International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Improving Performance using Relational and Graph Database

Harsha R. Vyawahare, Pravin P. Karde, Vilas M. Thakare

Abstract**:** ***Paper Relational database model (also called SQL databases) are one of the prevalent databases that are used with structured data. Currently news demands are arising owing to the magnitude with which the internet and social networks are getting used which brought importance to graph-structured data. Graph database (a nosql database) deal more naturally with highly connected data and are thus becoming popular and efficient choice. Due to limitations faced by relational databases in handling relationships (highly connected data), enterprise information systems find graph database as a promising alternative. According to the form of queries and property of data both relational and graph databases have vitality and flaws. Since most of the data is available in relational schema in this context, the conversion of an application from a relational to a graph format is very beneficial. Thus, this paper develops a dual database system through migration, which unifies the strengths of both relational databases and graph databases. Experimental results have shown that, this hybrid system has efficient performance.***

Index Terms**:** ***Graph database, Nosql database, Relational database, SQL databases.***

I. INTRODUCTION

Relational database is perhaps the most prominent approach for recording data. Taking into consideration, the growth in the use of social network and internet that has resulted into the exponential increase in data volume. These circumstances have made data more relationship rich, more connected and unstructured which again resulted into unsuitability of relational databases for storing it. The ultimate consequence is need of an alternative database. The emerging nosql databases have come up with many options with its characteristics, each specializing in an area where relational database failed. Graphs are known to be data representational model in many application domains such as computer communication networks, protein cell interaction, fraud detection social networks and the semantic web. All these applications require interacting with highly-connected data which ultimately led to the origination of a brand new category of storage systems under the umbrella of nosql databases, usually called graph database management systems (GDBMS). The key distinction between graph and relational databases is in terms of how they store relationships. Graph databases identify them in terms of connected edges between nodes whereas relational databases identify them through referential integrity constraints. The second key distinction between both databases is in terms of how they search for related records. Relational database searches the full required table which at times can be inefficient in the case where the table is large. Here graph databases turns out to be more beneficial as they range easily to large data sets without needing uneconomical join operations. Another advantage that graph database is having is a flexible schema where schema alterations can be done easily at any stage. The data form mainly available today is unstructured. Also every database is specializing in certain features which help them to deal with a particular kind of data. The term polyglot persistence is frequently coming into scene these days due to fact that many applications required multiple data storage. Thus in order to suffice to customer needs, companies is building applications that use multiple database systems. Here every database is chosen in such a way that one database covers strengths and the limitations are covered by another. This paper focuses on developing the an dual database system by using relational database (MySQL) and graph database (Neo4j) to work in unison using migration. As graph databases are meant for modelling relationships, a partial migration methodology is used. Also comparative analysis of migrated graph database using join sensitive and non-join sensitive queries is done. The remaining of the paper is organized as follows. Section 2 presents the basic details about both the models; Sect. 3 states some of the earlier work done in the area which serves as basic foundation. Sect. 4 discusses the actual concept and the various strategies available. Sect. 5 presents the proposed approach used and the experimental results are discussed in sect. 6; and Sect. 7 contains concluding remarks.

II. RELATIONAL AND GRAPH DATABASE ELEMENTS

This section briefly expresses the introductory details of relational and graph database which are used in the proposed approach.

Revised Manuscript Received on 30 July 2019.

***** Correspondence Author

Harsha R. Vyawahare*, PG Department of Computer Science, SGBAU, Amravati, Amravati, India.

Dr Pravin P. Karde, Department of Information Technology, Government Polytechnic, Amravati, India.

Dr Vilas M. Thakare, PG Department of Computer Science, SGBAU, Amravati, Amravati, India.

A. *Graph Database Model*

Graph databases are meant to deal with relationship rich data. Relationships are most important here and the basic reason to use it as an alternative to relational db is in doubtfully due to the fact that graph databases deal more naturally with relationships. Also most of the output of graph databases is obtained using these relationships. The graph model used in proposed approach is Neo4j which is a property graph model. It is a labelled, property graph database developed by Neo technology. The basic data entity in graph databases is nodes and edges represent association between nodes thorough relationships. An edge is comprised of a start node, end node, type, and direction. Figure 1 shows graph data model.

