Comparisons of Relational Databases with Big Data: a Teaching Approach

Ali Salehnia
South Dakota State University
Brookings, SD 57007
Ali.salehnia@sdstate.edu

Abstract

The paper's objective is to provide classification, characteristics and evaluation of available relational database systems which may be used in Big Data Predictions and Analytics. In addition, it provides a teaching approach from moving relational database to the Big Data environment.
In this study, we try to answer the question of why Relational Database Bases Management Systems such as IBM's, DB2, Oracle, and SAP fail to meet the Big Data Analytical and Prediction Requirements. The paper also compares the structured, semi-structured, and unstructured data as well as dealing with security issues related to these data formats. Finally, the operational issues such as scale, performance and availability of data by utilizing these database systems will be compared.

Introduction

It has been more than 40 years since the Relational Database Management Systems have been implemented and used by different organizations. This database approach was satisfying the needs of businesses dealing with static, query intensive data sets since these data sets were relatively small in nature [21, 22]. These traditional relational database systems do not answer the requirements of the increased type, volume, velocity and dynamic structure of the new data sets. So, organizations with lots of data, either have to purchase new systems or re-tool what they already have [1, 4, 5, and 8]. Big data deals with volume, velocity, variability and variety. Velocity obviously refers to how quickly the streaming data is captured [1, 6]. As more data are created and streamed the high variability and high volume as well as variety of formats are at issue [2, 5, and 7]. The incoming data from a Web log, unstructured content from the Internet, need to be captured, tagged with metadata and hierarchical file systems.
As the volume, complexity, variety and velocity of digital data grow faster by the day, we need to find solutions to use these data in productive, effective, and efficient ways. With shortcomings, the ACID approach associated with the relational in the BIG Data environment, we need to effectively take advantage and manage available tools to be able to utilize the huge volume and unstructured data residing in Big Data systems [23, 25].

When it comes to data collections and utilizations, these days, one can say: "It is Brave New World". For example, Google, Facebook, Governmental Agencies, and Financial institutions collect, transfer, manipulate, and analyze big sets of data on the daily basis [17, 21]. Creating (Collecting), Manipulating, analyzing and transferring, molecular modeling, medical images or DNA data require a newer approach of databases. Therefore, organizations need to adopt their data management practices as they load and analyze all these types of data. Whiles the traditional database approaches mainly deal with content data, new approaches call for dealing with context data [1, 11, 13, and 14].

Relational Database Management Systems (RDBMDS)

RDBMS is the standard language for relational database management systems. Data is stored in the form of rows and columns where each table must have one primary key.

Relational Database Model

The relational Database model (approach) was developed by E.F. Codd in 1970s. The most important feature of all relational databases is its support of ACID (Atomicity, Consistency, Isolation and Durability) properties which assures that all the transactions are reliably processed.

RDBMDS 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.

“1. The order of tuples and attributes is not important,

2. Every tuple is unique. This means that for every record in a table, there is something that uniquely identifies it from any other tuple,

3. Cells contain single values. This means that each cell in a table can contain only one value,

4. All values within an attribute are from the same domain,

5. Table names in the database must be unique and attributed names in tables must be unique. No two tables can have the same name in a database. Attributes (columns) cannot have the same name in a table. You can have two different tables that have similar attribute names.

6. The relations between tables are also stored in the form of the table SQL (Structured Query Language) is a programming language used to perform tasks such as update data on a database, or to retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, DB2, Sybase, Microsoft SQL Server, Access, etc.” [7, 8].

Limitations for SQL Database

“Scalability: Users have to scale relational database on powerful servers that are expensive and difficult to handle. To scale relational database it has to be distributed on to multiple servers.
Complexity: In SQL server’s, data has to fit into tables anyhow. If the data doesn’t fit into tables, then there is a need to design database structure that will be complex and again difficult to handle.” [7, 8].

Not Only SQL (NoSQL)
NoSQL deals with unstructured schema so, less data can be stored in multiple collections and nodes and it does not require fixed table sachems; in addition it supports limited join queries and it is scaled it horizontally[5, 7, 9].

Benefits of NoSQL

“Highly and easily scalable
Relational database or RDBMS databases are vertically scalable. When load increases on RDBMS database, we scale database by increasing server hardware power need to acquire expensive and bigger servers and NoSQL databases are designed to expand horizontally and in Horizontal scaling means that you scale by adding more machines into your pool of resources [18].

