BRAIN. Broad Research in Artificial Intelligence and Neuroscience
ISSN: 2068-0473 | e-ISSN: 2067-3957
Covered in: PubMed.gov; IndexCopernicus; The Linguist List; Google Academic; Ulrichs; getCITED; Genamics JournalSeek; J-Gate; SHERPA/RoMEO; Dayang Journal System; Public Knowledge Project; BIUM; NewJour; ArticleReach Direct; Link+; CSB; CiteSeerX; Socolar; KVK; WorldCat; CrossRef; Ideas RePeC; Econpapers; Socionet.

2019, Volume 10, Issue 4, pages: 56-72 | doi:10.18662/brain/05

Application of Non-Relational Databases in the Social Networks

Miroslav DECHEV[1],
Miroslav GALABOV[2],
Tsvetanka GEORGIEVA-TRIFONOVA[3]

[1] Faculty of Mathematics and Informatics, "St. Cyril and St. Methodius" University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria, miroslav.dechev@gmail.com

[2] Faculty of Mathematics and Informatics, "St. Cyril and St. Methodius" University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria, lexcom@abv.bg

[3] Faculty of Mathematics and Informatics, "St. Cyril and St. Methodius" University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria, cv.georgieva@live.uni-vt.bg

Abstract: *The purpose of this research is to present knowledge in the field of work of databases that contain semi-structured data, also known as NoSQL, in applications using big data. The characteristics of the main categories of the non-relational databases and some of their typical representatives (MongoDB, Redis) have been reviewed from the point of view of using them for storage and analysis of data in the social networks. The article presents a study, conducted on the already existing research in regards to the implementation of the non-relational databases in the social networks. Some of the perspectives of the future research are outlined as well.*

Keywords: *NoSQL; databases; social networks; big data; Redis; MongoDB.*

How to cite: Dechev, M., Galabov, M., & Georgieva-Trifonova, T. (2019). Application of Non-Relational Databases in the Social Networks. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience, 10*(4), 56-72. doi:10.18662/brain/05

1. Introduction

The object of the research is the non-relational database and its application in the social networks. A non-relational database (*Not only Structured Query Language, NoSQL*) utilizes a mechanism for storage and retrieval of data, that uses a freely-coordinated pattern in contrast to the widely distributed relational databases.

1.1. Non-relational databases

The general idea behind the creation of the non-relational database was to suit the needs of the internet applications, whose most typical trait is their flexibility when it comes to working with large quantities of semi-structured data. This article reviews the NoSQL applications on the social networks, where big data are used.

Big data is the designation of a technology utilizing a number of specific instruments and processes that aid the organizations and the companies in dealing with the ever-increasing quantities of information. The structure of big data changes very rapidly, which stems from the fact that new information generates constantly and originates from various different sources – social networks, video channels, GPS data, etc. Because of this, the structure of big data is very dynamic (Klisarova-Belcheva, 2015).

A serious upsurge has been registered in the last years in the development of databases that do not support SQL, or are non-relational. The main purpose of this is to achieve better productivity, refusal-steadfastness or to pursue the opportunity to deliver to a large number of systems.

One of the benefits of this approach is the acquisition of horizontal scaling. Most often, a non-relational database is a well-optimized storage space that contains information from the key-value type. Its purpose is to ease the processes of retrieval and addition of information, in order to enhance the productivity.

The NoSQL databases emerge at the beginning of the 21st century. They were created to answer the necessity of managing the large quantities of information in the modern web and mobile applications. Solutions such as MongoDB (2009), Redis (2009) and others begin to appear. The NoSQL databases are used to deal with big data and real-time web applications.

With the development of the cloud technologies, the role and the importance of the so-called NoSQL databases grow even further and further. This is due to the convenience that they offer when working in a real-time setting in web environment with big data (Cheresharov & Krushkov, 2016).

The non-relational databases play a substantial and ever-increasing role at the real-time web and the big data applications. The essence of the real-time web applications is in the processing and delivery of data in a limited and preferably a very short period of time. The real-time web applications deliver content in 'real time' – within few seconds or a fraction of the second. Their creation is a response to the contemporary business environment on the internet.

