Using Graph Databases for Querying and Network Analysing

Blagoj Ristevski
Department of Software Engineering and Information Systems
Faculty of Information and Communication Technologies - Bitola
"St. Kliment Ohridski" University - Bitola
Ul. Partizanska bb 7000 Bitola, Republic of Macedonia
blagoj.ristevski@uklo.edu.mk

***Abstract* - Huge amounts of data are generated and used on a daily basis. This has led to the new concepts: big data and hybrid databases that are becoming more popular and promising. Thus, there is a need of integrating and storing data from various heterogeneous data sources. With the growth of data size, NoSQL databases have outperformed traditional relational databases for analysis, access and querying on big data. Relational databases store data in multiple tables and they are not suitable to represent numerous kinds of complex relationships among entities, particularly in computer networks, biological and social networks. On the other hand, graph databases rely on graph structure and are able to handle complex relationships. These databases use nodes representing entities and edges representing relationships, to present and store data. Graph database are very suitable for storing and querying heavily interconnected data, especially for large-scale network data. In this paper, Neo4j database and Cypher query language are described, and their using for analysis, querying and effectively mining of biological network's data.**

Keywords: databases, NoSQL databases, omics data, graph databases, complex biological networks.

I. Introduction

Nowadays, a huge amount of data is created on a daily basis. These datasets can be a physical group of values of different attributes. To discover a helpful knowledge from these datasets, which is associated with relations between the attributes' values, numerous data mining algorithms can be applied on these datasets.

In relational databases, data are tabled and relationships among tables are achieved by foreign keys. The modeling of the relational databases requires obeying very strict rules and constraints that include primary key constraint and management, database integrity and database normalization.

In the big data area, creation of massive amounts of structured, non-structured or semi-structured data and application of data mining techniques requires another database models that will solve the challenges, which relational database model is not capable to resolve them in a suitable manner. Especially, the data that are suitable to be represented as networks and then to apply a proper network analysis requires different models, such as a graph database model.

The data representation as a graph, establishing of the relationships, the simplicity in the query formulation, data visualization modules, interoperability, acceptance of different standard data formats and the real programming experience on a logical database design make the graph databases to overcome the relational databases [1].

Differently from the relational databases, in graph databases it is possible to have a disciplined number of multivalued attributes, such as node name, node number or array of different attributes for a single node. As an analogue of connection between different tables (relations) through the foreign keys in relational databases, in graph databases each node has a fixed number of incoming and outgoing edges called as indegree and outdegree of the node, respectively. Another advantage of the graph database is the possibility to apply different graph algorithms for numerous experiments in data mining [1].

In graph databases, the relationships between two nodes are established using typical query syntax. There are different plugins, packages or modules for data visualization that can be easily integrated with the graph database workflow. Graph data visualization is an analogue to the select queries in the relational databases [1].

The remainder of the paper is structured as follows Section II compares relational and NoSQL databases. The main characteristics of graph databases and Cypher – query language for analyzing and network analyzing are described in Section III. The application of graph databases for biological data is given in the subsequent section. The last section provides concluding remarks and direction for further application of graph databases.

II. NoSQL Databases

Novel data storage systems, that are capable to deal with big data, are introduced and named as NoSQL databases. Many of NoSQL databases offer horizontal scalability and higher availability than relational databases [4].

As big data and hybrid database system are becoming popular, the popularity of NoSQL database is raising [2]. Companies and research organizations and institutes integrate relational and NoSQL databases. While relational databases are used to deal with small and middle scale data, NoSQL databases are used as system back-end data pool for batched operations for reading and writing, as well as analysis [2].

The recent advances of the social networks and modern Web with heterogeneous unstructured, semi-structured, continuous and high-order data require new database models that are different from relational database [9]. NoSQL databases show potential to deal with such datasets. There are four main group of NoSQL databases: graph, key-value, document and column family databases [9].

In key-value databases, data are stored in associative arrays and complex maps between sets of objects are allowed. In document databases, data are stored in JSON documents and they can deal with large document collections. Column family databases store data in tables with very long rows grouped into column families [9].

In graph databases, a graph is consisted of nodes, edges and properties. Nodes, that represent objects, contain properties in key-value pairs. Edges, that represent directed and labeled connections, have a start and an end node. Edges contain properties in key-value pairs, too [9]. Graph analytics and graph models can be extended to fuzzy graphs. In a fuzzy graph an edge between two nodes exists with a certain probability. This graph model extension has resulted with development of a fuzzy version of Cypher for graph querying [9].

