VOL 2 (2018) NO 2

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Comparison of NoSQL Database and Traditional Database-An emphatic analysis

M. Sandeep Kumar[#], Prabhu. J[#]

[#] *School of Information Technology and Engineering Vellore Institute of Technology, VIT University Vellore, India*
E-mail: sandeepkumarm322@gmail.com, j.prabhu@vit.ac.in

Abstract— **A Huge amount of data is manipulated by using the web application, Facebook, Twitter, social sites etc. Most of the data are unstructured data. It is not desirable for storing, performing and analyzing data in relational database for huge data. It affords way towards performing NOSQL database and uses fully for handling the big data. In this paper, we present the performance in store and query operation in NoSQL database, estimating the performance of both reads and write operation using simple and complex queries. Result represents that comparing Cassandra with relation database, Cassandra outperforms the relation database. Most of the organization used only Hbase and Cassandra for benefit of cost. Comparison Various NOSQL Database, issues while performing NOSQL database.**

Keywords— **Relational Database, Big Data, Store, Queries.**

I. INTRODUCTION

Now a day's most of the data are proliferation from various sources like the internet, social site, web application etc. All data are needed to store in some relational technology with known insufficient for the same [1]. Most data are moving towards cloud storage but it also provides more security issues. The main process of big data is like capturing, storage, processing, and interpretation. Big data comprises of four main characteristics like volume, velocity, veracity, and variety [2]. NoSQL plays a crucial role in analyzing and storage of data. Cloud computing acquired as a new platform, permitting the user to use their source based on their requirements. Some of the traditional methods to perform this are by using resources like grid and cluster computing. Usage of cloud based on elasticity it offers various services providers like Infrastructure as a service (IaaS), Platform as a service(PaaS), Software as a Service (SaaS) are the model used by both public and private. Elasticity, scalability, efficiency, and reusability are some of the advantages of using cloud computing. In this paper, we are discussing various NOSQL database like HBase, MongoDB, Cassandra. It presents some benefits and limitation of NOSQL database. The operation of read and write mainly depends on performance and scalability with both simple and complex queries. Comparison of various NOSQL database, issues while performing NOSQL database are discussed.

II. RELATED WORK

Hecht and Jablinski [3] presented vital characteristics provided by various NOSQL results like Availability and scalability. Konstantinou et al describe a clear study on Reading and updates operation perform in the Various database in Cassandra, HBase, and Riak. In HBase offers More Elasticity and quick read operation but in Cassandra has the ability to delivery quick insertion in a write operation. Whereas Meanwhile, Riak demonstrates in scalability and performance, considering the type of access. Most of the researchers focus on the performance of Evaluation.

Chieh-Ming Wu et al [4] put forth the performance of both NOSQL MongoDB and MS-SQL database. Index, Replication, Sharing, query, Modern Elastic data mode are some requisites provided by most of the internet application. By comparing the performance of both databases, NO-SQL provides more efficiency than MS-SQL. Due to fast improvement in big data, further trend depends on integration based on NOSQL. It performs with the various modern technologies used in SQL and NOSQL selection. Appliance and effectiveness

Veronika Abramova et al [5] describes two famous NOSQL database like MongoDB and Cassandra, Performance analyzes and evaluate by time, the size of the database. Read/update, read alone, read modify write, mix read/update .update alone are some test perform in workload. Due to increase in the size of data MongoDB begin to decrease performance, show low result but Cassandra while

using with more data. After performing with various workloads need to analyze both read/update performance, it has the potential for the update operation. Cassandra is quicker than MongoDB gives less performance time and size of the database are evaluated. But performing entire analysis MongoDB fails less with more record used, but Cassandra has more offers compare with MongoDB. In future author suggest performing more number of operation per second versus the size of the database. It provides in understanding more number of records ingrown data for reading and update operation.

Need Acknowledgement in performing the NOSQL system based on CAP theorem[6].Brewer describes the CAP theorem it states no distributed system like consistency, availability, partitioned tolerance. Consistency based on whole nodes view in similar data in the same duration [7]. Most of the Databased pick only" AP" availability and partition-tolerances, partition-tolerance will not tradeoff, availability, consistency is balanced, in most of the database providing more consistency compared to availability[8].Advance NOSQL system based on CAP theorem with a database like "CA-CP, and AP".