1.2. Social networks

The online gambling business, the financial applications, the games and the social networks are fully or partially based on real-time web implementations. The real-time web functionality allows the customers to obtain the actual information at the very moment when it s available on the server. This functionality is implemented through different techniques that are based on the different types of communication on the internet. There are numerous examples of real-time web applications and the challenges they face in the very environment around us (Nedyalkov, 2017).

Such an example is the social network Facebook and its notifications that are implemented in real-time. If a comment or a "like" occurs in our profile, we will be notified at the same moment as this occurrence. As one of the largest social networks with over 2 billion subscribers, Facebook is striving to respond to its users' requirement of a minimal delay. Facebook utilizes a combination of web sockets and long pooling in its real-time implementation (Nedyalkov, 2017).

Another example is the social network Twitter, which registers a large number of changes of the status of its users. Each user may have thousands or even millions of other users that are his/her followers. Namely, the followers are the ones who want to receive constantly and instantaneously updates for the user that they follow. In order to respond to this demand, Twitter uses a real-time implementation as well. The giant Google utilizes real-time implementation for its application "Google docs", where the users can share their documents with other users. The main goal is for everyone to follow the changes that are made in real time. The non-relational systems are called "Not only SQL" (*Structured Query Language*), or NoSQL, in order to emphasize on the fact that they allow the use of search languages that are different but similar to those of SQL.

The architecture of the non-relational database is distributed in a pattern, resistant to damage, as the data objects are kept in a few servers. Through this, the system may upkeep itself by adding more servers and if a server becomes damaged, the system will continue to operate successfully.

This type of database broadens in a horizontal manner and is used for the operation of large quantities of data when the productivity (in real time) is more important than the successiveness (as in the indexing of a large number of documents, the service pages of the websites with heavy traffic and the concession of streams) (Strozzi, 2017).

1.3. Categories of non-relational databases

The systems of management of the non-relational databases are generally intended to support allocated depositories where a high level of scalability is required. There are a few categories of non-relational databases as different methods with different categories and sub-categories for their classification exist. The most widely accepted classification is the one based on the data model. Some of the non-relational databases and their representatives, classified on the basis of the model of the data are presented in the following list: Column-oriented databases: Hbase, Cassandra, Accumulo; Document-oriented databases: MongoDB, Couch, Raven; Key-value databases: Dynamo, Riak, Azure, Redis, Cache, GT.m; Graph databases: Neo4J, Allegro, Virtuoso, Bigdata (http://nosql-database.org/, 2009-2011).

A typical trait of the databases of the key-value type is their use of associative arrays in the form of a hash-table holding a unique key and an index to a certain record. The data are presented in collections of sorted pairs of key-value; the key is unique within the borders of the collection.

The document-oriented databases provide storage, management and extraction of semi-structured data. MongoDB and Apache CouchDB are the more prominent representatives of this type of non-relational databases. The document-oriented model of data is similar to the key-value one with the sole difference that the recorded values represent documents and have an internal structure.

Compared to the relational model of data, the collections can be reviewed as tables and the documents can be reviewed as records (rows in tables). On the other hand, the documents in a single collection of document-oriented database do not necessarily share a common structure. All records of the table have the same successiveness of fields (attributes), while the separate documents in the collection may have fields that are completely different from one another.

The data of the column-oriented databases are saved in cells which in turn are grouped in columns, not rows (as is the case of the relational DBMS (*Database Management System*)). The columns on the other hand are grouped logically in families. Each family can contain practically an infinite number of columns. Data are saved and read by columns, not by rows. The advantage of the column-oriented format is the rapid search/access to the records, as well as the easy data aggregation. This is determined by the fact that the rows of the relational DBMS are saved in different partitions of the disc, while the column-oriented non-relational DBMS save all the cells of one column as a consecutive record in a single location of the memory, thus decreasing the time for input-output operations of a single read of the disc.

