International Electronic Journal of Pure and Applied Mathematics

Volume 10 No. 1 2016, 41-51
ISSN: 1314-0744
url: http://www.e.ijpam.eu
doi: http://dx.doi.org/10.12732/iejpam.v10i1.4

BIG DATA – SQL VS NOSQL

Petar Halachev

University of Chemical Technology and Metallurgy
Kl. Ohridski 8, 1756 Sofia, BULGARIA

Abstract: Big Data includes a huge volume, diverse in composition information from various sources, which is updated and accumulated with ever-increasing speed. Processing of Big Data with traditional methods is becoming more difficult to achieve, and in practice are imposed new technological IT solutions - NoSQL, Map Reduce, Hadoop, SAP HANA, etc. This article presents the main characteristics of Big Data, application areas and modern technologies in this field. The subject of the article is the NoSQL technology and the implementation of CAP-theorem that describes the relationship between consistency, availability and Partition Tolerance of the data. Use-cases have been developed to compare of the speed of execution of complex queries with a join clause and queries with grouping of data over SQL and NoSQL databases and are presented the obtained results. The respective conclusions have been drawn.

Key Words: big data, SQL, NoSQL, MongoDb, CAP-theorem

1. Introduction

In modern conditions the rapid increase in the volume of information (Big Data) is objective reality. Social networks, mobile devices, technologies like Internet of Things, business information are sources for generation of huge volumes of information. More than seven years Big Data is one of the key topics associated with promising IT technologies, the basis of which is likely to be transformed the business of companies from the financial, telecommunications and public sector, trade and others. Forms of application of Big Data are becoming more diverse and in parallel, the demand for technologies for general and concurrent processing of unstructured

Received: September 6, 2015

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

data is growing. According to data from a study of IDC Digital Universe, by the year 2020 the amount of data in the world will reach 40 Zb (zeta bytes), which is equivalent to 5200 Gb for every inhabitant of the planet.

2. Big Data - Characteristics, Application, Technology

Defining the concept Big Data is a subject of attention of many scientists and research institutes. According to J. Kalyvas Big Data is a "process uses people and technology to quickly analyze large amount of data different types (table structured and unstructured data, pictures, video, email, transaction data, social media interactions) from a variety of sources to produce a stream of actionable knowledge" [1].

In a technological aspect "Big Data is not entirely new technology, but rather a novel application of these existing technological trends" [2]. Gartner group define Big Data as "high volume, high velocity and high-variety information assets that demand cost-effective innovative form of processing for enhanced insight and decision making" [3]. Big Data is also defined in the 3Vs model Velocity, Variety and Volume. [4] Diya Soubra at ARM said "Big Data is expanding in multiple dimensions overtime" [5].

Compared to traditional databases Big Data is characterized by large volumes of information, decentralized storage of both structured and unstructured data with little or no relationships. Key features of Big Data are:

Volume - the accumulation of a large volumes of information makes it difficult for processing and storage with traditional methods. New approaches and advanced technologies are needed.

Velocity - Big Data is characterized by a high rate of accumulation of information (more than 90% of the data is from the last 2 years), which require timely processing in real time;

Variety - the diversity of Big Data and the increasing need to process both the structured and unstructured information (audio, video, text, etc.).

Veracity - it is necessary to ensure reliability of the information to the maximum extent;

Value - value of the accumulated information. Big Data should contribute benefits to organizations and to support the improvement of the business processes.

The fields of application of Big Data technology are broad and various. According to the IBM Institute for Business Value more than 53% of companies use Big Data in

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

customer service, 40% in operating efficiency and 7% in risk management. Technology related to Big Data were spread widely in various business areas - telecommunications, finance, trade, logistics, education, healthcare and government. Telecommunications operators keep some of the most demanding databases, which allows to make in-depth analyzes of the accumulated information. The main objective of the analysis is customer retention and attraction of new through market segmentation, analysis of customer traffic, defining the social identity of subscribers and others. In the area of financial services, Big Data allows to analyze the creditworthiness of customers, track the financial operations of each customer, to be offered appropriate banking services. In the field of trade accumulated information about customers and goods contributes to effective management of supply and sales, optimization of stocks of goods. According to Tech Pro Research [6] application of Big Data in many areas of the industry is more than 30%: Telecommunications - 58%, Engineering & Construction - 45%, Government - 38%, IT & Technology - 36%, Finance / Banking / Insurance - 33%, Logistics & Transport - 33%.

Along with the increased use of Big Data in all areas of life many problems arise and the need for their timely and adequate solving. Some of them are related to:

Storage and management of large volumes of data (thousands of terabytes and petabytes) as the relational database cannot effectively handle these processes.

Organization of unstructured information, consisting of text, images, video and more data types.

