

Survey of RDBMS vs. Graph Database Performance Using Time Series Data Manipulation

Paul Marquardt and Kristen Rollins

Abstract—Performance constitutes an important aspect of database management and application development. An interesting project consists of measuring the performance, data model differences, and capabilities between both a graphing database system and RDBMS. This project compares times series data with data manipulation to capture and measure DBMS internal and OS level metrics including CPU, RAM, and disk performance statistics.

Index Terms—databases, relational databases, MySQL, graph databases, Neo4j

I. PROBLEM STATEMENT

WE will test the suitability of an RDBMS system to manipulate times series data versus a graphing database to compare performance statistics.

We know that different types of database systems are more appropriate to use in some situations than others. Based on the interrelated nature of time series data, we suspect that a graph database system would provide an efficient and suitable representation for such data. Conversely, we believe that a relational database system would be an inefficient and inappropriate choice for storing and querying time series data. A suitable application to test this theory is creating a volatility index. Within this domain, we will perform functionally identical queries on the RDBMS and graph database representations and compare performance statistics. We will also evaluate how appropriately the two systems represent the relationships within the time series data.

II. RESEARCH METHODOLOGY

We will first identify data requirements, hardware, software, and configuration requirements for both the RDBMS and graphing database systems. Next, we will create logical data models to represent the time series data for both the relational and graph databases. When approaching the modeling stage, we will carefully consult previous work and examples to grasp how graph databases can effectively represent our data, as we are less familiar with modeling data graphically. We will finally calculate a volatility index from the last 22 days of data to test the claim that a graphing database system performs better with time series data versus a relational database management system.

Steps to complete this project are further broken down in our research timeline found in section IV.

This project proposal was submitted for a grade on February 3, 2019. The project was completed under the supervision of Dr. Daniel Engels at Southern Methodist University.

The authors can be contacted by email: Paul Marquardt at pmarquardt@smu.edu and Kristen Rollins at kmrollins@smu.edu.

III. PREVIOUS RELATED WORK

Based on a preliminary search, resources are available showing performance and usability comparisons between relational and graph databases for various cases, with one source reporting that the graph database was 1,000 times faster for their particular use case [1]. Neo4j itself has resources to help transition between relational and graph database models, which we will utilize while constructing our graphical data model [2]. In addition, research has been performed to measure performance statistics of Neo4j in particular for various use cases. These works include a performance comparison of Neo4j to other graph database systems [3], efficiency and usability comparisons between MySQL and Neo4j's query language Cypher [4], and an evaluation of the storage efficiency of Neo4j [5]. However, we did not readily find research in our specific domain of time series for explicit comparison between the two types of database systems.

IV. RESEARCH SCHEDULE

<i>Schedule</i>	<i>Objective</i>
Week 4	1. Obtain time series data (Quantsshare/Yahoo historical data/AMEX, NASDAQ, NYSE) 2. Obtain hardware (PowerEdge R720/Windows Server 2016) 3. Obtain software (MySQL/Neo4j) 4. Submit Final Project Proposal document
Week 6	5. Install/Configure software (MySQL/Neo4j) 6. Develop/document RDBMS data model (MySQL)
Week 7	7. Develop/document graphing data model (Neo4j) 8. Develop/document data manipulation to calculate volatility index (MySQL) 9. Develop/document data manipulation to calculate volatility index (Neo4j) 10. Create presentation slides for Project Initial Presentation
Week 9	11. Perform/document small test pilots to determine the calculation is correct and give a baseline of performance statistics for RDBMS (MySQL) 12. Perform/document small test pilots to determine the calculation is correct and give a baseline of performance statistics for graphing database system (Neo4j) 13. Compile documentation into Final Project Draft 1

<i>Schedule</i>	<i>Objective</i>
Week 10	14. Perform/document full test for RDBMS (MySQL) 15. Perform/document full test for graphing database system. (Neo4j)
Week 11	16. Complete/document a comparative analysis of results from RDBMS (MySQL) and graphing database system (Neo4j) performance statistics
Week 13	17. Complete documentation of analysis results from RDBMS (MySQL) 18. Complete documentation of analysis results from graphing database system (Neo4j)

V. RESOURCES NEEDED

We already have access to all of the resources we need to complete this project. The stock data we will use is publicly available. The database systems we have chosen to utilize, MySQL and Neo4j, are both open-source. Performance monitoring tools we will use include the Microsoft Performance Monitor as well as internal performance statistics from MySQL and Neo4j. We will also utilize hardware that Paul owns or has access to, including a PowerEdge R720 and Windows Server 2016.

REFERENCES

- [1] J. Adell, "Performance of graph vs. relational databases - dzone database," Feb 2013. [Online]. Available: <https://dzone.com/articles/performance-graph-vs>
- [2] "Relational databases vs. graph databases: A comparison." [Online]. Available: <https://neo4j.com/developer/graph-db-vs-rdbms/>
- [3] F. Holzschuher and R. Peinl, "Querying a graph database – language selection and performance considerations," *Journal of Computer and System Sciences*, vol. 82, no. 1, p. 45–68, 2016.
- [4] D. Dominguez-Sal, P. Urbón-Bayes, A. Giménez-Vañó, S. Gómez-Villamor, N. Martínez-Bazán, and J. L. Larriba-Pey, "Survey of graph database performance on the hpc scalable graph analysis benchmark," *Web-Age Information Management Lecture Notes in Computer Science*, p. 37–48, 2010.
- [5] F. Gong, Y. Ma, W. Gong, X. Li, C. Li, and X. Yuan, "Neo4j graph database realizes efficient storage performance of oilfield ontology," *Plos One*, vol. 13, no. 11, Nov 2018.