Enhancing Performance of Database system using Graph Data Modeling

Deepak Kumar1, Prashant Gupta2, Daniel Okunbor3

1Fayetteville State University, Fayetteville, NC, USA

2Amity Institute of Information Technology, Amity University Uttar Pradesh, Noida, India

3Fayetteville State University, Fayetteville, NC, USA

1dkumar@uncfsu.edu, deepakgupta\_du@rediffmail.com

2prashantdewalya@gmail.com

3diokunbor@gmail.com

***Abstract: Modern day business operations depend on database storage system. Relational database management systems (RDBMS) have been the power-horse of software applications for data storage since the 80’s. Relational databases store highly structured data in tables with predetermined columns of certain types and rows which contains similar types of information as records. With billions of devices connected through wide area network (Internet) the complexity of data management systems has increased tremendously, making it seemingly difficult and time consuming in executing queries using RDBMS query languages like Structured Query Language (SQL). Storage capacity and retrieval operations of connected data can be improved using graph database. In this paper, we focus on the adoption of graph modeling for improving storage capacity of database management applications. The graph-centric model using Neo4j (v2.1.5) was numerically compared to traditional database Oracle (11.g) system and the results are presented in this paper.***

**keywords:** Database, Relational databases, Graph, Storage system, Structured data.

1. INTRODUCTION

The traditional relational databases have been found to exhibit some drawbacks despite their ubiquitous nature in business enterprise applications. These drawbacks pertain to their rigidity and performance costs. It has been proven time and again that changes to the database generally result in performance degradation. Query complexity tend to grow with joins operations or other operations involving adding new type of data and relation. As a consequence, schemas are redesigned at the expense of compute time and data volume increases and exponential cost of join query operations[1]. An alternative to relational databases is the graph data modeling approached based on Neo4j. Neo4j is available on the internet DBMS system and it comes with numerous function constructs including read, update and delete operations.

Graph database models are designed to adopt online transactional processing (OLTP) systems and are engineered with transactional operational availability [21]. Graph database models are based on cipher query language. Cypher is a graph database query language which are used to stored a data into a node and data are highly connected in cypher query language. Cypher is a language of graph that is declarative and expressive, storing and updating the graph data as needed without additional costs. Cypher query language yields improved efficiency of database system compared to relational database systems[14,12,11]. Some of the reasons advanced for Neo4j for enterprise applications are[1]:

* It provides perfect (Atomicity, Consistency, Isolation, Durability) ACID transaction
* It creates more expansive opportunities
* It has higher range of nodes and relationships
* It provides higher velocity for query through traversal
* It supports data analytics

Graph data models accomplish operations by customizing data occurrences within transactions to ensure logical data. A graph representation has the capability to extend its capacity and complexity without any degradation on performance. Its only limitation is system hardware[12].

A typical server can manage a graph of thousands of millions of nodes and their relationships. The pictorial representation (graph) of database storage is generally employed to reveal defective nodes or data. Query operations are fast since it possible to perform several million traversals steps per second [14]. In this paper, we focus on the adoption of graph modeling for improving storage capacity of database management applications. The graph-centric model using Neo4j (v2.1.5) was numerically compared to traditional database Oracle (11.g) system and the results are presented in this paper.

The paper is divided in following sections: Section 2 represents issues in relational database storage model. Section 3 explains capacity model of graph database, and Section 4 provides a conclusion of the paper.

2. ISSUES IN RELATIONAL DATABASE STORAGE MODEL

In recent years, alternative approaches to traditional relational databases have been investigated. Majorities of these techniques (Neo4j, NoSQL, MongoDB) are based on graph data models. Some of these graph models have applications in may platforms. Social networking sites (e.g., Facebook) store data using connected graph. The popular department store, Walmart, in the United States uses graph database. As described earlier, data in today’s world, are more connected and complex resulting in frequent redesigns of schema in typical relational database that may degrade performance significantly. In contrast, graph databases are designed for highly connected data, consisting of nodes and their relationships. The cypher query language is designed specifically for graph databases to facilitate fast responses, versatility and with high accuracy and performance[12,17].

