Big Data Analytics Applied for Control Systems

Yousef Farhaoui(✉)

ASIA Team, Department of Computer Science, Faculty of Sciences and Technics,
Moulay Ismail University, Errachidia, Morocco
youseffarhaoui@gmail.com

Abstract. Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them. Challenges include analysis, capture, search, sharing, storage, transfer, visualization, querying, and updating and information privacy. However, these huge data cannot easily handle since the most of CS systems are relational, and an adjustment is needed before any processing. With emergence of Big Data, new NoSQL systems come to deal with this relational data issue. So, we propose an approach to migrate historical CS data from relational to NoSQL system, and use a distributed environment containing many nodes. As experimentation, we migrate data generated by an oil and gas CS to an appropriate distributed NoSQL system, and we perform some data mining experiments on them in order to compare results and prove the obtained performance.

Keywords: Big data · Data mining · Control system · NoSQL · NewSQL · Analytics

1 Introduction

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them. Challenges include analysis, capture, data curation, search, sharing, storage, transfer, visualization, querying, and updating and information privacy. The term "big data" often refers simply to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. "There is little doubt that the quantities of data now available are indeed large, but that's not the most relevant characteristic of this new data ecosystem."

Big Data could be defined as an emerging phenomenon which refers to the practice of treatment of large and complex data volumes, with technical systems associated such as algorithms used to visualize and analyze real-time or not (real-time or batch) these massive data, to create added value for the organization. In parallel, the Data Mining also experiencing a rapid development (neural networks, genetic algorithms…) and it is now possible to create self-learning IT structures. This is the Machine Learning: analysis and implementation of automated methods that allow a machine (at large) to evolve through a process of learning, and so perform tasks that are difficult or impossible to fill by more conventional algorithmic means.

M. Ezziyyani et al. (eds.), *Advanced Information Technology, Services and Systems*,
Lecture Notes in Networks and Systems 25, https://doi.org/10.1007/978-3-319-69137-4_36

ON-RELATIONAL and relational data models are different. The relational model takes data and separates it into many interrelated tables that contain rows and columns. Tables reference each other through foreign keys that are stored in columns as well. When querying data, the requested information will be collected from many tables, as if the user asks: what is the answer to my question?

Non-relational data models often start from the application-specific queries, as opposed to relational data models. Non-relational data model will be then driven by application-specific access patterns. An advanced understanding of data structures and algorithms is required [1], so that the main design would be to know: what questions fits to my data?

In this paper we will provide in the introduction, a preview of the different non-relational data models. Next, we will focus on document databases in Sects. 2 and 3, by discussing evaluation criteria related to this data model. Finally, we will address the evaluation results in the conclusion.

Industrial companies and manufacturers are increasingly equipped with control systems (CS) that generate very large amounts of real time data. These data are used by specific applications to provide real time critical information, in-time graphs of evolution, real-time alarms, etc. Later, these data are stored for historical archives, and in many cases they are often deleted later. However, lots of users are increasingly interested in these historical data in order to use them in many business processes, and especially in data mining area, like extracting useful knowledge, providing early feedback means, improving future requests, etc. Nevertheless, this huge accumulated data needs progressively capacity and power to support processing and storage. Every machine or server arrives at its limits to support the storage and processing of huge data, whatever its physical capacity CPU, memory, or disk. Also, machines' upgrade or extension cannot be considered as a permanent solution. So, distributed platforms contained a lot of servers (nodes), such as clusters, grids can be efficiently used to deal with this issue.

This article has five sections. The first section is an introduction, and in the second, we present relational data issue on distributed environment and some related work. We give in the third section an approach to use distributed NoSQL system and data migration. In the fourth section, we prepare CS data for a company [2] on an experimental platform. In the fifth section, we carry out some interesting queries to test and compare performance. A conclusion closes this paper at the last section.

2 Characteristics of Big Data

It is therefore important to understand the Big Data - Volume, Speed and Variety.

2.1 Volume

The volume describes the amount of data generated by companies or individuals. Big Data is usually associated with this feature. Firms across all sectors will need to find ways to manage the ever-increasing volume of data that is created daily.

Catalogs of more than 10 million products have become the rule rather than the exception. Some customers managing not only products but also their own customers can easily accumulate a volume exceeding the terabyte of data.

2.2 Speed

The speed describes the frequency at which data is generated, captured, and shared. Due to recent technological developments, consumers and businesses are generating more data in much shorter time frames. At this level of speed, companies can only capitalize on these data if they are collected and shared in real time. It is precisely at this stage that many analyses, CRM, personalization, point-of-sale and other systems fail. They can only process data in batches every few hours, at best.