The need to analyze Big Data, addresses the issues to work with unstructured information, generate analytical reports, create predictive models and others.

Directives in the collection, storage and processing of Big Data can be divided into 3 groups: software; technological equipment; repair services.

Most widely spread approaches for data processing are:

SQL - structured query language which uses SQL to create and modify data, and the management of data sets is performed by corresponding system for managing database.

NoSQL - the term most often means Not Only SQL and includes approaches that address the realization of the databases different from the models used in traditional relational DBMS. NoSQL are suitable for use in constantly changing database structures, such as collection and storage of information from social networks.

Map Reduce - distributed computing model, which is used for parallel computing on very large volumes of data (petabytes and larger). The principle of operation consists in sequential processing of data by two methods Map and Reduce - Map selects the preliminary data and Reduce aggregates them.

Hadoop - is used for the realization of requests and contextual mechanisms of highly loaded sites - Facebook, eBay, Amazon and others. Hadoop system is

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

protected from failure of each of the nodes of the cluster, as each block has at least one copy of the data from another node.

SAP HANA - high-performance new SQL platform for storage and processing of data. Provides high processing speed of requests, simplifies the system landscape and reduces maintenance costs of the analytical system.

Technological equipment includes: servers (data warehouses) and infrastructure equipment to which are related the resources to accelerate the platforms, the sources of uninterrupted power supply, sets of server consoles and others.

Support services are related to the construction of architectural database systems, optimization of the infrastructure and provision of safe data storage.

Software, equipment and support services together form a complex platforms for storage and analysis of data. Large companies such as Microsoft, Hewlett-Packard, EMC Corporation and others offers services on development and implementation of solutions for Big Data and its management.

3. NoSQL and CAP-Theorem

In 2009 in San Francisco for the first time Eric Evans from RackSpace offers the term "NoSQL" to name the new open source products like BigTable and Dynamo. NoSQL has a spontaneous origin and characterizes the development of IT beyond relational databases. P. Sadalage and M. Fowler attempts to systematize and group the knowledge of the NoSQL in their book "NoSQL Distilled" [7]. The common characteristics of NoSQL and the many disparate systems with such a name can be found on the website [8]. Recently variety of programming solutions related to NoSQL are offered. Synonyms of the term NoSQL are huge volumes of data, clusters, linear scalability, no-relationally, fault tolerance and others.

The relational model is not always the most appropriate for presentation of the data. When developing applications with NoSQL is available a wide range of services for data processing of the type "documents" records "key-value" graphs and others. According to the CAP-theorem it is impossible to build such a database, which simultaneously have the following three properties:

Consistency. In the formal proof of the theorem this property is called atomicity [9] and it is expressed in the existence of a distributed system with general order of execution of operations. E. Brewer [10] interprets the property Consistency as the presence of data for a particular object in each node.

Availability means that all data is available in any point of time. According to E. Brewer "almost all requests must be answered." [11].

Partition Tolerance implies that the system continues to work after a failure of one or several distributed nodes. System having such properties should preserve its efficiency in conditions with loss of any number of messages sent between nodes.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

[12]. Practically it is considered that messages which delivery exceed the time limit are lost.

NoSQL databases, Google Big Table, HBase and Hypertable are systems of similar type [13]. Theorem CAP (Consistency, Availability and Partitioning tolerance) was proposed by Eric Brewer, a professor at the University of California, Berkeley and one of the founders of Google, in 2001 in the keynote of Principles of Distributed Computing.

NRW (Node, Read and Write) allows to analyze and tune how a distributed database will trade off consistency and the performance of reads and writes.

N is the number of nodes that keep copies of a distributed record.

W is the number of nodes that confirm a successful write.

R is the number of nodes that send back the read data.

The majority of NoSQL databases use $N > W > 1$ - more than one write must complete, but not all nodes need to be updated immediately.

When:
$W < N$ there is high write availability
$R < N$ there is high read availability
$W + R > N$ there is a strong consistency, read/write are fully overlapped
$W + R \leq N$ there is an eventual consistency, meaning that there is no overlap in the read and write set [14] ;

4. SQL vs NoSQL

The speed of processing of complex SQL queries is essential when are carried out analyses on large databases. In the presented example there is a relation of the type "one to many" (one order contains several products: Gasoline 95, Diesel Fuel and Gas).

The objective is to compare the performance of relational and non-relational database management systems. When joining two tables on a key field with a SQL statement, the scheme of the data can be built on both the RDBMS SQLite [15], and on NoSQL system - MongoDB node [16]. Table. 1 and Table. 2 gives the three types of fuels, 2 orders, quantities of the fuels purchased and their respective ITEM_ID.