III. Characteristics of Graph Databases

One of the main drawbacks of the relational model is its limitation to explicitly cover requirement semantics [11]. Where relational databases are optimized to deal with aggregated data, graph databases are very suitable to handle highly connected data. A common graph type supported by most graph databases is the property graph. Property graphs are labeled, attributed and directed multi-graphs [11]. The graph database design is suitable to construct predictive models and to detect correlations and patterns that exist in the data.

Graph databases support storage of unstructured data that can be integrated through contemporary forms of data interchange by using REST and JSON interfaces. They provide a natural manner for graph storage as well as for graph analytics such as link prediction and minimum spanning trees [9].

A graph database is represented by graph consisted of sets of nodes and edges. Nodes represent biological entities, whereas edges represent the connections or interactions. In addition, the nodes can have properties and nodes that have shared property can be comprehended that they are linked, as shown on Fig. 1. Nodes linked through edges to an intermediate node indicate that they are sharing common property [5].

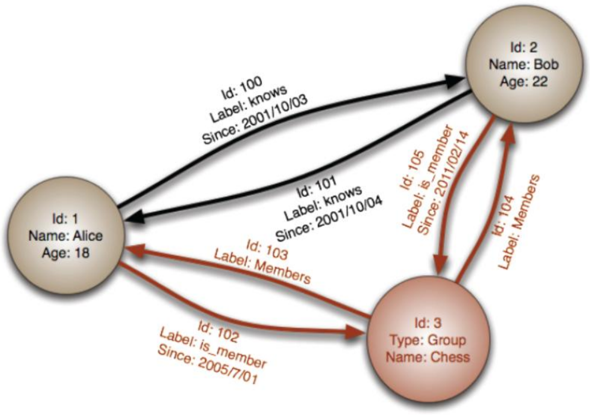

Figure 1: A graph with nodes and edges, where the edges represent the properties between the nodes [16].

The graph databases are optimized to store and query graph structures and huge amounts of data containing interconnected data, which are not suitable to be in represented by the relational databases [14]. Following characteristics make graph data to be the fastest growing category in database management systems [14]:

- Data representation as graph makes them to be quite intuitive.
- They support using new data types.
- They are able to store huge volume data in the order of petabytes.
- The information searching and data mining techniques are more optimized; especially they show high querying performances in very deep searches, compared to the relational databases.
- During development they can be easily adapted in regards of data updating and hence they are very agile.
- They are very suitable for highly interconnected data, such data in social networks, computer networks, biological networks and web semantics.

Examples of graph databases are: Neo4j, Allegro Graph, Hypergraph DB, Bitsy, GraphBase and TigerGraph [13].

Nowadays, with creation of huge amounts of unstructured data and using big data analytics applications in biology, as well as in many other domains, Neo4j is a very suitable platform that provides solutions to numerous challenges in databases, data science and data mining.

One of the best representatives of graph databases is Neo4j [14]. The horizontal scalability in Neo4j allows easily adding new nodes in a database. It has its own query language – Cypher. Cypher query language provides deletion, insertion and creation of the basic elements: nodes, edges and properties The data storage is disk-based via proprietary file systems. It fully supports ACID (atomicity, consistency, isolation and durability) transaction properties and it has very intuitive and accessible interface.

Graph databases, whose schemas are free and flexible, by using graph model can effectively store, manage, update data and relationships [7]. Most commonly used graph database is Neo4j that is a robust and supports ACID transaction properties [7]. The data of Neo4j are stored on a disk and they are loaded into memory when a query is executed and then cashed. For a Neo4j high availability (HA) cluster, each Neo4j instance data are consistent and the corresponding queries are based on each Neo4j instance of independent occupied memory space [7].

Neo4j graph database can be run as desktop database or database server and uses Cypher, an SQL-like versatile query language for graphs. Neo4j has user friendly interface that can show results as graphs or tables and has interfaces to the essential programming languages used in bioinformatics: R, Java and Python [5]. Hölsch et al. has shown that Neo4j is more efficient for path queries that do not filter on specific types of edges, differently for analytical queries, where the relational database has shown better performances [8].

To retrieve information out of a graph, a traversal is required. A graph traversal, that includes “walking” along the elements of a graph, is a crucial operation for data retrieval [11]. Differently from the SQL queries, the traversal is a localized operation.

For instance, Cypher graph query language has a function for finding shortest paths, disregarding the edge weights. This leads to a huge speed improvement when Neo4j finds the shortest path with no length constraint, compared to SQL querying [3]. The result from a SQL query is available only in tabular form, but graph-view results are not available. In Cypher query language the results can be shown as graphs and as tables [13].

