A Review on Big Data Management and NoSQL Databases in Digital Forensics

Nikhil Mangle[1], **Prof. Praful B. Sambhare**[2]

[1]Student, Dept. of computer Science & Engg., P.R.Pote Patil) College of Engineering & Mang Amravati.

[2]Asst. Prof. Dept of Computer Science & Engg., P.R.Pote (Patil) College of Engineering & Mang Amravati.

Abstract: ***The huge growth in the Internet and the emerging of the new web technologies and the trend toward come with a new challenges, new applications and new concepts of database such as NoSQL databases which is recently becomes a very popular as alternative to the relational databases mostly in dealing with large data which is one of the most common features of web today, in the providence of high availability and scalability to the distributed systems which need fast access time and also can't tolerate any down time during failures and have been used heavily by the big organizations and web companies as Digital forensics data is much complex and heterogeneous in that it can be structured, unstructured and semi-structured data. Relational database management systems (RDBMS) typically expose a query interface based on SQL (Structured Query Language). The RDBMS are mainly for management of structured data and hard to scale out to the ever growing size of data sets. This paper reviews the features of NoSQL database technologies as an alternative to RDBMS for management of Big Data. It evaluates and performance of a RDBMS in comparison with two NoSQL database systems. Addressing 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: Digital forensics, NoSQL database, big data, NoSQL Security, RDBMS

1. Introduction

In the past few years have witnessed an exponential growth in the volume of data on digital forensics leading to big data issues [1]. The dimensions of digital evidence supports have grown exponentially due to the factors that the price drop of hard drives, an ever increasing size of computer storage, a widespread use of mobile devices, and the ubiquity of network connections. In [2], Big Data is defined as a term that encompasses the use of techniques to capture, process, analyze and visualize potentially large datasets in a reasonable timeframe not accessible to standard IT technologies. Basically Big Data is characterized with three techniques:

- Volume, the sheer amount of data generated[7],
- Velocity, the rate the data is being generated,[10] and
- Variety, the heterogeneity of data sources [14].

Big Data also poses challenges to data management. Traditional relational database management systems (RDBMS) such as MySQL are mainly employed for management of structured data. They typically expose a query interface based on SQL (structured query language). Postrelational database systems which are coined as NoSQL (Not Only SQL) [4] have emerged for management of semistructured and unstructured data. NoSQL database systems mainly have the following advantages over traditional RDBMS:

- Scale-out onto economical commodity servers. When the data size grows, instead of installing expensive large database servers, the horizontal scalability of NoSQL database systems proves cost effective by making use of clustered inexpensive commodity servers.
- Handling big data, including the ability to replicate and to distribute data over many commodity servers.
- Dealing with faults in server failures.
- Support of flexible data models. In focus with the structured data requirements of RDBMS, NoSQL databases virtually support any data structure.

Redis forensics is of great importance in many aspects.

First, Redis is widely used in many companies to store large amount of data and would thus be a primary target in a forensic investigation. Second, some data in Redis might be mistakenly removed by database users. Third, Redis is a potential target of database intrusions that involves stealing or tampering the database data. The data recovered in Redis could be used to prove a database security breach and determine the scope of a database intrusion. The study on R.F could help the study on some other NoSQL databases with similar key-value storage mechanism like Riak and Cassandra. The study on Redis forensics also provides insight into the forensic techniques for some other memory databases. We could analyze the disk backup file instead of the memory to extract the database data. This paper reviews two techniques to facilitate scalability in data management. MongoDB and Riak evaluate their performance in data management in comparison with MySQL which is a traditional RDBMS.

2. Related Work

Many papers that issued the relationship between Relational and NoSQL databases were given 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] also in [12] the discussion about the structured and non-structured database also explained paper 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. 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.

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.

Also in demand of Cloud computing is a model for enabling ubiquitous, convenient, network access to a shared pool of configurable computing resources (e.g., network, storage, servers, applications, and various services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [19].

