NoSQL Databases as Social Networks Storage Systems

Dražena Gašpar
University of Mostar – Faculty of Economics, Bosnia and Herzegovina
Mirela Mabić
University of Mostar – Faculty of Economics, Bosnia and Herzegovina

Abstract

The paper presents analysis of the storage systems used by social network sites. Namely, the social networks are one of the main driving forces behind the NoSQL database development. Facebook and Twitter were, together with other the Big Data players like Google and Amazon, first faced with the limitations of relational databases in solving their needs related to unprecedented transaction volumes, expectations of low-latency access to massive datasets, and nearly perfect service availability while operating in an unreliable environment. The first NoSQL databases arose as internal solutions created out of necessity, and not with the intention to abandon relational databases. But the main question is if, after more than ten years of development, NoSQL databases proved that they could be valuable storage solutions for social networks' data. The paper shows that there is still a lot of room for improvement in the use of NoSQL in social networks and provides some suggestions on how NoSQL databases can bring additional value to social network sites.

Keywords: NoSQL, social networks, social network site, social networks analysis
JEL classification: O39

Introduction

The development of NoSQL databases is tightly coupled with the Big Data phenomenon. This phenomenon is related to extensive requests for the storage and management of enormous quantity of very complex, dynamic, evolving, distributed and heterogeneous data from different sources and platforms. Since relational databases did not prove that can adequately answer to Big Data challenges, new databases – NoSQL – had to be developed.

The first use of the term NoSQL in the present sense of the word was registered in 2009 as the name of a meetup organized by Johan Oskarsson. The original term NoSQL Meetup was about open-source, distributed, nonrelational databases. The original idea was only to name the meetup, but the term NoSQL unexpectedly quickly spread up and became generally adopted by the IT community to designate the new trend in the development of databases.

Although the term NoSQL has become generally accepted, no universally-adopted definition exists for it. The NoSQL archive defines NoSQL databases as "non-relational, distributed, open-source and horizontal scalable" (NoSQL, 2017). Essentially, NoSQL is not about abandoning some software and hardware database architectures, but it is about a specific technology because NoSQL solutions are based on a different set of objectives and hardware models than it was the case with relational databases.

The social network sites (SNSs), like Facebook, Twitter, Youtube, were among the first to face Big Data challenges. SNSs are web-based services that allow individuals

to construct a public or semi-public profile within a bounded system, to articulate a list of other users with whom they share a connection, and to view and traverse their list of connections and those made by others within the system (Boyd et al., 2008). They are a special-purpose software (or social media tool) designed to facilitate the creation and maintenance of social relations (Khan, 2015). Facebook, Twitter, LinkedIn and Google+ are some of the most popular social network sites.

SNSs provides to users a possibility to create their own social network in digital environment. A social network is a group of nodes and links formed by social entities where nodes can represent social entities such as people and organizations. Links represent their relationships, such as friendship and trade relations. Social networks can exist both in the real and online worlds (Khan, 2015). The subject of this paper is online social networks. Online social networks enable simple and fast forming, expanding, and maintaining social relations and they very quick attract a huge number of users. The explosive growth of SNSs users as a consequence have the rapid and enormous growth of the data SNSs operate with and they very soon faced the challenges brought by Big Data, primarily unprecedented transaction volumes, expectations of low-latency access to massive datasets, and nearly perfect service availability while operating in an unreliable environment. In order to keep and further develop the level of their services, SNSs had to find answer to those challenges. Since relational databases could not offer the adequate solution, SNSs, like other big Internet companies (Google, Yahoo, Amazon) had to find their own ways to cope with Big Data challenges.

The paper presents the results of analysis of the storage systems used for the five top social network sites.

NoSQL databases

NoSQL is an umbrella term related to numerous databases. NoSQL databases differ in architecture and purpose. For NoSQL supporters it is natural because they believe that a universal solution which could apply to all data types, volumes, and objectives does not exist. Despite differences, NoSQL databases have following common features (McCreary et al., 2014):

- **Tables are not basic structures.** NoSQL databases store and work with data in different formats (key-values, graphs, column family, documents, and tables).
- **There are no joins.** NoSQL databases allow data processing through simple interfaces, without the need for joins.
- **They are schema-free.** NoSQL databases allow data manipulation without the need for their previous modelling (e.g., entity-relational model).
- **There are many processors.** NoSQL databases allow storage on multiple processors while keeping high levels of performance.
- **They use shared-nothing commodity computers.** Most NoSQL databases are based on hardware architecture consisting of low-cost commodity processors that have separate random access memory (RAM) and disk.
- **They support linear scalability.** The addition of a larger number of processors is manifested in a consistent increase in performance.
- **Innovation** NoSQL databases offer several options to store and process data, including SQL. NoSQL supporters advocate an inclusive approach, aware that there is not only one solution to any problem. For them, NoSQL means "not only SQL."