IV. Graph Databases for Biological Data

The absence of database schema makes Neo4j to be much more flexible, but it eliminates the data interoperability standards, which can be solved by introducing consistent semantics [5]. Moreover, the Cypher graph query language in the biological graph databases is able to generate a plethora of common biologically-based queries [5].

The main concerns in big omics data in biology are dealing with their rich semantics and complexity. The principles for graph database modelling are very similar to those employed for design of ontology. The ontology in graph databases is analogous to the schema in a relational database.

One of the major goals in biology is to understand complex interactions among entities that exist and contribute to the living cell functions [6]. These interactions can be measured experimentally and are stored in heterogeneous data formats. Revealing of these complex interactions among these voluminous heterogeneous biological data is very difficult task. Although biological network data can be stored in relational databases, multiple join queries make them to be too computational demanding and to have too complex design.

Currently, to understand molecular basis of various disease, huge amounts of high-throughput data, so called omics data, are generated. Understanding the underlying diseases’ processes requires integration of numerous heterogeneous data and then to study the complex interactions and relationships among entities (i.e., genes, proteins, non-coding RNAs, metabolites). These relationships might belong to the different types, such as participation (a gene/protein participates in a particular pathway), sequence similarity (one protein is similar to another) or interactions (one protein/gene interacts with another protein/gene) [5]. In such a way, these relationships depict complex biological networks.

Information contained in the biological data is typically highly connected, semi-structured or unstructured, as well as unpredictable. Therefore, these data features are crucial to choose a suitable manner to represent and query the complex biological networks [5]. Moreover, the amount of information associated to a particular entity may not be uniform, which makes the standard relational database model to be non-suitable to handle cases with a large number of uniform entries associated with a limited number of data types. In relational database model all data should be transformed to predefined schema, which makes concerns for representing of semi-structured and unstructured data. Biological research can be unpredictable, especially when new techniques or new data resources are introduced. To add these new data into a relational database that can be very important to understand the roles and involvements of particular biological entries in a particular pathway, re-design of the database schema is needed, which is a complex and demanding task [5]. Graph databases are capable to represent non-uniformly distributed, highly connected, semi-structured and unpredictable data which are founded in the biological research studies [5]. The graph databases are scalable and allow adding new data types.

One approach to detect new oncogenes is the extraction of subgraphs with four nodes from a protein-protein interaction network that contain transcription of the gene expressions [15]. As a result of the huge amounts of subgraphs formed by this complex network of interactions, the processing and querying when relational databases are used is very time demanding and not effective.

Genome corresponds to the whole genetic material of an organism, whereas the transcriptome determines which genes are expressed and these expression levels change during the organism’s life. Both genome and transcriptome are related to the proteins that demonstrate big variations among the cells of a particular organism [15]. The proteome of an organism analyzes the proteins, enables identification of proteins and understanding of their functions.

Proteome is the total set of proteins, their variations and modifications with other proteins and interactions network.

The transcriptome is the set of all RNA messenger transcriptions produced within the cells of a tissue. To obtain transcriptome data, a high-throughput technology is used to determine how the transcriptome changes in time lags or in a determined biological state of the organism [15]. Thus, transcriptome allows clarifying a disease pathogenesis.

The protein-protein interaction networks are interactome networks consisted of all interaction among proteins that happen in a variety of protein's physiologically relevant concentration [15].

Mattioli and Gubitoso have shown that the identification and detection of four-node networks motifs by using graph database was the optimal option [15].

Shoshi et al. have developed a Neo4j graph database called GenCoNet that integrates different associations between genes, diseases, variants and drugs for the essential hypertension and bronchial asthma [10].

Have and Jensen have proposed an integrated graph database for biological data, called BioGraphDB [12]. BioGraphDB can be applied for various clinical research analysis, such as:

- analysis of gene functions and pathways
- analysis of protein motifs linked to cellular pathway
- analysis of tumor-suppressor/oncogenic microRNA
- target analysis of differentially expressed microRNAs in cancer.

To analyze the forehead-mentioned biological network interactions/pathways, various biological database should be used such as: Entrez gene, RefSeq, Gene Ontology (GO), Reactome, UniProt, miRNA-target interactions, mirBase and mirCancer [12].

V. Discussion and future work

Graph databases have outperformed relational databases when working with data where data topology or interconnectivity is important [11].

Graph databases are very suitable to apply for biological and semantic networks and recommender systems. Bioinformatics uses graph databases to represent complex biological networks that include proteins, genes, enzymes, metabolites and microRNAs. A typical example is the open source bioinformatics platform Bio4j that uses Neo4j as a framework. Bio4j has integrated data from diverse sources (Gene Ontology, RefSeq, UniREf, Uniprot KB, NCBI Taxonomy, Expasy Enzyme Database) into one data source.

