International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

An Efficient Graph Database Model

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

Abstract*: Relational databases are holding the maximum amount of data underpinning the web. They show excellent record of convenience and efficiency in repository, optimized query execution, scalability, security and accuracy. Recently graph databases are seen as an good replacement for relational database. When compared to the relational data model, graph data model is more vivid, strong and data expressed in it models relationships among data properly. An important requirement is to increase the vast quantities of data stored in RDB into web. In this situation, migration from relational to graph format is very advantageous. Both databases have advantages and limitations depending on the form of queries. Thus, this paper converts relational to graph database by utilizing the schema in order to develop a dual database system through migration, which merges the capability of both relational db and graph db. The experimental results are provided to demonstrate the practicability of the method and query response time over the target database. The proposed concept is proved by implementing it on MySQL and Neo4j.

Index Terms*: Graph Database, Migration, Neo4j, Relational Database.

I. INTRODUCTION

Recent years graph databases have drawn a mass attention. The reason behind their glory is principally due to fact that data in current applications can be obviously modeled as graphs. Example is social media applications which are resulting in exponential growth of interconnected data which can be better modeled as graphs. A graph is defined as a group of nodes and edges, where nodes correspond to entities and edges stand for relationships between nodes. The most popular variants of graphs models today are the property graph, the resource description framework (RDF) and the hypergraph. The data is naturally saved in graphs in graph databases. Also querying is accomplished using traversals [1]. Graph databases are many times compared with relational databases as both are meant for storing relationships. Literature survey states that interconnected data can be better modeled as graphs. However relational databases cannot be avoided and put aside due to its justifiable history of matureness, security, accuracy, scalability, ease and convenience in storage and optimized query execution. The term polyglot persistence is frequently coming into scene these days. The concept behind is using more than one database depending upon the manner whether the data is accessed by complete application or modules of individual application. Due to these circumstance and market demands, many companies are employing the polyglot persistence concept. Here databases are selected so that shortcoming of one database is overcome by another in order to achieve efficiency [2]. MySql (open source Relational Database) and Neo4j (open source Graph Database) is used for evaluating the outcomes of the research. The ultimate objective of this paper is to consider strengths of Mysql (relational db) and Neo4j (graph db) and use them together to work in unison using migration. Partial migration concept is used in order to leverage the benefits of neo4j. Also comparative analysis on response times between mysql, fully and partially migrated neo4j is done to show the feasibility of said approach. Finally visualization of migrated neo4j is also shown. The rest of this paper is organized as follows: Section 2 presents the available deployment strategies. Section 3 shows the proposed methodology. Section 4 states the implementation environment. Section 5 evaluates the performance of the proposed approach. Finally, Section 6 concludes the paper.

II. DEPLOYMENT PARADIGMS

There are three important paradigms related to relational to graph migration. These paradigms decide which source data is to be moved to target. The figure 1 shows the required strategies.

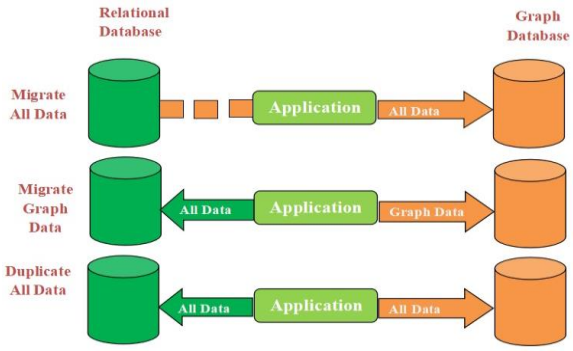

Fig. 1. Paradigms for deploying relational to graph database

Case 1 is full migration situation where the relational database is altogether discarded. In case 2 is partial migration situation where the graph data is only migrated. Where as in case 3 all data is duplicated on both dbs. The second and third paradigms are considered as polyglot persistence since it often results in getting the most optimized results from the best database for the query. Paper focuses on case 2 strategy also literature review does not focus on partial migration strategy. Paper uses this strategy in order to leverage the strengths of both databases in order to develop efficient system.

Manuscript published on 30 August 2019.
*Correspondence Author(s)
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.