2.3 Variety

The proliferation of data types from sources such as social media, Machine to Machine interactions and mobile devices creates a great diversity beyond traditional transactional data. The data is no longer part of a clear, easy-to-use structure.

New data types include content, geo-spatial data, hardware data points, geolocation data, connection data, machine-generated data, measurement data, mobile data, physical data points, processes, RFID data, Research data, trust data, flow data, social media data, text data, and web-based data.

Why is it important to understand all this?

Because Big Data helps us to get a better representation of customer interaction with the company. It allows a better understanding of what customers would like to achieve at each point of contact.

It minimizes the risk of losing these customers when switching from one point of contact to another and ensures the relevance of the information that is delivered to them. Thus, to improve both the quality of service, key aspect for customers, and the transformation rate of these customers, it is important for the company not to lose sight of the Big Data.

3 Characteristics of NoSQL Databases

The term “NoSQL” was invented in 2009 during an event on distributed databases. The term is vague, incorrect (some NoSQL engines use variants of the SQL language, for example Cassandra), but has the advantage of having a certain marketing and polemic effect. In this part, we will discuss the general characteristics of NoSQL engines, historically, conceptually and technically, in relation to relational databases, but also independently of this reference.

3.1 Principle of NoSQL Databases

The NoSQL databases, especially the document-oriented ones, neglect the strengths of relational databases, namely the notion of registration and the relations between elements, in order to focus on the notion of a document. NoSQL databases are much more flexible and more scalable. The organizational structure is no longer linked to a relational scheme that is difficult to modify, and the basis can therefore grow without constraint.

On the other hand, the "document" orientation facilitates the deployment of the database on multiple machines. Automatically, of course. The developer is not concerned with the location of documents, split or not. When the database becomes too large, it is enough to define new machines connected on the network, and the NoSQL database gets by.

This is the answer to new applications demanding speed of processing and quantity of data managed. Quantities of the order of several hundred terabytes. Note that there are also the NOSQL bases of type "columns" and bases of type "graph". Column bases are an excellent solution for massive analysis. The graphic bases, more delicate to be apprehended, are, as their denomination indicates, more adapted to the resolution of the questions of organization in network (structure in arcs and nodes).

NoSQL systems use replication to achieve multiple objectives.

- Availability. Replication ensures that the system is always available. In the event of a server, node or disk failure, the task performed by the defective component can be immediately supported by another component. This technique of failover is an essential asset to ensure the stability of a system that can include thousands of nodes, without having to swallow a monstrous budget in monitoring and maintenance.
- Scalability (reading). If data is available on multiple machines, it becomes possible to distribute (read) requests on these machines. This is the typical scenario for the scalability of Web applications.
- Scalability (writing). Finally, one can think of distributing also the requests in writing, but there one is faced with delicate potential problems of competing writings and reconciliation.

The technique is very classic and used by all the DBMS of the world. Instead of repeatedly writing to the disk without a pre-defined order (so-called "random" accesses), which each time require a displacement of the read head and therefore a latency of a few milliseconds, one writes sequentially in a file of Log (log) and the data is also placed in RAM memory.

3.2 The Emergence of Big Data and NoSQL Databases

Software evolutions follow naturally the material evolutions. The first DBMSs were built around mainframes and depended on the storage capacities of the time. The success of the relational model is due not only to the qualities of the model itself but also to the optimization of storage that allows the reduction of the redundancy of the data. With the widespread use of network interconnections, increasing Internet bandwidth and

lowering the cost of moderately powerful machines, new possibilities have emerged in the area of distributed computing and virtualization, for example.

The shift to the twenty-first century has seen the volume of data manipulated by some organizations, especially those related to the Internet, increase dramatically. Scientific data, social networks, telephone operators, medical databases, national territorial defense agencies, economic and social indicators, etc., the increasing computerization of all types of processing implies an exponential increase in the volume of data Now in petabytes. This is what the Anglo-Saxons called the Big Data. Managing and processing these data volumes is seen as a new IT challenge, and traditional, highly transactional relational database engines appear to be completely outdated.

4 Distributed Data Issue and Related Work

In this section, we present the constraints of the relational model in distributed world, the NoSQL technology, and some related works given to deal with this issue. A distributed environment is a set of physical machines (nodes) which participate jointly to accomplish parallel processing and storage. All resources of the nodes (CPU, memory, disks) can be shared or not. The number of nodes can differ from a few to thousands of nodes. So, we can find simple clusters with few nodes which typically share disks, or more complex like grids that contain hundreds or thousands of nodes where resources are often not shared.

