ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

NOSQL DATABASES: NEW TREND OF DATABASES,EMERGING REASONS, CLASSIFICATION AND SECURITY ISSUES

Jeelani Ahmed*, Raafiya Gulmeher

* Student, Computer science and Engineering KBN college of Engineering Kalaburagi, Karnataka, India
Assistant Professor, Computer Science and Engineering KBN college of Engineering Kalaburagi, Karnataka, India

ABSTRACT

The recent growth in the internet market and the emerging of new IT technologies with new challenges and new concepts such as NoSQL which is now becomes a very popular as a replacement to the relational databases specially when working with the big data. This paper includes the introduction of NoSQL databases along with the important differences between traditional relational databases and NoSQL databases. This paper also addresses the emerging resources, classification and few applications of the NoSQL databases. It also focuses on the security mechanism that must be implemented at the middleware by the developers to overcome the security issues of NoSQL databases.

KEYWORDS: Relational vs. NoSQL database, NoSQL Security.

INTRODUCTION

Relational Database (RDB) which is based on the relational model has been created more than 30 years ago for the purpose to serve business data processing since then it has become the best place for storing information that include business data, personal data and much more. While, the user requirements and various characteristics of hardware have evolved from that time to including data warehouses, text management, and stream data processing, these kinds of processes have very different requirements than traditional business data processing. Also the web 2.0 came by many applications that depend on storing and processing big amount of data and it requires high availability and scalability which added more challenges to the Relational database [1]. And because of that a growing number of industries have adopted different types of non-relational databases, commonly known as NoSQL databases as the applications they serve emerge [2] like Yahoo with their PNUTTS to meet parallel and geographically distributed database system for their web applications [3], Facebook with their Cassandra and Google with BigTable.

NoSQL as term was first used by Carlo Strozzi in 1998 as name of file-based database he was developing, and then from that time it has being used for the relational databases that exclude the use of Structured Query Language (SQL). However, it was not before 2009 that it became a serious competitor to the term RDB. In today's 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 not good fit for it" [4]. The uncontrollably usage of these NoSQL products inspired other companies as well to develop their own solutions and led to emerge of generic NoSQL database systems, today more than 150 NoSQL products are available [5]. These products come with suitability issues to some areas of application, security issues and reliability issues.

RELATIONAL VS NOSQL DATABASES

Transaction reliability

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

Data Model

Sets in mathematics are the driving force for relational database, all the data represented as mathematical n-ary relations, an n-ary relation is a subset of the Cartesian product of N domains. The data represented as tuples inside the

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

database and grouped into relations. The relation (represented by table) contain set of tuples (rows) with sequence of attributes named column in the relation table, the type of an attribute is 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 rows are described by well-defined schema.

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

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

Cloud
In cloud environments the relational databases are not well suited because they do not support full content data search and are hardly scalable beyond a limit. Whereas NoSQL databases are the best option for cloud databases because all the NoSQL database characteristics are very desirable for cloud databases. The cloud databases are not ACID compliant and it provide improved scalability, availability, performance and flexibility also it deals with unstructured, semi-structured data or structured data.

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

Data warehouse
For data warehousing Relational databases are used which is known as - resulting of gathering data from many sources and over the size of stored data increases and this results in big data problem which increases other problem like performance degradation when doing an OLAP, statistical process or data mining. Whereas NoSQL databases is not designed to used for data warehouse applications because the designers focused on scalability, availability, high performance.

Complexity
In relational databases complexity is rises because data must be convert into tables by user and the structure of the database could be difficult, quit complex and slow working with when the data does not fit into those tables, whereas the unstructured, semi-structured or structured data are stored in the NoSQL database.

Crash Recovery
Crash recovery is grantee in relational databases via recovery manager which is responsible for ensuring durability and transaction atomicity by using log files and ARIES algorithm. Whereas in NoSQL databases crash recovery depend on replication to recover from the crash.

Security
To provide the security services very secure mechanisms has adopted by Relational databases even they faces many security problems like Cross Site Scripting, SQL injection, Weak communication protocols, Root Kits and much more. Today many of studies investigate and try to resolve these vulnerabilities. NoSQL databases came to increase

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

performance of databases that effect on security side and to solve problem of big data storing in it but many of Current NoSQL product try to solve this security issue.

