Revista de la Facultad de Ingeniería Industrial
18(2): 135-28 (2015) UNMSM
ISSN: 1560-9146 (Impreso) / ISSN: 1810-9993 (Electrónico)

Literature review about Neo4j graph database as a feasible alternative for replacing RDBMS

Recibido: 11/03/15 Aceptado: 04/11/15

Félix Melchor Santos López*
Eulogio Guillermo Santos De La Cruz**

ABSTRACT

Neo4 is a NoSQL graph database that has been emerging in the fields of social networks and web applications with high concurrency. The characteristics of supporting technological transactions and a high scalability have been attracting the attention of the academic world. Therefore, this literature review focuses in analyse four research papers developed in USA, Spain, India and Germany. The authors show the results of benchmarking between Neo4j and other models, including relational databases. The aim of this paper is to illustrate the features of Neo4j, the architecture and advantages. Also, the purpose is to identify if Neo4 is a reliable alternative for replacing the RDBMS (Relational Database Management System) and offer suggestions for carrying out better experiments.

Keywords: benchmarking, graph, Neo4j, NoSQL

Revisión bibliográfica sobre la factibilidad de la base de datos Neo4j orientado a grafos como alternativa de reemplazo de RDBMS

RESUMEN

Neo4j es una base de datos gráfica NoSQL que viene emergiendo en los campos de las redes sociales y las aplicaciones web de alta concurrencia. Las características tecnológicas de soportas transacciones y una alta escalabilidad están atrayendo la atención del mundo académico. Por tanto, esta revisión bibliográfica se enfoca en el análisis de cuatro investigaciones recientes realizadas en Estados Unidos, España, India y Alemania. Los autores muestran los resultados de experimentos de comparaciones entre Neo4j y otros modelos, incluyendo las bases de datos relacionales. El objetivo de este artículo es ilustrar las características de Neo4j, su arquitectura y ventajas. Además, el propósito es identificar si Neo4j es una alternativa confiable para reemplazar a las RDBMS (Sistema de Gestión de Bases de Datos Relacionales). Finalmente, se ofrecen recomendaciones para llevar a cabo experimentos comparativos más precisos.

Palabras clave: comparación, grafo, Neo4j, NoSQL

1. INTRODUCTION

The spread of the Internet nowadays includes social networks, sharing of pictures, online movies, online video games and other topics related to the Web 2.0. Those trendy applications have demanded a dramatic change in how organizations manage information in the databases. Alternatives databases named as Not Only SQL or NoSQL have been emerging in recent years, and they have been monopolizing the market of web applications with high demand. This movement appeared for the first time in a conference of non-relational databases in San Francisco in 2009 [1]. It was described for pointing out the set of databases that do not use SQL (Structure Query Language) of the classic relational models. NoSQL divides around of one hundred and fifty different types of databases in four big groups: Colum, Documental, Key-Value and Graph databases. This division is based in architectural features and how the NoSQL databases treat the four basic operations denominated CRUD (Create, Read, Update and Delete). The aim of this literature review is the evaluation of a specific graph NoSQL database called Neo4j in comparison with other databases, and explains the architecture, characteristics, advantages and disadvantages.

This paper reviews the definition of Neo4j graph database, its architectural components and the mechanisms for manipulating the data. The first point gives an overview of Neo4j features, then the structure and how it records and retrieves data. The following section evaluates four research papers and the main findings, and each article shows a benchmarking between Neo4j and relational databases as MySQL and Postgres. After, the review criticizes how the previous experiments were carried out, focusing on the methodologies, the quality of the simulated data, the sample size and the IT infrastructure used. The literature review concludes with recommendations about how to achieve deeper and precise benchmarking of Neo4j with relational databases. It also highlights the importance of using real social web portals or information systems currently working in production environments. More research about the performance with the Delete and Update operations is required in order to claim that Neo4j is a realistic and strong candidate for replacing the relational databases that have been hoarding the market of enterprise applications.