Maintaining NoSQL Servers is Less Expensive
Maintenance of high-end RDBMS systems is expensive and needs trained manpower for database management, but NoSQL databases require less management. It supports many features like automatic repair, easier data distribution and in NoSQL, the simpler data models make administration and tuning requirements reduced.

Lesser Server Cost and open-Source

NoSQL databases are cheap and open source and their implementation is easy and typically uses cheap servers to manage the exploding data and transaction while RDBMS databases are expensive and requires big servers and storage systems. So, the storing and processing data cost per gigabyte in the case of NoSQL can be many times smaller than the cost of RDBMS [19, 20].

No Schema or Fixed Data model
NoSQL database is schema less so data can be inserted in a NoSQL database without any predefined schema. So, the format or data model can be changed at any time, without any application disruption and change management.

Support Integrated Caching
NoSQL database supports caching in system memory, so it increases data output performance and SQL database where this has to be done using separate infrastructure." [5, 7, and 8].

Limitations & disadvantage of NoSQL:

1. "NoSQL databases are open source which is its greatest strength but at the same time, it can be considered as its greatest weakness because there are not many defined standards for NoSQL databases; so, no two NoSQL databases are equal.
2. No stored procedures in Mongodb (NoSql database).
3. GUI mode tools to access the database is not flexibly available in market.
4. It is so difficult to find NoSQL experts because it is the latest technology and NoSQL developers are in learning mode [5, 6, 10, and 11].

Differences between Relational Database and Big Data

Personal user information, geographic location data, user-generated data and text, machine-logging data, sensor-generated data, and satellite images are just a few examples of the ever-expanding arrays being captured. Enterprises are also relying on Big Data to drive their mission-critical applications. Across the board, organizations are turning to NoSQL databases because they are uniquely suited for these new classes of data emerging today [5, 7, 10, and 11].

Relational and NoSQL data models are very different. The relational model takes data and separates it into many interrelated tables that contain rows and columns. These tables reference each other through foreign keys that are stored in columns as well [10, 13].

NoSQL is increasingly being considered a viable alternative to relational databases, especially for Big Data applications, as more enterprises recognize that operating at scale is better achieved on clusters of standard, commodity servers. In addition, a schema-less data model is often preferable for the variety and type of data captured and processed today.

In the NoSQL space, we read and write from traditional or operational databases while the activities to access and manipulation of Big Data are very different. Operational databases generally host large datasets with a large numbers of users that are constantly accessing the data to execute on transactions in real time [17, 19, 22, and 25]. When dealing with Big Data, we have to consider scale and volume of the data set and how to organize them effectively. Therefore, NoSQL is critical for scalability [5, 7, 10-12].

A distributed scale-out approach also usually ends up being cheaper than the scale-up alternative. This is a consequence of large, complex, fault-tolerant servers being expensive to design, build and support. [7, 10, and 11].

When a user needs to run a query on a set of data, the desired information needs to be collected from many tables and joined before it can be provided to the application. Similarly, when writing data, the write needs to be coordinated and performed on many tables. When data is relatively low-volume and when it is flowing into a database at a low velocity, a relational database is usually able to capture and store the information. But, today's applications are often built on the expectation that massive volumes of data can be written (and read) at speeds near real-time [1, 2, and 3].

Big Data Performance and Scalability

In order to effectively and efficiency use the underlying databases of Big Data, we need to either scale up approach or scale out approach to deal with the concurrent global users, commonly referred as Big Users [3, 4, and 26].

Scale up approach refers to a centralized architecture in which functionalities are added to existing servers based on the increase number of global concurrent users and these servers becomes bigger and bigger [3,4]. While scaling-out refers to a distributed architecture where a commodity of servers are added to meet the requirements of global users [3, 6, and 6]. The scale-out approach of NoSQL databases is very much easier. If huge number of users start using application then another commodity server is added very simply. There is no need to modify the application since the application always sees a single (distributed) database [16].

Along with performance, cost and scalability of NoSQL databases, the flexibility is also equally attractive. As users come and go, commodity servers/virtual machines can be quickly added or removed from the server pool by keeping track of the user population and thus operating cost is also reduced. The NoSQL databases are highly fault tolerant databases because the load is distributed across many commodity servers and thus support in continuous operations [10, 11, and 13].

