JOURNAL OF SCIENCE OF THE MILITARY ACADEMY OF LAND FORCES

Volume 49 Number 3 (185) 2017

DOI: 10.5604/01.3001.0010.5128

NOSQL DATABASES AS A DATA WAREHOUSE FOR DECISION SUPPORT SYSTEMS

Jarosław KURPANIK[*]

[*] Faculty of Informatics and Communication, University of Economics in Katowice
jaroslaw.kurpanik@ue.katowice.pl

Received on 21st March ; accepted after revision June 2017

Abstract:

Nowadays, to some extent decision support systems are forced to base their operation on large data warehouses whose analysis is difficult and time consuming. This is why where data are stored becomes vital. The use of an efficient and productive data warehouse for this purpose can significantly improve application/system operation. Currently one of the most common solutions used in Big Data storage and quick processing are non-relational databases NoSQL. They are a relatively new solution, however, their development is dynamic and their market share is increased on a daily basis, which means that it worth investigating what they offer.

***Keywords**:*

Big Data, NoSQL, ACID, CAP

INTRODUCTION

These days decision making processes are supported by specialised computer systems which on the basis of the analysis of collected data indicate the best solution to a given problem. This process should be quick, particularly in the case when decisions are made in emergency situations, it is also vital that the proposed solutions are accurate. One of the factors which influences response/decision accuracy is the amount of data on the basis of which decisions are made – the larger the amount of data, the higher the conclusion accuracy. Such an approach practically imposes using Big Data by such systems [7].

However, to expedite the data searching process, it is possible to increase the computing power of servers or use software solutions allowing for quick data record and readout. The software based acceleration of data readout and record in decision supporting systems can be obtained thanks to the use of quick access databases such as NoSQL which, thanks to their properties, are a perfect solution for Big Data storage and analysis.

The goals of this study is the theoretical and practical comparison of NoSQL type databases with their relational competitors. The theoretical comparison was conducted on the basis of literature studies and the practical one was based on the comparison of the productivity of the record and readout of the representatives of various types of databases.

1. BIG DATA

Big Data is a relative notion because it describes situations when the size, speed and variety exceed the computational ability and capacity of an organisation, which has an adverse effect on a decision making process [8]. Although the data processing and analysis are complex processes, data are collected whenever there is a possibility of revealing new information or acquiring knowledge, as in today's world it is data access to create a competitive advantage in nearly every area of life [10]. This is why it is worth taking considerable interest in the ways they are processed, and NoSQL databases undoubtedly are solutions which not only can be, but have to be taken into account in such cases.

2. NOSQL

The term NoSQL was sued for the first time in the late 90's with reference to a relational database, developed by Carlo Strozzi, which did not use the SQL query language. The notion has remained in use until today, however, its significance changed in 2009 during a conference in San Francisco organised by Johan Oskarsson where it was used to refer to various non-relational database management systems, used mainly for the analysis and mining large, not necessarily orderly databases, which differed from the traditional ones.

Today the term NoSQL can be rephrased as "Not only SQL" and is used to refer to non-relational databases used to store large amounts of data on distributed servers and in data centres. Such solutions need a model which does not require any schema, avoids connections and usually is scaled horizontally. Although the model profile is slightly different from the commonly used relational approach, this type of databases are perfect for data storage in all forms: orderly, partly orderly and disorderly [3].

NoSQL databases are used mainly for Big Data storage because they ensure a high level of flexibility, relatively small delay during readout and the high productivity of data recording which can be achieved thanks to the fact that they operate on the basis of cluster environment [5].

3. COMPARISON OF ACID AND CAP

Data in NoSQL databases do not have to be orderly and connected in relationships, this is why they cannot be described and evaluated using ACID (Atomicity Consistency Isolation Durability), a well-known set of properties used in relational databases, which ensures the correctness of transactions.

Every relational database using the ACID model ensures:

- Atomicity – every transaction must be completed;
- Consistency – certainty that a transaction will not interfere with data integrity;
- Isolation – certainty that transactions are separated from one another and will not modify the same data at the same time;
- Durability – after an unplanned stoppage a database can restart and offer access to consistent, up-to-date data.

These properties are not necessary in non-relational databases operating on the basis of the CAP theorem [4], also called Brewer's theorem after the name of its creator Eric Brewer who presented its fundamentals in 2000. Its validity was confirmed by Seth Gilbert and Nancy Lynch in 2002.

