CLASSIFICATION, CHARACTERISTICS AND EVALUATION OF RDBMS IN BIG DATA PREDICTIONS AND ANALYTICS

D. Venu Gopal\textsuperscript{1}  
Research Scholar  
Department of Computer Science & Engineering,  
Sunrise University, Alwar, Rajasthan, India

Dr. Shalini Goel\textsuperscript{2}  
Research Supervisor  
Department of Computer Science & Engineering,  
Sunrise University, Alwar, Rajasthan, India

ABSTRACT

The purpose of this paper is to discuss elaborately the process of characterization, classification, and evaluation of existing relational database systems which are often used in various aspects of Big Data like in analytics and forecasts. Moreover, the paper also shows a learning method for shifting from relational database system to Big Data system. In this paper, we have dealt with the basic problems that the RDMSs of some famous companies like Oracle, DB2, SAP, and IBM often fail to meet the basic criteria for forecast and analytics in Big Data. The paper then discusses the security issues to different unstructured, semi-structured, and structured data formats and makes a comparison among these three data formats. At last, the paper will take into account the operational aspects of performance, measurement, and accessibility of data which are important while dealing with different database systems.

Keywords: Big data, RDBMS, database model, SQL

INTRODUCTION

The invention and application of Relational Database Management Systems have crosses almost 40 years now. Various organizations have been using RDBMS for their operations. During its early phase of applications, the organizations used it for processing static and query-intensive small data sets. These early phased types of RDBMSs are not efficient enough to meet the basic needs of the organizations today which use mostly dynamic and complex data sets. In this condition, the organizations which need to handle lots of data either going to reengineer their existing systems or have to look for completely new systems [1]. The purpose of Big Data is to manage 4Vs – Volume, Variety, Velocity, and Variability. Velocity is a new term in database management which means how promptly the streaming data is apprehended by the system. With time, various types of data are created and streamed in large volumes whose variability is high [2]. The incoming flows of data available from the internet and weblog which are mostly unstructured are to be trapped and tagged with the structured existing file system of the concerned organization as also with the respective metadata.

As the above-mentioned 4Vs of digital data proliferate with time, we need to look for the solutions to manage and apply this huge database productively in a more efficient and effective way. To overcome the shortfalls associated with these databases, the ACID approach linked with the Big Data
environment becomes suitable. We should know the application of various newer tools to handle effectively the enormous data available with Big Data systems [3].

As far as the data collection and uses are concerned, we can say that the world is residing on data; with the big players like Google, Facebook, Twitter, Instagram, various financial institutions, and government agencies accumulating extensively large data every day, data management has reached a different height these days [4]. Collecting, manipulating, managing, molecular modeling of DNA data or medical data need a different approach to databases available today. So, the organizations need a different data management practices today. Traditional data approach focuses on content data whereas the modern approach deals with context data [5].

RELATIONAL DATABASE MANAGEMENT SYSTEMS (RDBMS)

These days, we frequently come across the term RDBMS which is the standard language used for mentioning relational database management systems. Here, data is stored in rows and columns with each table possessing a primary key.

RELATIONAL DATABASE MODEL

RDBMSs allow the definition of data structures, storage and retrieval operations and integrity constraints. In this model, there are five substantial rules. By following these rules, the data integrity will be insured. These rules are also called characteristics of RDBMS.

E.E. Codd developed the concept of a relational database system in 1970. A reliable and approachable relational database model is based on ACID or Atomicity, Consistency, Isolation, and Durability. These are the basic properties that ensure smooth transactions with the available database [6]. RDBMS lets the definition of data structures, storing, and recovery or other operations and defines the integrity parameters. This model contains five important rules that ensure the integrity of data. These are also the defined features of RDBMS:

1. Every tuple is different and distinguishable which mean every record in a table is easily identifiable.
2. The order and attributes of the tuples in an RDBMS are not important.
3. Cells are meant for bearing just one value.
4. In the same attribute, all values belong to the same domain.
5. The names of the table must be unique; the attributed names (columns) in a table must also be unique.

In an RDBMS, the tables are interlinked. The relation between two tables is also stored in another table in SQL. This is a programming language that is used to perform some basic tasks like data upgradation and retrieving or recovering data in a database. Some widely-used RDBMSs that mostly use SQL (Structured Query Language) are Oracle, Microsoft SQL Server, DB2, etc. [7].

LIMITATION FOR SQL DATABASE

Here are the two primary limitations of SQL database:

1. Scalability – The users of an RDMS where SQL is applied needs to measure the database through highly capable servers. These servers are expensive and tough to handle. For a smooth-sailing measurement, the relational database needs to be spread over multiple servers.
2. Complexity – In a SQL server, it is mandatory that the data has to be appropriate for the concerned table. If this is not so, the database may need remodeling which will aggravate the complexities even further.
NOSQL OR NOT SQL

This SQL is appropriate for an unstructured schema. This is unstructured since this does not require any specific table schemes; moreover, fewer data can be stored in different nodes and collections. Apart from that, it supports some specific join queries and can only be scaled horizontally [9].