It denotes a model in which a computing infrastructure is viewed as a "cloud", from which businesses and individuals can access applications on demand from anywhere in the world [21]. Essential characteristics of the cloud-computing model, according to the U.S. National Institute of Standards and Technology (NIST), include [19]:

- On-demand self-service, enabling a user to access cloud provider services without human interaction;
- Broad network access that enables heterogeneous thick and thin client applications to access the services;
- Pooling of service provider computing resources to serve multiple consumers;
- Automatic, rapid, and elastic provisioning of resources, Measured service.

Overall, a cloud computing model aims to provide benefits in terms of lesser up-front investment in infrastructure during deployment, lower operating costs, higher scalability, ease of access through the Web, and reduced business risks and maintenance expenses [20].

Pokorny [17] have also reviewed NoSQL data stores. They portrayed a number of NoSQL data stores, describing their data models and their main underlying principles and features. However, in contrast to this work, they did not perform direct feature comparison among data stores.

Sadalage and Fowler [18] described the principles on which NoSQL stores are based and why they may be superior to traditional databases. They introduced several solutions, but they did not compare features as is done in this work.

3. Data Replication and Sharding

Data replication and sharding are two techniques that can be used to enforce scalability.

3.1 Replication

With replication, the same portion of data is replicated to one or more data nodes. The redundancy of data in a computer cluster helps to achieve availability, scalability and to improve the performance of a system. NoSQL databases normally follow either master-slave architecture or a peer-to-peer (P2P) architecture. The master-slave architecture is composed of one master (also called primary) data node and a number of slave (also called secondary) nodes. All write operations are only served by the master node and read operations can be served by both the master and the slaves. Asynchronous replication on the other side provides lower latencies because data is replicated later on. But asynchronous replication could lead to inconsistent reads.

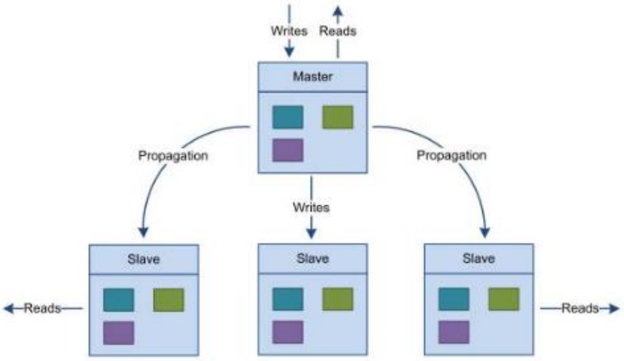

Figure 1: The master-slave architecture for data replication.

An alternative architecture is the P2P model as shown in Fig.2. With this architecture, all nodes play an equal role and can serve read and write requests.

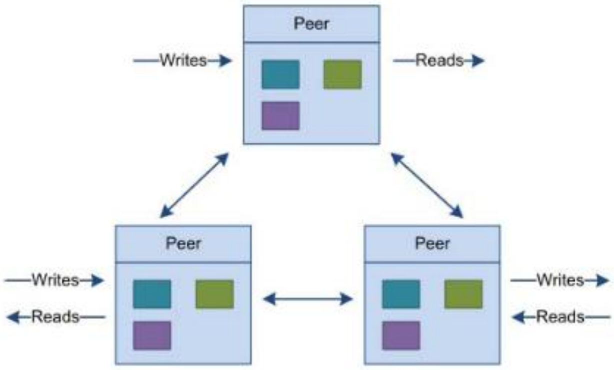

Figure 2: The peer-to-peer architecture for data replication.

In contrast to the master-slave architecture, the P2P architecture can improve the write performance by adding more nodes to the system. Also, the P2P architecture does not have a single point of failure and a bottleneck in data management.