NOSQL EMERGING REASONS

Migrating towards Nosql databases

Since the 80s the relational data model architecture has been dominating out of the many different data-model architectures, with the implementations like MySQL[7], Microsoft SQL Servers [8] and Oracle database [9]. Lately, however, because of its data modeling techniques the relational databases leads to the problems in many cases. Organizations demands evolution of new data management model because of exponential growth in complexity of data generated by real time systems, sensors, global users and social networks etc and the storage of this big amount of data on big distributed system[10]. Organizations are increasingly turning to non-relational or NoSQL databases [11] because of collecting large amount of unstructured and ever changing data as shown figure 1

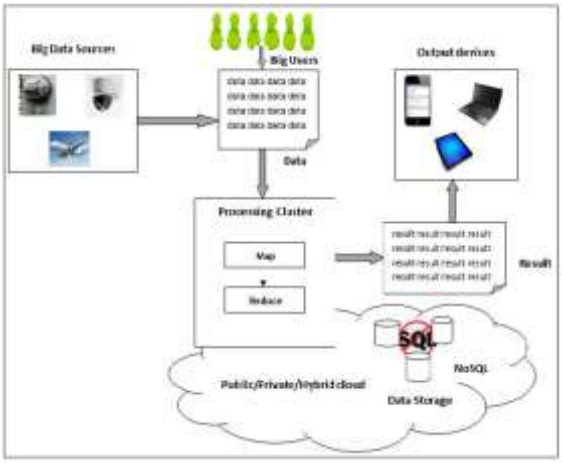

Fig -1: Organizations migrating towards NoSQL database

Analytical processing of large scale datasets in warehouses focused in NoSQL databases that offer increased scalability over commodity hardware and servers [12]. Storage and computational requirements of applications such as for Business Intelligence [13], Big Data Analytics [14], and social networking over peta-byte datasets have published SQL-like Centralized database to their limits [15]. This results in development of non-relational data stores called NoSQL databases which are horizontally scalable and distributed like Google's Bigtable[16] and its open source implementation HBase[17] and facebook Cassandra[18]. The emergence of distributed key-value stores proves the efficiency and cost effectiveness of their approaches[19] such as Cassandra and Voldemort [20]. The non-relational databases also have limitations like it is hard to scale with Data warehousing, Web 2.0, Grid and cloud applications [21]. Web applications like blogs, which include various kinds of attributes, for these, strict relational schema of relational databases can be a burden. The fast changing information and other data like text, pictures, audios, videos and real time data have to be stored within multiple tables. NoSQL systems have the ability to store and index Big Data sets while enabling a large amount of concurrent user requests.

Adoption of NoSQL Database

The acronym NoSQL was defined in 1998, which means Not Only SQL. As many of the Web 2.0 leaders have been adopted a NoSQL technology the NoSQL movement has been in the news from the past few years. All the companies use NoSQL in one way or another such as Twitter, Google, Amazon, Digg, LinkedIn and Facebook. A survey was conducted in the year 2012 called Couchbase Survey [22]. Key points from the Couchbase NoSQL survey include:

- Out of 1,300 respondents nearly half indicated they have funded NoSQL projects in the first half of this year. NoSQL projects in companies with more than 250 developers getting funding about nearly 70%, over the course of 2012.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

- 49% responded that the primary driver for their migration from relational to NoSQL database technology is the rigid schema. Other highly ranked reasons given for migrating to NoSQL are high latency/low performance and lack of scalability (see chart below for more details).
- 37% indicating NoSQL becoming more important with 40% said that it is very critical or important to their daily operations. As shown in figure 2

Fig-2: Key problems-driving to NoSQL databases

NoSQL databases are getting serious attention by the organizations that have massive data storage. And the experts of NoSQL Database are highly demanded for most of the developing Companies. This graph highlights the job trends of five NoSQL Databases from Indeed.com shown in figure 3

Fig-3: job trends of five NoSQL Databases (Source: Indeed.com)

MongoDB's growth means that it has gained its place as the most famous NoSQL database, as mentioned by the LinkedIn profile. As the chart below describe, it now holds 45% of all mentions of NoSQL technologies in LinkedIn profiles. See figure 4.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