NoSQL has some benefits:

1. NoSQL is extremely scalable and comparatively easy to scale. Relational database (RDBMS) is vertically measurable. Vertical measurement or scaling means, the users have added several machines with the available resources. It is necessary that when load increases, server hardware capacity should be increased too [18].

2. It is evident that the maintenance of highly capable RDBMS systems is rather expensive and complicated but NoSQL requires less management. It possesses many important features like smooth data distribution and data transfer, automatic data repair.

3. Less expensive server and open-source are two attractive features of NoSQL. As their operation is easy and they require cheaper servers, users prefer it more than RDBMS servers. The later ones are comparatively expensive and need bigger storage space. It is found that the storage and processing data cost per gigabyte in NoSQL is lower than the cost for same data processing per gigabyte of RDBMS [19].

4. NoSQL database does not possess any schema. So, these kinds of databases could be used without defining any schema. Hence, the application or any change could be adopted at any time.

5. Data output performance is better than other systems as NoSQL supports integrated caching in system memory [8].

LIMITATIONS AND DISADVANTAGES OF NOSQL

Here are the most obvious limitations and disadvantages [10]

1. The same open-source that is considered as the greatest strength of NoSQL can also be a weakness here. As there is no defined standard for the system, no two databases in NoSQL are equal.

2. The availability of GUI mode tools is limited in the market.

3. Availability of experts is also very limited as the system is still in its developmental stages.

4. There is no a specific stored process Mongo DB or NoSQL database.

DIFFERENCES BETWEEN RELATIONAL DATABASE AND BIG DATA

Data from different sources or fields are collected for different purposes. User information, Machine-logging data, User-generated data, Satellite images, Sensor-generated data, etc. just a few examples of different sources. Often these data are interlinked to fulfill some common purposes. These days, organizations are using Big Data to design their operation-critical applications. It is found that the organizations are now depending more on NoSQL to utilize various data sources or data for meeting newer requirements in their business environment [11].

NoSQL and Relational data models are quite different from one another. In the latter types of models, several interrelated tables are containing rows and columns accept data and then create interrelated tables as per the requirements. These tables are oriented in such a way that they can refer each other according to the situation with the help of external keys that are preserved in the columns also [13].

With the passage of time, NoSQL has been witnessing increasing demand across the world as the users realize that it is a good alternative to a relational database system. In Big Data to, modern organizations see NoSQL helps in better operation when they have to use clusters and community
servers in different situations. As NoSQL fetches schema-less data models, it is often found to be more productive in several types of data capturing operations.

While using NoSQL, the users take into account traditional or functioning databases, the activities related to the data access and manipulation of data in Big Data are somewhat different. Operational databases generally accumulate huge datasets and involve a large number of users. In the latter case, the users are constantly accessing the data for transactions at any point of time [17]. In Big Data, we have to take into account the scalability aspects and a suitable way of handling the large data. In this situation, NoSQL is the best option [12]. It is found that a disseminated scale-out method is economical than a complete scale-up alternative. This is a result of huge, complicated, fault-resistant servers being expensive to develop and maintain [7].

While fetching information against a query, the system links several tables and data as required. This happens as per the pre-defined structure. So, at the time of writing data, similar data may be required to be written and coordinated in several tables. If such volume is low or the flow of data is relatively at a low velocity, a relational database can comfortably store the information. But, in the present-day applications, data flow is quite voluminous, it is necessary that any volume of data can be written, stored, and read at really high speed [3].

**BIG DATA PERFORMANCE AND SCALABILITY**

In Big Data, we need either scale-up or scale-out approach to manage a huge number of concurrent global users who are known as big users [26].

In Scale-up approach, a centralized architecture is applied. Here, all the basic and advanced functionalities are added to the functional servers. Each time a new user is added, some additional functionality may be developed. So, the servers grow in size with the increasing number of users [4].

On the other hand, Scale-out works on a distributed architecture. Here, a commodity of servers is used to meet the needs of the global users [6]. In a Scale-out approach, NoSQL seems more fruitful. In a Scale-out system, a commodity server is used at the initial phase. As the number of users increases, another commodity server may be required. No modification of the existing application accepts the entire database as a unified system [16].

NoSQL provides a high performing environment. At the same time, the expenses and scalability of the database system become highly flexible. The number of users goes on changing; keeping track of the users’ number commodity servers can be added or reduced. So, the operating cost always remains within the range. Overall, NoSQL databases are highly fault resistant, as the total load is evenly spread over several community servers. It is an efficient way of ceaseless operations [13].

**COMPARISONS OF STRUCTURED AND NON-STRUCTURED**

Organized data sets are those where the processing activities are predetermined and outcomes are structured. Application of structured systems is immensely beneficial in many circumstances like in inventory control, customer relations, payroll, booking systems, etc. In all these environments in inputs and outputs are structured or predetermined. The output data are displayed systematically so that the user can understand it easily.

On the unstructured data sets are also used in a variable environment like in contacts, blogs, emails, and in various other communication systems. In this forms of communications, the user a wide range of functionalities to structure a message before communicating with the other one. The rules and regulations in the unstructured forms are highly flexible.

