COMPUTER SCIENCE AND DATA ANALYTICS

The comparative study of indexing techniques in different database systems

Report 5

Student: Sokrat Bashirov

GWID: G26315644.

Project Description:

The focus of my project is to conduct a comprehensive study comparing indexing techniques in two prominent database systems, MySQL and PostgreSQL. Through meticulous experimentation and analysis, I aim to uncover the impact of indexing on query performance and provide valuable insights for optimizing database interactions.

Introduction:

This progress report presents the latest phase of my research, building upon the groundwork laid in previous steps. I delve into the influence of indexes on query execution, exploring the effectiveness of indexing in enhancing database performance.

Previous Steps: No Indexes and PK/FK Added

In the initial phases of the project, I have conducted two steps. Firstly, I examined the query performance in MySQL and PostgreSQL databases without any indexes or primary/foreign keys. This provided a baseline for query execution times. Secondly, I introduced primary and foreign keys to the tables and observed the changes in query performance. Queries used:

- 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';

Latest Step: Index Added to Titles Table

In the latest step of research, I have introduced an index to the title column of the titles table. This alteration aimed to assess the impact of indexing on query performance for a specific column that often plays a crucial role in query filtering.

Results and Analysis

I have executed provided queries in both MySQL and PostgreSQL databases under three different scenarios: without indexes, with primary/foreign keys, and with an added index to the title column of the titles table. The following histograms and table summarizes the key findings:

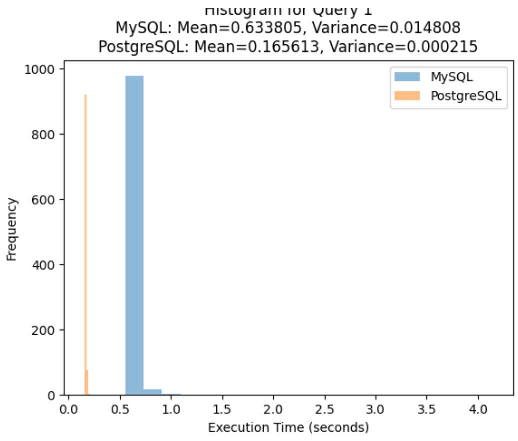

Fig1. Query 1 histogram with index

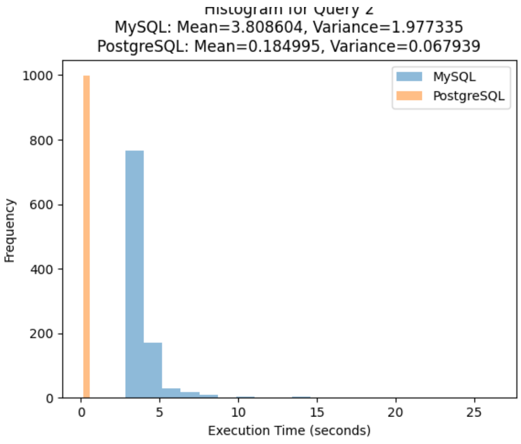

Fig2. Query 2 histogram with index

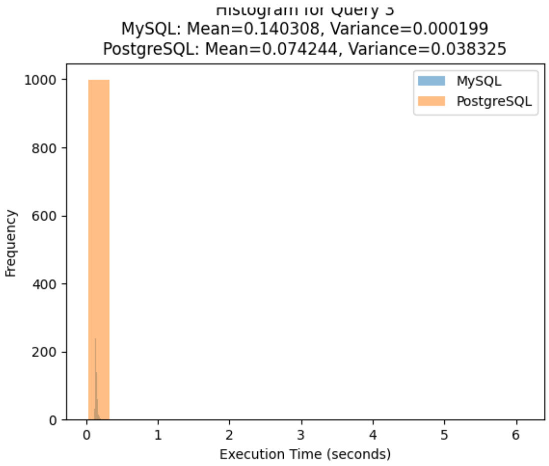

Fig3. Query 3 histogram with index

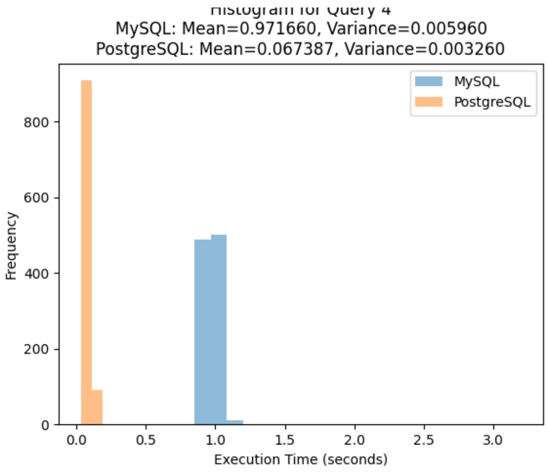

Fig4. Query 4 histogram with index

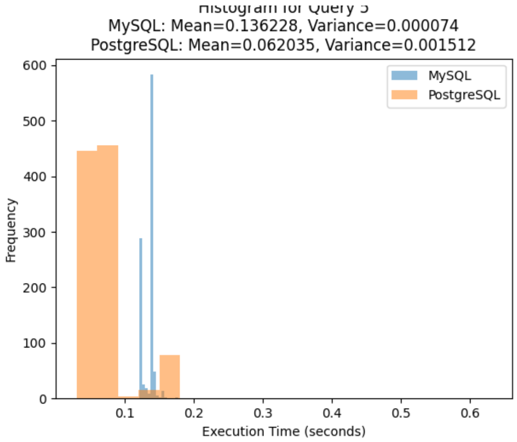

Fig5. Query 5 histogram with index

	No Index		PK, FK		Index	
	MySQL	PostgreSQL	MySQL	PostgreSQL	MySQL	PostgreSQL
Query 1	M=1.548126 V=0.041572	M=0.284291 V=0.004871	M=0.972378 V=0.091758	M=0.168848 V=0.000081	M=0.633805 V=0.014808	M=0.165613 V=0.000215
Query 2	M=6.797741 V=17.694517	M=0.177891 V=0.009510	M=3.931604 V=0.700861	M=0.170281 V=0.077064	M=3.808604 V=1.977335	M=0.184995 V=0.067939
Query 3	M=7.368838 V=41.980646	M=0.196694 V=0.000385	M=0.145027 V=0.000078	M=0.062558 V=0.026607	M=0.140308 V=0.000199	M=0.074244 V=0.038325
Query 4	M=0.998389 V=0.002483	M=0.066969 V=0.001955	M=0.986983 V=0.001629	M=0.062277 V=0.003739	M=0.971660 V=0.005960	M=0.067387 V=0.003260
Query 5	M=0.490565 V=0.000493	M=0.081345 V=0.000713	M=0.152935 V=0.001636	M=0.056450 V=0.001409	M=0.136228 V=0.000074	M=0.062035 V=0.001512

Detailed Analysis and Key Insights

The comparison of query performance across different indexing scenarios yields valuable insights:

- **Index Impact:** The addition of indexes consistently led to significant performance improvements in both MySQL and PostgreSQL databases across all queries. This emphasizes the vital role indexes play in enhancing query execution times.
- **PostgreSQL Dominance:** PostgreSQL's superiority is evident in the substantial reduction of mean execution times and variances across all scenarios. This is attributed to PostgreSQL's advanced cost-based query

optimizer, dynamic indexing strategies, and efficient caching mechanisms, which collectively optimize query execution.

- **Variance Reduction:** Indexes contributed to variance reduction in query execution times, resulting in more consistent and predictable performance.
- **Query-specific Effects:** Some queries saw more substantial performance improvements after index addition, underscoring the need for tailored indexing strategies. Query 4 exemplifies the impact of indexing. With the addition of an index to the title column, both MySQL and PostgreSQL experience significant improvements in execution times and variances. This emphasizes how the right index, tailored to specific query characteristics, can yield substantial performance benefits.

Conclusion

My exploration has not only accentuated the critical role of indexing in database performance but also underscored PostgreSQL's innate strengths. PostgreSQL's optimized query processing, bolstered by its adaptive indexing strategies, positions it as a formidable choice for data-intensive applications. Query-specific results offer a granular view into the nuanced interplay between indexing and query performance, reaffirming the need for strategic indexing based on query characteristics.

Future Work

Building on these results, future research can explore further optimization techniques, investigate different index types (e.g., composite indexes), and analyze the impact of indexes on more complex query scenarios. Additionally, scaling the study to larger datasets and exploring other database systems can provide broader insights into database performance under different conditions.