Comparisons of Structured and Nonstructured

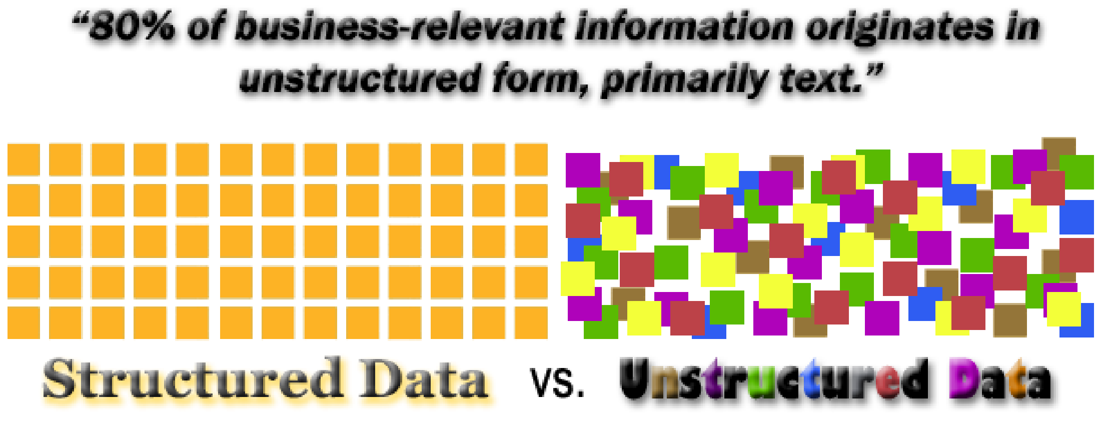
Structured data sets are those where the activity of processing and output is predetermined and highly organized. Structured systems are designed. Payroll, Inventory control systems, point of sale systems, airline reservations are all forms of structured systems since they are using structured data- the data which is stored and displayed as a set of rows and tables. In contrast, unstructured data sets are the data that have little or no predetermined form or structure. Unstructured data sets include email, contracts, blogs, and other communications. A person who performs a communications activity in an unstructured system has wide latitude to structure the message in whatever form is desired. The rules of unstructured systems are fewer and less complex [9, 10, and 13]. The structured and nanostructured data can be different from technical, organizational, structural, functional point of view. Each has its own environment and needs to be treated and used accordingly. The structured and nanostructured data can be different from technical, organizational, structural and functional point of view. [4, 11, 15, 18, 19].

Relational databases are highly structured: all the data in the table are stored as rows and columns. Each column has a data type which is mostly normalized. The SQL is suitable to relational databases to store and retrieve data in a structured way. Queries are Plain English commands. There are always fixed number of columns although additional columns can be added later. Most of the tables are related to each other with primary and foreign keys thus providing “Referential Integrity” among the objects. The major vendors are ORACLE, SQL Server, MySQL, PostgreSQL, etc. [7, 9].

What is the unstructured data?

Unstructured data sets are the data that have little or no predetermined form or structured and are raw and unorganized. Ideally, all of this information would be converted into structured data. However, this would be costly and time consuming. Also, not all types of unstructured data can easily be converted into a structured model [11, 15, and 31]. A few examples of unstructured data include: Emails, Word processing files, PDF files, spreadsheets, digital Image files, video and audio files as well as social media posts. A person, who performs a communications activity in an unstructured system, has wide latitude to structure the message in whatever form is desired. The rules of unstructured systems are fewer and less complex [9, 10, and 13].

Seth Grimes reported that “80% of business-relevant information originates in unstructured form, primarily text.” The following graph shows the data representation of each type [11, 31]:


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

Relational Database Systems Security

Database security is an important issue in database design. Availability, confidentiality, and integrity are properties used to evaluate databases’ security level. Methods such as authentication, auditing and access control could be utilized. Oracle, as a most used commercial database, provides a powerful protection against data penetration. Specifically, the following features are provided to protect data.

- Authorization;
- Authentication;
- Discretionary & Mandatory Access Control;
- Data Integrity;
- Data Privacy.