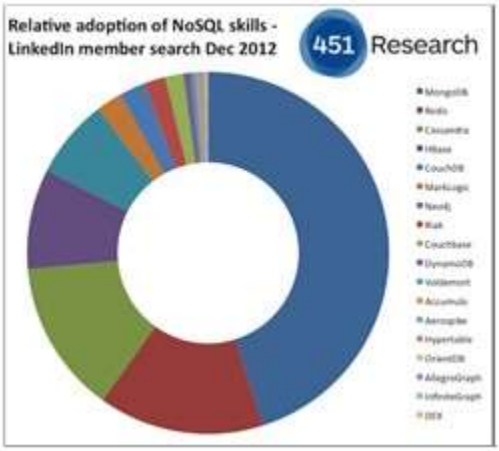

Fig-4: NoSQL LinkedIn Skills Index – December 2012
(Source: http://blogs.the451group.com)

CLASSIFICATION OF NOSQL

Leavitt, N. classifies the NoSQL databases into three types:

Key Value Store Databases

With these NoSQL databases developers develop their applications with unformatted data storage approach, schema-less, resulting in elimination of fixed data model. Hence these are the simplest NoSQL databases. In this type of database the data is stored in the form of key-value pair. In that every single item in the database is associated with key and this key represents an attribute name together with its value. Compared to non relational databases these type of database support faster execution of queries and high concurrency. See figure 5
Example: Redis [23], TC and TT.

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

Fig-5: Key/Value Store NoSQL Database
(Source: www.readwritewebcomimages.com)

Column Oriented Databases

The database which stores their data in the form of columns is known as Column Oriented Database. This make it faster read a particular mentioned column to memory and making calculations on all values in a column. These are used for optimization of queries over large datasets, and these databases stores column of data together.
Example: Cassandara [24], Hypertable [25] etc…

Document Oriented Databases

As their name implies designed to store and manage documents. These databases are encoded in JSON (JavaScript option notation) or XML format to store the values which is called as document. These databases helps in easy debugging, conceptualizing data by supporting complex data structures. See figure 6

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

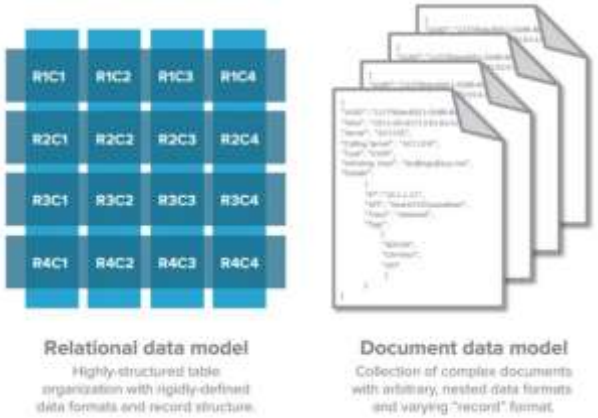

Fig-6: Document Store NoSQL Database (Source: http://gigaom.com/2011/07/29/couchbase-2-0-unql-sql-nosql/)

In NoSQL Database market there are many products that now claim to be part of the NoSQL database, far too many to mention here or describe in any detail.

APPLICATIONS

NoSQL is a critical component of the data architecture in some companies. Here are a few examples of such company's use cases.

Facebook: HBase for Messages

When Facebook decided to explore its messaging services architecture [26] to encompass email, instant messages, text and more, they knew it had to have a distributed fault-tolerant database which able to manage peta bytes of messages. And to send over 15 billion messages from one person to another person over 350 million users per month" and a chat service that "supports over 300 million users who send over 120 billion messages per month" to handle these existing services it needed something. Meanwhile they tried MySQL, a solution with which they have extensive experience [27], they created Cassandra [28]—an open source project combining elements of Google's BigTable[29] and Amazon's Dynamo [30] designs—and Apache HBase, a distributed database that closely mimics Google's BigTable implementation and is tightly integrated with Apache Hadoop and ZooKeeper. Facebook favored the strong consistency model, automatic failover, load-balancing, compression, and MapReduce support of HBase for their new production messaging service.