Another type is the graph databases. They are intended for data whose relations will be best presented through a graph. The data and the connections between them are presented as nodes and edges of a graph. These databases bear similarity to the document-oriented ones but they maintain connections between the separate objects. This database implements the idea of "object" of data known as node, which can have properties and edges with other nodes. Examples for such databases are: Infinite Graph, InfoGrid (https://help.superhosting.bg/sql-nosql-databases.html, 2019).

A subcategory of the graph databases are the RDF (*Resource Description Framework*) depots (RDF triplestores). The RDF standard, developed by W3C, allows for the semantic description of the web resources and their mutual connections in a manner that is simultaneously understandable to machines and humans alike. RDF presents the meaning of the data through triples; and every triple gives a statement that a certain resource (subject) has a property (predicate) with a certain value (object).

According to the CAP theorem it is impossible for a theoretically perfect working distributed data store to simultaneously provide more than two of the following three guarantees (Brewer, 2012):

- *Consistency* (*C*) – consistency of the data. As a result of every read of data, their latest version must be available;
- *Availability* (*A*) – it is accepted, as a guarantee, that every request to DBMS will receive a result. If the time period for the receiving of the result is too long, this may be understood as an impossibility for a response or unavailability of the data;
- *Partition tolerance* (*P*) – the ability to divide between numerous servers. The system can perform data distribution between few independent servers and it will continue to operate despite the failure of one of the servers or of the software.

The relational databases allow for the realization of the CA and CP (which is more difficult) combinations, but they are not intended for the AP combination. On the other hand, and especially with the internet services (the social networks included), the availability is quite important, and the lack of possibility for distribution between numerous servers, serves as a limiter of the growth of the service.

Distributed databases are those that can be divided and replicated. The innovatory NoSQL solutions can also be allocated to this type of databases. They provide the property "stability to separation". The main task of the NoSQL is to overcome the limitations of the relational model. The main problem of the relations becomes apparent when the quantities of information increase. The more the quantity increases, the more the time required for access to the entered information rises as well.

The characteristics of the NoSQL DBMS make them suitable for storage and operation of large quantities of data. Despite the many positive sides and advantages of the cloud technologies in the solutions of the companies that develop software, a certain amount of disadvantages is also present. In terms of security and confidentiality, it still cannot be certainly determined whether the service providers ensure adequate data protection and the business applications require a high level of reliability as they cannot bear crashes while operating. A prerequisite for acceptance would be the eventual creation of a reliability register.

2. Comparison of the characteristics of MongoDB and Redis

With the apparition of the social networks, the usage of the NoSQL database rapidly increases. The contemporary social networks grow constantly more and more and the data generated are enormous, volumetric and complicated. The big data are a new solution for this phenomenon. Compared to the traditional SQL technology, NoSQL has numerous positive sides – simplicity of projection, scalability of the data model, competitiveness, control, successiveness in the storage included. In addition, NoSQL has many types of databases – for example, the abovementioned contemporary networks MongoDB and Redis. For the social networks the large quantity of data is always related to more development and a day-to-day support service. The relational database is vertically scalable, which means that the data can be scaled only through hardware enhancement. In this manner, however, the technology becomes quite expensive for the processing of big data. NoSQL enables for the support of a large quantity of read – save operations and saving of joint objects, which contributes for the co-ordination of the data in the distribution system. This shows that the NoSQL database is a necessary choice for storage of large unstructured amounts of data, taking in consideration the quantity of data, the data access model and the characteristics of the social networks. NoSQL has its irreplaceable advantages. NoSQL is already widely used in the modern media business.

Two popular NoSQL databases MongoDB and Redis already boast a broad application within the contemporary social media that work with enormous quantities of data and possess a number of advantages. NoSQL plays an important role in the big data of the social networks. Fundamentally, the website for social networks consists of 5 main components: Pro, Post, Relations, Media, API (Zhang & Zhang, 2017).

Apart from the access to information, the users are also given the opportunity to share their thoughts and messages, to connect with one another, to share photos and videos. API provides the social networks with the opportunity to connect to other websites and applications. As the social network deals with big data, the time of usage cannot be too long. When extracting different data from different tables in a relational database, the execution of SQL requests containing JOIN operations, executing calculations for search and comparison – that would lead to extension of the time for data extraction is required; especially if the data are distributed to different servers.