III. METHODOLOGY: THE DATA MIGRATION MODULE

A methodology is proposed to build hybrid database by partly migrating relational to graph database which utilizes the schema and the constraints of the source. Rows are mapped to nodes and foreign keys are mapped into edges. The architecture of proposed system is as depicted in figure 2. The SQL queries executing on relational db can be executed on graph db using CQL.

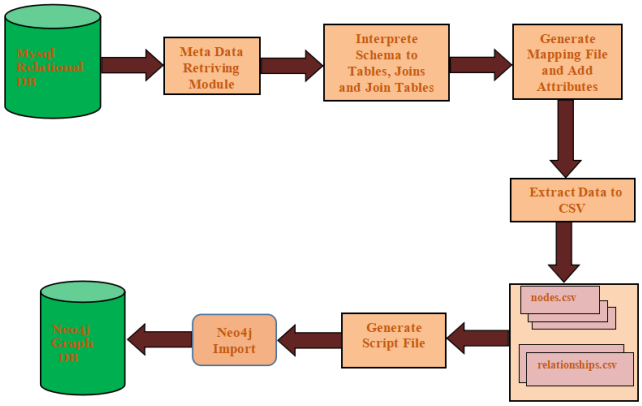

Fig. 2. Proposed Architecture

A. Meta Data Retrieving Module

Deals with retrieving the meta data information. The meta data about the foreign keys and referenced tables can be retrieved using an sql command as shown in Figure 3.

```
select table_name, column_name, constraint_name,
referenced_table_name, referenced_column_name
from information_schema.key_column_usage
where referenced_table_schema = 'input db';
```

Fig. 3. Meta Data Retrieving Command

B. Generation of Mapping File

Once the meta data information about the source relation database is known, the mapping file can be generated. Mapping file makes most important decisions about migration. The mapping is done as per the procedure given in next part.

C. Identification and Mapping of Tables, Joins and Join Tables

- Once Primary keys are stored in nodes with respective table properties in neo4j.
- Relationships (foreign keys) are principally stored as relationships.
- Representation of the constraints as Joins or Join Tables
  - i. Joins (foreign keys) are converted as relationships (edges).
  - ii. Tables that have two foreign keys are converted as relationships (edges).
  - iii. Tables that have more than two foreign keys get converted to nodes and the foreign keys to the other tables are converted as relationships.

As the migration needed is not a full migration. User interface is provided for selecting few attributes along with the attributes obtained through mapping rules. This stage decides how the final graph would look like.

D. Extraction of Data To CSV

Since the graph schema is decided in previous stage selected data need to be extracted from mysql. An obvious way to execute this is to first export the tables in CSV format. Following Figure 4 is an example of export_csv.sql.

```
COPY (SELECT * FROM customers) TO
 '/tmp/customers.csv' WITH CSV header;
COPY (SELECT * FROM suppliers) TO
 '/tmp/suppliers.csv' WITH CSV header;
COPY (SELECT * FROM products) TO
'/tmp/products.csv' WITH CSV header;
COPY (SELECT * FROM employees) TO
 '/tmp/employees.csv' WITH CSV header;
COPY (SELECT * FROM categories) TO
'/tmp/categories.csv' WITH CSV header;
COPY (SELECT * FROM orders LEFT OUTER JOIN
order_details ON order_details.OrderID = orders.OrderID)
TO '/tmp/orders.csv' WITH CSV header;
```

Fig. 4. Export CSV Command

E. Generation of Script File

In order to support dynamic conversion, the scripting file which contains the commands needs not to be static. Thus a script file is generated at run time. An example script file generated is as in figure 5.

```
// Create customers
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:customers.csv" AS row
CREATE (:Customer {companyName: row.CompanyName, customerID:
 row.CustomerID, fax: row.Fax, phone: row.Phone});
// Create products
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:products.csv" AS row
CREATE (:Product {productName: row.ProductName, productID:
row.ProductID, unitPrice: toFloat(row.UnitPrice)});
CREATE INDEX ON :Product(productName);
CREATE INDEX ON :Customer(customerName);
USING PERIODIC COMMITLOAD CSV WITH HEADERS FROM "file:orders.csv"
AS rowMATCH (order:Order {orderID: row.OrderID})MATCH
(product:Product {productID: row.ProductID})
MERGE (order)-[pu:PRODUCT]->(product)ON
CREATE SET pu.unitPrice = toFloat(row.UnitPrice), pu.quantity = toFloat(row.Quantity);
```

Fig. 5. Script File

F. Import Process

The script file contains the necessary commands to fetch the data into neo4j. The script file is executed o complete the import process.

