IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 8 (August. 2013), ||V2|| PP 25-27

Comparative analysis of Relational and Graph databases

Garima Jaiswal, Arun Prakash Agrawal
[1](*Amity School of Engineering & Technology Noida, India)*
[2](*Amity School of Engineering & Technology Noida, India)*

Abstract: The relational model has dominated the computer industry since the 1980s mainly for storing and retrieving data. Lately, however, relational database has been losing its importance due to its reliance on a strict schema which makes it difficult to add new relationships between the objects. Another important reason of its failure is that as the available data is growing manifolds, it is becoming complicated to work with relational model as joining a large number of tables is not working efficiently. One of the proposed solutions is to transfer to the Graph databases as they aspire to overcome such type of problems. This paper provides a comparative analysis of a graph database Neo4j with the most widespread relational database MySQL.

Keywords: - *Flexibility, Maturity, Security, Retrieval, Schema*

I. INTRODUCTION

A relational database is a database that has a collection of tables of data items, all of which is formally described and organized according to the relational model. In the relational model, each table schema must identify a primary column used for identifying a row called the primary key. Tables can relate by using a foreign key that points to the primary key. The relational model offers various levels of refinement of the table relations called database normalization. The database management system (DBMS) of a relational database is called an RDBMS. Relational Databases have been providing the storage space support for many decades now with implementations like Oracle, MySQL, etc. [1].

People used database just for storing tabular data like purchase reports and finance records. But in today's environment, these relational representations are not efficient in performing frequent operations for example the World Wide Web exhibits far more complicated networks of relationships than were expected when SQL was designed. The network of hyperlinks connecting all the pages on the World Wide Web is highly complex and almost impossible to model efficiently in a relational database [2].

Similar issues are involved in modeling the social network like Twitter, Facebook, etc. Implementing such problems in relational databases involves large number of joins which is costly to be calculated [3]. With the intense increase in the usage of the internet leading to need for storing large amounts of interconnected data, there was a clear desire for a data store modified to the needs of graph data. Graph databases are optimized for these types of networks (social networking and website link structure), as the graph is a usual way of storing connections between users. Relational databases are not helpful when the data model evolves over time [4], which means relational databases depend on stiff schema and make it complicated to add new relationships between objects. All these boundaries of relational databases led to the innovation of graph databases.

A. Graph Databases

Graph Database is a database system where the associations between objects or entities are similarly as important as the objects themselves [5]. In a graph database, data are represented by nodes, edges and properties. Nodes are represented as objects and edges manifest the relationship between nodes. There are several implementations of graphical database. Both nodes and edges can have properties that illustrate their particular characteristics. Some of the best known graph databases are: Infogrid, Hypergraph DB, Jena, DEx, FlockdB and Neo4j [6] -[8]. Out of these only Neo4j is discussed here.

B. Neo4j

Neo4j is an open-source graph database supported by Neo Technology.Neo4j stores data in nodes connected by directed, typed relationships with properties on both, also known as a Property Graph [9].
Main features of Neo4j are:

- Intuitive, using a graph model for data representation.
- Reliable, with full ACID transactions.
- Durable and fast, using a tradition disk-Based, native storage engine.
- Extraordinarily scalable, up to several billion nodes/relationships/properties.
- Highly-available, when distributed across multiple machines.

- Expressive, with a powerful, human readable graph query language.
- Fast, with a powerful traversal framework for high-speed graph queries.
- Embeddable, with a few small jars.
- Simple, accessible by a convenient REST interface or an object-oriented Java API [10].

It is comprised of two parts:
(1) **-** A client that sends instructions to the server via RMI.
(2) **-** A Server that processes these commands and sends back the processed result to the client.

The database is queried through Cypher Query Language. It is a latest query language that has been recently added to the Neo4j. Unlike imperative languages like Java and scripting language like Gremlin and the Ruby Neo4j bindings, Cypher is a declarative language. Using Cypher, efficient querying of the graph is possible, without having to write traverses in the code.

II. ASSESMENT PARAMETERS FOR GRAPH DATABASE AND RELATIONAL DATABASE

The assessment between MySQL and Neo4j is based upon different criteria [4]. These criteria are the pillars to decide which database should be adopted for implementation.

A. Level of Support/Maturity

Maturity refers to how well the system is tested. If a system has been tested, number of times, it means it is more secure and more bugs have been found out. Maturity of a system is proportional to level of support. Relational Databases have been providing storage support for decades now. So they are more stable and mature.
Relational databases have a unified language SQL. As SQL does not differ much between implementations, support for one implementation is applicable to others as well. Since Neo4j version 1.1 was released in February 2010, it is less stable and less mature.
Neo4j is still growing and maturing and has not undergone the same rigorous performance testing as relational databases. Most of the support comes from its parent company's website www.neo4j.org and is limited from outside of Neo4j site.