One of the popular DBMS is MongoDB (from English - “humongous”- “enormous”) - a document-oriented system for databases, that works on different platforms. As it is classified as a non-relational database, MongoDB evades the traditional table, based on the relational structure and utilizes JSON-similar documents with dynamic schemes (MongoDB uses BSON format) which allows for the data integration within some applications to be easier and faster. This DBMS was developed by 10gen (now MongoDB Inc.) in October 2007 as part of a platform initially planned to be a service, but in 2009 the company focused on a model of software development with open-source and it began to offer paid support and other services. Since then, MongoDB is accepted as back-end software by a number of large web sites and services (Koleva, 2014).

Firstly, it does not require a lot of time in order for one to learn how to write code for MongoDB. Afterwards, the scaling is one of the significant advantages of the use of this system for management of databases.

When large quantities of data are present, they can be saved in numerous nodes, which is a good method to decrease the data in the social networks. The main structure of MongoDB is the document and it generates automatically a primary key (id) in order to establish unique identification for each document. The identification and the document are conceptually similar to the key value. MongoDB attempts to hold most of its data in the memory thus the simple requests require less time and it becomes faster. Its advantage in comparison with the other NoSQL is that it is easy to use and allows the users to benefit from the advantages of the cloud; with its horizontal scaling as well as its good compatibility with the various types of data that are used today (Asay, 2014).

MongoDB is used for a number of applications in the real world. However, the security restrictions vary depending on the occasions for its usage and the amount of the data sets used for the security characteristics that are presented in the two of the most popular NoSQL databases. The authors present security threats, which include lack of support for encryption of files of data (Gupta & Gugulothu, 2018).

Another database that is rather popular to use for social networks is Redis. It has an open source and uses key-value for storage. This non-relational database utilizes many types of data structures, instead of just one – for example interconnected lists, sets, sorted sets, hashes etc. The main purpose of its applications is to decrease the loading times of the websites. Redis saves all of the data in the RAM memory, which makes it quite useful as a cashing instrument. There is a certain restriction to the amount of data that can be saved, as the RAM memory does not allow the storage of large quantities of data. With the NoSQL databases the data can be saved on a drive.

Therefore, Redis is a database that saves data in the RAM memory of the machine with assured persistency. It is written in C and is funded by VMWare. The work logic of Redis as a key/value storage is similar to MemcacheDB (Zhang & Zhang, 2017).

Redis stores the data to the memory and periodically makes records of the changes of the disc. It is used for extraction of atomic values in a very fast manner. However, Redis may only be used when the data have lesser volume so that they can be saved to the memory of the server. The loss of a little quantity of information is acceptable at power shortages for example; but for situations when there is a greater loss, it is preferable not to use Redis. It allows the user to save lists in the value field, that are not arrays, but solely interconnected lists, which ensures the possibility even when many elements are present in a list, a new element to be added with LPUSH in a constant time. Arranged lists are also supported, as well as a SET object /a unique list of strings/.

There are a few popular applications of Redis in the social networks nowadays. The most common solution is MySQL or another relational database, which can serve as a main data depot.

Meanwhile, Redis might be devoted to work fields that require more flexible and effective kind of work. This relates to a real project of development, usually with a lesser number of data. Redis became popular among many social media and companies, some big names such as Twitter, Weibo, Pinterest and Snapchat and etc. (Zhang & Zhang, 2017).

Generally speaking, the social networks have two major requirements for the database system; the first one is the ability to save a large quantity of data, the second – a responsive server and user. In other words the usage of NoSQL is much more suitable for the processing of big data. The rapid pace in the increase of data of the social networks requires simplicity of the design work, better control and continuity in the storage process, things that the SQL technology cannot fully respond to, especially in terms of the storage of this enormous quantity of data.

