Relational Vs. NoSQL databases: A survey

Mohamed Ahmed Mohamed
The National Assembly, Sudan

Relational vs. NoSQL Databases: A Survey

Mohamed A. Mohamed   Obay G. Altrafi   Mohammed O. Ismail

Department of Computer Science
University of Khartoum
Khartoum, Sudan

Abstract— **The huge growth in the Internet market and the emerging of the new web technologies and the trend toward what is called web 2.0 and recently web 3.0 come with a new challenges, new applications and new concepts such as NoSQL databases which is recently becomes a very popular as an alternative to the relational databases specially in dealing with large data which is one of the most common features of web today, providing high availability and scalability to the distributed systems which need fast access time and can't tolerate any down time during failures and have been used heavily by the big enterprises and web companies such as Facebook, amazon and google. Every new technology faced many challenges like Security vulnerabilities. This paper addresses the concepts of NoSQL, the movement, motivations and needs behind it, and reviews the types of NoSQL databases and the issues concerning to these databases mainly areas of application and the security issues compared with traditional relational databases.**

Keywords— "Relational vs. NoSQL database" "NoSQL Security"

I. INTRODUCTION

Relational Database (RDB) which based on the relational model has been architected more than 30 years ago mainly to serve business data processing since then it has become the best option for storing information that range from financial records, personal data and much more. However, the user requirements and hardware characteristics have evolved from that time to including data warehouses, text management, and stream processing, these kind of process have very different requirements than traditional business data processing. Also the web 2.0 came by many new applications that depend on storing and processing big amount of data and it needs high availably and scalability which added more challenges to the RDB [1]. And because of that a growing number of companies have adopted various types of non-relational databases, commonly referred to as NoSQL databases as the applications they serve emerge [2] like Yahoo with their PNUTTS to meet massively parallel and geographically distributed database system for their web applications as they said [3], Facebook with Cassandra and Google with BigTable.

NoSQL as term was first used in 1998 by Carlo Strozzi as name of file-based database he was developing , since that time it has being used for the relational databases that omit the use of Structured Query Language (SQL). However, it was not before 2009 that it became a serious competitor to the term RDB. In present Eric Evans an employee in Rackspace Company described the ambition of the NoSQL movement, as "the whole point of seeking alternatives is that you need to solve a problem that relational databases are a bad fit for" [4]. The wildly usage of these NoSQL products encouraged other companies to make their own solutions and led to emerge of generic NoSQL database systems, now there is more than 150 NoSQL product [5]. These products come with issues like suitability to some areas of application, security and reliability.

There are some Fundamentals must be aware of, *ACID* used to refer to the four properties of transactions (atomicity, consistency, isolation, durability).

- *Atomicity*: stands for 'everything or nothing'. If any part of the transaction left incomplete then the entire transaction is considered failed.
- *Consistency*: ensures that a database before and after any transaction is stable at a valid state.
- *Isolation*: ensures that multiple transactions executing at the same time do not affect one another's execution. Thus, requiring the concurrent transactions to be serialized.
- *Durability*: ensures that once a transaction has been committed it will remain in the same state i.e. stored permanently even if there are some errors, or even if the system crash or power loss occurs [6] [7].

BASE (Basically available, Soft state, Eventual consistency) It is intended that the consistency after a transaction is not a solid state anymore (soft state). It shall be reached not right after finishing the transaction, but rather in some time during the operation (eventually consistent). The focus of BASE is the permanent availability. BASE is the opposite of ACID. NoSQL databases are classified in-between the spectrum from ACID to BASE. In the case of a bank, the eventual consistency is not what you want, thinking about two different balances on different servers! The balance must be equal just in time in every database involved in a money transaction session. In the case of an online book trade, the "just-in-time consistency" becomes less important. It does not matter if a book's price on one replication differs from another during a short time like a few hours. [8]

In addition, the *CAP* theorem must be mentioned, it first appearance was in year 2000; Eric Brewer introduced the idea that there is a fundamental trade-off between consistency, availability, and partition tolerance. These terms explained below:

- *Consistency*: The data is always the same in every replication on every server.
- *Availability*: The data must always be accessible (permanently available).
- *Partition Tolerance*: The database works fine despite network and machine failures.

The theorem says that only two of these aspects can be guaranteed at the same time in a distributed system. You have to "pick" two of them. In this paper, we will not discuss the proof of this theorem; we will just accept it as a matter of fact. [8] [9]

In this paper we compared between the concepts of the two technologies in the form of data model and areas of application its support to the cloud and we will focus on the security issues concerning with both databases. Existing related works are described in section II. In section III we discuss main different between two databases. We conclude in section IV.