IV. IMPLEMENTATION ENVIRONMENT

The Graph databases are ideal use case for application domains like networking, social networking, fraud analysis, semantic web, recommendation system, blogs, social networking, etc. This paper implements the partial migration of relational specifically mysql northwind. Databases and technologies used for implementation are stated ahead.

Neo4j is a popular open source variant of graph database by Neo Technology. It is robust, scalable and written in java also suited for holding and managing of relationship rich and networked data. It serves as efficient alternative for relational databases which are showing join pain. For implementation Neo4j- community-2.1.6 version is used is acts as storage medium for migrated mysql data. The default query language supported by neo4j is cypher which can be used conveniently. A browser based user interface is also provided by neo4j. Mysql is also a popular and open source varinat of relational database management system. SQL is the default query language supported which can be very conveniently used.

Retrieval Number: I7805078919/19©BEIESP
DOI: 10.35940/ijitee.I7805.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

It runs on many platforms including Microsoft Windows, Linux, etc. Implementation is done using MySQL server 5.6. MySQL phpmyadim has been used as third party user interface is used. The system configuration used for implementation is as follows.

- Microsoft Windows 8.1
- Inter(R) Celeron(R) CPU N2940 @ 1.83GHz 1.83 GHz
- 4 GB RAM
- 64-bit OS
- x64-based processor

The data set used in this experiment is the Northwind dataset, a commonly-used SQL datasets which are graphic enough. The details are stated in Table 1.

TABLE I. DETAILS OF DATA SET USED

Data Set	Size	Relations	Tuples
Northwind	1.01 MB	13	3308

V. RESULTS

The usefulness and efficiency of proposed method is demonstrated by comparing the neo4j graph data model generated by partial migration with mysql relational db and neo4j graph model (having full data). Following queries have been utilized for the same purpose as shown in Table 2.

TABLE II. QUERIES USED

Sr. No	Queries
	Northwind Data Set
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?

The response times obtained using the proposed method are as shown in table 3 and the analysis is as shown in Figure 6

TABLE III. RESPONSE TIME FOR NORTHWIND DATA SET

Sr.No	Query Type	On MySQL (ms)	On Neo4j Partial (ms)	On Neo4j Complete (ms)
Q1	Join Sensitive	8.72	7.8	11.4
Q2	Join Sensitive	7.54	4.2	5.4
Q3	Non Join Sensitive	2.6	4.2	3.6
Q4	Non Join Sensitive	4.2	4.2	5.2
Q5	Join Sensitive	6.42	2.4	4