B. *Relational Database Model*

Relational Database exists from 1970s, which organizes data into tables i.e. it stores data in 2-dimensional tables. A table is a two-dimensional structure made up of rows (tuple, records) and columns (attributes, fields). SQL is use to retrieve data from relational database. It does not support scalability, also not well to handle huge and more connected data as it requires too many join operations. Example: RDBMS (open source MySql). In relational database, relational schema is denoted as Ri(Ai), where Ri is ith relation and Ai is set of attributes of ith relation. Attributes are properties of entities also called as columns. The Relational Database R is set of tuples over all the relational schema R1(A1)….Rn(An). Relation is also referred to as a table. Every tuple in the table is uniquely identified with the help of key called primary key. Another table in the database can refer to this primary key through the use of foreign keys [13]. Figure 2 shows relational data model.

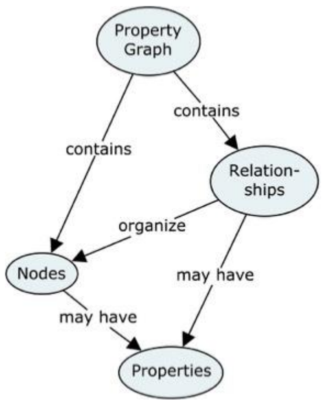

Fig. 1. The Graph Model

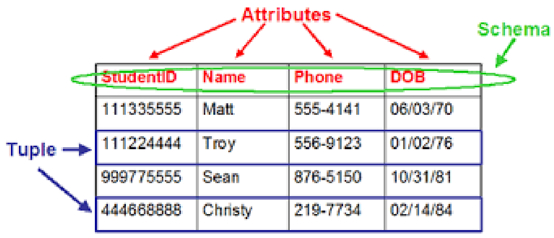

Fig. 2. The Relational Model

III. GROUNDING RESEARCH

As per the literature review much of the work is done about comparing relational data models and graph data models. For instance, [8][9] have evaluated Mysql and Neo4j based on subjective as well as objective parameters and experimental comparisons shows Neo4j to perform better. There has been work done towards handling vast volume of data in RDBMS by integrating it and working in unison with GDBMS. Authors in [4] have worked on devising a middleware layer between the PostgreSQL and Neo4J which optimizes the response times. Authors in [9] created a generalization between SQL and NoSQL databases. Requests are reviewed in order to determine which model would be perfect for processing it. An approach by authors in [10] manages graphs as the first class citizens inside the relational database structure calling this phenomenon as GRFusion. Many algorithms proposed for data migration and schema-mapping between Relational to non-Relational and between non-Relational to non-Relational. While migrating from Relational to non-Relational the biggest challenge is to transfer schema information to schema-less databases. Also majority of the data available on semantic web is stored in RDF format. As most of the current web data is stored in RDB. Thus to make data available on semantic web, there is the need to bridge gap between RDB and RDF. Recent research has shown that semantic web technologies are better for web, particularly in the case where data has to be integrated and exchanged among different sources. So, many organizations needed to convert their data from relational to RDF to make it available on semantic web. This increased the need to convert relational data into graph modelled data [17]. The work nearest to the proposed approach regarding transforming relational database to graph database has also been undertaken by many authors. Soussi [11] suggests that relational database data could be transformed to other model like ER, RDF or XML which can later be used to create a graph in a more simple way. The work R2G by [12][15] is about automatic conversion of entire relational database to a graph database along with its conjunctive queries using schema graph, schema paths and data mapping. Authors in [13] used web based ETL approach that uses integrity constraints defined over source relational database to transform into target graph database without any data loss. An approach called FD2G that leverages the existence of functional dependencies information inside the input relational database to automatically perform the conversion to property graph databases has been proposed by [14]. The proposed work uses partial migration method in order to leverage the graph databases fully.