Today, different Big Data challenges are, with more or less success, resolved with different NoSQL database architectures. According to that, NoSQL databases can

be classified into four basic categories, each resolving a different type of big data problems:

- Key-Value.
- Column-Family.
- Document.
- Graph.

The key-value type of NoSQL databases uses a key to locate a value (e.g., traditional data, BLOBs – Binary Large OBjects, files) in simple, standalone tables, known as hash tables. In this case, searches are performing against keys, not values, and they are restricted to exact matches. Some of the best-known key-value stores are Amazon DynamoDB, Berkeley DB, Redis and Riak.

Column-family or column-oriented NoSQL databases have been named for their design with data stored in columns. In contrast, a row-oriented database (relational database) keeps information about a row together. Some of the well-known column-family stores are Google BigTable, Apache Cassandra, HBase, Hypertable and Amazon SimpleDB.

Document NoSQL stores have been designed to store and manage documents. The documents are encoded in standard data exchange formats, including XML, JSON (JavaScript Object Notation), and BSON (Binary JSON). Some of the best-known document stores are MongoDB, CouchDB, Terrastore, and RavenDB.

Graph NoSQL databases excel at dealing with highly interconnected data. They focus on relationships, rather than data. A graph store consists of nodes and relationships between nodes. Both nodes and relationships have properties (or key-value pairs) to store data. Some of the better-known graph stores are Neo4J, Infinite Graph, and FlockDB.

Social networks sites and data storage

Today there is a number of different SNSs, although one of the most known and with the most users worldwide is still Facebook. To better understand SNSs data storage challenges in Table 1 are presented the top five SNSs according to estimated unique monthly visitor (eBizMBA, 2017), together with data from http://www.internet livestats.com/ about active users and uploads on a daily basis.

Table 1
The top five social network sites

No	Social network site	Estimated unique monthly visitors (eBizMBA, 2017)	Active user and uploads on a daily basis (http://www.internetlivestats.com/)
1	Facebook	1,500,000,000	1,900,000,000 Active users
2	YouTube	1,499,000,000	5,000,000,000 Videos
3	Twitter	400,000,000	550,000,000 Tweets 310,000,000 Active users
4	Instagram	275,000,000	56,000,000 Photos uploaded
5	LinkedIn	250,000,000	n/a

Source: (eBizMBA, 2017; http://www.internet livestats.com/)

Table 1 shows complexity related to number of SNSs users, the quantity and variety of data that SNSs have to process (videos, text and audio).

The SNSs listed in Table 1 were analyzed related to data storage systems they used for fulfilment of the requirements and expectations of hundreds of millions users on the daily basis.

At the beginning of year 2004 Mark Zuckerberg register thefacebook.com domain and started the era of Facebook. Very soon Facebook became a hot topic and by the end of 2004 it had over one million of registered members. From the very beginning, Facebook relied on relational MySQL database as data storage. However, Facebook's engineers, inspired by Google's paper about Google BigTable NoSQL database, developed Cassandra, column-family store, Facebook's NoSQL database. In 2008 Facebook open sourced Cassandra, but it was not moving it forward. Although Cassandra gained attraction outside Facebook, Facebook had not built community around it, moreover it decided to replace Cassandra with HBase for its messaging system (Finley, 2014). Over the last years Facebook has been continuously worked on improving HBase. The result of that work is HydraBase that developed with aim to avoid HBase limitations (Fong et al., 2014). Facebook decided to merge its internal HBase branch with the current HBase Open Source branch. With this merge, Facebook is also planning to integrate HydraBase into Open Source HBase. (Facebook, 2014). Today, Facebook still primarily uses MySQL for structured data storage such as wall posts, user information, timeline and similar, but it also use HBase, MongoDB, Memcached databases. Facebook developers have added a variety of other systems to make it truly web scalable over their 1.5 billion users (CIMS, 2015).

YouTube era officially started in December 2005, although the first video is uploaded in April 2005 by Jawed Karim, one of the YouTube founders. But in February 2006 YouTube had 20000 uploads on a daily basis (Telegraph, 2010). Related to databases, YouTube, similar to Facebook started with MySQL databases. In the next years YouTube developed a custom tool on top of MySQL database named Vitess. Vitess has been serving all YouTube database traffic since 2011 (Kumar, 2017).

Twitter story begun with the first tweet in March 2006 sent by Jack Dorsey, one of the Twitter co-founders. But in 2007 the huge explosion of Twitter usage was noticed (MacArthur, 2016). Over the years, Twitter has used and made significant contributions to many open source databases. But the result was not satisfactory. Far too much time was spent firefighting production systems to meet the performance expectations of Twitter various products, and standing up new storage capacity for a use case involved too much manual work and process. Twitter experience developing and operating production storage at Twitter's scale made it clear that the situation was simply not sustainable. So in Twitter decided to build by themselves Twitter's next generation distributed database called Manhattan. Today Twitter uses Manhattan distributes database as one of the primary data stores serving Tweets, Direct Messages and advertisements (Kotian, 2016).