Craigslist: MongoDB for Archived Postings

Craigslist has recently migrated over 2 billion archived postings from its MySQL clusters into a set of replicated MongoDB servers [31]. But they still use MySQL for all active postings on their site, and now days when you sign in and review old posts that are now expired, you are about to accessing their new MongoDB-based service. For Craigslist, scalability and reliability were central requirements, but one of the more interesting features they gained by going to MongoDB was schema flexibility. Their MySQL databases were so large that any schema change (ALTER TABLE) would take around two months to complete across the replicated database set. Contrast that with MongoDB, where data is stored as JSON documents with no schema enforcement at all. Craigslist simplified their architecture by separating archived postings from live postings and made it easier to change their production schema as requirements changed.

SECURITY ISSUES

In this section we will focus on security built-into the NoSQL databases environment which are entrusted in handling Big Data, we will further see the weaknesses of these systems. Our motive is to discover the security problems inherent in the NoSQL database system environments and exploring how best to secure these environments. Data need to be safely secured that is being generated for processing and storage results in terms of amount, diversity and rate of data. A survey of the top big data vendors as well as deployments for services have revealed that huge amounts of sensitive and valuable data are generated by machines or by human that routinely use and access mobile and web applications

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

being handled across many platforms through the various applications all over the world. The data is highly valuable that is available at the organizations, and must be protected and a subject to privacy laws and compliance regulations.

Threats Posed By Distributed Environments
The increased attack surfaces across several distributed nodes are created which makes it very complex to secure the NoSQL database system because in NoSQL database environment nodes are distributed subsequently resulting in parallel computation. The probability of unauthorized access increases when deciding where to grant database system Access, whether at the remote location or at the Clients home locations.

Safeguarding Integrity
Because of its heterogeneous nature than in homogeneous environments the protection of integrity in NoSQL database system is difficult. It is difficult to implement integrity constraints because of its schema-less nature and absence of central control.

Communication between Nodes
All communication protocols as DataNodes talk to the NameNode are placed on top of the TCP/IP mainly relying on RPC over TCP/IP. In the NoSQL distributed environment a Remote Procedure Call (RPC) abstraction wraps both the DataNode Protocol and the Client Protocol. NoSQL database Systems with RPC ports are especially vulnerable that are exposed to the Distributed environment.

Shared Data/Fragmented Data
Data is horizontally segments in the form of slices in NoSQL databases and share them across multiple servers. In the NoSQL database environment which is distributed across multiple servers, data from a variety of nodes move from one node to another node. An example is that at one organisation which has clusters with up to 4000 nodes, and about 65 million files and 80 million blocks. The maintenance is computationally expensive of replicated shards of data that includes passwords and more prone to error and increases the risk of theft.

Compromised Clients
NoSQL databases accessed by clients are in contact with various nodes and resource managers directly. The entire system is compromised when malicious data gets propagated from a single compromised location. When there is no central management security point, protecting the name servers, nodes and those clients become difficult.

Protection of Data at Rest
When it comes to protecting data in storage, most NoSQL databases are found wanting, by employing encryption techniques only a few types of NoSQL databases provide mechanisms to protect data at rest. Encryption is widely referred as the de-facto standard for safeguarding data in storage. Malicious intruders will find the data unintelligible who intend to steal from archives or with intention to read data directly from the disk. With the help of decryption keys encrypted data will be accessed by users, but however transparency required in the NoSQL environment and encryption services offered by most industry solutions lack horizontal scaling.

Challenges In Enforcing Access Control
Role-based access control is difficult to enforce because of NoSQL database's schema-less structure. For example data store by Key-Value store by means of a distributed index for object storage. Different data are stored in one huge database in this type of database. As heterogeneous data is stored together in one database as opposed to relational models this becomes a challenge.

Configuration and Patch Management
Existing configuration management tools work for underlying platforms. Different clusters of servers or nodes may have different patch revisions. The challenge may be created in enforcing security continuously across the NoSQL database environment when added nodes may have newer patches than existing nodes.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

Firewalls
In NoSQL database environment firewalls cannot protect data at rest or in-transit. The database is immediately exposed to attacks when a firewall gets breached. Firewall breaches emanating cannot be avoided like attackers from the firewall perimeter who get into data centres electronically or physically can get access to data.