3. CAPACITY MODEL OF GRAPH DATABASE

1. File Capacity- Graph technology is based on Java for all file handling and non blocking I/O file systems. It is able to handle huge data file whenever needed. It allows ACID velocity to be reduced easily since restriction is only on primary memory storage [12].
2. Read velocity**-** Graph database allows businesses to realize maximum value from available limited resources [11, 14]. Graph database does not chunk or latch any read operations. There is no danger for deadlocks in read transactions. Queries are done in parallel on distributed computing systems with many processors. That contribute great improve scenarios with larger server.There are two different kind of applications [17, 13]:

* Continuous sustained action
* Large volume access

To support the various requirements of these applications, Neo4j applies these two applications by writing to the storage layer. In transactional, ACID –compliment normal operation remoteness layers is maintained and read operation can appear at the same times as the writing process [5,11]. This requires disk write, access and actual flushing of data. The write speed of Neo4j on a individual server in continuous mode is bound by the I/O capacity of the hardware. Neo4j provides huge performance by allowing data to be written deliberately and never flushed to consistent logs. It has a batch inserter that is used enhance non transactional bulk import of large amount of data [12,18].

3.1 Advance database modeling based friend finding based on social network using graph query

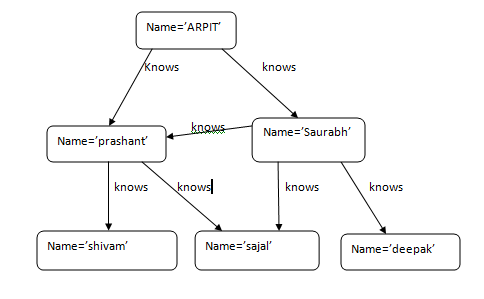


Fig 1.This figure shows graph based data where each node is connected to different node in a relationship manner. In this graph, friend of Arpit is find out.

**Create statement .**

create (\_0 {`name`:"Prashant"})

create (\_1 {`name`:"saurabh"})

create (\_2 {`name`:"shivam"})

create (\_3 {`name`:"sajal"})

create (\_4 {`name`:"deepak"})

create (\_5 {`name`:"arpit"})

create \_0-[:`know`]->\_3

create \_0-[:`know`]->\_2

create \_1-[:`know`]->\_4

create \_1-[:`know`]->\_3

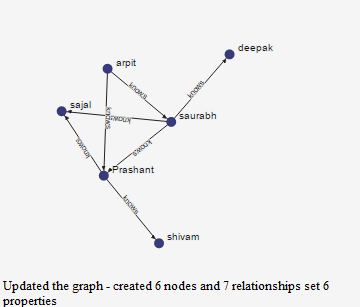
create \_1-[:`know`]->\_0

create \_5-[:`know`]->\_1

create \_5-[:`know`]->\_0

This Query took 32 ms and returned no rows. Updated the graph - created 6 nodes and 7 relationships set 6 properties

Fig 2.Graph based schema- This Query took 32 ms and returned no rows. Updated the graph - created 6 [12].



To search out the friends of Arpit friends that are not already his friends, the query looks like

Graph statement

MATCH (arpit { name: 'arpit' })-[:know\*2..2]-(f\_of\_f)

WHERE NOT (arpit)-[:know]-(f\_of\_f)

RETURN friend\_of\_friend.name, COUNT(\*)

ORDER BY COUNT(\*) DESC , friend\_of\_friend.name

Result of Query-here is result of query.

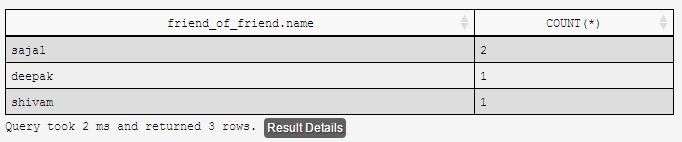
  
Fig 3.Arpit friend details. It took 2ms to find the result and stored 3 row of data.

