Sketch-based Join Order Selection for In-Memory Database Systems

Yesdaulet Izenov, Asoke Datta, Jun Hyung Shin and Florin Rusu
University of California, Merced
yizenov, adatta2, jshin33, frusu@ucmerced.edu

Optimal Join Order Selection

Goal: selecting an optimal order of relations involved in a given query so that the system consumes less amount of memory and cpu resources

Major variables that affects the performance of query execution:

• accurate selectivities • cost model • size of join order search space

Our Contribution

Pre-computation: represent relations in matrix forms that is build on true selectivities and capture appropriate join attribute values
Estimation: estimate join cardinalities via relations' sketch representations
Graph-based search: exaustive depth-first search on each node in join graph

Fast-AGM Sketch for Join Size Estimation

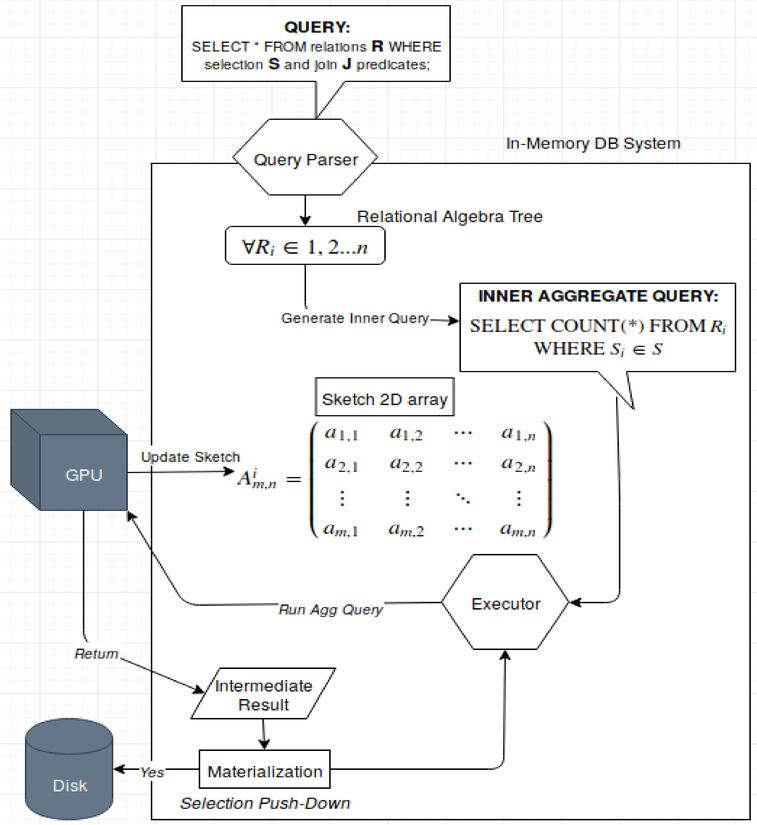
- 4-wise independent $\pm\mathbf{1}$ random variables
- 2-universal hash function h: I $\rightarrow \mathbf{1}\cdots\boldsymbol{n}$

hash functions | global randomized variables | sketch representation of T_i

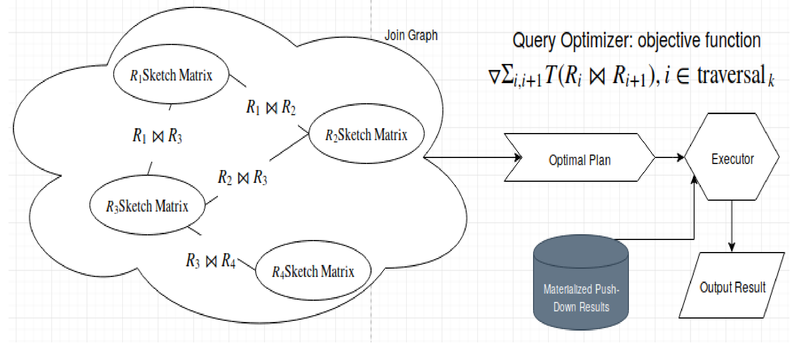
$$\sigma(T_i) \xrightarrow{\text{column value}} h(x) = \begin{pmatrix} h_{1,1} \\ h_{2,1} \\ \vdots \\ h_{m,1} \end{pmatrix} \rightarrow X_{m,n} = \begin{pmatrix} -1 & +1 & \cdots & +1 \\ -1 & -1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ +1 & +1 & \cdots & -1 \end{pmatrix} \rightarrow \vec{z} = \begin{pmatrix} +1 \times x \\ -1 \times x \\ \vdots \\ -1 \times x \end{pmatrix} \rightarrow A^i_{m,n} = \begin{pmatrix} 0 & 0 & \cdots & a_{1,n} + z_1 \\ a_{2,1} + z_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{m,n} + z_m \end{pmatrix}$$

$$R_1 \bowtie R_2 = Median(\vec{s}), \vec{s} = \begin{pmatrix} \Sigma_i^n (A^1_{1,i} \times A^2_{1,i}) \\ \Sigma_i^n (A^1_{2,i} \times A^2_{2,i} \\ \vdots \\ \Sigma_i^n A^1_{m,i} \times A^2_{m,i} \end{pmatrix}$$

Sketch Preparation Phase



Join Order Selection Phase



Dataset and Setup

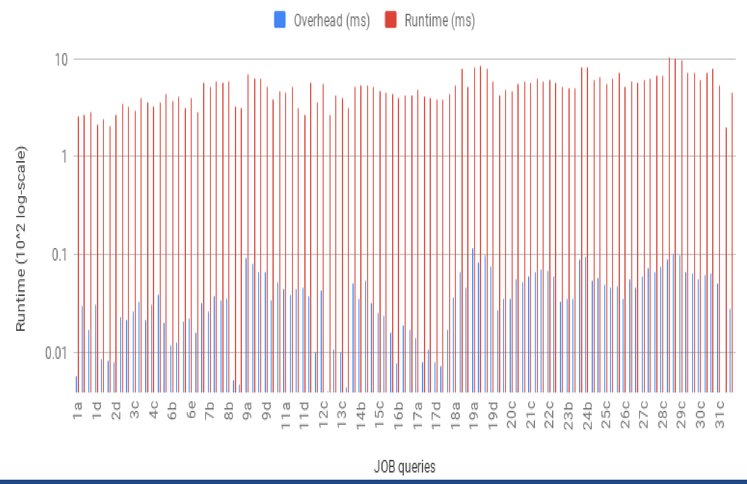
IMDB benchmark dataset: real-world dataset containing correlations and non-uniform data distributions
Join Order Benchmark: challenging realistic workload

- multi-join cyclic queries
- 28 join predicates in the most complicated query
- 17 involved relations in the most complicated query

MapD: in-memory, GPU-accelerated, column-oriented database system
Server Environment

- 2 Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz
- 1 Tesla K80 GPU
- 8 DDR4 memory 32GB @ 2400 MHZ (total: 256GB)

Runtime Overhead



Intermediate Join Cardinalities