B. Security

MySQL has extensive multi user support. However Neo4j does not have any built in mechanisms for managing security restrictions and multiple users [11]. It presumes a trusted environment. Although there are Access Control List security mechanisms but even Access Control List management is handled at the application layer. On the other hand, there is extensive support for ACL based security in MySQL.

C. Flexibility

Although relational databases are more mature and secure as compared to graph databases, but its schema is fixed, which makes it difficult to extend these databases [12] and less suitable to manage ad-hoc schemas that evolve over time

III. IMPLEMENTATION DETAILS

The evaluation between MySQL and Neo4j is based upon a set of predefined queries.
To implement relational databases, MySQL version 5.1.41 was used. The database was queried using PHP scripting language.
Graph databases are implemented using Neo4j Community version 1.6. The database is queried with Cypher Query Language.
Queries were designed to analyze the performance difference between a relational database and a graph database.

Schema for relational database included the following tables:
1) Student: stu_id, stu_name
2) Friends: stu_id, friend_id
3) Department: stu_id, dept_name
4) Faculty: dept_name, fac_name

The three queries defined here:
S0: Find all friends of Nimika.
S1: Find the department name of Nimika's friends.
S2: Find the faculty name who worked in Nimika's friends department.

IV. EXPERIMENTAL RESULTS

Execution times were collected after executing the queries and noted in milliseconds. The results have been tabulated in table 1. It can be easily observed from the values retrieved (TABLE 1.) that the retrieval times of graph databases are less than relational databases. This is because relational database search all of the data to find the data that meets the search criteria. The larger the data set, the longer it takes to find matches, so when the number of users get increased from one hundred to five hundred, the retrieval time gets increased manifold. On the other hand graph database looks only at records that are directly connected to other records; it does not scan the entire group to find the nodes that meet the search criteria. So, increasing number of nodes from one hundred to five hundred does not increase the retrieval time much as can be visualized from the graphs.

Table 1: Query Results In Milliseconds

No_of_Objects	MySQL:S0	Neo4j:S0	MySQL:S1	Neo4j:S1	MySQL:S2	Neo4j:S2
100	19.56	8	33	12.65	111.34	19.57
500	281.38	10	333.96	17	620.56	21

V. CONCLUSION

Both systems performed well on the objective and subjective benchmarks. In general, graph databases performed better when objective tests were performed. This implies that graph databases retrieve the results of the set of predefines query faster than relational databases.

Not only this, graph databases are more flexible than relational databases as new relationships can be added to graph databases without the need to restructure the schema again. With such a difference in the query retrieval time of MySQL and Neo4j, Neo4j can be used for commercial purposes like website link structures and social networking.

REFERENCES

[1] Chad Vicknair, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, Dawn Wilkins "A Comparison of a Graph Database and a Relational Database", ACM Southeast Regional Conference, 2010

[2] Database Trends and Applications. Available http://www.dbta.com/Articles/Columns/Notes-on-NoSQL/Graph-Databases-and-the-Value-They-Provide-74544.aspx,2012.

[3] M. Kleppmann. Should you go beyond Relational databases ?Available http://carsonified.com/blog/dev/should-you-go-beyond-relational-databadat.

[4] Neo4j Blog, Available http://blog.neo4j.org/2009/04/current-database-debate-and-graph.html.

[5] M. I. Jordan (Ed). (1998),"Learning in Graphical Models". MIT Press.

[6] HypergraphDB website, Available http://www.kobrix.com/hgdb.jsp

[7] Jena documentation, Available http://jena.sourceforge.net/documentation.html

[8] Infogrid. Blog, Available http://infogrid.org/blog/2010/03/operations-on-a-graph-databae-papa-4.

[9] Neo4j. Home. http://neo4j.org, 2012

[10] D. Dominguez-Sal, P. Urban-Bayes, A.Gim´enez-Va˜n´o, S.G´omez-Villamor, N. Mart´ınez-Baz´an, and J.L. Larriba-Pey, "Survey of Graph Database performance on the HPC Scalable Graph Analysis Benchmark", Proceeding WAIM'10 Proceedings of the 2010 international Conference on Web-age information Management.

[11]` T. Ivarsson. [Neo] security, Available http://lists.neo4j.org/pipermail/user/2009-November/001955.html,2011.

[12] R. Angles and C. Gutierrez," Survey of Graph databasemoDels", ACM Comput. Survey., 40 (1):1–39, 2008.

[13] Neo4j. The neoz database (2006), Available http://dist.neo4j.org/neo-technology introduction.pdf