In the social networks, the quantity of data is related to the everyday maintenance and developments. Taking into consideration the quantity of data, the model of access to the data and the characteristics of the social networks, NoSQL has its proven advantages and as it became clear from the exposition above – two of the popular NoSQL databases are MongoDB and Redis. Both systems of database management are widely applied in the contemporary social networks, because of their ability to work with the large quantities of data that are generated in these applications. One of the biggest disadvantages of the relational database is the longer exploitation period because while sharing and uploading videos and photos, the posting of statuses and the addition of new friends to our lists in the social networks adds up to the large quantity of data and delays the data extraction. The advantages of MongoDB compared to the relational database are firstly the time and secondly the scaling. The transition from one type of document to another, the addition or deletion of data is much easier, because here we talk about the property to divide the documents to different nodes.

Due to its model of data collection, MongoDB allows the users to take advantage of the benefits of the cloud in the infrastructure with its horizontal scalability and to utilize easily the various types of data, that are used today (Zhang & Zhang, 2017).

It can also be said about Redis that it is a flexible type of database. Typical for it is the assigning of a pending key, comparatively lesser saved data and a high speed.

It seems unrealistic to store large quantities of data in the memory. Redis helps us with a solution for optimization, by providing us with the key EXPIRE, which clears-up the current lifecycle of the key (Zhang & Zhang, 2017).

One of the users of Redis is the social network Pinterest, one of the popular media whose main activity is the sharing and publishing of photos and videos. The data stream there is very large and it requires fast updating of the images, especially for the mobile phones.

Redis works in the background and stores several types of lists for a given user *p,* a list of users, which *p* follows, a list of user walls, which *p* follows, a list of the followers of *p* (Zhang & Zhang, 2017). A summarized comparison of the main characteristics of MongoDB and Redis is presented in table 1.

Table 1. Comparison of MongoDB and Redis characteristics

DBMS NoSQL	**Type of software**	**Structuring and storing**	**Requests**	**Distribution of the information**
MongoDB	Document-oriented database	Document collections in JSON format. It utilizes a file system that allows for the eventual connection to many servers.	Specific fields of documents. There are various types of queries, supported by MongoDB. Among them are the field queries, range queries, as well as searches with regular expressions (regex).	Distributes the information in collections evenly between set of keys (shard).
Redis	Key-value type of database	Strings and more complex data structures – linked lists, sets, sorted sets, hash tables. It stores the data in the RAM memory, which requires small quantity data, that can be saved on a disk.	Redis might be used for storage and extraction of atomic values in an exceptionally fast manner.	Supports the establishment of clusters, which is automised by Redis sharding itself.

3. Related works on application of non-relational databases in social networks

The existing research on the application of the non-relational databases in the social networks may be divided in two groups depending on the functionality for which they are intended – storage of the data in the social network or their analysis. Apart from that, the non-relational databases may contribute to the successful implementation of additional functional characteristics of the social networks.

From the point of view of the non-relational databases that are used, it is important to mark that certain research interest was noted in regards to the application of semantic technologies for modelling and analysis of social networks (Bontcheva & Rout, 2014), (Mika, 2007).

3.1. Application of the non-relational databases for scalable storage of the data in a social network

The research of this group could be summarized as related to:

• Finding of problems and offering of solutions with the help of non-relational databases;

In (Faisal, Chaudhary & Mumthas, 2015) there are certain problems that are defined, that stem from the development of a social network with the aid of a relational database. The established problems are related to provision of scalability and productiveness of the requests. Certain solutions are offered through graph databases.

• Advantages and disadvantages of the application of non-relational databases for social networks.

Some general conclusions were drawn up, regarding the advantages and disadvantages of some specific non-relational databases from the point of view of their use for social networks – the document-oriented MongoDB (Kanoje, Powar & Mukhopadhyay, 2015) and the database from the key-value type Redis (Zhang & Zhang, 2017).

In the publication of (Mathew & Madhu Kumar, 2015) the non-relational databases used in popular social networks are studied and the characteristics of the different categories are compared (document, key-value, graph, column-based types).

The advantages of the non-relational databases are reviewed – in order to overcome some of the problems that come from the necessity of both storage and management of the big data for the development of applications for the social networks (Gašpar & Mabić, 2017). This work contains a detailed conclusion on why a great number of the popular social networks do not choose to switch completely to non-relational storage.

