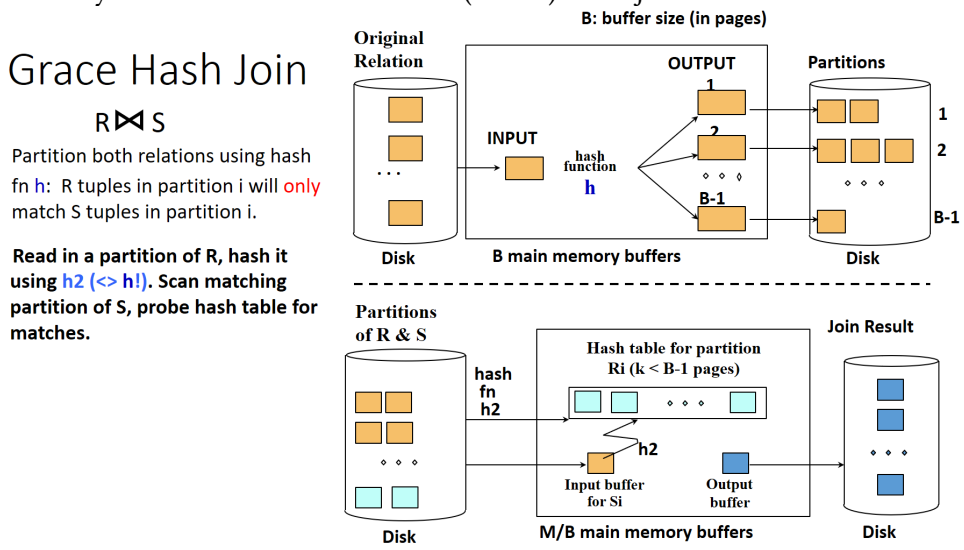
CAS CS 561: Data Systems Architectures
Data-intensive Systems and Computing Lab
Department of Computer Science
College of Arts and Sciences, Boston University
http://bu-disc.github.io/CS561/

CS561 Spring 2025 - 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:



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 uses some heuristic threshold to cache the most frequent items in memory to avoid redundant I/Os. However, our algorithm [1] shows that a better partitioning strategy can further reduce required I/Os 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) Read the implementation of hybrid hash join in Postgres [2,3].
(b) Refine an existing partitioning strategy for partition-wise joins according to the existing skew optimization [1] and integrate it into Postgres
(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*

Postgres Repo: https://github.com/postgres/postgres
NOCAP Repo: https://github.com/BU-DiSC/NOCAP-join

References:

[1] NOCAP: Near-Optimal Correlation-Aware Partitioning for Joins
[2] PostgreSQL Source Code: src/backend/executor/nodeHashjoin.c File Reference
[3] PostgreSQL Source Code: src/backend/executor/nodeHash.c File Reference