Benchmarking Study – Checkpoint 1

Identifying Use Case, Sample Data Sources, and Database Systems Study

Pooja Bendre, Ezhilarasu R, and Ritesh Kumar

2024WI_MS_DSP_420-DL_SEC61_SEC62: Database Systems

Benchmarking Study – Term Checkpoint 1

Abid Ali and Jaya R

February 4, 2024

Introduction:

In the rapidly evolving landscape of financial markets, the capacity to efficiently manage and analyze large datasets, such as daily equity Futures and Options (F&O) data from India's largest stock exchange, has become paramount. This necessity underscores the critical role of database systems, which serve as the backbone for storing, retrieving, and processing financial information. With a plethora of database technologies available—ranging from traditional relational databases like PostgreSQL and MySQL to NoSQL options such as MongoDB and graph databases like Neo4J—selecting the optimal system for specific financial data analysis needs poses a significant challenge. Consequently, there is a pressing need for a comprehensive benchmarking study that evaluates these diverse database systems based on their performance, efficiency, and resource utilization when handling complex financial datasets.

This study aims to bridge this gap by employing Python and SQL to conduct a thorough comparison of selected database technologies. By focusing on key performance indicators (KPIs) such as database and table creation time, query performance, resource utilization, and data handling efficiency, the research intends to offer invaluable insights into the most suited database technology for financial market data analysis. The outcome of this study will not only assist database administrators and financial analysts in making informed decisions but also contribute to the broader field of financial technology by elucidating the strengths and weaknesses of each database system in the context of high-volume, dynamic financial data.

For the benchmarking study on database performance using daily equity Futures and Options (F&O) data from India's largest stock exchange, here's a detailed plan:

Database Systems:

- PostgreSQL: An open-source, object-relational database system known for its robustness, scalability, and support for complex data types and advanced queries.
- MySQL: Another popular open-source relational database management system, widely used for its performance, reliability, and ease of use.
- Neo4J: A graph database that excels in storing and querying interconnected data, which could provide unique insights for complex financial market relationships.

Data Set:

The NSE India Futures & Options Daily (2000-20) dataset encompasses two decades of comprehensive trading data from the National Stock Exchange of India, one of the largest stock exchanges globally. This rich dataset includes detailed records of daily transactions in the futures and options market, covering a wide array of financial instruments. Key data points include opening and closing prices, high and low values, trading volumes, and open interest figures, providing an invaluable resource for financial analysis, market prediction, and academic research. By capturing the evolution of market dynamics over twenty years, this dataset offers unique insights into the trends, behavior, and fluctuations of the Indian financial markets, serving as a crucial tool for traders, investors, and policymakers alike.

Languages for the Study:

- Python: Will be the primary language due to its widespread use in data science, availability of powerful libraries (pandas for data manipulation, SQLAlchemy for database connections, matplotlib for data visualization), and its ability to integrate with various database systems efficiently.
- SQL: Essential for creating, querying, and managing database systems. It will be used to interact with relational databases like PostgreSQL and MySQL directly.

- R (Optional): Might be considered for specific statistical analyses or if specific financial analysis packages in R offer advantages over Python counterparts.

Benchmark Study Design:

Phase 1: Setup and Configuration:

- Select and configure the database systems on a standardized testing environment.
- Prepare the datasets, ensuring they are normalized and indexed appropriately for each database type.
- Define the schema for relational databases and the document/graph structure for databases.

Phase 2: Execution of Benchmark Tests

- Database and Table Creation: Measure the time and resources required to create databases and tables/collections/nodes.
- Data Import/Loading: Evaluate the efficiency of importing the dataset into each database system.
- Query Performance: Execute a series of SQL and NoSQL queries ranging from simple data retrievals to complex aggregations and joins. Measure the response time and resource utilization.
- Concurrency and Scalability Testing: Simulate multiple concurrent accesses to assess each database's handling of simultaneous requests.

Phase 3: Analysis and Reporting

- Collect and analyze the data on performance metrics like execution time, CPU, and memory usage.

- Compare the performance across different database systems and identify trends and outliers.
- Compile the findings into a report, highlighting the strengths and weaknesses of each database system in handling financial market data.

Concerns About Conducting This Research:

- Data Volume and Complexity: Handling large volumes of financial data can be challenging, especially ensuring the data's integrity and relevance throughout the testing phases.
- Fair Comparison: Designing tests that fairly compare relational and NoSQL databases, given their different data models and use cases.
- Resource Availability: Ensuring access to sufficient computational resources to accurately simulate real-world database loads and performance.
- Keeping Tests Up-To-Date: Database technologies evolve rapidly, and maintaining the relevancy of the benchmark tests over time can be challenging.
- Interpretation of Results: Drawing meaningful conclusions from the data, considering the multitude of factors that affect database performance, and translating these findings into actionable recommendations.

References:

1. Data: NSE India Futures & Options Daily (2000-20) https://www.kaggle.com/datasets/tanay001/nseindia-futures-options-daily/data
2. NSE Historical Reports https://www.nseindia.com/resources/historical-reports-capital-market-daily-monthly-archives
3. Practical SQL – Second Edition, Debarros A., https://nostarch.com/practical-sql-2nd-edition
4. Data Engineering with Python, Crickard P., https://www.packtpub.com/en-fi/product/data-engineering-with-python-9781839214189?type=ebook
5. Efficient MySQL Performance, Nichter D., https://www.oreilly.com/library/view/efficient-mysql-performance/9781098105082/#:~:text=Daniel%20Nichter%20shows%20you%20how,the%20most%20important%20MySQL%20metrics
6. Building Knowledge Graphs: A Practitioner's Guide, Barrasa J. and Webber J, https://neo4j.com/knowledge-graphs-practitioners-guide/?utm_source=google&utm_medium=PaidSearch&utm_campaign=GDB&utm_content=AMS-X-SEM-Category-Expansion-Evergreen-Search&utm_term=&gad_source=1&gclid=CjwKCAiAiP2tBhBXEiwACslfnlvKh8bgC8n93lU8rhc_BJk5IR-eFhNCW5BNieF3L8AWfPS8rEagZBoC6ugQAvD_BwE