This paper describes in depth comparative analysis of NOSQL Database with a traditional database. Remaining organized as follows: In next section 2: related work. In Section 3: Important characteristics of NOSQL, various NOSQL Database, and advantage and disadvantage over NOSQL Database. In section 4: Query difference and comparison of various NOSQL database. In Section 5: Evaluation and result of NOSQL Database. In Section 6: Conclusion describes best performs result in NOSQL database over the traditional database

III. Characteristics Of NOSQL Databases

In NOSQL database first introduced Big Table [9], it is based on three key values like row, column, and timestamp. It used for making an effective multidimensional mapping, Column key is categories into groups; it performs by a single unit. Big Table and Amazons' Dynamo [10] are some of the non-relational databases, it performs in both source like open and closed. Ease access, velocity, scalability are growing popularity in NOSQL database. NOSQL data based mainly depends on key-value pairs; it has potential to make as secondary key contains values. Column family data based depends on key value pair. The key address consists of the column and super column. Super column consists of the column and it performs only in solitary units some other key type based on the document-oriented database, it consists of an easy value and has the capability to preserve object. XML, JSON, and BSON are some of the format used in Objects.

The unique characteristics of NoSQL Database are listed below.

- Large amount of data set stored in NO SQL compared to SQL
- SQL language not used by NO SQL
- NO SQL perform without help of inconsistency in distributed platform
- It will not affirm with ACID Properties
- It gives more flexible structure compare to SQL
- It leads to more performance due to horizontal scalability.
- All data stored in NOSQL without featuring fixed Schema

A. MongoDB

Mongo DB stores data depends on the document in the database where document are group into collections, it depends on structure only but in some scenario, the document can store in a different structure. Mongo DB follows some standard format in storing document like BSON-Binary JSON, the size of each restricted to 16MB.It finds based on defined type. Durability, Concurrency is important features of Mongo DB. The introduction of replication generally allows the data in durability. It employs by using Master-slave replication Mechanism, it grants a master and one or more slaves. Master will read or write files when slave assists in supporting reading operations. Once master fails, the slave will raise to master whole replicates will be asynchronous, entire updates will not separate directed. Entire replication member will be configured by the administrator in many ways as follows.

Secondary Only Member: Replication will store data but it will note courage to master under some condition.

Hidden Member: Hidden replication will not get primary and unseen to the user application. Most of the member allows committing backup and it read testing only.

Delayed Member: It copies primary unit from replica by a particular delay when data replica equivalence with the previous master, it will not similar to the previous update.

Arbiters: Only internal participate only with communicate with other members.

Non-Voting members: In election replica will not take part it many performing huge cluster. It may have more than 7 members.

B. Cassandra

It is open source database it determined by key value, column family database due to its consistent like dynamo has stores data in column family like the big table. Cassandra implements for managing the large structured data and it usable in Apache. Some properties in Cassandra-like it capable of making elastic and linearly. Cassandra performance will increase when a number of nodes present in the cluster. Similar to a relational database, Cassandra affirms ACID properties, it performs quickly in write. It also affirms data distribution in replicating the data over data centers. The language used in Cassandra is Java.

C. Hbase

It is column family database it determines by using Java, similar to Big data table[9].It implements on top of HDFS, it allows feature of the big table like Fast in processing both structured and unstructured storage data in HDFS. Meanwhile, it determines only in batch processing. It also has additional functionality increasing the column index similarity. Hbase is similar to Cassandra, it is linear and scalable determine them by master-slave. When Various HMaster server applied on the failure of HMaster and then remaining load in region server another Hbase automatically does their work. HBase allows to auto failure affirm single fails occur in some case Hmaster also perform in some scenario. When both read and write operation performing application using big data.

Table 1 shows the advantages and disadvantages of these systems over NoSQL database.

TABLE 1
ADVANTAGES AND DISADVANTAGES OVER NOSQL DATABASE

Advantages	Disadvantage
Simple in using Scalable	Immature
It does not need database administrators	Quick, flexible and high efficient
It performs with more Space	Difficult in maintenance
Huge range of data model	Not having standard query language
NoSQL, DBaaS gives like Riak, Cassandra is programmed for dealing with the failure of hardware.	Few NoSQL database are not having complaint

IV. QUERYING DIFFERENCE IN NOSQL DATABASE

MySQL, Oracle are some of the relational databases, it used to perform an operation like storage, retrieval, data manipulation but in NoSQL, it performs with solitary query language it receives variation in user demands. In NoSQL database are stored with the particular data model.

Table 2 illustrates the difference in read, write, delete operations performed in various NoSQL databases like MySQL, MongoDB, Cassandra, and HBase. The sample queries for each operation are portrayed in the table.