In this implementation the normalization of the data is not complete - the name of the same product is repeated. This requires when updating the data (change of product name) to make changes to all locations in the database, where the name of the product presents, which slows down performance. The result is the accumulation of an excessive amount of data compared to the relational model, and hence slower at handling the processing, because is treated more data.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

Items	
ITEM_ID	NAME
55	Gasoline 95
56	Diesel Fuel
57	Gas

Table 1: Items

Orders		
ORDER_ID	ITEM_ID	QUANTITY
1	55	575
1	56	345
2	56	223
2	57	375
2	55	350

Table 2: Orders. The model from Table 1 and 2 is realized with NoSql in Listing 1.

```
[ { order_id:1,
    items : [
        {name:"Gasoline 95", quantity:575},
        {name:"Diesel Fuel", quantity:345} ] },
  { order_id:2,
    items : [
        {name:"Diesel Fuel", quantity:223},
        {name:"Gas",         quantity:375},
        {name:"Gasoline 95", quantity:350} ] }
]
```

Listing 1. Code of the model with NoSQL and JSON

5. Example 1: SQL vs MySQL and JOIN

The implementation of the relational model with SQL was prepared the following code (listing 2).

```
SELECT A.ORDER_ID, A.QUANTITY,
       B.NAME
FROM   ORDERS A
JOIN   ITEMS B ON A.ITEM_ID = B.ITEM_ID
```

Listing 2. Query with join through SQL

The results from the query is given in Table 3.

The application of NoSQL requires processing of the results on the server. The given example in listing 3 is written in NodeJs [17], and makes a request to the server

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

ORDER_ID	QUANTITY	NAME
1	575	Gasoline 95
1	345	Diesel Fuel
2	223	Diesel Fuel
2	375	Gas
2	350	Gasoline 95

Table 3: Result of the query with the join through SQL

for each record of the Table. 1 or calls the function findOne 5 times, and whenever a call is made NodeJS address the database via HTTP protocol. The execution time with the use of NoSQL significantly exceeds the execution time for the relational model in which the result is obtained with only one HTTP request to the server. The development of MongoDB, which is a relatively new product, continues and is expected in the near future to maintain joins with only one request.

```
// Javascript NodeJS MongoDb
var items  = [ { item_id:55, name="Gasoline 95" } ,
               { item_id:56, name="Diesel Fuel" } ,
               { item_id:57, name="Gas"         } ],
    orders = [ { order_id:1, item_id: 55, quantity:575 } ,
               { order_id:1, item_id: 56, quantity:345 } ,
               { order_id:2, item_id: 56, quantity:223 } ,
               { order_id:2, item_id: 57, quantity:375 } ,
               { order_id:2, item_id: 55, quantity:350 } ];

db.items.insert(items);
db.orders.insert(orders);

db.orders.find().forEach(
    function (order) {
        var join = db.items.findOne({ "item_id": order.item_id} );
        if (join != null) {
            printjson(join);
        }
    });
```

Listing 3. Query with join by using NoSQL and Java Script. The result through NoSQL is analogous to the result of the query through SQL (Table. 3).

6. Example 2: SQL vs NoSQL with Aggregation Clause

The performance of DBMS can be measured with linear query through summing and grouping SQL statement that has analog in the NoSQL system MongoDb. To find the sum of the group, the system must read every record in the database and accumulate the amount. This ensures that it will be performed a read operation available to each item.

The implementation of the task with SQL use the following code with a summing

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

function (Listing 4).

```
SELECT SUM(A.QUANTITY)
FROM   ORDERS A
GROUP  BY ORDER_ID
```

Listing 4. SQL query with a summing and grouping function.

When implementing the task with MongoDb is used the following code with a summing function (Listing 5). The execution of the query and the result from

```
db.orders.aggregate(
   [
      {
        $group : {
           _id : { order_id: "$order_id" },
           orderQuantity: { $sum: "$quantity" },
        }
      }
   ]
)
```

Listing 5. MongoDb query with a summing and grouping function.

Listing 5 is presented in Listing 6:

```
[ { order_id:1, quantity:920 } ,
  { order_id:2, quantity:948 } ]
```

Listing 6. Execution of the query with a summing and grouping function under MongoDb

When conducting the test for performance was used Processor Core 2 Duo E4300, with a SSD hard drive and 4 GB of RAM. Tests were carried out both with a relatively small number of records and with tables with a size of 10 million lines of records, and the implementation of the join clause took up to one hour and 40 minutes. The results of these tests are presented and summarized in Table 4.

		Number of records (thousands)			
		10	100	1000	10000
JOIN clause	SQLite	5.50	87.67	837.61	4203.85
	MongoDb	7.43	103.22	1475.26	6584.78
GROUP BY and SUM clauses	SQLite	4.25	17.23	427.36	1424.38
	MongoDb	5.85	14.81	542.34	1231.82

Table 4: Time for processing in seconds (the lower value means higher performance)

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

Chart 1. Represents the dependency between the number of the processed requests, and the time needed for their processing.