* Post graduate student, Master of Information Systems, The University of Melbourne, Australia. Informatics Engineer, Pontificia Universidad Católica del Perú. E-mail: fsantos@pucp.edu.pe
** Industrial Engineer, Universidad Nacional Mayor de San Marcos. Professor at Industrial Engineering School, UNMSM. E-mail: esantosd@unmsm.edu.pe

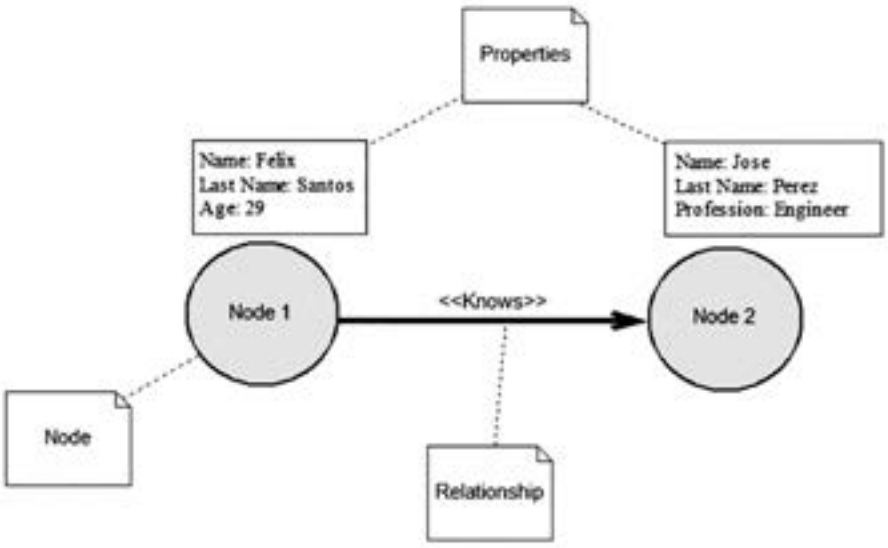
2. OVERVIEW AND ARCHITECTURE

Neo4j is a NoSQL database that belongs to the category of graph databases, and it follows the mathematical theory of trees. According to Justin Miller [2], the nodes and the relationships between them is the base for the concept of graph. The author states that nodes and relationships or arista have properties where the information systems or users can store the data within the pattern key-value. Hence, Neo4j follows a structure where the nodes are represented as vertices and the relationships or arista as edges. This organization applies sophisticated algorithms and mathematical calculations for efficient data retrieval. It also keeps the use of a dynamic structure, the assignation of values only when it is necessary and a more precise design aligned to the business rules [3]. Furthermore, Neo4j warranties ACID (Atomicity, Consistency, Isolation and Durability) behaviour and this becomes Neo4j in the few NoSQL databases that support transactional operations. Figure 1 illustrates the components of a graph, how are distributed and the storage of the information.

Another main characteristic of Neo4j is a robust architecture implemented under the concept of high availability (HA). The master-slave cluster is the most important consideration in the model of HA [4]; it divides Neo4j in two parts: the database itself and the cluster management component. Into that component, there is a mechanism that provides constant synchronization of all instances and it ensures that the master cluster election is automatic. This process permits the master cluster to handle all the write operations, and it gives a centralized control for achieving scalability. Moreover, all the graphs are replicated in each instance of each cluster, and this feature provides security of continuing work and response despite of possible failures in some clusters. This redundancy also works in a quorum where most of the clusters should be online for writing operations; otherwise the read-only state is immediately activated. Figure 2 illustrates the high availability architecture in Neo4j.

