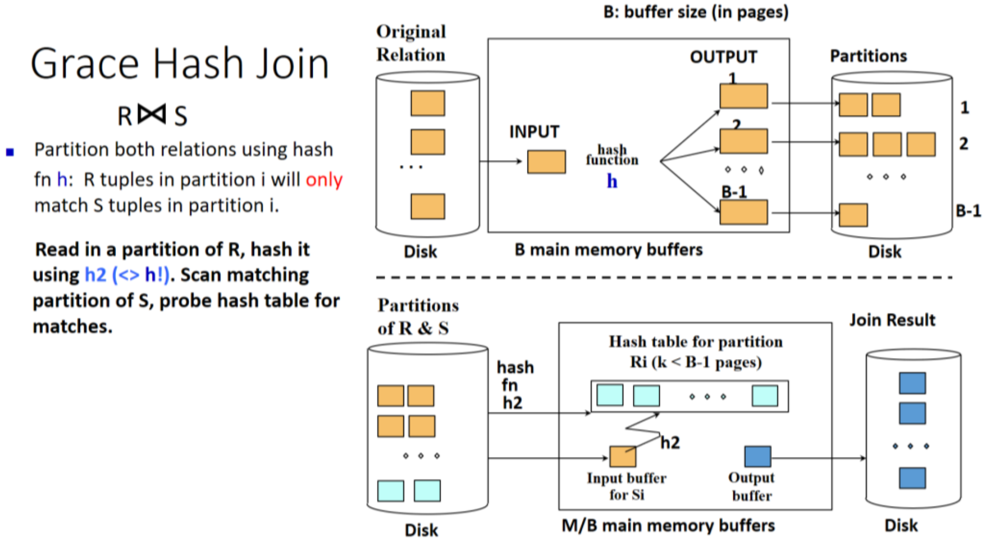
CAS CS 561: Data Systems Architectures
Data-intensive Systems and Computing Lab
Department of Computer Science
College of Arts and Sciences, Boston University
http://bu-disc.github.io/CS561/

CS561 Spring 2024 - Research Project

Title: *Boosting Join Implementation for Skew Correlation in Postgres*

Background: Join is a fundamental operator in relational database systems. Among different join implementations, hash join offers efficient performance and judicious use of memory. Below is the workflow of (Grace) hash joins:



Grace Hash Join

R⋈S

- Partition both relations using hash fn h: R tuples in partition i will only match S tuples in partition i.

Read in a partition of R, hash it using h2 (<> h!). Scan matching partition of S, probe hash table for matches.

Problem: Hash join uniformly distributes all the records regardless of the correlation of the join attributes. When there is a skew correlation during the join execution, a few partitions might be larger than the available memory which results in unnecessary I/Os. Existing skewness-based optimization in Postgres only caches the most frequent items in memory to avoid redundant I/Os. However, our analysis shows that only caching the most frequent items in memory may not be optimal for storage-based PK-FK (primary-key foreign-key) joins.

Objective: The objective of the project is to boost the hybrid hash join implementation for skew optimization in Postgres. The workflow is as follows:

(a) Understand the implementation of hybrid hash join in Postgres [1,2].
(b) Refine an existing partitioning strategy for partition-wise joins according to the existing skew optimization and integrate it into Postgres [3].
(c) Benchmark the performance for skew join under memory pressure (a modified TPC-H generator is given to produce skew correlation between tables `orders` and `lineitem`).

Responsible Mentor: *Zichen Zhu (zczhu@bu.edu)*
Postgres Repo: https://github.com/postgres/postgres
References:

[1] PostgreSQL Source Code: src/backend/executor/nodeHashjoin.c

[2] PostgreSQL Source Code: src/backend/executor/nodeHash.c

[3] Zhu, Z., Hu, X., & Athanassoulis, M. (2023). NOCAP: Near-Optimal Correlation-Aware Partitioning Joins. Proceedings of the ACM on Management of Data, 1(4), 1-27.