Table 3 shows the comparative analysis of various NoSQL databases by considering various analytical parameters including programming language used, fault tolerance, storage media used, community genre, MapReduce framework that is followed, Type of query language used, various available modes of replication, and protocol used in transport and application layers of the system.

Various issues of NOSQL database like Authentication, Authorization, Attacks, Protocols [13] is shown in Table 4. It discusses the various issues occurred in Redis, DynamoDB, Voldemort, MongoDB, CouchDB, Cassandra, HBase, HyperTable, and Neo4j

TABLE 2
QUERY DIFFERENCE IN READ, WRITE, DELETE

Database	Read	Write	Delete
My SQL	Insert into journal values (‘scope’,234, both open and close)	Update set of Journal Id=345where e357	Delete from list of journal where Name=’pharmacy’;
MongoDB	Db .journal .insert ({jname:”scopex”,id:235,type:”both open and closed”})	Db. journal up data({}(‘$set ’ :id’ :id’});	Db. Journal .delete ();
Cassandra	Insert into journal values(‘scopex”.234,’both open and close’);	Update Set of journal Id=234where id=350;	Delete set of journal id=234whereid350
Hbase	Assign journal’,’row2’,’id: a’,’scope’,	Similar to insert operation	Disable ’set of journal’

TABLE 3
COMPARISON OF VARIOUS NO SQL DATABASE

DB/ Properties	MongoDB	Cassandra	Accumulo	Couch DB	Hbase	Redis	Riak
Language	C++	Java	Java	Erlang	Java	C,C++	Erlang
Data Model	BSON	Big Table	Big Table	JSON	Big Table and Dynamo	Data Structure	Data structure
Fault Tolerance	Replication	Partitioning and replication	Replication	Replication	Partitioning and replication	Replication	Replication
Data Storage	Memory, file framework	Dynamo for storing data	HDFS	Memory, File framework	HDFS	File system	Bit cast, Memory
Community	AGPL	Facebook	Apache	Apache	Apache	BSD	Apache
MapReduce	YES	YES	Yes	YES	YES	NO	YES
Query Language	-	API calls	Java API, Thrift API	-	XML, Thrift API	API calls	Javascript
Replication Modes	Master-Slave Replication	Master-Slave replication	Multi-master replication	Multi-Master Replication	Master-Slave Replication	Master-Slave Replication	Multi Master Replication
Protocol	TCP/IP	Thrift	Thrift	HTTP/ REST	Thrift, API, tradition	Binary, Similar to telnet	REST

TABLE 4
ISSUES IN NO SQL DATABASE

NOSQL DB	TYPE	AUTHENTICATION	AUTHORIZATION	ATTACKS	PROTOCOL
Redis	Key Value	Small layer	It will not support	-	No Encryption
DynamoDB	Key Value	It will gives upport	-	-	Https
Voldemort	Key Value	It will not support	It will not support	-	-
MongoDB	Document Based	It will not support	It will not support	Script Injection	SSL
CouchDB	KeyValue	It will give support	-	DOS	SSL
Cassandra	Document Based	It will give support	-	Both	SSL
Hbase	Column Based	It will give support	It will give support	Doesn't give any report to DOS and Injection	SSH
HyperTable	Column Based	It will not support	-	-	-
Neo4j	Graph-based	-	It will not support	-	SSL

V. EVALUATION OF RESULT COMPARING WITH NOSQL AND MY SQL STORAGE DATA

Evaluation based on a number of records used in both read and write operation in Cassandra and MySQL. Cassandra show improved result compared to MySQL. Moreover, other types of NoSQL database not used because of time constraints.

A. Workload generator

It based on analysis and performance of a benchmark, one application requires performing continuous with more stream data. We can with generate any application having more unstructured data. Both read and write operation perform in an application based on benchmarking.

B. Workload Executor

Based on two type

- Write
- Read

a. Write phase:

It loads set of record to both databases using JDBC connectivity.The user creates a various thread to load data in parallel with both databases like Cassandra, MY SQL. A thread increases the number of throughputs.

b. Read phase:

Data Load in the database when load phase occurs. It performs some queries while reading data's in the cluster. Queries can be retrieval in both datasets using simple query. "select to a complex query.

C. Metrics collection

It collected by logging and writes an application to storage and dashboard. Monitoring performance the time using a time stamp. Record with varies in write and read operations.

Bench Marking in NO SQL database

YCSB

YCSB executed based on command lines and it produces a number of threads and queries for framework below test. It evaluates based on throughput and it operation per second and Latency performs in this operation based on record. YCSB run based on Performance, scalability, elasticity availability, replication.