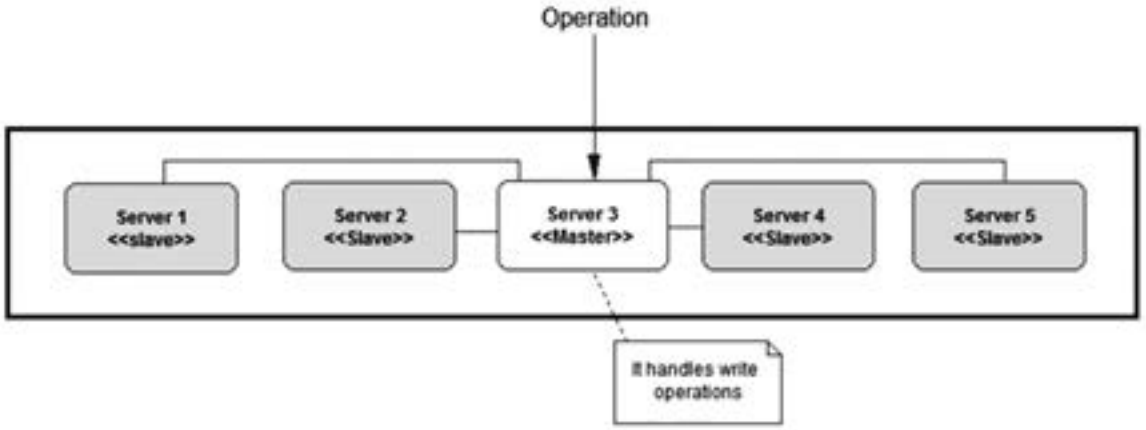
On the other hand, in the classical model of relational databases the data is inserted and manipulated in a set of rigid tables. It is bound through relationships, indexed and identified by primary and foreign keys using logical foundation of mathematics and relational algebra [5]. Meanwhile, Neo4j follows a non-structured repository which gives the advantage of not needing neither a previous database design nor a rigid and difficult manipulation of set of nodes. The article titled “Facebook Graph Search with Neo4j” [6] written by Jiepeng Zhang, Zhenhua Li and Sha Liu is an outstanding example how a dynamic structure is useful in specific projects. Jiepeng et al. (2013) developed a social network search based on Facebook. This application collects all the friends who like a specific thing. This application requires navigating and searches into a big set of people and people’s friends in order to collect the target requirement. Therefore, this complex search can only be understood and visualized like a big tree or grid. There are several many-to-many relationships

Figure 1. Graph illustration.



Source: Own elaboration

Figure 2. High Availability of Neo4j.



Source: Own elaboration

in case of implementing a relational structure for recording the results of the search. Thus, this complexity was solved by the authors using a non-structured database like Neo4j.

3. BENCHMARKING WITH RELATIONAL DATABASES

The benchmarking evaluation between Neo4j and relational databases must be accurate and it should explain clearly which relational databases brands are involved in the experiments. For instance, a research benchmarking between Neo4j and MySQL relational database was presented in the 16th International Conference on Extending Database Technology in 2013 by Florian Holzschuher and Dr. René Peinl at the Institute of Information Systems at Hof University in Germany. A second study was carried out in India at Thapar University by Shalini Batra and Charu Tyagi. The researchers also only compared Neo4j and MySQL databases. Similarly, Chad Vicknair and his colleagues presented an article in the Association Computing Machinery Southeast Conference in 2010. The study focused on MySQL and Neo4j databases. In Spain Renzo Angles and his colleagues at DAMA (Data Management group) presented an evaluation including Neo4j and Dex graph databases, PostgreSQL and Virtuoso relational databases, and an RDF store called RDF-3X.

4. STRUCTURES EVALUATED

Data types and structures are relevant in benchmarking evaluations and those variables could affect future performance results. Florian Holzschuher and Dr. René Peinl evaluated a real social web portal and the performance when a simulated information system retrievals data. The relational model designed tables called first name, gender, and target. Meanwhile, in Neo4j those tables were established as nodes. Other authors evaluated three queries where the complexity and amount of objects were incremented in each case, but details about the structure were not presented [7]. The peculiarity of other evaluation was based on using a social network, but taking as reference a report of the use of Facebook in 2012 [8]. In addition, nine queries were evaluated. The study also implemented and evaluated the loading time of the data, as well as the recovery. Another research showed the importance of the provenance of the data, defined as its origin and how it was collected [9]. The authors highlight the DAG (Directed Acyclic Graph) considered the structure, simple or complex, to store data. Vicknair et al. (2010) state that DAG is the most common way to store data in either relational or graph database. Nonetheless, relational databases are inefficient and slower than graph for processing complex relationships and joins.