IV. DATA MIGRATION CONCEPT & DEPLOYMENT PARADIGMS

Data migration is defined as the process of transferring data from one database, called source, to other database called target. Some typical steps performed during the process are:

A. Connect to the source database

B. Connect to the destination database

C. Read the data from the source

D. Transform the data to the destination format/model

E. Perform the insertion of data into the destination

The foremost thing to be decided during any migration is its strategy. There are three important paradigms related to relational to graph migration. These paradigms decide which source data to be moved to target. Figure 3 shows the required strategies.

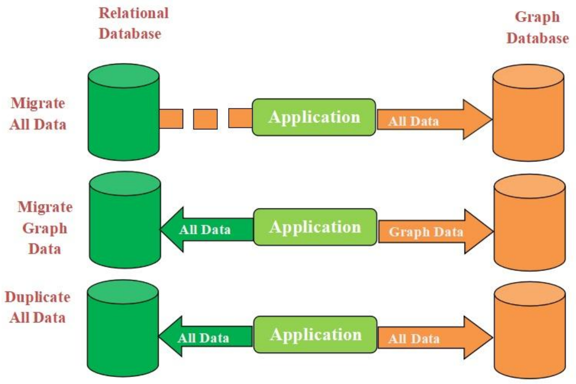

Fig. 3. Paradigms for deploying relational to graph database

Case 1: This paradigm is appropriate when the relational model does not suffice the application needs and it does not model the data properly. Thus the development teams decide to abandon their relational database altogether and migrate all of their data into a graph database. This is a one-time, bulk migration and all relational data is moved to graph database.

Case 2: This paradigm is appropriate when the relational database is showing poor performance especially in data retrieval. Thus the development teams decide to continue to use their relational database for any use case that relies on non-graph, tabular data and, for any use cases that involve a lot of JOINs or data relationships use graph database. This is a partial migration scenario where relationships are fetched and moved to graph database.

Case 3: This paradigm is appropriate when the both databases with complete data are needed. Thus the development teams decide to duplicate all of their data into both a relational database and a graph database. Here data can be queried in whatever form is the most optimal for the queries to run. The second and third paradigms are considered as polyglot persistence, since these approaches use a data store according to its strengths. While this introduces additional complexity into an application's architecture, it often results in getting the most optimized results from the best database for the query. Actually none of these is the correct paradigm for deploying an RDBMS and a graph it depends on application goals, frequent use cases and most common queries. The deployment strategy paper focuses on is case 2. The literature review does not focus on partial migration strategy. Paper uses this strategy in order to leverage the strengths and weakness of both databases in order to develop efficient hybrid system.

V. THE PROPOSED APPROACH

An approach that supports dual databases is proposed by migrating relationships from source Mysql database to Neo4j graph database. The migration is performed using following steps like:

A. Meta Data Retrieving

This step retrieves essential meta data information from the source. The required meta data includes list of primary keys and foreign keys along with referenced and referencing tables.

B. Generation of Mapping File

This step decides how data is modeled in Neo4j i.e. what are nodes and which relationships are.

C. Extraction of Data to CSV

As the data can be migrated through csv format, the selected data is converted into csv files.

D. Generation of Script File

This step generates necessary commands needed to carry out migration.

E. Neo4j Import Process

This step performs the final import process and the data is loaded into neo4j.

The data set used to experiment is the Northwind dataset, a commonly-used SQL dataset which is grapy enough. The Northwind database is a sample database used by Microsoft to demonstrate the features of some of its products. The database contains the sales data for Northwind traders, a fictitious specialty foods export import company. The ER diagram for the sample database is as shown in Figure 4. The dataset has 13 relations and approximately 3308 tuples. The queries used to experiment are as depicted in table 1.

Table 1 Queries Used

Sr. No	Queries
Q1	What are Steven Buchanan's top 5 selling products?
Q2	What is the reporting structure two levels deep?
Q3	Which are 10 high price products?
Q4	Which products names starting with C with prices greater than 100?
Q5	Which employees ordered Chocolade?