Big Data Security

While Security and privacy issues are big concerns in the relational database approach, the challenges for the organizations and individuals dealing with Big Data sets become more complex. The vast amount of Big Data and their speed and velocity make security and privacy a major concern. Availability and accessibility of this type of data to organizations, at rapid speed create an opportunity for hackers and criminals to have faster access to governmental and organizations' entity data. Nowadays, massive quantities of heterogeneous data are collected, manipulated, analyzed, and transferred rapidly [4, 5, 8, and 9].

With the advent of Big Data comes the risk of greater security breaches as data volumes increase. Many companies are still trying to evaluate the potential of Big Data, let alone investigate the risks associated with Hadoop and the Cloud. Big data does bring new security challenges to the table, such as the fact that most existing security solutions we use in the traditional database and data processing approaches world might not work in this environment. The main reasons for this are the scale, speed and variety of data. Most existing security solutions were not built with Big Data in mind even most of those were developed as afterthoughts. The fast-increasing amount of personal-related data produced and consumed around the planet adds challenges to users' privacy considerations. Due to increase in hacking activities users' privacy is also more important than before [8, 9, 23, 27].

J. Hurwitz et al. suggests a set of protection options for Big Data Security: "Managing the storing, archiving, and accessing of the keys is difficult. To alleviate this problem, generate and compute encryption keys as needed to reduce complexity and improve security. One way to safeguard Big Data is software and hardware encryption technology that operates on selected data on the fly or across an entire disk. However, this method increases the load on a database server's CPU. This increases costs and overall complexity.

Here are some other available data-safeguarding techniques:

- **Data anonymization:** When data is anonymized, you remove all data that can be uniquely tied to an individual. Although this technique can protect some personal identification, hence privacy, you need to be really careful about the amount of information you strip out.
- **Tokenization:** This technique protects sensitive data by replacing it with random tokens or alias values that mean nothing to someone who gains unauthorized access to this data. This technique decreases the chance that thieves could do anything with the data.
- **Cloud database controls:** In this technique, access controls are built into the database to protect the whole database so that each piece of data doesn't need to be encrypted." [28].

Colin White of BI Research [29] indicates that "New and evolving analytical processing technologies now make possible what was not possible before. Examples include:

- **New systems** that handle a wide variety of data from sensor data to web and social media data.
- **Improved analytical capabilities** (sometimes called: *advanced analytics*) including event, predictive and text analytics.
- **Operational business intelligence** that improves business agility by enabling automated real-time actions and intraday decision making.
- **Faster hardware** ranging from faster multi-core processors and large memory spaces, to solid-state drives and virtual data storage for handling hot and cold data.
- **Cloud computing** including on-demand software-as-a-service (SaaS) analytical solutions in public clouds, and data platforms and virtualization in private clouds."

She also suggests that supporting Big Data involves combining these technologies to enable new solutions that can bring significant benefits to the business. In another study, [30] it is indicated that "Today's threat environment imposes the three vs of big data: volume, variety, and velocity. Each of these is increasing at an astounding rate and has required a shift in how security vendors manage threats." [24, 29].

Conclusion

In this paper the characteristics of Relational Database Systems and Big Data were presented. Also, the types of data each approach has to deal which were discussed. The security issues in each model were presented. The operational issues such as scale, performance, and availability of data by utilizing these database systems were also compared.