5. DATA SAMPLE SIZE

Another key characteristic in benchmarking experiments is the data sample size which should be relevant for simulating or representing real scenarios. One experiment mentioned that the data set contained 2011 people 1126982 messages, 25365 activities, 2000 addresses, 200 groups and 100 organizations [10]. Likewise, the evaluation tested a larger dataset with 10003 people and the respective amount of other nodes based on Slashdot from the 2008 Stanford Large Network Dataset

Collection [10]. Other authors limited their sample size to one set of one hundred objects and other of five hundred [7], without giving a clear specification or justification if those sizes are relevant in real contexts. Other scholars evaluated bigger sample size, which included run in twelve MySQL and Neo4j databases, providing constant increment of 1000, 5000, 10000 and 100000 nodes subsequently [9]. In other experiment the sample size was aligned to the demand of real social web networks like Facebook. The study used twelve queries and the simulation of the names of persons and locations were selected randomly from dictionaries including 5494 first names, 88799 last names and 656 locations [8].

6. OUTSTANDING PERFORMANCE

There is a clear consensus among the investigations that the theoretical superiority of graph oriented database, in terms of performance, is better than relational models. According to the findings of the four experiments explained in this literature review, there is a better time response in the case of Neo4j measured in milliseconds. Additionally, all authors claim that the relational model needs a longer searching time when the data is incremented, conversely Neo4j only seeks on the set of nodes linked, and it provides a shorter path to cover. However, one author stated that there were issues in the performance for the cases of numeric type data. The study also pointed out that another data type like double, outside the scope of that experiment, could works slower [9]. Also, in one experiment authors evaluated Dex, another graph database. The results led to better performance to Dex, followed by Neo4j and below RDF3 and PostgreSQL [8].

7. CRITIQUE OF THE INVESTIGATIONS

In terms of the data quality used on the previous papers, only Angles et al. (2013) argued the use of a real dictionary. The final data is based on the result of a real social network like Facebook that provided the sample size. The experiment adopted a Recursive Matrix Model and probabilities [8], where 80% of the nodes were assigned to the people and 20% for the webpages. The context was the case of a group people who liked a web site. Additionally, the authors focused in a micro benchmarking of atomic operations instead of queries with high difficulty. On the other hand, Holzchuher and Dr. Peinl also evaluated a real social web, and they made sure that their data were most similar to those used in information systems in real scenarios. However, they did not mention any neither statistical operation nor strong criteria for choosing the sample size, but they generated the random data based on Stanford Large Network Dataset Collection [10]. In the other experiment, Vicknair et al. (2010) focused on data types, and they did not evaluate a real social network or application. They did not explain how the sample size was calculated and the approach on DAG looked more theoretical than realistic. Likewise, Batra and Tyagi (2012) evaluated three queries using one hundred and five hundred objects in each run. The experiment did not explain how that sample size was calculated. There was not a social network or another real program as an input for this evaluation. As a whole, most executions should require more real datasets and an accurate justification for the sample size.

Another controversial point on the investigations is the incomplete evaluation of CRUD operations. Holzchuher focused only in the retrieval of data, as well as Batra and Vicknair et al. (2010). Angles et al. (2013) carried out data loading time and query execution for recovering data. The operations of Create and Read were tested, but the operations of Update and Delete were omitted in every comparison. The IT infrastructure and the operative system were relevant, and every execution could respond different depending on different environments. Angles et al. (2013) used an Intel Xeon E5530 CPU of 2.4 GHz, 32GB DDR3 memory at 1066 MHz, 1Tb hard drive and Linux Debian 2.6.32-5-amd64 kernel. Meanwhile, Oracle Virtual Box and a Proxmox/KVM server, using Two AMD 6-core of 3.1 GHz, 64 GB Ram and RAID 5 with 4x15 TB hard disc was used in Holzchuher evaluation. Vicknair et al. (2010) used Ubuntu Linux System 9.2, with an Intel Core 2 Duo CPU, 3.00 GHz and 4GB RAM. Batra did not mention neither the operative systems nor the hardware used. These characteristics are important because they simulate real scenarios based in hardware and software similar to production environments are compulsory for achieving reliable conclusions.