So, the structured and unstructured systems may be different from organizational, technical, and functional perspectives [10].
Relational databases are normally structured. The written data there are maintained in rows and columns. In each column, mostly normalized data types are used. The SQL is appropriate to accumulate and recover or retrieve data in a planned way.

Queries are placed in common English terms. Queries possess a specific number of columns but any number of columns can be added later. Most of the tables are interlinked. The most popular vendors in this field are SQL Server, Oracle, MySQL, etc. [9].

A. What is the unstructured data?

The data sets that don’t possess any predetermined structure or possess very little structure are commonly known as unstructured data sets. In general, most these unstructured data sets can be structured at any moment. However, the whole affair may be a bit expensive [15]. Some, emails, word, spreadsheets, PDF, social media posts, video or audio files, etc. Users who perform communication activities within some unstructured systems get a wide range of options to structure the massages in their own ways. As such, the rules in the unstructured systems are not so stringent; the users here enjoy more freedom than the structured systems. This why the unstructured systems are so popular among all types of users [13].

According to Seth Grimes, more than 80% of the information in the business world is created through the unstructured systems. The graph given below shows the data representation in each type [11]:

“80% of business-relevant information originates in unstructured form, primarily text.”

Figure 1: Structured data vs. unstructured data

RELATIONAL DATABASE SYSTEMS SECURITY

Among all important aspects in a relational database system, database security is the most important one. The level of security and its authenticity is scaled through three primary aspects, viz. confidentiality, integrity, and availability. Some important methods utilized here are auditing, access control, and authentication. In this context, the capability of ORACLE is worth mentioning. It provides most powerful protection against all kinds of security threats in this realm. The following features are note-worthy in data protection in any system: Authorization, Authentication, Data Privacy, Data Integrity, and Data access control.

A. Big Data Security

The increasing applications of Big Data in various industries have created many new challenges making the whole system of relational database approach more complicated. The safety and privacy in this matter are big concerns these days. With the increasing speed of availability of Big Data, the security matters getting more and more important across the industries. These days, a vast array of data in huge quantities are accumulated, operated, analyzed, and transferred from one destination to another. The fast pace of data transfer and data processing give an opportunity to the hackers to access
highly confidential data from various sources especially from government and financial institutions’ data sources [8].

As data volume increases, as it continuously happens in Big Data, the security issues get more complicated. There are several companies still trying to evaluate the huge potential of Big Data. The investigation on the probable security threats associated with Hadoop and Cloud are underway. As stated above, Big Data poses various security threats with ever increasing newer threats that were non-existing even a few months ago. It is believed that many security solutions appropriate for the traditional systems might not work with Big Data because of changing speed, scale, and data characteristics. The existing security solutions are developed keeping the focus on the common characteristics of Big Data. The first challenge in this realm is to keep people’s personal data secured. A remarkable increase in data hacking, the threat to privacy has also increased manifold. So, this is the primary concern today [23].

Hurwitz et al. have suggested some essential options in Big Data security. According to these researchers, accessing, storing, and archiving of the keys are a really difficult task in Big Data. The issue can be solved to a large extent by generating and computing the encryption keys. One of the best solutions in this regard is inventing and applying software and hardware encryption methods. These encryption methods or technologies operate on some specific data or on the whole disk. However, these methods can increase the load on the CPU, increase the cost, and increase the overall complexity of the system.

Following are the most applied data security processes:

1. Data Anonymisation: In data anonymisation, the user can remove those data that can be exclusively linked to an individual. This provides some sorts of data privatization.

2. Tokenization: This process safeguards highly sensitive data by substituting it with a random token; this prevents the unauthorized access to important data. This procedure decreases the risk associated with data transfer.

3. Cloud-based database management: In this process, the whole database is encrypted so that no individual data requires encryption [26].

Collin White (27) describes that innovative and highly efficient analytical technologies used in data processing are doing marvelous jobs in this field. They are doing so many things that were never seemed possible even a decade ago. Some examples are as follows:

1. Some new systems are efficiently managing varieties of data that are constantly formed in social media and World Wide Web.

2. More efficient analytics known as advanced analytics like text analytics and event analytics are doing wonderful jobs.

3. Upgraded hardware systems including multi-core processors, solid-state drives, huge memory spaces, etc. are making the processing really fast.

4. Cloud computing systems including SaaS, Data platforms, Virtualization in personal clouds, etc. have brought higher flexibility in work.

5. Operational intelligence in the various business environments has improved productivity, instant action, and real-time decision-making processes.

Collin White further suggests that combining these above-mentioned innovative and efficient functionalities improve the overall business agility in any industry. According to another research in today’s business environment Big Data is indispensable, but it inevitably brings three aspects, viz. volume, variety, and velocity which increase the threat to any forms of database. Each of these aspects is getting stronger with time thereby creating new challenge to the security vendors [24].
CONCLUSION

In this paper, we saw the basic characteristics of RDBMS and Big Data. We also discussed the data types attributed to RDBMS and Big Data. Comparisons among various operational issues like scale, data availability, data performance, etc. Are also made. The security issues and how these issues are dealt with are also discussed elaborately.

REFERENCES