The main parts of the CS databases are based on the relational model which is built on the concept of table (relation between data) and operations of set-algebra.

This model is suitable for transactional needs due to the ACID properties (Atomicity, Consistency, Isolation, and Durability) [3], and it works very well in a single-node environment. Conversely, relational data cannot be well distributed and deployed in a distributed multi-nodes environment. The ACID properties become constraints for the model and then prevent data from being distributed effectively between nodes [4].

Note also that ACID constraints, although they ensure consistency, may become in some cases a blocking factor [5, 6]. For instance, an investigator in internet is often interested in having immediate response even if that response is not up-to-date. Consequently, new systems need to be created in order to dynamically distribute and manage data between nodes with more efficiency and usefulness. New systems for Big Data have been emerged like NoSQL and NewSQL.

5 Data Model Evaluation Criteria

The following criteria will be used to evaluate the document oriented model:

1. Nature of data (structured, semi-structured).
2. Data relationship (referential integrity, hierarchical relationships).
3. Data life-cycle (versioning, TTL).
4. Dataview (CRUD operations).
5. Data consistency (ACID properties).

6. Performance (indexing, partitioning).
7. Storage volume (BigData).
8. Data analysis (data aggregates).
9. Persistency and fault tolerance (data replication).
10. Data security (access rights, data encryption).

In general, data can be structured or semi-structured, depending on the related use-case.

In the relational data model, the tables contain a set of rows where every row is grouping a set of values. Existing relationships are based on joins between rows in different tables and they are not based on the semantic relationships between keywords (values).

6 Proposal

As proposal, we suggest migrating CS data to a suitable NoSQL system. We choose a NoSQL system according to specific criteria which are predominantly based on the fitting of NoSQLclass to CS data. Also, since CS data must be consistent, the type of the chosen NoSQL system must be CP (Consistency/Partitioning). Migrated NoSQLCS

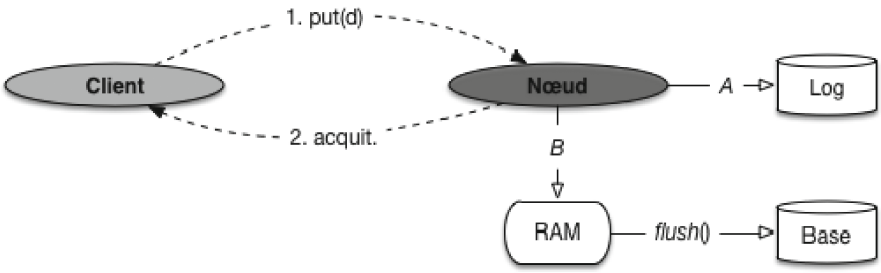

Fig. 1. Writing with logging

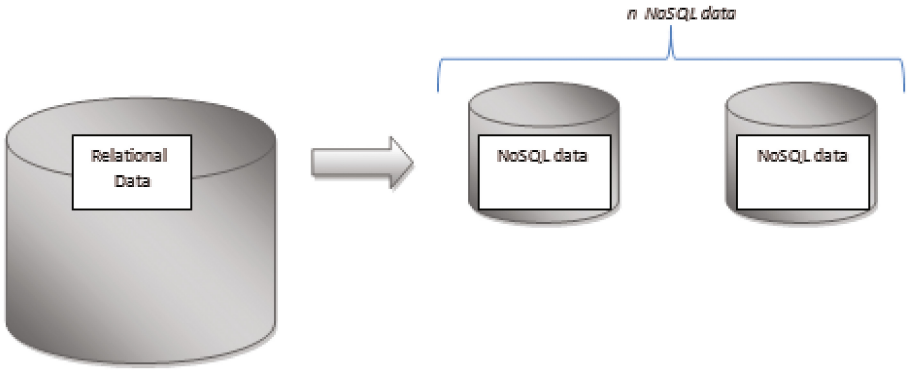

Fig. 2. Proposed approach

data will be distributed on a platform consisted of many nodes. The Sharding is a NoSQL property which distributes and allocates dynamically resources between nodes and adapt them according to data needs. "Figure 1," represents a schema for our proposed migration. This can greatly improve the performance on either the data storage or the processing time for ad hoc queries (Fig. 2).