References

[1] "Big Data: Volume, Velocity, Variability, Variety", http://nosql.mypopescu.com/post/6361838342/bigdata-volume-velocity-variability-variety, Accessed April 2015.
[2]. B. Wiederhold, "18 essential Hadoop tools for crunching big data", Network World, www.googletagmanager.com/ns.html. Accessed April, 2015.
[3]. A. K. Zaki, "NoSQL Database: New Millennium Database for Big Data , Big Users, Coud Computing and Its Security Challenges," http://esatjournals.org/Volumes/IJRET/2014V03/I15/IJRET20140315080.pdf, Accessed May 2015.
[4]. L. Arthur, "What is Big Data", htpp:://www.forbes.com/sites/liasaarthur/2013/08/15/what-is-big-data/, Accessed May 2105
[5]. S. Penchikala, "Virtual Panel: Security Considerations in Accessing NoSQL Databases‖, Nov. 2011. http://www.infoq.com/articles/nosql-data-security-virtual- panel., Accessed May 2015.
[6].Oracle Databases from web: http://www.oracle.com/us/products/database/overview/index.h tml, Accessed May 2015.
[7]. Relational Database Management System (RDBMS) vs noSQL. http://openproceedings.org/html/pages/2015_edbt.html, Accessed April 2015.
[8]. "Relational -database-management-system-rdbms-vs-nosql/, http://www.loginradius.com/engineering/relational-database-management-system-rdbms-vs-nosql/ Accessed April 2015.
[9]. M. Ramachandran "Relational Vs Non-Relational databases", http://bigdata-madesimple.com/relational-vs-non-relational-databases, Accessed May 2015.
[10]. L. P. Issac "SQL vs NoSQL Database Differences Explained with few Example DB", http://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/, Accessed May 2015.
[11]. Sherpa Software. "Structured and Unstructured Data: What is It? http://www.sherpasoftware.com/blog/structured-and-unstructured-data-what-is-it/, Accessed May 2015.
[12]. F. Chang, et al. "Bigtable: A distributed storage system for structured data.", ACM Transactions on Computer Systems (TOCS) 26.2 (2008): 4.
[13]. D. Gosain, "A survey and comparison of relational and non-Relational Databases‖. IJERT,Vol 1,Issue 6,2012.
[14]. L. Okman, , N. Gal-Oz, Y. Gonen, E. Gudes, and J. Abramov, , "Security Issues in NoSQL Databases," Trust, Security and Privacy in Computing and Communications (TrustCom), 2
[15]. IEEE 10th International Conference on , vol., no., pp.541-547, 16-18 Nov. 2011 doi: 10.1109/TrustCom.2011.70
[16].Find cloud security alliance https://cloudsecurityalliance.org/research/big-data/

[17].B. Sullivan, "NoSQL, But Even Less Security‖, 2011. http://blogs.adobe.com/asset/files/2011/04/NoSQL-But-Even- Less-Security.pdf. Accessed April 2015.
[18].Find Source: www.couchbase.com/why-nosql/nosql- database.
[19]. K. Madia, "The Five Most Interesting Talks at Black Hat, DEF CON and BSides", https://securityintelligence.com/five-steps-for-better-security-analytics-in-2015, Accessed August 2015.
[20]. "The Four Pillars of Big Data for the CMO", http://www.certona.com/the-four-pillars-of-big-data-for-the-cmo/, Accessed August 2015.
[21]. Unlocking the Magic of Unstructured Content", *IDS White Paper*, http://inmoncif.com/registration/whitepapers/unlocking-final.pdf, Accessed August 2015.
[22]. Z.Liu' B. .Jiang^z and J. Heer, "imMens: Real-time Visual Querying of Big Data" Eurographics Conference on Visualization (EuroVis) 2013 Volume 32 (2013), Number 3.
[23]. S. Durbin**,** **"**A CEO's Guide to Big Data Security", http://www.infosecurity-magazine.com/view/32736/a-ceos-guide-to-big-data-security/ Accessed May 2015.
[24]. E. Weindruch"Big Data brings privacy, security concerns",http://www.bizjournals.com/charlotte/print-edition/2013/09/20/big-data-brings-privacy-security.html?page=all, Accessed May 2015.
[25]. "Enhancing Big Data Security", www.advantech.com, Accessed May 2015.
[26]. H. Chen, R. H. L. Chiang, V. C. Storey, "Business Intelligence and Analytics: From Bog Data to Big impact " , *MIS Quarterly*, Special Issue: Business Intelligence Research.
[27]. **"**Holistic Approach Needed for Big Data Security"**,** http://www.theiia.org/intAuditor/feature-articles/2013/february/holistic-approach-needed-for-big-data-security/, Accessed May 2015.
[28]. J. Hurwitz, A. Nugent, F. Halper, and M. Kaufman , "Security Considerations with Big Data", http://www.dummies.com/how-to/content/security-considerations-with-big-data.html, Accessed May 2015.
[29]. C. White, "Using Big Data for Smarter Decision Making", BI Research, July 2011.
[30]. "Addressing Big Data Security Challenges: The Right Tools for Smart Protection", *Trend Micro*.
[31]. S. Grimes, "Unstructured Data and the 80 Percent Rule", http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/, Accessed May 2015.