Review of Business and Technology Research, Vol. 13, No. 1, 2016, ISSN 1941-9414

IMPROVING STORAGE CAPACITY OF DATABASE SYSTEM USING GRAPH

Deepak Kumar, Amity University Uttar Pradesh, India (deepakgupta_du@rediffmail.com)

Prashant Gupta, Amity University Uttar Pradesh, India (prashantdewalya@gmail.com)

Daniel Okunbor, Fayetteville State University, USA (diokunbor@gmail.com)

ABSTRACT

Today database is an important part of electronic computing device like desktop, laptop, mobile device system, etc. All devices are deal with database. Electronic computing device depends on database storages system. Data base is backbone in any electronic computing device in 21st century. When any electronic computing device interacts with computing machine, electronic data are grow day by day. We expect that data to be stored for future use. Relational databases management systems (RDBMS) have been the power-horse of software applications since the 80s, and continue at this stage. They store highly structured data in tables with predetermined columns of certain types and rows which contains similar type of information. Now a day, data are more complex as billions of devices are connected through wide area network (internet). Also, it is more complex to handle by RDBMS query language. Storage capacity of connected data can be improve by using graph database. Graph database is a buzz word in the Graph data that overcome the issue which are faced by traditional database system. In this paper, we have shown that the storage capacity of connected database can be improve by using graph database. Numerically these conditions have been described by Oracle (11.g) and Neo4j (v2.1.5).

Keywords: Graph Database, Neoj4j, Database, Storage System, Efficiency

INTRODUCTION

The limitations of traditional databases storage model in particular the relational model performance degrades with number and level of relationship and database size increases, if any changes made in database to cover the requirements of current application domains. In traditional database query complexity grow with needs of joins operations. Adding new type of data and relation in database requires schema redesign (Angles Renzo & Gutierrez C., 2008). Data volume and join number affected cost query operation exponentially. These are the issue faced in traditional database systems, which affects storage capacity of connected database. Neo4j is a world largest graph database that provides graph base database system. A graph database on the internet DBMS system having the functions like create, read, update and delete (Neo4jmanual, 2010). Graph databases are built to use in transactional (OLTP) systems and are engineered with transactional integrity and operational availability. Graph data modeling can improve the connected database using graph database. Cypher query based language improves the efficiency of database compare to relational database system. Cypher is a graph database query language which is used to store a data into a node and data are highly connected in cypher query language. Cypher is a graph query language which is declarative and it allows for efficient expressive querying as well as updating of the graph data store (Angles Renzo & Gutierrez C., 2008).

In recent years, software developers have been investigating storage alternatives for relational databases system. Neo4j, cypher query is a term for some of those new systems. Connected database graph based projects like Walmart and facebook, social networking site are used to stored database. neo4j is used to store data. facebook is social networking site and Walmart is store projects, which uses high-volume data storage in the system. So, they reject the object and relational model (Paredaens J. & Kuijpers B., 1998). Relational database management system deals with data to store the data in rows and columns in the table. When data are more complex and highly connected then the performance of computer degrades. Today data are more connected and more complex. When data are highly connected in relational database management system, schemas are redesign, which is typical to

handle. To overcome this problem we can use graph database. Neo4j database is used to stored data into node and relational form, which can be responsible for highly connected data. In cypher query language, data is stored in graph form. Data are highly connected, which overcome the issue of relational database management system. Neo4j graph database is a strong, versatile and superior database. Neo4j is suitable for full undertaking organization or a subset of the full server can be utilized as a part of lightweight (Neo4jmanual, 2010).

Rest of the paper is as follow. Next section deals with capacity model of graph database. Followed by this section, it deals with compression analysis of traditional database and graph database query language. After this section, it deals with the performance based evaluation. At last section we conclude the paper.

CAPACITY MODEL OF GRAPH DATABASE

A. File capacity

Neo4j technology is based on java for all files handling non blocking I/O file system. At the same time, the storage records, blue print is optimized for mutually dependent data. Neo4j does not require order device. Hence file capacity to handle huge file can be done at all time. Hence ACID velocity reduces easily as RAM becomes the limiting factor (Paredaens J. & Kuijpers B., 1998).

B. Read velocity

Business Company wants to enhance the use of hardware to distribute the maximal business output from available resources (Angles Renzo, 2013). Neo4j does not chunk or latch any read operations. Hence, there is no danger for deadlocks in read and transaction operation. Along threaded read approach to the database, queries can be run together on as many as processors available at that time. That contributes very good scale-up scenarios with large server.