Instagram is relatively new SNS. It is the first photo social platform launched in October 2010. From the very beginning Instagram had explosive growth. It had one million users just two months after its launch and it kept on growing ever since (Desreumaux, 2014). In order to ensure adequate data storage for such huge growth Instagram combines relational database – PostgresSQL – with NoSQL databases Redis and Cassandra; taking advantage of each tech's strength per use case (DataStax, 2016).

LinkedIn was launched in 2003. Growth was slow at first, but accelerated with the introduction of address book uploads in late 2003 (LinkedIn, 2017). LinkedIn has continuously worked on building data infrastructure that enables long term growth.

As data storage solutions LinkedIn use Espresso and Voldemort databases. Espresso is LinkedIn's online, distributed, fault-tolerant NoSQL database that currently powers approximately 30 LinkedIn applications including Member Profile, InMail (LinkedIn's member-to-member messaging system), portions of the Homepage and mobile applications, etc. (Auradkar, 2015). One of the many projects LinkedIn has contributed to open source is Project Voldemort, a distributed key-value storage system. LinkedIn developed a massive offline workflow using Hadoop and Voldemort data store to precompute data insights like People You May Know, Similar profiles, Notable Alumni, and profile browse maps (Clemm, 2015).

Discussion: NoSQL databases as SNSs storage system

Analysis of a top five SNSs and their storage systems showed that most of them still extensively used relational database – mainly MySQL. However, all analyzed SNSs are using NoSQL databases at some extent. Most of them developed their NoSQL databases (Facebook, Twitter) and intensively contributing development of open source NoSQL databases (HBase, Cassandra, Voldemort). But the question is why SNSs did not make complete transitions to NoSQL databases which were primarily developed as answer to Big Data problems with which huge web sites were faced? The answer to this question probably lies in the following characteristics of NoSQL database:

- Huge number of diverse NoSQL databases. Although after ten years, NoSQL databases are becoming mature, most important features are still not implemented or tested in real environment.
- Lack of standardization. Since NoSQL is umbrella term for many diverse products, the design, data store, query languages and other features of NoSQL databases vary considerably between different NoSQL products. The consequence is that the learning curve for NoSQL databases is slower, since a developer who is familiar with one type of NoSQL database is not always prepared to work with a different one. That is serious barrier to wider NoSQL adoption.
- Resolve demands of the Web 2.0 applications providing very fast and efficient "insert-read-update-delete" cycle. But the problem arises if business intelligence and analytics tools have to be used.
- Vendors support on a global scale. Most NoSQL databases are open source, with just few firms handling the support, so they lack the credibility that established relational database vendors (Oracle, Microsoft, IBM) enjoy (Richards, 2015).

Regardless of NoSQL database disadvantages, there are two main reasons crucial for the growth of NoSQL popularity, and that reasons are forcing the big relational database vendors (Oracle, Microsoft, IBM) to go in the direction of convergence toward NoSQL databases (Kernochan, 2016):

1. *Scaling to handle massive numbers of transactions*. NoSQL databases proved to be good in this case because they, opposed to relational databases, relaxed strict consistency and avoid use of tables for storing the data.
2. *Delivering "almost real time" performance* for the large amounts of distributed transactions (particularly writes) associated with Big Data, such as the Internet of Things (IoT). Here, NoSQL databases are well suited to scaling almost real time access to data. One of the key benefits of NoSQL databases is that they allow users to tune the tradeoff between scalability and data quality dynamically.

The analysis of databases used by top SNSs showed that these SNSs used the both, relational and NoSQL databases, for resolving different user requirements. Most of them use NoSQL database as a complement to relational database that is still more powerful in deeper data analytics.

Conclusion

The presented analysis of databases used by the most popular social network sites (Facebook, YouTube, Twitter, Instagram and LinkedIn) showed that most of them use combination of different databases, both relational and NoSQL, in order to resolve diverse needs and requirements of their users. The main motif behind NoSQL database development is finding solutions for Big Data challenges (volume, velocity and variety of data). NoSQL databases answered to those challenges through distributed, cluster-oriented, horizontally scalable and lately tunable consistency features. But when data querying and analyzing is in question, NoSQL databases with use of different programming languages and APIs turned out to be inadequate solution, opposed to standardized SQL (Structured Query Language) used by relational databases. In meantime, the largest database management vendors (Oracle, Microsoft, IBM) realized that NoSQL databases bring some innovative and good solutions to the problems they had been facing for years (Big Data, high availability, distribution). They innovated and expanded their relational databases and thus brought them closer to NoSQL databases. On the other hand, NoSQL database vendors are developing support for SQL in order to make data analysis easier for their users.