3.2. Application of the non-relational databases for analysis of the data in the social network.

In (Warchal, 2012) the opportunities of the graph database are studied and the existence of functionalities is affirmed for the graph database Neo4j when analyzing the social networks.

In order to conduct analysis of data in social networks through application of semantic technologies:

- RDF (*Resource Description Framework*) data presentation of a social network is proposed as well as the implementation of SPARQL (*SPARQL Protocol and RDF Query Language*) requests (Ereteo et al., 2009);
- Ontology is defined (Li, Yang, He & Ai, 2010);
- The architecture of the application of the RDF storage is described as well as the extraction of data with the SPARQL queries (Srivastav & Chauhan, 2017).

In (Angles, Prat-Pérez, Dominguez-Sal & Larriba-Pey, 2013) a study was conducted on the productivity of the different types of queries for extraction of useful information from social networks, executed in graph (Dex, Neo4j), RDF (RDF-3X) and relational databases (Virtuoso, PostgreSQL). Through an experiment it was confirmed that the graph databases, including RDF provide better productivity.

3.3. Application of the non-relational databases in the provision of additional functionality for the social network.

The prototypes and the ontologies of the social networks were developed with the aid of the semantic technologies in order to provide them with additional functionality:

- For social data portability, reuse and integration;

A prototype of a social network through the application of semantic technologies – in order to overcome the limitations, related to the social data portability, the reuse and the integration is presented in (Razmerita, Jusevičius & Firantas, 2009).

- For exchange, compatibility, conversion and extraction of data;

The prototype of a social network, described in (Martín & Gutierrez, 2009), fulfills the listed functionalities with the aid of RDF – for presentation of the data model and SPARQL for transformation and extraction of the data.

- Semantic compatibility.

In (Tserpes, Papadakis, Kardara & Papaoikonomou, 2012) a piece of ontology is defined – in order to achieve semantic compatibility – interoperability between different social networks.

3.4. Summary and perspectives

Frequently the non-relational database of choice for the support of a social network is not just one. On the one hand, the benefits and the characteristics of each database are used to serve to a specific aspect of the functionality of the social network.

On the other hand, this choice represents a way to overcome some of the disadvantages of the non-relational databases. As a general disadvantage of most of the databases we can point out the lack of standardization and the limited abilities for requests. The problems that stem from that are related with: data transfer, compatibility between the non-relational databases and dependency on the supplier. An attempt to solve these problems is the use of the RDF databases (AllegroGraph, 4store, Virtuoso, Stardog, GraphDB, etc.), which explains the noticeable research interest, aimed at the application of the semantic technologies for modeling and analysis of the social networks and represents the perspectives for a possible future research:

- Definition and enrichment of the RDF descriptions and ontologies;
- Provision of a semantic interoperability and integration of the data between the different social networks;
- Application of algorithms for data mining on a RDF graph – in order to extract useful knowledge from the data on the social network.

4. Conclusion

NoSQL will continue to have an important role for the future development of the social networks and will receive ever-greater attention from them. The academic circles, as well as the business appreciate highly the future of NoSQL. The purpose of NoSQL is not to fully replace SQL, but to support the relational database to work better and better. As it became clear in the aforementioned, the advantages of the NoSQL databases over the relational ones are the high level of scalability and their productivity. NoSQL can be developed on a completely new level. After all, the 'wheel of technology' goes only in one direction – and it is forward. The combination of big data and Cloud-based calculations is the latest technological challenge for the new technological world and the IT specialists. The general focus is on the creation of an allocated work environment for big data. In the last years, the cloud technologies and big data provoke great interest in the business as well as among the software-developers.

For the implementation of the mobile applications, for example, there is the necessity of centralization on the basis of the data in order to enable the data exchange between users that use different devices, or to improve the accessibility of the users' data for different devices and applications. For the maintenance of the data, a cloud-based service type BaaS (Backend as a Service) is used for the development. Most of the BaaS suppliers offer non-relational databases (Antonova & Valchev, 2015).