II. RELATED WORK

Many early papers that issued the relationship between Relational and NoSQL databases were give an overview of NoSQL database its types and characteristics, they were so enthusiastic about NoSQL and how it declined the dominance of SQL like in [10] [11] however in [12] there discussion about the structured and non-structured database also the paper explained how the use of NoSQL databases like Cassandra improved the performance of the system, in addition it can scale the network without changing any hardware or buying bigger server. The result is improving the network scalability with low-cost commodity hardware.

In [7] which a survey paper issue relational databases, there features and shortcomings also NoSQL and its features, however there shortcoming and Issues with NoSQL databases has been mentioned in [13] as serious concerns and doubts about it like it's complexity, consistency, its limited Eco structures , and most of the developer is unfamiliar with the technology. Also in [14] the authors give statement that the demand for relational database will not go away anytime soon and it will exclusively serve in line of application that support business operations however NoSQL databases will serve the large, public and content centric applications. In addition in [2] there where analysis for the security issues with NoSQL databases considered in Cassandra and MangoDB as example.

III. RELATIONAL VS. NOSQL DATABASES

A. *Transaction reliability:*

Relational databases guarantee very high transaction reliability because they fully support ACID unlike the **NoSQL databases** because they range from BASE to ACID.

B. *Data Model:*

Relational databases based on the concepts of sets in mathematics, all the data represented as mathematical n-ary relations, an n-ary relation being a subset of the Cartesian product of N domains. The data inside the database represented as tuples and grouped into relations. The relation (represented by table) contain set of Tuples (rows) which sequence of attributes named column in the relation table, the type of an attribute it identified by the domain which is set of values that have a common meaning. This data model is very specific and well organized. Columns and are described by well-defined schema. The set of related data stored in rows has the same structure.

NoSQL databases take many modelling techniques like key value stores, graph, and document data model. NoSQL is classification took its name of types from their data model but sometimes we find NoSQL database system using two or more of the data models to represents the data. The common and the main feature that distinguish the NoSQL data model is it doesn't use the table as storage structure of the data also it's schema-less and very efficient in handling the unstructured data like word or pdf files, images, and video file ,etc…

C. *Scalability:*

Scalability in **relational databases** is greatest challenge that faces it; because it depends on the vertical scalability (by adding more hardware resources like RAM, CUP, etc…) however vertical scalability dependence on improving hardware is very costive and actually impractical for the reason of hardware limitation. Other type of scalability is horizontal (in which more commodity nodes or system unites are added) but when the relational databases are created it wasn't in the mind to give the support for the web applications that spread among many server and service millions of users like what happening nowadays so it doesn't support horizontal scalability very well. **NoSQL databases** depend on the horizontal scalability.

D. Cloud:

The **relational databases** are not well suited for cloud environments because they do not support full content data search and are hard to scale them beyond a limit. However **NoSQL databases** are the best solution for cloud databases because all the characteristics that define the NoSQL databases are very desirable for cloud databases. The cloud databases are not ACID compliant and it provide improved availability, scalability, performance and flexibility also it deals with unstructured, semi-structured data or structured data.

E. Big data handling:

Big data handling is very big issue in **relational databases** and the solution was and will always be the scalability and data distribution which take two forms vertical or horizontal in which data must be portioned into multiple servers which raise an issue of complexity in the joining for these data and the performance related to this operations. NoSQL databases designed to handle the big data so they implemented methods to improve the performance of storing and retrieving data.

F. Data warehouse:

Relational databases used for data warehousing which - as known - resulting of gathering data from many sources and over time the size of stored data increases and this lead to big data problem which raises other problem like performance degradation when doing an OLAP, data mining or statistical process. In the other hand **NoSQL databases** is not designed to severe data warehouse applications because the designers focused on high performance, scalability, availability and storing big data which is may be benefited by data warehouse to solve the increasing size of stored data problem.

G. Complexity:

Complexity in **relational databases** rises because the user must convert data into tables and when the data does not fit into those tables the structure of the database could be quit complex, difficult, and slow working with, unlike the **NoSQL databases** which have the capabilities to store unstructured, semi-structured or structured data.

H. Crash Recovery:

Relational databases grantee crash recovery via recovery manager which responsible for ensuring transaction atomicity and durability by use log files and ARIES algorithm. On the other hand crash recovery in **NoSQL databases** depend on replication as backup to recover from the crash, however some has other mechanism to as Journal file in Mangodb.

I. Security:

Relational databases has adopted very secure mechanisms to provide the security services although they faces many security threads like SQL injection, Cross Site Scripting, Root Kits, Weak communication protocols and much more. Many of studies today investigate and try to solve this Vulnerabilities. NoSQL databases came manly solutions to solve problem of big data storing and increase performance of databases that effect on security side in it but many of Current NoSQL product try to solve this security issue. As showing in table (1) we compare between relational databases security and NoSQL databases in some security services.