The CAP theorem (Consistency Availability Partition tolerance) states that none of distributed systems cannot ensure consistency, high availability and data partition at the same time, only two of these criteria can be met simultaneously [1].

In the comparison of the CAP theory with ACID properties one should remember that the common notions of consistency and data availability have different meanings in these two cases. In the CAP theorem they carry the following meanings:

- Consistency – all cluster nodes have access to the same data at the same time;
- Availability – every request is guaranteed to receive a response regardless if it is successful or not;
- Partition tolerance – a system continues operation despite the faults of one or a few cluster nodes [9].

Therefore, based on the statement from the CAP theorem according to which the management systems of non-relational databases can implement only two out of three available guarantees, the following division of these systems becomes natural (see Fig. 1):

- CA (consistency and availability);
- AP (availability and partition tolerance);
- CP (consistency and partition tolerance) [6].

Table 1 presents similarities and differences between relational (RDBMS) and non-relational (NoSQL, nonRDBMS) database management systems.

Table 1. Comparison of ACID and CAP properties

RDBMS	nonRDBMS
ACID	CAP
Constructed on the basis of schema	No schema

RDBMS	nonRDBMS
Vertical scaling by increasing computational power per unit	Horizontal scaling by increasing the number of nodes per cluster
Use of joins in SQL queries	No joins
Permitted operations: CRUD (create, read, update, delete)	Permitted operations: CRUD (create, read, update, delete)
Recommended for small databases, below 1TB	Recommended for big databases, above 1TB

Source: Author's own study on the basis of [4]

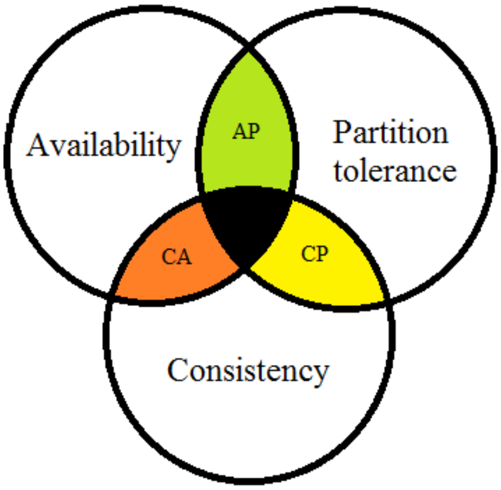

Fig. 1. CAP theorem

Source: Author's own study on the basis of [2]

4. NOSQL CLASSIFICATION

NoSQL databases are divided with regard to the implemented data model which determines the logical organization of data in a database – it defines the way of acquiring and updating data. Currently the most common classification of this type is a four-segment classification in which the following bases can be distinguished: column-oriented, key-value, document-oriented and graph [1].

4.1. Column-oriented databases

These databases store data in column families which have the form of a row with a key for each of them. Column families are groups of related data which usually are extracted together. This type of databases offer perfect storage capacity for systems requiring quick recording. The most well-known representatives of this group are such database environments as: Cassandra, HBase, Hypertable[1,5,6].

4.2. Key-value databases

Key-value databases are the simplest with respect to data availability because they are composed of hash tables with only two columns: key and value. The key is a kind of value ID identifier while the value is a blob type field in which various types of data are stored. The most common representatives of this group are such database environments as: Redis, BerkleyDB, LevelDB[1,5,6].

4.3. Document-oriented databases

This type of databases store and return XML, BSON, JSON documents which are hierarchically distributed in server memory creating tree structures, these in turn can be composed of collections, maps and scalar values. The most common representatives of this group are such database environments as: CouchDB, MongoDB [1,5,6].

4.4. Graph databases

Graph databases are constructed on the basis of graphs containing nodes and edges. The most common representatives of this group are such database environments as: HyperGraph, InfoGrid, Neo4 [1,5,6].

5. CASE STUDY

In the case study the data recording and readout speed in relational and non-relational technologies used in databases were compared. The selected representative of relational databases was an Oracle database, Oracle12c instance, while it competitor was a non-relational database made by Apache – Cassandra. Both instances were installed on Linux CentOS7 operating system which was implemented on the virtual machines of Oracle Virtual Box Manager. The created machines were assigned the following hardware: RAM - 2GB, CPU – 1 core 2.6 GHz.

For the needs of the tests tables (Table 2) were made in both databases, they were composed of three columns: *ID, name and surname.* Various numbers of rows were uploaded to them using *insert into* commands – 50 000, 100 000, 200 000, 400000. The values in the ID column were unique values in the whole table.