Big data represents a bulk of data, whose quantity and complexity turn their management and processing into a difficult task, when the traditional methods of processing and storage are at use. Although the term big data is often used in relation to the quantity of the data, it also bears a different meaning. Big data designates the technology that includes instruments and processes that the companies use to deal with the ever-increasing quantity of information. The progress of the informational and the communicational technologies, the digitization of the commerce and the distribution of the social networks are a prerequisite for the creation of large quantities of information in a rising pace. According to a recently conducted study, over 90% of the data worldwide were generated in the last 2 years. Various sensors, the websites of the social networks and other sources generate an enormous data flow – such data as digital photographs, videos, purchases, sales GPS coordinates etc.

Acknowledgment

A pre-print version of this article was available on ResearchGate at the following link: https://www.researchgate.net/publication/337186961_Application_of_non-relational_databases_in_the_social_networks
The current paper contains the final version of the article.

References

Angles, R., Prat-Pérez, A., Dominguez-Sal, D., & Larriba-Pey, J.L. (2013). Benchmarking database systems for social network applications. *Proceedings of the First International Workshop on Graph Data Management Experience and Systems - GRADES '13*. New York, U.S.A. doi:10.1145/2484425.2484440

Antonova, V. & Valchev, P. (2015). *Android mobile application for a personal library catalog in cloudy environments*. Scientific papers at the University of Rousse. Retrieved from: http://conf.uni-ruse.bg/bg/docs/cp15/3.2/3.2-28.pdf.

Asay, M. (2014). *MongoDB, Cassandra, and HBase - the three NoSQL databases to watch*. [Internet source]. Retrieved from: https://www.infoworld.com/article/2848722/mongodb-cassandra-hbase-three-nosql-databases-to-watch.html

Bontcheva, K. & Rout, D. (2014). Making sense of social media streams through semantics: A survey. *Semantic Web Journal*, *5*(5), 373-403.

Brewer, E. (2012). CAP twelve years later: How the "rules" have changed. *Computer*, *45*(2), 23-29. doi:10.1109/MC.2012.37

Cheresharov, S., & Krushkov, H. (2016). NoSQL Approaches in SQL Databases, *Scientific Conference "Innovative ICT in Business and Education: Future Trends, Applications and Implementation"*, Pamporovo, Retrieved from: http://fmi-plovdiv.org/GetResource?id=2519.

Ereteo, G., Gandon, F., Buffa, M. (2009). Semantic Social Network Analysis. *Web Science*, *WebSci'09*, pp.1–5. Athènes, Greece. Retrieved from: https://hal.inria.fr/inria-00378174/document

Faisal, A., Chaudhary, A., & Mumthas, S. (2015). *Role of graph databases in social networks*. Retrieved from: https://www.researchgate.net/publication/304462174_Role_of_graph_databases_in_social_networks (accessed 13 July 2019).

Gašpar, D. & Mabić, M. (2017). NoSQL databases as social networks storage systems. *ENTerprise REsearch InNOVAtion Conference*. Dubrovnik, Hrvatska. Retrieved from: https://www.researchgate.net/publication/319944521_NoSQL_Databases_as_Social_Networks_storage_systems

Gupta, S. & Gugulothu, N. (2018). Secure NoSQL for the social networking and e-commerce based bigdata applications deployed in cloud. *International Journal of Cloud Applications and Computing*, *8*(2), 113-129. doi:10.4018/IJCAC.2018040106.

Kanoje, S., Powar, V., Mukhopadhyay, D. (2015). Using MongoDB for social networking website: Deciphering the Pros and Cons. *IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems*. doi: 10.1109/ICIIECS.2015.7192924.

Klisarova-Belcheva, S. (2015). Implementation of Cloud Computing in Big Data Applications, *Management and Sustainable Development*, 52 (3), 51-

58. Retrieved from: http://oldweb.ltu.bg/jmsd/files/articles/52/52-08_S_Klisarova-Belcheva.pdf.

Koleva, P. (2014). *Open source database management systems. NoSQL databases.* Retrieved from: https://informatika.dokumentite.com/art/sistemi-za-upravlenie-na-bazi-ot-danni-s-otvoren-kod-nosql-bazi-ot-danni-/86605.

