COMPUTER SCIENCE AND DATA ANALYTICS

The Comparative Study of Indexing Techniques in Different Database Systems

Interim Report

Student: Sokrat Bashirov

GWID: G26315644.

Introduction

This report provides an overview of the current progress in research project, "The Comparative Study of Indexing Techniques in Different Database Systems." The primary goal of this project is to evaluate the impact of indexing on query performance in MySQL and PostgreSQL databases using a real-world dataset. By running a set of five representative queries 1000 times, the aim is to analyze the execution time for each query and compare the performance of both databases without any indexes.

Problem Description

Efficient data retrieval and query performance are crucial aspects of modern database management systems. Indexing is a fundamental technique employed to optimize data retrieval and improve overall database efficiency. The implementation of indexing can significantly impact the performance of database systems.

The primary objective of this research project is to conduct a comparative study of query performance in MySQL and PostgreSQL databases without any indexes and after the incremental addition of indexes. I aim to evaluate how indexing affects the execution time of five representative queries using a real-world dataset.

Current Progress

As of the current stage of my project, I have accomplished the following key milestones:

Data Preparation: I have successfully imported the "employees" database into both MySQL and PostgreSQL databases. All indexes, primary keys, and foreign keys were removed to ensure a level playing field for the comparative study.

Query Execution and Metric Collection: I have designed a set of five representative queries that cover a range of common use cases. Using the Python script provided in GitHub repository's code folder, I executed each query 1000 times on both MySQL and PostgreSQL databases without any indexes. I have also collected the query execution time metrics for each query.

Queries that I chose:

- SELECT emp_no, COUNT(*) AS count FROM employees GROUP BY emp_no;
- SELECT * FROM salaries WHERE salary = 94443 OR salary = 59571;
- SELECT E.*, S.* FROM employees E JOIN salaries S ON E.emp_no = S.emp_no WHERE E.first_name = 'Duangkaew';
- SELECT * FROM titles WHERE title LIKE 'senior%';
- SELECT E.*, T.* FROM employees E JOIN titles T ON E.emp_no = T.emp_no WHERE E.first_name = 'Duangkaew';

Histogram Visualization: Leveraging the collected metrics, I have created histograms to visualize the distribution of query execution times for each query on both databases without indexes. These histograms provide valuable insights into the performance differences between MySQL and PostgreSQL.

The resulted histogram for each query:

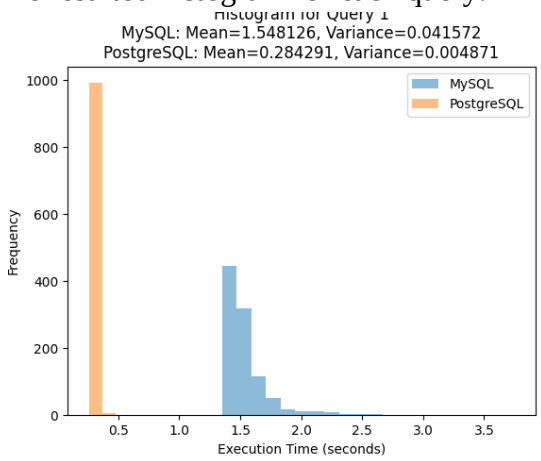

Fig1. Query 1

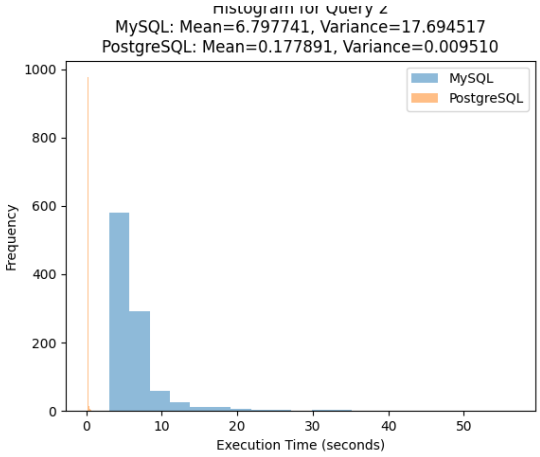

Fig2. Query 2

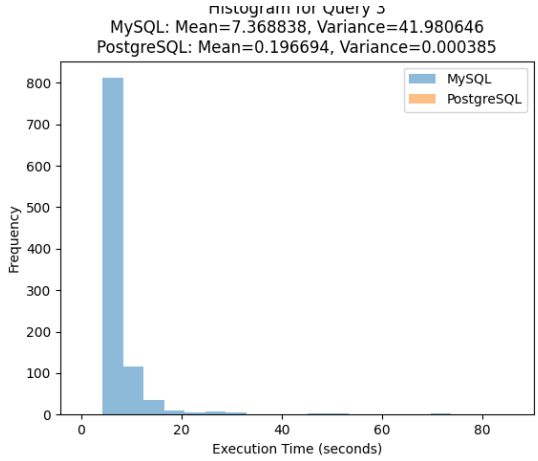

Fig3. Query 3

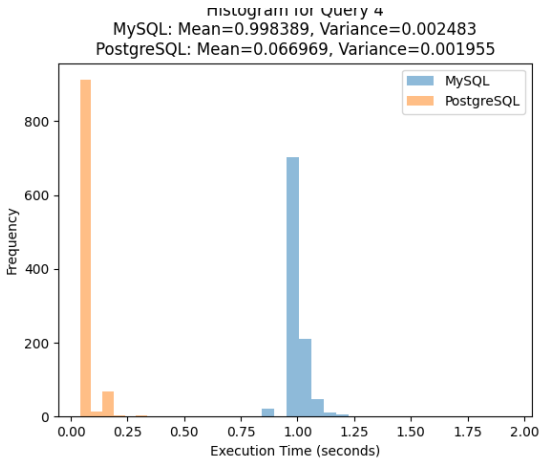

Fig4. Query 4

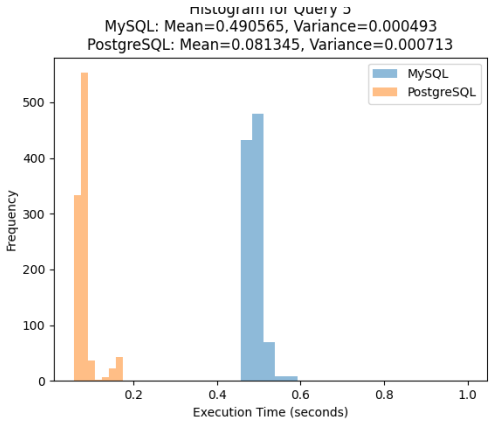

Fig5. Query 5

Query 1 - MySQL: Mean=1.548126, Variance=0.041572
Query 1 - PostgreSQL: Mean=0.284291, Variance=0.004871

Query 2 - MySQL: Mean=6.797741, Variance=17.694517
Query 2 - PostgreSQL: Mean=0.177891, Variance=0.009510

Query 3 - MySQL: Mean=7.368838, Variance=41.980646
Query 3 - PostgreSQL: Mean=0.196694, Variance=0.000385

Query 4 - MySQL: Mean=0.998389, Variance=0.002483
Query 4 - PostgreSQL: Mean=0.066969, Variance=0.001955

Query 5 - MySQL: Mean=0.490565, Variance=0.000493
Query 5 - PostgreSQL: Mean=0.081345, Variance=0.000713

Analysis

Upon analyzing the query execution time histograms for both MySQL and PostgreSQL databases, a noticeable performance disparity has emerged. The results indicate that PostgreSQL consistently outperforms MySQL, demonstrating significantly lower query execution times. There are several reasons that could explain this observed performance difference:

- Query Optimization: PostgreSQL is renowned for its advanced query optimizer, which intelligently chooses the most efficient execution plan for queries. This may lead to faster execution times compared to MySQL, especially for complex queries.
- Concurrency Control: PostgreSQL's handling of concurrent transactions and locking mechanisms may also contribute to its superior performance. Depending on the workload and query concurrency, PostgreSQL may handle multiple queries more efficiently.

Future Work

While I have made progress in the project, there are several important tasks ahead:

Indexing Strategy Implementation: The next phase involves incrementally adding indexes to the tables in both MySQL and PostgreSQL databases. After each index addition, I will re-run the queries and collect performance metrics to assess the impact of indexing on query execution times.

Statistical Analysis: Once we have data on query performance with and without indexes, I will perform a comprehensive statistical analysis to determine the significance of the performance differences and assess the effectiveness of indexing techniques.

Conclusion

The current progress in my research project has laid a strong foundation for the comparative study of indexing techniques in MySQL and PostgreSQL databases. By executing representative queries 1000 times and collecting essential metrics, I have obtained valuable insights into the query execution times without any indexes. The histograms provide clear visualizations of the performance distribution for each query.

Moving forward, I will implement indexing strategies and conduct further analysis to draw conclusive insights into the impact of indexing on query performance. I anticipate that the results of this study will contribute valuable knowledge to the field of database optimization, aiding developers and administrators in making informed decisions regarding indexing strategies for improved query performance.