C. Write speed

Write speed is an attention for many business firm applications. There are two different kind of plot

- Continuous sustained action
- Large volume access

To backing the various requirement of these cases. Neo4j support two modes of writing to the storage layer.

In Transactional, an ACID compliment normal operation remote layer is maintained. Read operation can be appear at the same times with the writing process. At each one act as the data is persisted to disk and can be reborn to a consistent state upon system failure. These required disks write access and an actual flushing of data. Hence the write speed of Neo4j on an individual server is in continuous mode and it is bound by the I/O capacity of the hardware. Therefore, the use of rapid SSD is highly recommended for production scenarios (Paredaens J. & Kuijpers B., 1998).

Neo4j operate directly on the storage file and node does not contribute transactional security. So, it can be only used where it requires native write thread. As a result of data is written deliberately and never flushed to the consistent logs. Huge performance boosted is achieved. The batch inserter is enhancing for non transactional bulk of large amount of data.

Data size- In neo4j data size is mainly defined by the address space of fundamental keys of nodes relationship properties and relation types. The main address space as follows (Neo4jmanual, 2010).

Nodes	2^{35} (~ 34 billion)
Relationships	2^{35} (~ 34 billion)
Properties	2^{36} to 2^{38} depending on property types (maximum ~ 274 billion, always at least ~ 68 billion)
relationship types	2^{16} (~ 65 000)

Table 1: The main address space of neo4j

COMPRESSION ANALYSIS OF TRADITIONAL DATABASE AND GRAPH DATABASE QUERY LANGUAGE

It can be compare with SQL to analysis the efficiency of database system with the previous method. The wasteful of the traditional database system can be understood as the queries around connected data. In compression analysis, we compare query performance of data and relational database management system (SQL query).

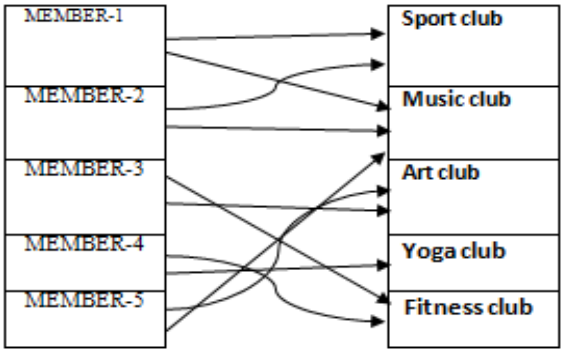


Figure 1: Relation between members and different club-society

We can take club-society as an example. Here, one member can be associated with many different clubs and similarly each club can associate different members. This figure shows the connection between different tables of database. The same data has been shown in graph database as figure 2.

Graph database as a RDBMS form

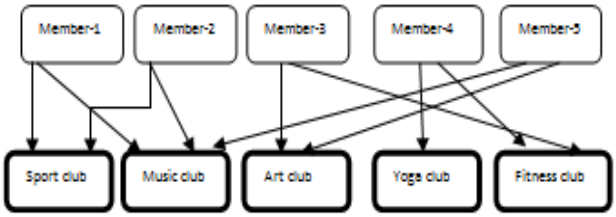
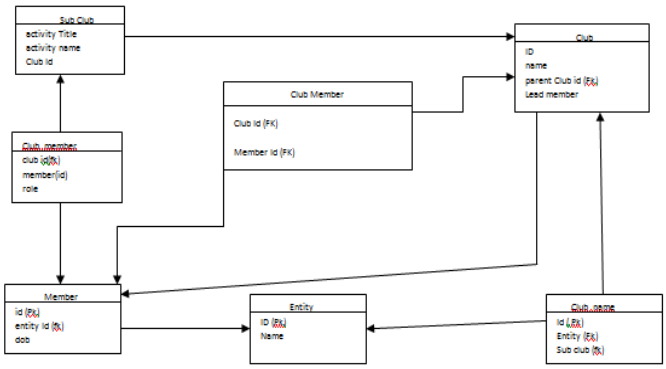


Figure2: Graph database as a RDBMS form

The graph form of database is shown in figure 2. Here, it can understand as graph database of the relational database management system.

Put the club society statement data into relational database and cypher query. Then find the optimum solution and calculate time and space for database system.

Figure 3: Club society data of relational database management system



In above figure 3, we can simplify club society data into relational database management system and find query time and performance of statement of database system.