Table 2. Table scripts used in productivity tests

Cassandra	Oracle
CREATE TABLE test.test(id int, name varchar, surname varchar, PRIMARY KEY(id));	CREATE TABLE test.test(id number(10), name varchar2(50), surname varchar2(50), PRIMARY KEY(id));

Source: Author's own study

Chart 1 presents data recording time in newly created tables for a changeable number of rows.

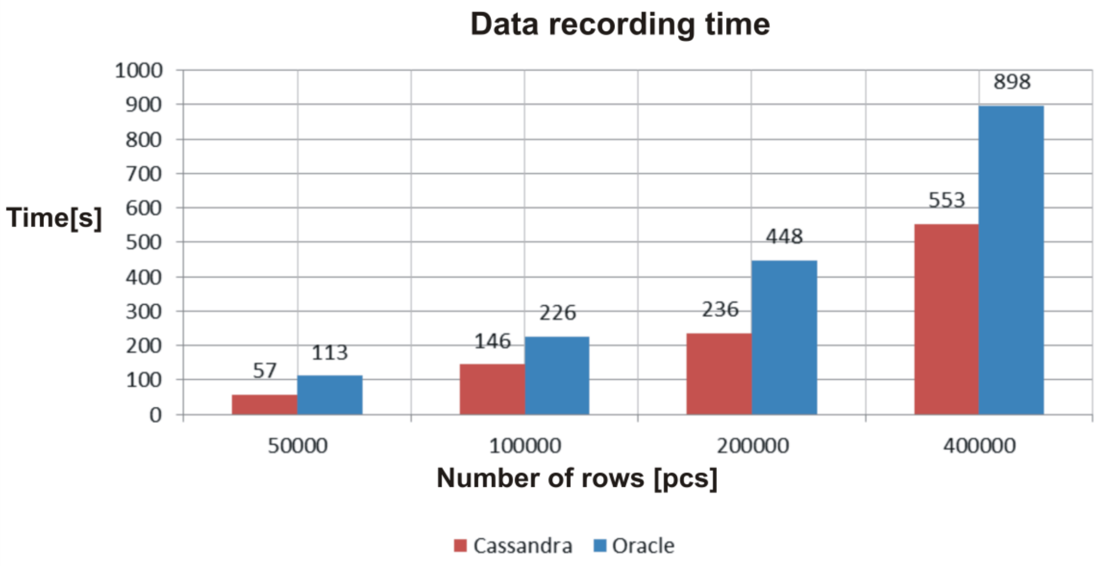

Fig. 1. Recording efficiency comparison between Oracle12c and Cassandra

Source: Author's own study

On the basis of the conducted experiment one can conclude that data recording times in each of the presented cases were twice shorter for non-relational Cassandra.

Additionally, readout productivity tests were conducted for each set of rows. The results are presented in Chart 2. In the test a simple query was made on each of the presented sets:
*select * from t800c where id=45358;*