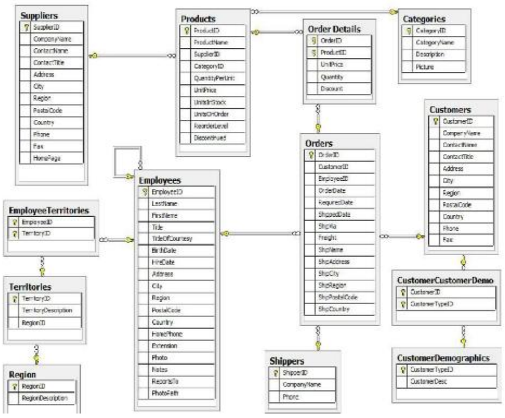

Fig. 4. ER Diagram for Northwind Dataset

VI. RESULTS

For illustration of proposed method, a comparison is presented between the Neo4j graph data model generated by full migration method with Mysql relational database is presented. The queries depicted in above table have been utilized for the same purpose. Following results are obtained using the proposed method as shown in Table 2 and Figure 5.

Table 2 Response Time In Ms For Northwind Dataset

Sr.No	Query Type	On Mysql	On Neo4j Complete
Q1	Join Sensitive	8.72	11.4
Q2	Join Sensitive	7.54	5.4
Q3	Non Join Sensitive	2.6	3.6
Q4	Non Join Sensitive	4.2	5.2
Q5	Join Sensitive	6.42	4.0

Fig. 5. Response Times Obtained

VII. CONCLUSION

Graph database is seen as a promising alternative for relational database. This paper has presented a concept to use both databases together in order to get benefits of both. The response time results of complete migration and the source relational database executed on selected queries is shown. Results show that complete migration does not serve the purpose. The future scope includes the implementation of said approach for partial migration.

REFERENCES

1. H. R. Vyawahare, Dr P. P. Karde, Dr.V.M.Thakare, "Survey on Comparison of Graph Database Model",In 2nd International Springer Conference on Data Management, Analytics & Innovation (ICDMAI-2018) at IICMR, Pune, Dt 19-21 Jan 2018.
2. R. D. Virgilio, A. Maccioni and R. Torlone, "R2G: a Tool for Migrating Relations to Graphs", In Proceeding of the 17th International Conference on Extending Database Technology (EDBT 2014), pp. 640-643, 10.5441/002/edbt.2014.63, March 2014.
3. N. Spyratos and T. Sugibuchi, "Proper-a graph data model based on property graphs", International Workshop on Information Search, Integration, and Personalization, (ISIP 2015), Communications in Computer and Information Science, vol 622, pp. 23-45, Springer, Champp. , 2015.
4. R. Ettrup and L. Nielsen, "Bridge-DB- Query Optimization in a Multi-Database System", Master's Thesis, Software Mater, Aalborg University, June 2015.
5. R. Angles, C. Gutierrez, "Survey of graph database models", ACM Computing Surveys (CSUR), vol. 40, no. 1, article no 1, February 2008, doi:10.1145/1322432.1322433.
6. O. Orel, S. Zakosek and Prof. M. Baranovic, "Property Oriented Relational-To-Graph Database Conversion," Automatika, vol. 57, no. 3, pp. 836– 845, 2016.
7. S. Batra and T. Charu, "Comparative analysis of relational and graph databases", International Journal of Soft Computing and Engineering issn: 2231-2307, Vol. 2, No.2, pp 509-512, May 2012.
8. C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen and D. Wilkins, "A comparison of a graph database and a relational database: a data provenance perspective" , In Proceedings of the 48th Annual Southeast Regional Conference ACM SE,10, pp.42:1-42:6, April 2010, doi:10.1145/1900008.1900067.
9. J. Roijackers, "Bridging sql and nosql", Master's thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, pp.25-32, May 2012.
10. M.S. Hassan, T. Kuznetsova, H. C. Jeong and M. Sadoghi, "Extending In-Memory Relational Database Engines with Native Graph Support", In Proceedings of the 21st International Conference on Extending Database Technology, (EDBT), March 2018, ISBN 978-3-89318-078-3 on OpenProceedings.org ISSN: 2367-2005, pp 25-36
11. R. Soussi, "Querying and extracting heterogeneous graphs from structured data and unstrutured content", PhD thesis, Ecole Centrale Paris, 2012.
12. R. D. Virgilio, A. Maccioni and R. Torlone, "Converting Relational to Graph Databases", Proc. ACM First International Workshop on Graph Data Management Experience and Systems, Article No.: 1, pp. 1-6, June 2013 doi:http://doi.acm.org/10.1145/2484425.2484426.
13. S. D. Chaure, M. U. Kulkarni and P. M. Jadhav, "Web based ETL Approach to Transform Relational Database to Graph Database", International Journal of Computer Sciences and Engineering, Volume-3, Issue-7, pp 92-97, July 2015, E-ISSN: 2347-2693.
14. Y. A. Megid, N. El-Tazi and A. Fahmy, "Using Functional Dependencies in Conversion of Relational Databases to Graph Databases", International Conference on Database and Expert Systems Applications DEXA, Lecture Notes in Computer Science, vol 11030, pp. 350-357, Springer, Cham, August 2018.
15. M. Singh and K. Kaur, "SQL2Neo : Moving Health-care Data From Relational To Graph Databases", In IEEE International Advance Computing Conference (IACC), pp. 721-725, June 2015, doi: 10.1109/IADCC.2015.7154801.
16. S. Bordoloi and B. Kalita, " Designing Graph Database Models from Existing Relational Databases", International Journal of Computer Applications (0975 – 8887) Volume 74– No.1, pp. 25-31, July 2013.
17. Mehak Gupta, "A Novel Approach to Transform Relational Database into Graph Database using Neo4j", Master's thesis, Thapar University, June 2014.
18. H. R.Vyawahare, Dr P. P. Karde "A Short Survey on Graphical Database", Indian Journal of Computer Science and Engineering (IJCSE) ISSN : 0976-5166, Vol 6, Issue 4, Page No 142-146, Aug-Sept 2015.