Fig 4. Value count of friend of friend

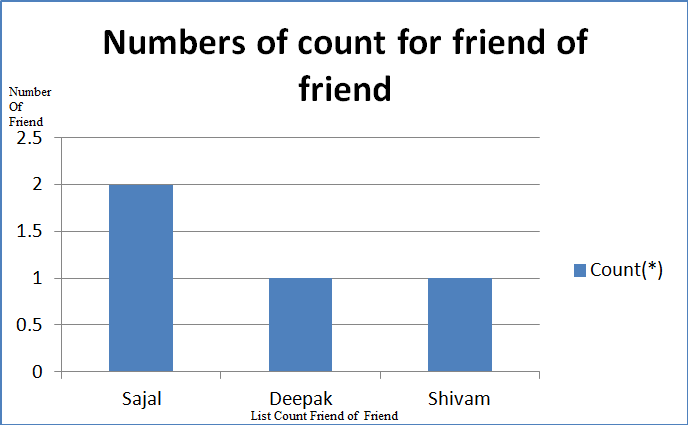


Fig 5.Details study of query-



3.2 COMPRESSION ANALYSIS OF TRADITIONAL DATABASE AND GRAPH DATABASE QUERY

This compression analysis compare query performance of dataand compare relational database management system SQL query.

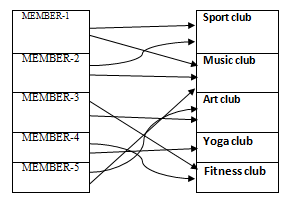


Table 1.

Compare query performance data

As an example, we can take club-society architecture. One member can be associated with many different clubs. Similarly, each club can have different members. So this figure shows the connection between different tables of database and the same data has been shown in graph database form.

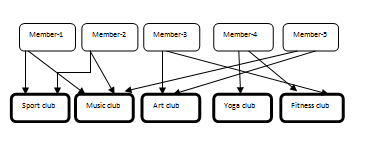
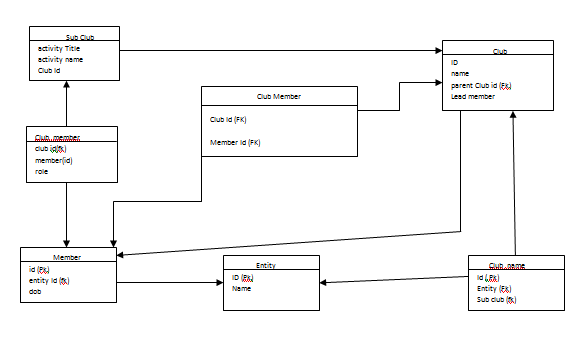


Fig 6.Graph database as a RDBMS form-

The graph database is relatively easy to understand compared to the relational database management system. We model the club society statement data using relational database and cypher query, determine optimum solution and calculate time and space.

 Fig 7. RDBMS query based schema

The corresponding SQL query statement block is:

SELECT name FROM Member

LEFT JOIN Member\_club

ON Member.Id = Member\_Club.MemberId

LEFT JOIN Club

ON Club.Id = Member\_Club.ClubId

WHERE Club.name = "Sport Club"

##### The corresponding cypher query statement block is:

MATCH (M:Member)<-[:cmember]-(c:club)

WHERE c.name = "Sport Club"

RETURN C.member

Design Model Differences: We observed the following differences between the two models above:

1. Cypher query has less statements than that of SQL query.
2. The execute time for Cypher query is half of that of SQL query[12].
3. Cypher query reduces the chance of error [7].

3.3. PERFORMANCE BASED EVALUATION

Following parameters are evaluated for capacity system.

1. Time. Cypher query comparatively take less time than structure query language. Cypher query execution time is fast.

2. 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 stored in database system.

3. Performance assessment was conducted on windows 10, with 3GB primary memory and 2.60GHz core i3 processor. Neo4j (v2.1.5) and Oracle (community 11.g). The test cases were run 4 times.

Following table shows the different parameter of performance for graph based database .