8. CONCLUSIONS AND RECOMMENDATIONS

To sum up, in three of the four articles analysed, Neo4j showed better results in terms of time response, compared with relational databases. Only Angles et al. (2013) evaluated DEX, another graph database that offered a better performance than Neo4j. The findings show that the superiority of graph databases is demonstrable through benchmarking experiments. Additionally, the evaluation of different data types and structures

did not alter the final results in each study. The data sample did not modify the tendency in the conclusions, but the authors did not explain neither a methodology nor a formal method for sample size justification.

There are four gaps that should be solved in order to achieve more conclusive results. The quality of the data impacts in the final results because if Neo4j is a feasible alternative for replacing database models, the comparison executions should use real datasets. It is highly recommendable for instance to evaluate also data from banking, insurance and selling processes because those scenarios use a high rate of relational databases. As a second point, the operations of Update and Delete data should be evaluated and compared because there are very few databases that are only used for insertions and readings. Therefore, scientific evidence of those operations are mandatory. After that, in enterprise contexts operative systems like UNIX, Solaris and Linux Red Hat are more common. It is recommended to compare the performance in more commercial operating systems. Nevertheless, this should be combined with other hardware configurations such powerful servers and devices used in high availability systems in the financial industry for example. Finally, further research should follow the Gartner Magic Quadrant for Operational Database Management Systems released in 2013 [11]. Gartner is a non-biased company that performs surveys in the technological market. MySQL, Postgres and Virtuoso do not lead the market of relational databases. A future comparison between Neo4j and relational model should evaluate the most popular relational databases like Oracle, SAP, Microsoft and IBM.

9. REFERENCES

[1] C. Strauch and W. Kriha, “NoSQL Databases: Selected Topics on Software-Technology Ultra-Large Scale Sites,” Stuttgart Media University, Stuttgart, 2011.

[2] M. Justin, “Graph Database Applications and Concepts with Neo4j,” in Proceedings of the Southern Association for Information Systems Conference, Atlanta, USA, 2013.

[3] E. Redmond and J. Wilson, “A Guide to Modern Databases and the NoSQL Movement,” in Seven Databases in Seven Weeks, USA, Pragmatic Bookshelf, 2012, p. 220.

[4] D. Montag, “Understanding Neo4j Scalability,” NeoTechnology, USA, 2013.

[5] H. Darwen, An Introduction to Relational Database Theory, United Kingdom: Bookboon. com, 2010.

[6] J. Zhang, Z. Li and S. Liu, “Facebook Graph Search with Neo4j,” Department of Computer Science, Georgia State University, Atlanta, USA, 2013.

[7] S. Batra and C. Tyagi, “Comparative Analysis of Relational and Graph Databases,” in International Journal of Soft Computing and Engineering, India, 2012.

[8] R. Angles, A. Prat-Pérez, D. Dominguez-Sal and J.-L. Larriba-Pey, “Benchmarking database systems for social network applications,” in First International Workshop on Graph Data Management Experience and Systems, New York, USA, 2013.

[9] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen and D. Wilkins, “A Comparison of a Graph Database and a Relational Database,” in ACMSE 10’, Oxford, MS, USA, 2010.

[10] H. Florian and P. René, “Performance of Graph Query Languages: Comparison of Cypher, Gremlin and Native Access in Neo4j,” ACM, pp. 18-22, 2013.

[11] R. Zicari, “ODBMS.org,” Operational Database Management Systems, 23 03 2013. [Online]. Available: http://www.odbms.org/2014/03/2013-gartner-magic-quadrant-operational-database-management-systems/. [Accessed 17 10 2014].