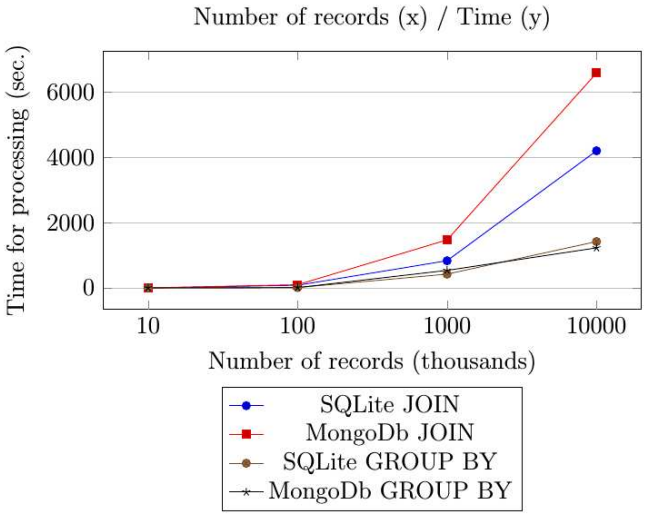

Chart 1: Graphical representation of the results of the test

7. Conclusion

The chart shows that SQLite has better performance when dealing with queries with join clauses, particularly on large datasets. The performance of queries with aggregation function the capabilities of both systems (SQLite and MongoDB) are relatively equal. Can be evaluated more complex queries (embedded SQL statements), but the result in terms of performance will be in favor of SQL, because they will cause a much greater number of sub queries over HTTP to the server, if realized with MongoDb. With respect of the aggregate function, the capabilities of relational databases are aligned in flexibility with the NoSQL databases. From Chart. 1 is observed a trend of progressive increase in the time required for the processing with increasing number of records, which is more explicit at the join clause.

The growing use of Big Data, the introduction of new approaches to use and management would help companies to increase their efficiency and competitiveness. Key levers that in the future will help the growing extension of Big Data are: increased interest from consumers, the development of methods for processing Big Data, the creation of technology parks, development of the market of IT. In parallel, the application of Big Data is associated with a number of problems and limitations: ensuring the safety and privacy of the data, shortage of skilled personnel, the complexity in the implementation of new technologies and their high value and others. NoSQL - solutions will not necessarily and entirely replace relational DBMS. However, the

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)

popularity of the NoSQL increases, which is related to the need to process huge volumes of data, scalability and availability. That is why when applying modern technologies it is necessary to take into account their advantages, to work on elimination of the problems by assessing each case and choose the appropriate means for solving specific tasks.

References

[1] Kalyvas J., M. Overly, Big Data: A Business and Legal Guide, CRC Press, Taylor & Francis Group, LLC, 2015, pp. 1

[2] Pries K., R. Dunnigan Big Data Analytics: A Practical Guide for Managers, CRC Press, Taylor & Francis Group, LLC, 2015, pp. 67

[3] Gartner, IT Glossary, 2013. http://www.gartner.com/it-glossary/big-data/

[4] Frampton M, Big Data Made Easy: A Working Guide to the Complete Hadoop Toolset, Apress, Springer Science+Business Media New York, 2015, pp. 1

[5] The 3Vs that Define Big Data, Diya Soubra, July 5, 2012, datasciencecentral.com

[6] http://www.techproresearch.com/article/research-29-using-iot-to-collect-big-data/, Tech Pro Research report, The Power of IoT and Big Data

[7] Sadalage P., M. Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Library of Congress Cataloging-in-Publication Data, 2013

[8] http://nosql-database.org/

[9] Abadi D., Consistency Tradeoffs in Modern Distributed Database System Design: CAP is Only Part of the Story, IEEE Computer, 45 (2), 2012, pp. 37-42

[10] Apache Apache Cassandra 1.1 Documentation, http://www.datastax.com/docsource/pdf/cassandra11.pdf.2

[11] Brewer E. CAP Twelve Years Later: How the "Rules" Have Changed, IEEE Computer, 45(2), 2012, pp. 23-29

[12] Abadi D., Consistency Tradeoffs in Modern Distributed Database System Design: CAP is Only Part of the Story, IEEE Computer, 45 (2), 2012, pp. 37-42

[13] Brewer E., A. Fox, Harvest, Yield, and Scalable Tolerant Systems, Proceedings of the Seventh Workshop on Hot Topics in Operating Systems, IEEE Computer Society, 1999, pp. 174-178

[14] http://ivoroshilin.com/2012/12/13/brewers-cap-theorem-explained-base-versus-acid/

[15] https://www.sqlite.org/

[16] https://www.mongodb.org/

[17] https://nodejs.org/en/

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 10, No. 1 (2016)