TABLE I
SECURITY SERVICES IN RELATIONAL & NOSQL DATABASES

Category	**Relational databases**	**NoSQL databases**
Authentication	All relational databases came with authentication mechanism, and can choice any of that mechanism to use.	Many NoSQL databases by default does not come with authentication or authorization mechanism, but can use some of external method to perform this operation.
Data Integrity	ACID properties that used in relational databases guarantee database transactions are processed reliably that ensure data integrate.	Eventually consistent is one of BASE properties principle ttherefore data integrity is not always achieved in NoSQL databases.
Confidentiality	Data confidentiality is often achieved in relational database because it was use encryption techniques, to store data encrypted.	Data confidentiality is not achieved, because usually data is store clear.
Auditing	Provide mechanisms to audit that allow writing to the database	Most of NoSQL databases don't provide auditing.

	syslog or xml files, and some relational database give more advanced auditing like oracle Fine Grained Auditing which allow audit under certain circumstances. For example record an entry to the log file if customer deposited more than 1,000,000$.	There some databases that provide auditing with issues like Couchdb which store the user name and password in the log file which of course compromises the security
Client communication	Relational databases provide secure client communication mechanism via using encryption and SSL protocols.	Most of NoSQL databases do not provide mechanisms of secure client communication

IV. CONCLUSION

We reviewed the concepts of the relational databases and NoSQL database, motivation behind NoSQL databases and why many of big companies using them. NoSQL databases different in many aspects from traditional databases like structured schema, transaction methodology, complexity, crash recovery and dealing with storing big data which the feature lead to use NoSQL in cloud computing and may be data warehouses.also paper focused in Security because it became most undertaken feature today, in relational databases these feature covered very well however NoSQL has shortage in security mainly because their designer focuses on other purposes than security and generally the NoSQL databases solution still fresh it didn't reach the full maturity yet, for all that we can find many security vulnerabilities in it.

Finally NoSQL has well experience big evolution in the near future because most of current applications and software are tend to depending on web also size of data need to store is in continues increasing rapidly, that convince us to believe that NoSQL databases well face huge growth and improvement and well solve its security problems soon or later.

V. REFERENCES

[1] Stonebraker, Michael; Madden, Samuel; Abadi, Daniel J.; Harizopoulos, Stavros, "The end of an architectural era: (it's time for a complete rewrite)," Proceedings of the 33rd international conference on Very large data bases, VLDB, p. 1150–1160, 2007.

[2] N. G.-O. Y. G. E. G. J. A. Lior Okman, "Security Issues in NoSQL Databases," in 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11, 2011.

[3] Brian F. Cooper, Raghu Ramakrishnan, Utkarsh Srivastva, Adam Silberstein and others, "PNUTS:Yahoo!'s Hosted Data Serving Platform," ACM, no. 08, 2008.

[4] P. W. Kriha, "NoSQL Databases," [Online]. Available: www.christof-strauch.de/nosqldbs.pdf. [Accessed 2 2013].

[5] "NoSQL databases," [Online]. Available: nosql-database.org. [Accessed 10 6 2013].

[6] J. G. Raghu Ramakrishnan, Database Management Systems, McGraw-Hill, 2002.

[7] Nishtha Jatana, Sahil Puri, Mehak Ahuja, Ishita Kathuria, Dishant Gosain, "A Survey and Comparison of Relational and Non-Relational Database," International Journal of Engineering Research & Technology (IJERT), vol. I, no. 6, 2012.

[8] S. Weber, "NoSQL Databases," University of Applied Sciences HTW Chur, Switzerland, 2010.

[9] N. A. L. Seth Gilbert, "Perspectives on the CAP Theorem," Singapore, 2012.

[10] V. Sharma and M. Dave, "SQL and NoSQL Databases," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 8, pp. 20 - 27, 2012.

[11] R. P. Padhy, M. R. Patra and S. C. Satapathy, "RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database's," INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES, vol. 11, no. 1015, pp. 15 - 30, 2011.

[12] A. Bhatewara and K. Waghmare, "Improving Network Scalability Using," International Journal of Advanced Computer Research, vol. 2, no. 6, pp. 488 - 490, 2012.

[13] N. Leavitt, "Will NoSQL Database Live Up to Their Promise?," IEEE computer society, vol. 10, no. 9162, pp. 12 - 14, 2010.

[14] C. Nance and T. Losser, "NOSQL VS RDBMS - WHY THERE IS ROOM FOR BOTH," in Proceedings of the Southern Association for Information Systems Conference., Savannah, GA, USA, 2013.