7 Preparation of Target CS Data

To prepare CS data, we start implementing a distributed platform, and we perform migration data from Oracle 12c to MongoDB version 2.61. Some experiments and their results will be done in the next section. Firstly, we configure a distributed platform managed directly by MongoDB.

8 Results and Discussion

The new CS MongoDB database is now ready; our goal is to prove performance acquired by using elasticity given by distributed NoSQLdata. This elasticity signifies the ability to distribute data and queries processing on multiple nodes, contrary to the rigidity of relational data. So theoretically, good performance is expected. In the following, we use the same data mining queries for both the distributed MongoDBdata and the Oracle mono-node data. We will compare and analyze the results of the storage, run time, and shared query processing by varying each time the number of shares from 2 to 10. As experiments, we use collections which containing realtime data used frequently in looking for frequent itemsets. For data storage, we can see in Fig. 3 a graph showing

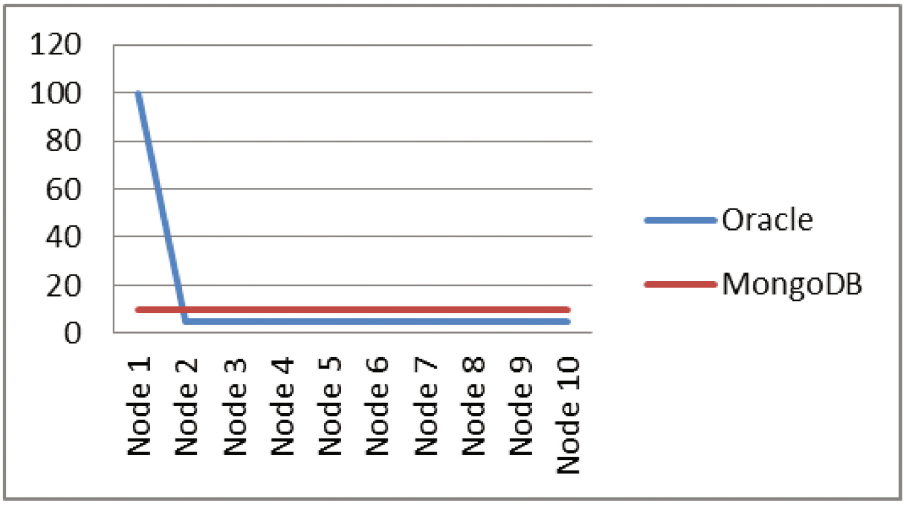

Fig. 3. Comparison the CS data storage repartition between MongoDB and Oracle

that MongoDB has shared automatically CS data on different shards, but for Oracle only one node supports data storage, the other nodes remain idle.

9 Conclusion

As experiment, after configuring a distributed environment with a number of nodes, we have installed MongoDB and migrated CS data. The NoSQLsharding property allows repartition of storage and processing between all nodes. Some experiments Data Ming queries have been done in order to compare performance of results between multi-nodes NoSQL data and mono-node relational data. Overall, the final results demonstrate an interesting improvement in run time, in addition of the data storage gained by joining all disk nodes. Finally, in perspective we look forward to enlarge this experimentation in widespread platforms environments with many nodes like Hadoop platform. Also, since NoSQL is new-fashioned and still in development, we are hearing of potential new NoSQL systems to test them and find the most appropriate for CS data. Document oriented clusters provide highly scalable architecture and better system availability. For this reason this model is one of the most used NoSQL models on the worldwide.

References

1. Kaur, K., Rani, R.: Modeling and querying data in NoSQL databases. In: BigData, 2013 IEEE International Conference. INSPEC Accession Number 13999217 (2013)
2. Hashem, H., Ranc, D.: An integrative Modeling of BigData Processing. Int. J. Comput. Sci. Appl. Print ISSN 0972-9038 (2014)
3. Sharma, V., Dave, M.: SQL and NoSQL Databases. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **2**(8), 2–8 (2012). ISSN:2277 128X. Research paper available: www.ijarcsse.com
4. Degroodt, N.: L'élasticité des bases de données sur le Cloud Computing. Master thesis in Sciences computer, FreeUniversity of Bruxelles, pp. 12–20 (2011)
5. Li, Y., Manoharan, S.: A performance comparison of SQL and NoSQLdatabases. In: Communications, Computers and Signal Processing, 2013 IEEE Pacific Rim Conference (2013). ISSN 1555-5798
6. Farhaoui, Y.: Big data and NoSQL system for control system. IJEFT **14**(2) (2017)