The future will prove is the bridging the gap between relational and NoSQL databases the path that can lead to new database evolution. However, that new databases should be able to provide support for different and often opposite users requirements by enabling combinations of both approaches through tunable and configurable capabilities that will give the users the opportunity to use databases on the way that best suits their needs.

References

1. Auradkar, A. (2015), "Introducing Espresso - LinkedIn's hot new distributed document store", available at: https://engineering.linkedin.com/espresso/introducing-espresso-linkedins-hot-new-distributed-document-store (02 March 2017)
2. Boyd, D.M., Ellison, N.B. (2008), "Social Network Sites: Definition, History, and Scholarship", Journal of Computer-Mediated Communication Vol. 13, pp. 210–230.
3. CIMS (2015), "What database actually FACEBOOK uses?", available at:
4. https://www.linkedin.com/pulse/what-database-actually-facebook-uses-e-courts-and-e-filing-software (06 March 2017)
5. Clemm, J. (2015), "A Brief History of Scaling LinkedIn", available at: https://engineering.linkedin.com/architecture/brief-history-scaling-linkedin (02 March 2017)
6. DataStax (2016), "Go with proven and solid technologies when you can", available at: https://medium.com/@DataStax/instagram-engineerings-3-rules-to-a-scalable-cloud-application-architecture-c44afed31406 (02 March 2017)
7. Desreumaux, G. (2014), "The Complete History of Instagram", available at: http://wersm.com/the-complete-history-of-instagram/ (02 March 2017)
8. eBizMBA (2017), "Top 15 Most Popular Social Networking Sites: April 2017", available at: http://www.ebizmba.com/articles/social-networking-websites (16 April 2017)

9. Finley, K. (2014), "Out in the OpeN: The Abandoned Facebook Tech that now Helps Power Apple", available at: https://www.wired.com/2014/08/datastax/ (22 March 2017)
10. Fong, Z., Shroff, R. (2014), "HydraBase – The evolution of HBase@Facebook", available at: https://code.facebook.com/posts/321111638043166/hydrabase-the-evolution-of-hbase-facebook/ (05 March 2017)
11. Khan, G.F. (20015), Seven Layers of Social Media Analytics: Mining Business Insights from Social Media Text, Actions, Networks, Hyperlinks, Apps, Search Engine, and Location Data, Kindle Edition.
12. Kernochan, W. (2016), "Using NoSQL Databases to Handle Fast Data", available at: http://www.enterpriseappstoday.com/print/datamanagement/usingnosqldatabasestohandlefastdata.html (17 March 2017)
13. Kotian, A. (2016), "Manhattan software deployments: how we deploy Twitter's large scale distributed database", available at: https://blog.twitter.com/2016/manhattan-software-deployments-how-we-deploy-twitter-s-large-scale-distributed-database (10 March 2017)
14. Kumar, R. (2017), "What database is YouTube using to store its videos?", available at: https://www.quora.com/What-database-is-YouTube-using-to-store-its-videos (10 March 2017)
15. MacArthur, A. (2016), "The Real History of Twitter, In Brief", available at: https://www.lifewire.com/history-of-twitter-3288854 (10 March 2017)
17. McCreary, D., Kelly, A. (2014), Making Sense of NoSQL: A Guide for Managers and the Rest of Us, Manning Publications Co, USA.
18. NoSQL (2017), http://nosql-database.org/ (15 December 2016)
19. Richards, J. (2015), "Advantages and Disadvantages of NoSQL databases – what you should know", available at: http://www.hadoop360.com/blog/advantages-and-disadvantages-of-nosql-databases-what-you-should-k (29 January 2017)
20. The Telegraph (2010), "YouTube: a history", available at: http://www.telegraph.co.uk/finance/newsbysector/mediatechnologyandtelecoms/digital-media/7596636/YouTube-a-history.html (28 February 2017)
21. http://www.internetlivestats.com/ (20 April 2017)

About the authors

Dražena Gašpar is full time professor of Database Systems and Business Information Systems at the Faculty of Economics, University of Mostar. Her research interests include databases, data warehouse, business information systems and software application in business and education. She is co-founder of a "Hera" software company in Mostar and has almost two decades of experience in developing and implementing business information systems. Author can be contacted at **drazena.gaspar@sve-mo.ba**.

Mirela Mabić works at the Faculty of Economics, University of Mostar, as an assistant at the Department for Business Informatics. Her research interests include business information systems, the practical application of software and web technologies both in business and in education, quality of higher education and applied statistics. Author can be contacted at **mirela.mabic@sve-mo.ba**.