A. SQL Statement

SELECT name FROM Member

LEFT JOIN Person_Department

ON Member.Id = Member_Club.MemberId

LEFT JOIN Club

ON Club.Id = Member_Club.ClubId

WHERE Club.name = "Sport Club"

B. Cypher query statement

MATCH (M:Member)<-[:cmember]-(c:club)
WHERE c.name = "Sport Club"
RETURN C.member

In above example, when we compare SQL methodology with cypher based query, the following is observed:

cypher query language takes half length of statement and structure in compared to SQL.

cypher query take half of time to execute the statement as compare to SQL statement.

cypher query reduces the chance of error as compare to SQL statement.

The Cypher query is half the length of the SQL statement and structure is simplified to be compared to SQL. Not only would this Cypher query is faster to create and run it, but it also reduce chances for error. Cypher query language also take less space and time rather than SQL query.

PERFORMANCE BASED EVALUATION

Time – cypher query comparatively take less time than structure query language. Cypher query execution time fast.

Space-cypher query take half length of statement comparatively structure query language. So graph data take less line of code and data takes less space to store in database.

Machine analysis setup:

Performance assessment was conducted on window 10, with 3GB primary memory and 2.60GHz core i3 processor. Neo4j (v2.1.5) and Oracle (community 11.g) has been taken as tools. The test cases were run 4 times. Comparison between traditional database and graph database is shown in table 2. Here, we found that execution time is less for Neo4j (Graph database).

Depth	RDMS execution time(S)	NEO4j Execution Times(S)	Records Returned
1	0.019	0.01	2500
2	15.67	0.189	11000
3	22.95	0.687	68000
4	56.89	1.031	91000

Table 2: Comparison between traditional RDBMS and Graph database

CONCLUSION

In this research paper, we analyze the basic of graph and look into database concepts for storage system of database system. RDBMS uses the tabular database forms to store information. New kind of storage system in the block form is provided in neo4j. We saw later is a superior in which data are more connected. Neo4j is graph storage type database and the cypher query language is traditional methods. Graph based structure uses nodes and edges for database storage system. Query and graph based language are used to create connection with developing large data in Internet. Relation database suffer execution degradation as huge number of node are added due to large number of entries in join table. Therefore, based on network nature of internet activities, graph data base designed can be used for fast access of complex data. So, it increases the storage capacity of database. In above example, when we compare SQL methodology with cypher based query. It is observed that cypher query language takes half of statement and structure in compared to SQL, cypher query take half of time to execute the statement in comparison to SQL statement and cypher query reduces the chance of error as compare to SQL statement.

REFERENCES

Angles Renzo, Gutierrez C.(2008). Survey of Graph Database Models. ACM Computing Surveys, Vol. 40, No. 1, Article 1.

Angles Renzo, Gutierrez Claudio (2008). Survey of Graph Database Models. ACM Computing Surveys, Vol. 40, No. 1, Article 1.

Angles Renzo (2013). Comparison of Current Graph Database Models. 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW), pp: 171-177.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., And Wiener, J.(2000). Graph structure in the Web. Proceedings of the 9th International World Wide Web conference on Computer Networks: The International Journal of Computer and Telecommunications Networking, North-Holland Publishing Co., p.p. 309–320.

Codd, E. F. (1980). Data models in database management. in the Proceedings of the 1980 Workshop on Data abstraction, Databases, and Conceptual Modeling. ACM Press, 112–114.

Chandra, A. K. (1988). Theory of database queries. in the Proceedings of the 7th Symposium on Principles of Database Systems (PODS). ACM Press, p.p. 1–9.

Ehrig H., Prange U., Taentzer G.(2004). Fundamental theory for typed attributed graph transformation. in Proc. of the 2nd Int. Conference on Graph Transformation (ICGT), ser. LNCS, no. 3256, pp. 162–177, Springer.

Jadhav Pradeep, Obero Ruhi (2015). Comparative Analysis of Graph Database Models using Classification and Clustering by using Weka Tool. International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 2,p.p. 438-445.

Neo4jmanual (2010). Internet: http://docs. neo4j.org/chunked/stabl e/graphdb- neo4jnodes. html

NOSQL Databases, http://nosql-database .org/

Paredaens J. and Kuijpers B.(1998). Data Models and Query Languages for Spatial Databases. Data & Knowledge Engineering (DKE), vol. 25, no. 1-2, pp. 29–53.

Short overview on the emerging world of graph databases, http://www.graph-database.org/overview.html.