Li, T., Yang, H., He, J. & Ai, Y. (2010). A social network analysis methods based on ontology. *Conference Paper of the Third International Symposium on Knowledge Acquisition and Modeling* (pp.258-261). Wuhan, China. doi 10.1109/KAM.2010.5646196.

Martín, M. S. & Gutierrez, C. (2009). Representing, querying and transforming social networks with RDF/SPARQL. In: Aroyo L. et al. (eds), *The semantic web: Research and applications. Lecture notes in computer science*, (vol. 5554, pp. 293–307). Berlin, Germany: Springer.

Mathew, A. B. & Madhu Kumar, S. D. (2015). Analysis of Data Management and Query Handling in Social Networks using NoSQL Databases. *International Conference on Advances in Computing, Communications and Informatics* (pp. 800-806). Kochi, India. doi: 10.1109/ICACCI.2015.7275708.

Mika, P. (2007). *Social networks and the semantic web: Computing for human experience*. New York, U.S.A.: Springer.

Nedyalkov, I. (2017). *Models of web applications processing large data arrays in real time*. [Doctoral thesis]. University of National and World Economy, Sofia. Retrieved from: http://konkursi.unwe.bg/documents/585Avtoreferat-2017_05_13.pdf.

Razmerita, L., Jusevičius, M., & Firantas, R. (2009). New generation of social networks based on semantic web technologies: The importance of social data portability. *Workshop on Adaptation and Personalization for Web 2.0*. (pp.79-87). Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.352.8469.

Relational and nonrelative database (SQL/NoSQL), Retrieved from: https://help.superhosting.bg/sql-nosql-databases.html, 2019.

Srivastav, A. & Chauhan, A. (2017). Social network data retrieval using semantic technology. *Asian Journal of Pharmaceutical and Clinical Research*, *10*(13), 31-35. doi: 10.22159/ajpcr.2017.v10s1.19541.

Strozzi, C. (2017). *NoSQL relational database management system*, Retrieved from: http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/NoSQL/Home%20Page.

Tserpes, K., Papadakis, G.A., Kardara, M., & Papaoikonomou, A. (2012). An ontology for social networking sites interoperability. *Proceedings of* *the International Conference on Knowledge Engineering and Ontology Development* (pp. 245-250).doi: 10.5220/0004142202450250

Warchal, L. (2012). Using Neo4j graph database in social network analysis. *Studia Informatica*, *33*(2A), 271-279.

Zhang, Y. & Zhang, C. (2017). *Applications of NoSQL database in modern social media.* Retrieved from: http://www.cs.rochester.edu/courses/261/fall2017/termpaper/submissions/12/Paper.pdf (accessed 13 July 2019).

Biodata

Miroslav DECHEV was born in Veliko Tarnovo on 4.11.1979. He obtained his bachelor's degree in Book Publishing (2005), master's degree in Journalism; he is a PhD student in Computer Sciences (2018) at the "Faculty of Mathematics and Informatics" of the Veliko Tarnovo University "St. St. Cyril & Methodius" in Veliko Tarnovo, Bulgaria. He currently works as a journalist at a regional media. His research interest includes different aspects of the information security, the artificial intellect, the Internet of Things, the cloud technologies, the social networks. He is the author of over 5000 interviews and two books /fiction/.

Miroslav GALABOV was born in Veliko Tarnovo. He received his Ph.D. degree in Automation Systems for Processing of Information and Control in 1999. Since 2002 he serves as an assistant professor; In 2005 he became associate professor and in 2015 - professor in the Computer Systems and Technologies Department of the St. Cyril and St. Methodius University of Veliko Tarnovo. He is the author of 10 textbooks, and over 50 papers. His current interests are in signal processing, 3D technologies and multimedia.

Tsvetanka GEORGIEVA-TRIFONOVA received her MSc degree in Mathematics and Informatics in 1997 and her PhD degree in Computer Science in 2009 – both from the University of Veliko Tarnovo, Bulgaria. Currently she serves as an Associate Professor at the University of Veliko Tarnovo, Bulgaria and teaches Databases, Information Systems Modeling, Data Warehousing and Mining, and Artificial Intelligence. Her research interests include data mining, text mining, non-relational databases, and information systems.