Moreover, social networks, such as LinkedIn and Facebook and online airplane booking companies utilize graph databases, too [6].

Furthermore, for storing of highly-interconnected data, voluminous and unstructured data graph databases should be used. The developed software solutions should be based on the graph databases and their properties, particularly when those solutions are focused on social network analysis and biological big omics data.

Acknowledgement

This paper was supported by the Ministry of Education and Science of the Republic of Macedonia and the Ministry of Science and Technology (MOST) of the Government of the People's Republic of China within the bilateral project "Modeling, Simulation and Analysis of Complex Networks in Biology".

References

[1] Johnpaul, C. I., and Tojo Mathew. "A Cypher query based NoSQL data mining on protein datasets using Neo4j graph database." In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1-6. IEEE, 2017.

[2] Liao, Ying-Ti, Jiazheng Zhou, Chia-Hung Lu, Shih-Chang Chen, Ching-Hsien Hsu, Wenguang Chen, Mon-Fong Jiang, and Yeh-Ching Chung. "Data adapter for querying and transformation between SQL and NoSQL database." *Future Generation Computer Systems* 65 (2016): 111-121.

[3] Have, C. T., & Jensen, L. J. (2013). Are graph databases ready for bioinformatics?. *Bioinformatics*, *29*(24), 3107.

[4] Gessert, Felix, Wolfram Wingerath, Steffen Friedrich, and Norbert Ritter. "NoSQL database systems: a survey and decision guidance." *Computer Science-Research and Development* 32, no. 3-4 (2017): 353-365.

[5] Lysenko, Artem, Irina A. Roznovăţ, Mansoor Saqi, Alexander Mazein, Christopher J. Rawlings, and Charles Auffray. "Representing and querying disease networks using graph databases." *BioData mining* 9, no. 1 (2016): 23.

[6] Yoon, Byoung-Ha, Seon-Kyu Kim, and Seon-Young Kim. "Use of graph database for the integration of heterogeneous biological data." *Genomics & informatics* 15, no. 1 (2017): 19.

[7] Huang, Hongcheng, and Ziyu Dong. "Research on architecture and query performance based on distributed graph database neo4j." In *2013 3rd International Conference on Consumer Electronics, Communications and Networks*, pp. 533-536. IEEE, 2013.

[8] Hölsch, Jürgen, Tobias Schmidt, and Michael Grossniklaus. "On the performance of analytical and pattern matching graph queries in neo4j and a relational database." In *EDBT/ICDT 2017 Joint Conference: 6th International Workshop on Querying Graph Structured Data (GraphQ)*. 2017.

[9] Drakopoulos, Georgios, Aikaterini Baroutiadi, and Vasileios Megalooikonomou. "Higher order graph centrality measures for Neo4j." In *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1-6. IEEE, 2015.

[10] Shoshi, Alban, Ralf Hofestädt, Olga Zolotareva, Marcel Friedrichs, Alex Maier, Vladimir A. Ivanisenko, Victor E. Dosenko, and Elena Yu Bragina. "GenCoNet–a graph database for the analysis of comorbidities by gene networks." *Journal of integrative bioinformatics* 15, no. 4 (2018).

[11] Miller, Justin J. "Graph database applications and concepts with Neo4j." In *Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA*, vol. 2324, no. S 36. 2013.

[12] Fiannaca, Antonino, Massimo La Rosa, Laura La Paglia, Antonio Messina, and Alfonso Urso. "BioGraphDB: a new GraphDB collecting heterogeneous data for bioinformatics analysis." *Proceedings of BIOTECHNO* (2016).

[13] Jaiswal, Garima, and Arun Prakash Agrawal. "Comparative analysis of Relational and Graph databases." *IOSR Journal of Engineering (IOSRJEN)* 3, no. 8 (2013): 25-27.

[14] Guia, José, Valéria Gonçalves Soares, and Jorge Bernardino. "Graph Databases: Neo4j Analysis." In *ICEIS (1)*, pp. 351-356. 2017.

[15] Mattioli, Diogo, and Marco D. Gubitoso. "Application of Graph Database in the Storage of Heterogeneous Omics Data for the Treatment in Bioinformatics." In *Proceedings of the 2018 10th International Conference on Bioinformatics and Biomedical Technology*, pp. 51-56. ACM, 2018.

[16] Goel, Ankur. *Neo4j cookbook*. Packt Publishing Ltd, 2015.