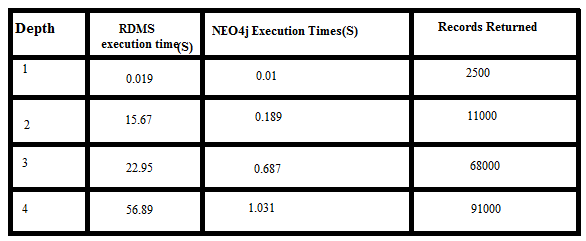


Table -2: Performance for different system

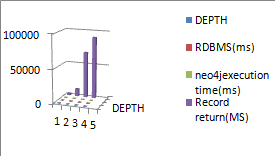
Table 2 shows the values of parameter of performance for different system.****

Fig -8. Show the parameter value in different database

**4.** CONCLUSION

In this paper, we consider and analyze the performance and scalability of relational database to graph database. Also, we compare them according to data model of database. As per result, we conclude that graph database improves the performance of database management system compared to traditional database. Database that use graph based structured with node edges and store data property can be defined as a graph database. Interpretation query and graph are used to create connection. For Internet applications, relational databases suffer execution degradation when large number of nodes are added; this is due largely to join operations. For this reason and because of the network nature of Internet activities, graph databases are generally considered appropriate. We have compared SQL methodology with cypher based query language. It is observed that cypher query language takes half of statements and structure compared to SQL. Cypher query take half of execution time compared to SQL statement. Cypher query reduces the chance of error compared to SQL.

References

[1] Angles Renzo, Gutierrez C., “Survey of Graph Database Models”, ACM Computing Surveys, Vol. 40, No. 1, Article 1, Publication date: February 2008.

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

[3] Angles Renzo, Gutierrez Claudio , “Survey of Graph Database Models” ACM Computing Surveys, Vol. 40, No. 1, Article 1, Publication date: February 2008.

[4] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., And Wiener, J., “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., 309–320,2000.

[5] Codd, E. F. “ Data models in database management” In Proceedings of the 1980 Workshop on Data abstraction, Databases, and Conceptual Modeling. ACM Press, 112–114,1980.

[6] Chandra, A. K., “Theory of database queries” Proceedings of the 7th Symposium on Principles of Database Systems (PODS). ACM Press, 1–9, 1988.

[7] Jadhav Pradeep, Obero Ruhi, “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, 2015.

[8]“NOSQL Databases”, http://nosql-database .org/

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

[10] Neo4jmanual, Internet: http://docs. neo4j.org/chunked/stabl e/graphdb- neo4jnodes. html ,2010

[11] Angles Renzo, “Comparison of Current Graph Database Models”, [2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)](https://www.computer.org/csdl/proceedings/icdew/2012/4748/00/index.html), pp: 171-177, 2013.

[12] Paredaens J. and Kuijpers B., “Data Models and Query Languages for Spatial Databases,” Data & Knowledge Engineering (DKE), vol. 25, no. 1-2, pp. 29–53, 1998. “Neo4j,” http://neo4j.org/.

[13] Williams David W., Huan Jun, Wang Wei “Graph Database Indexing Using Structured Graph Decomposition”, 2013 IEEE 29th International Conference on Data Engineering (ICDE), http://doi.ieeecomputersociety.org/10.1109/ICDE.2007.368956

[14] Johansson O., “Graph Decomposition Using Node Labels”, Doctoral Dissertation, Royal Institute of Technology, Stockholm, 2001.

[15] Rajbhandari P., Shah R. C., Agarwal S., “Graph Database Model for Querying, Searching and Updating”, 2012 International Conference on Software and Computer Applications (ICSCA 2012) IPCSIT vol. 41, IACSIT Press, Singapore, 2012.

[16] Mannino M.V. Shapiro, “L.D.: Extensions to Query Languages for Graph Traversal Problems,” IEEE TKDE 2, pp. 353–363, 1990.

[17] Martinez-Bazan, N., Gomez-Villamor, S., and Escale-Claveras, F., “DEX: A high-performance graph database management system,” Data Engineering Workshops (ICDEW), 2011 IEEE 27th International Conference on April 2011, pp. 124 – 127, 2011.

[18] J. Huan, D. Bandyopadhyay, W. Wang, and J. Snoeyink,“Comparing graph representation of protein structure for mining family-specific residue-based packing motifs,” Journal of Computational Biology (JCB), 12(6): pp. 657– 671, 2005.