3.2 Data Sharding

Sharding is a technique of distributing different pieces of data onto a number of data nodes which are in this case called shards. Each data shard is independently from others which is often referred to as shared-nothing architecture as shown in Fig.3

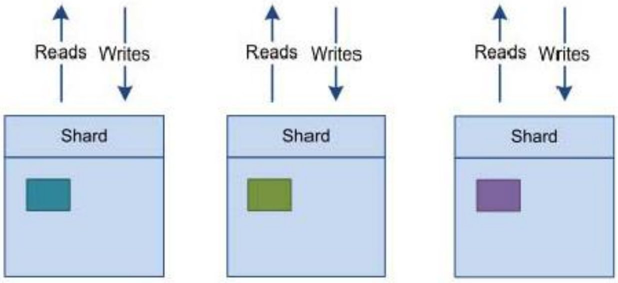

Figure 3: Shared nothing data shards.

A point out that data sharding provides a way of scaling out write and read operations also mentioned that the arrangement of data is important. Data should be placed close to the point where it is being accessed to improve performance. Obviously, sharding has some benefits but it can potentially cause some problems on data resilience. Once a shard fails the data on this shard is lost.

4. Consistency

4.1 The CAP theorem

In order to store and process massive datasets, a common employed strategy is to partition the data and store the partitions across different server nodes. Additionally, these partitions also be replicated in multiple servers that why the data is still available even in case of servers' failures. Many modern data stores, such as Cassandra and BigTable, use these and others strategies to implement high-available and scalable solutions that can be leveraged in cloud environments. Nevertheless, these solutions and others replicated networked data stores have an important restriction, which was formalized by the CAP theorem: only two of three CAP properties (consistency, availability, and partition tolerance) can be satisfied by networked shared-data systems at the same time.

4.2 Quorums

Quorums is a technique that some P2P NoSQL databases use to implement the level of consistency in a replicated database. The first algorithms in this field were proposed by them. The basic question with quorums is how many nodes need to be contacted to get a read or write quorum, the three values N (number of replicas), W (number of nodes that must acknowledge a successful write), and R (number of nodes that must acknowledge a successful read) need to be considered. To achieve strong consistency, the rule is defined using Eqs.(1) and (2)

$$W + R > N \qquad (1)$$

$$W > N/2 \qquad (2)$$

5. 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. The data inside the database represented as tuples and grouped into relations. This data model is very specific and well organized. 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.

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.

D. Cloud:
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.

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.

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.

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.

6. Conclusion

In this paper we reviewed the concepts of NoSQL databases different in many aspects from traditional databases like structured schema, transaction methodology, complexity, crash recovery and dealing with storing big data.

Also this paper reviewed NoSQL database technologies which can be used for management of Big Data. It introduced MongoDB and Riak as two representatives NoSQL databases and evaluated their performance in read and update operations. The testing results showed that Riak performs better than MongoDB in dealing with large datasets, but MongoDB outperforms Riak on reasonably small datasets due to its in memory processing.

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-

[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.

[15] Amazon EC2 Cloud, https://aws.amazon.com/ec2/

[16] D. Ghoshal, R. Canon and L. Ramakrishnan, "I/o Performance of Virtualized Cloud Environments", Proc. of the 2nd International Workshop on Data Intensive Computing in the Clouds, DataCloud-SC '11, pages 71–80, 2011.

[17] Pokorny J (2011) NoSQL Databases: a step to database scalability in Web environment. Int J Web Info Syst 9(1):69–82

[18] Sadalage PJ, Fowler M (2013) NoSQL distilled: a brief guide to the emerging world of polyglot persistence. Addison-Wesley, Upper Saddle River, NJ

[19] Mell P, Grance T (2011) The NIST definition of cloud computing. NIST special publication 800–145, http://csrc.nist.gov/publications/nistpubs/800145/SP800-145.pdf. Accessed on 29 Sep 2013

[20] Zhang Q, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. J Intern Serv Appl 1:7–18.10.1007/s13174-010-0007-6

[21] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gen Computer Syst 25(6):599–616, http://dx.doi.org/10.1016/j.future. 2008.12.001