Authentication Clients
To authenticate clients, DataNode, NameNode in the NoSQL database environment Kerberos can be used. Upon duplicating or stealing the Kerberos ticket malicious Clients and Nodes can gain unauthorised access to the NoSQL database system. These credentials can be obtained from system snapshots as well as virtual images. To generate malicious services into the databases environment when exact copies, clones and imposter nodes can be used, then situation has worsened in this Big Data environment.

Audit And Logging
To aid in discovery of malicious activities in the database system Audits and logs are performed. However logging is not useful to detect malicious activities without actually looking at the developing policies and data. Audit may have impact on their effectiveness by frequency at which the Audits are carried out. There will be serious problems for the organization if audits are performed quarterly that means malicious activities can harm the organization and it may be lately identified.

CONCLUSION

Since from many years relational databases have dominated the industries but now NoSQL databases are getting attention of developers due to the following reasons:

- NoSQL databases provides dynamic flexible schema-less data model, which is most suited for the big data and big users.
- To support global users and big data NoSQL databases have an ability to scale up dramatically.
- To satisfy big users expectation NoSQL databases provide an improved performance without compromising scalability.

The security mechanism must be implemented at the middleware by the developers to overcome the security issues of NoSQL databases. And In comparison with the relational databases we need to strengthen the database without compromising the performance features and scalability.

Finally NoSQL has big evolution in the future because most of the software and current applications are moving to depending on web also size of data need to store is in increasing rapidly. That proved us to believe that it will face improvement and huge growth and soon or later will solve its security problems.

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] Mohamed A Mohamed, Mohammed O Ismail; "Relational vs. NoSQL Databases: A Survey", International Journal of Computer and Information Technology, volume 03, May 2014.
[7] MySQL Databases from web: http://www.mysql.com/
[8] Microsoft SQL Server Databases from web: http://www.microsoft.com/en-us/sqlserver/default.aspx.
[9] Oracle Databases from web: http://www.oracle.com/us/products/database/overview/index.html.
[10] A B M Moniruzzaman and syed Akhtar Hossain, NoSQL database: New Era of databases for Big data Analytics-Classification, Characteristics and comparison.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

[11] Levih,N(2010). ―Will NoSQL databases live up to their promise?‖ computer43(2), 12-14.
[12] Konstantinou,I.,Angelou, E. Boumpouka,C., Tsoumakos,D., andKoziris,N(2011) October. ―On the elasticity of NoSQL databases over cloud management platforms.
[13] Luhn, H. P. (1958). A business intelligence system. IBM Journal of Research and Development, 2(4), 314-319.
[14] Russom, P. (2011). big data analytics. TDWI Best Practices Report, 4 th Quarter 2011.
[15] Abadi, D. J. (2009). Data management in the cloud: Limitations and opportunities. IEEE Data Eng. Bull, 32(1), 3-12.
[16] Chang, Fay, et al. "Bigtable: A distributed storage system for structured data."ACM Transactions on Computer Systems (TOCS) 26.2 (2008): 4.
[17] HBase Databases from web: http://hbase.apache.org
[18] Lakshman, A., & Malik, P. (2010). Cassandra—A decentralized structured storage.
[19] Use relational DBMS, N. (2009). Saying good-bye to DBMSs, designing effective interfaces. Communications of the ACM, 52(9).
[20] http://www.slideshare.net/adorepump/voldemort-nosql
[21] Padhy, R. P., Patra, M. R., &Satapathy, S. C. (2011). RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database_s. International Journal of Advanced Engineering Science and Technologies, 11.
[22] Couchbase Survey from web: http://www.couchbase.com/press-releases/couchbase-survey-shows-accelerated-adoption-nosql-2012.
[23] Redishttp://redis.io/
[24] AvinashLakshman, Prashant Malik, "Cassandra-A Structured Storage System on a P2P Network", http://cassandra.apache.org/
[25] Hypertable, http://hypertable.org/
[26] The Underlying Technology of Messages: https://www.facebook.com/notes_/facebook-engineering/the-underlying-technology-of-messages/454991608919.
[27] MySQL at Facebook: http://www.facebook.com/MySQLatFacebook.
[28] Note on creating Cassandra: https://www.facebook.com/note.php?note_id =24413138919.
[29] Google BigTable: http://labs.google.com/papers/bigtable.html.