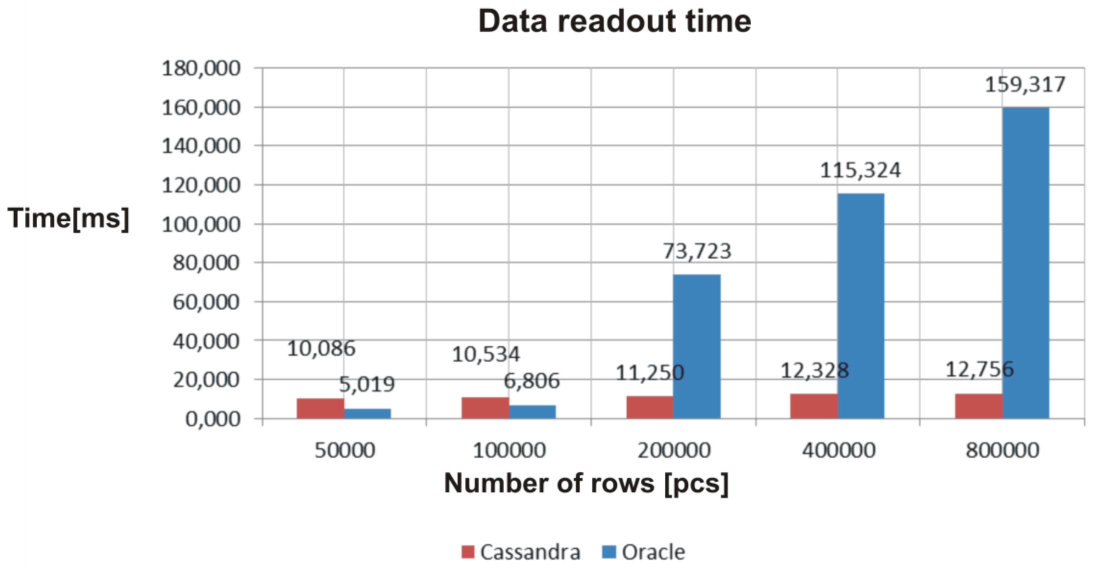

Fig. 2. Readout efficiency comparison between Oracle12c and Cassandra

Source: Author's own study

The obtained results are not as explicit as in the case of the recording time because for two smallest sets the relational database turned out to have a better result, while for larger sets with over 200 thousand rows non-relating Cassandra again proved to be better.

CONCLUSIONS

The article describes NoSQL databases which due to their productivity should be used as warehouses for systems operating on Big Data. Undoubtedly decision support systems belong to this group as in their case operation speed and accuracy are deciding factors when it comes to gaining a competitive advantage, and in the case of the military they may influence such important issues as the element of surprise and a preemptive action.

With reference to the conducted experiment, it can be stated that non-relational solutions allow to accelerate the recording and readout process even in the case of relatively small sets.

This article is an introduction to the research on the productivity of NoSQL databases which is already in progress. At the beginning the productivity was tested for solutions operating with one server so as to check what acceleration can be obtained if the same solution is implemented in cluster environments in the future, and also which of the above mentioned databases would be best for decision support processes.

BIBLIOGRAPHY

1. He Ch. (2015), *Survey on NoSQL Database Technology*, Journal of Applied Science and Engineering Innovation, Information Technology Service Center of Hexi University, Gansu Zhangye, China, vol.2, no 2, 50-54.
2. Kirti, Maan P. (2015), *Database for Unstructured, Semistructured data-NoSQL*, International Journal of Advanced Research in Computer Engineering & Technology, Vol.4 Issue 2, 466-469.
3. Ming Ch. Wu., Huang Y. F., Lee J. (2015), *Comparisons Between MongoDB and MS-SQL Databases on the TWC Website*, American Journal of Software Engineering and Applications, USA, 4(2), 35-41.
4. Nataraja Sekhar G., Saritha S. S. J., Penchalaiah C. (2015), *HBase Performance Testing On Multi-node Cluster Setup*, International Journal Of Engineering And Computer Science, Mandsaur, India, Vol. 4, Issue 4, April, 11272-11278.
5. Pańskowska M., Kurpanik J., NoSQL Problem Literature Reviews, *Social Media & Creativity Support Systems* pod red. Pańkowska M., Palonka J., UE Katowice, Katowice, 2015.
6. Sadalage P., Fowler M. (2013), NoSQL Distilled: *A Brief Guide to the Emerging World of Polyglot Persistence*, Pearson Education – Addison Wesley, Boston.
7. Shibata T., Kurachi Y. (2015), Big Data Analysis Solution for Driving Innovation in Onsite Decision Making, Fujitsu Science Technology Journal, vol.51, no 2, 33-41.

8. Tinkhede S. A., Deshpande S. P. (2015), *Big Data - The Vast Growing Technology with its Challenges and Solutions*, International Journal of Computer Science and Mobile Applications, vol.3, Issue 1, 33-38.
9. [Online]. [access: 29.09.2015]. Available on the Internet: https://en.wikipedia.org/wiki/CAP_theorem.
10. [Online]. [access: 29.09.2015]. Available on the Internet: https://pl.wikipedia.org/wiki/Big_data.

BIOGRAPHICAL NOTE

Jarosław KURPANIK – scientific and didactic employee at the Faculty of Informatics and Communication, University of Economics in Katowice, and a member of the Economic IT Science Association. His interest cover such scientific and professional areas as databases and operating systems. In addition to this, he is interested in water rescue services and crisis management.

HOW TO CITE THIS PAPER

Kurpanik J., (2017)., NOSQL databases as a data warehouse for decision support systems. *Zeszyty Naukowe Wyższa Szkoła Oficerska Wojsk Lądowych im. gen. Tadeusza Kosciuszko Journal of Science of the gen. Tadeusz Kosciuszko Military Academy of Land Forces,* 49 (3), p. 124-131, DOI: 10.5604/01.3001.0010.5128