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: *Project Q-QUEST: Are Questionable QUEry Selectivity Tolerable?*

Background: Query optimization plays a major role in the performance of databases and can potentially make or break usability. Generally, query optimizers rely on an estimated cost to select the best potential query plan among a set of candidate query plans. The cost is mostly dominated by the estimating the *query selectivity*, equivocally this problem is known as the cardinality estimation problem.[1] Several techniques have been developed to estimate cardinality, ranging from clever heuristics that leverage statistics from the database and replacing heuristics with learned cardinality estimators [1]. In practice, heuristics work well enough and provide decent performance [2]. However, query optimizers assume cardinality estimations are absolute, and do not consider potential errors that come from the fact the value is an *estimation.* Taking on

Objective: This project aims to study the performance sensitivity with respect to a databases cardinality estimation, and potentially implement a solution. If query optimizers consider cardinality estimations as absolute, we want to answer the question “How robust are query optimizers to incorrect cardinality estimations?”. We can achieve this by looking at the true cardinality, defining how far away our estimations are from our true cardinality, and increasing/decreasing the cardinality to understand performance. Lastly, we can work towards robust query plan selection; rather than selecting the query plan that minimizes an estimated cost, we can select the query plan that minimizes a maximum estimated cost.

Technical: The project will utilize PostgreSQL and the cardinality estimator they use. For reference, [3] mentions the files that show the logic PostgreSQL uses for cardinality estimation (e.g. *src/backend/utils/adt/selfuncs.c*). The following flow of tasks also act as potential goals/objectives for project QUESST:

(a) Create a pipeline in PostgreSQL artificially adds **parameterized** noise to an existing cardinality calculation (either estimate or true).
(b) Run experiments sweeping different “distances” away from the true cardinality value and check the performance.
(c) Implement a “robust query plan selector”, now that we understand the effect of wrong cardinality estimates lets improve the selection process starting with a heuristic. Generate multiple plans and test them with varying selectivity values centered around our estimation and pick the one that minimizes our maximum cost.

[1] Cardinality estimation specifically refers to estimating the absolute count of the output of a query, while query selectivity refers to estimating the percentage of tuples that satisfy a query predicate.

Responsible Mentor: *Andy Huynh*

References:

[1] Xiao Hu, Yuxi Liu, Haibo Xiu, Pankaj K. Agarwal, Debmalya Panigrahi, Sudeepa Roy, and Jun Yang. 2022. "*Selectivity Functions of Range Queries are Learnable."* Proc. of the 2022 International Conference on Management of Data (SIGMOD '22), 959–972. https://doi.org/10.1145/3514221.3517896

[2] Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter Boncz, Alfons Kemper, and Thomas Neumann. 2015. "*How Good Are Query Optimizers, Really?"* Proc. VLDB Endow. 9, 3 (November 2015), 204–215. https://doi.org/10.14778/2850583.2850594