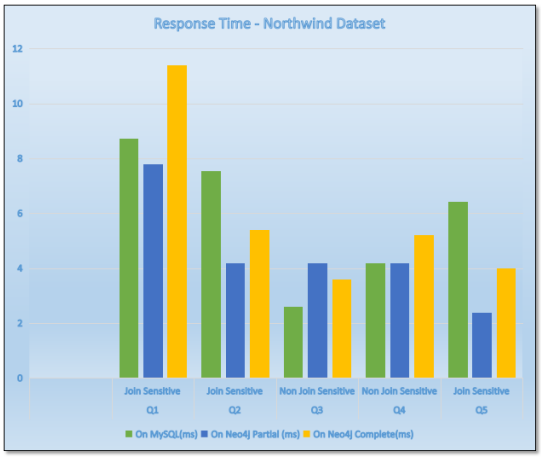

Fig. 6. Result analysis of Northwind and Data Set

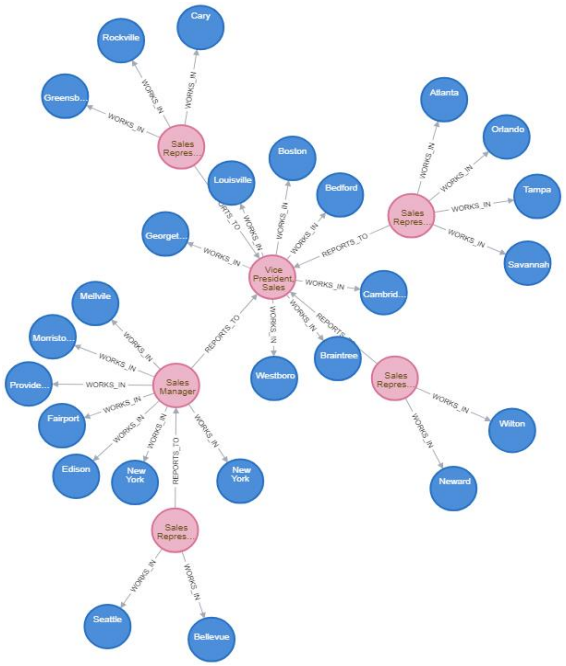

Fig. 7. Visualizations of part of graph db generated

VI. CONCLUSION

Graph databases are basically meant for managing relationships has been proved by the results also. Non join sensitive queries are showing better performance on the traditional tabular database as compared to the graph database. At the same time join sensitive are showing better performance on the graph database storing relationships. So it will not be incorrect to state that both dbs. can be used together to achieve efficient performance.

REFERENCES

1. 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.
2. R. Ettrup and L. Nielsen, "Bridge-DB- Query Optimization in a Multi-Database System", Master's Thesis, Software Mater, Aalborg University, June 2015.

3. 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.
4. M. Gupta, “A Novel Approach to Transform Relational Database into Graph Database using Neo4j”, Master’s thesis, Thapar University, June 2014
5. L. Rocha, F. Vale, E. Cirilo, D. Barbosa and F. Mourao, “A framework for migrating relational datasets to NoSQL” Procedia Computer Science, vol 51, pp. 2593-2602, 2015.
6. M. Singh and K. Kaur, “SQL2Neo: Moving health-care data from relational to graph databases”, In EEE International Advance Computing Conference (IACC), pp. 721-725, doi: 10.1109/IADCC.2015.7154801, 2015.
7. O .Orel, S. Zakosek and Prof. M. Baranovic, “Property Oriented Relational-To-Graph Database Conversion,” Automatika, vol. 57, no. 3, pp. 836– 845, 2016.
8. M. Hunger, R. Boyd and W. Lyon, “The Definitive Guide to Graph Databases for the RDBMS Developer”, Ebook Neo4j, 2016.
9. D. Gaspar, M. Mabic and T. Krtalic, “Integrating Two Worlds: Relational and NoSQL”, In Proceedings of the Central European Conference on Information and Intelligent System, pp. 11-18, Sept. 2017.
10. Y.Unal and H. Oguztuzun, “Migration of data from relational database to graph database”,Proceedings of the 8th International Conference on Information Systems and Technologies, Article No. 6, ACM, March 2018.
11. J. Pokorny, “Integration of Relational and NoSQL Databases”, In: Ngoc Thanh Ngu-yen, et al (eds), ACIIDS (2), LNCS, vol. 10752, pp. 35-45. Springer 2018 https://doi.org/10.1007/978-3-319-75420-8_4.
12. 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.
13. L. Liu and M. T. Ozsu, “Encyclopedia of Database Systems”, Springer Nature 2018 https://doi.org/10.1007/978-1-4614-8265-9.
14. Y. Khan, A. Zimmermann, A. Jha, V. Gadepally, M. D’Aquin and R. Sahay, “One Size Does Not Fit All: Querying Web Polystores”, In IEEE Access, vol. 7, pp. 9598-9617, 2019.
15. M. Hunger, “From Relational to Graph: A Developer’S Guide”, Neo4j White Paper, Dzone Refcardz
16. H. R.Vyawahare, Dr P. P. Karde “An Overview on Gaph Database Model ”, International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE) ISSN(Online): 2320-9801, Vol 3, Issue 8, Page No 7455-7457, August 2015 DOI: 10.15680/IJIRCCE.2015. 0308071 .
17. 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.
18. 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.
19. 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.
20. H.R. Vyawahare, P.P. Karde, V.M. Thakare , “Graph Database: An Alternative for Relational Database”, International Journal of Computer Sciences and Engineering, Vol.7, Issue.2, pp.685-691, Feb 2019.
21. H. R.Vyawahare, Dr P. P. Karde, Dr V. M. Thakare, “Hybrid Database Model For Efficient Performance”, In Elsevier International Conference On Pervasive Computing Advances & Applications(PerCAA), January 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.