19. H. R.Vyawahare, Dr P. P. Karde, Dr V. M. Thakare, "Graph Database:The Next Frontier in managing relationships", Journal of Applied Science and Computations",Issn no:1076-5131, Vol.5,Issue 10 , pp562-569,Octomber 2018.
20. H. R.Vyawahare, Dr P. P. Karde, Dr V. M. Thakare, " Survey On Comparison of Graph Database Model", In International Journal of Research and Analytical Reviews, Volume 6, Issue 1, pp 1600-1608, February 2019.

AUTHORS PROFILE

H.R.Vyaawahare pursed Bachelor of Engineering and Master of Engineering from Sant Gadge Baba Amravati University. She is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Computer Science and Engineering in Sipna College of Engineering and Technology, Amravati since 2000.
She is a life member of ISTE, IETE, IE and CSI. She has published more than 15 research papers in reputed international journals including conferences. She has 19 years of teaching experience.

Dr P. P. Karde is Professor and currently heading the department of information technology in government polytechnic college Amravati. He has pursed Ph.D from Sant Gadge Baba Amravati University, Amravati. He is member of many professional bodies including ISTE, IEEE computer society since 2013, a life member of the ISROSET since 2013 and ACM since 2011. He has published many research papers in reputed international journals and conferences including IEEE and it's also available online. His main research work focuses on Selection & Maintenance of Materialized View. He has 23 years of teaching experience.

Dr V. M. Thakare is Professor and Head of PG department of computer science in Sant Gadge Baba Amravati University, Amravati. He has pursued Ph.D from Dr B.A.M University, Aurangabad. He is one of senior most professor of Computer Science among all the 11 Government Universities in Maharashtra. He has guided many Ph.D scholars. He acts as expert and member in many bodies and advisory committees. He is also member of many professional bodies. He has delivered many keynote addresses and invited talks throughout India on variety of subjects related to computer science and engineering. Also he has organized number of International and National Conferences, Workshops and Seminars. He has published many research papers in reputed international journals including and conferences including IEEE and it's also available online. He has experience of about 30 years in teaching field. His interest of research is in Computer Architecture, Artificial Intelligence and Robotics, Database and Data warehousing & mining.