D. Process of load

By using benchmarking huge data load in each workload. In database permit to generate non-durable write operation for this process for data need to quick as much as possible.

TABLE 5
PERFORMANCE OF WRITE OPERATION

Record	Cassandra(MS)	record	MYSQL
100	1	100	5
200	2	200	9
500	4	500	19
1000	8	1,000	43
10,000	60	10,000	400
100,000	456	100,000	3,000
200,000	918	200,000	6,000
500,000	2280	500,000	15,000
100,0000	4560	100,0000	30,000

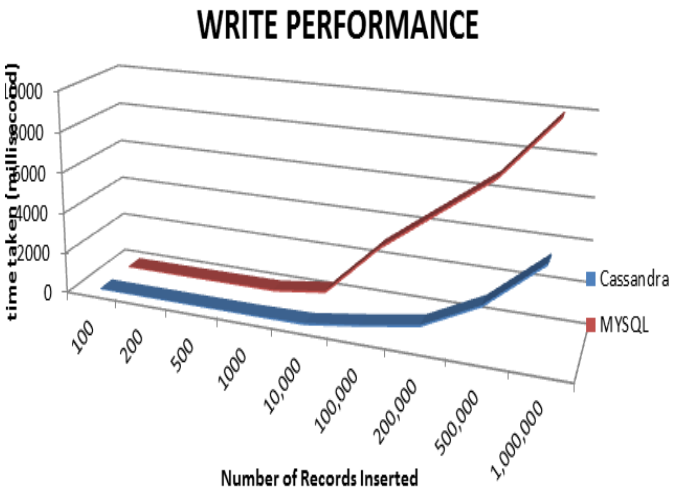

Fig: 1 Write Performance

a. Process of Retrieval

Duration of retrieval record is more in Cassandra, it gradually more in MySQL with more number of records places in hardware configuration.Moreover, MySQL gives better result in retrieval process compare to Cassandra.

TABLE 6
PERFORMANCE OF READ OPERATION

Record	Cassandra(MS)	Record	MYSQL(MS)
100	2	100	1
200	3	200	2
500	5	500	2
1,000	8	1,000	5
10,000	10	10,000	6
100,000	12	100,000	8
200,000	24	200,000	16
500,000	60	500,000	40
100,0000	120	100,0000	80

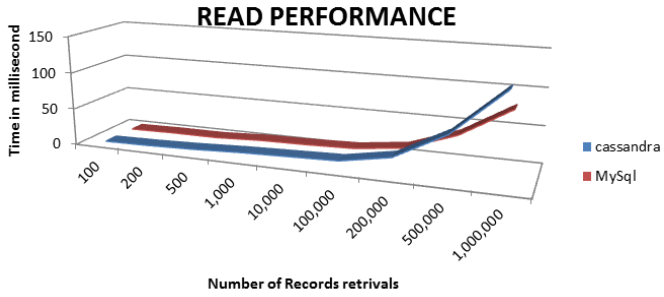

Fig: 2 Read Performances

VI. CONCLUSIONS

Many organization depends on constructing of database like MYSQL it does not tackle the demands of scalability and availability of real data. NoSQL database affirms with scalability, consistency, availability and fault tolerance. In this paper discussing various NoSQL database and comparing advantage, limitation, a solution with MYSQL and NOSQL. Now a day most companies using NOSQL database like Cassandra, Mongo dB etc. The advance world demands similar to big data it has the ability to perform, analysis and interpreted by combining with NOSQL database based on analyzing queries.

ACKNOWLEDGMENT

A special note of thanks to Vellore Institute of Technology (VIT) University for providing necessary infrastructure facilities to carry out the research work and my friends who have helped me to complete this study.

REFERENCES

[1] Gupta, S., & Narsimha, G. (2017). Efficient Query Analysis and Performance Evaluation of the NoSQL Data Store for Big Data. In Proceedings of the First International Conference on Computational Intelligence and Informatics (pp. 549-558). Springer Singapore.

[2] Gudivada, V. N., Rao, D., & Raghavan, V. V. (2014, June). NoSQL systems for big data management. In Services (SERVICES), 2014 IEEE World Congress on (pp. 190-197). IEEE.

[3] Hecht, R., & Jablonski, S. (2011, December). NoSQL evaluation: A use case oriented survey. In Cloud and Service Computing (CSC), 2011 International Conference on (pp. 336-341). IEEE.

[4] Wu, C. M., Huang, Y. F., & Lee, J. (2015). Comparisons between MongoDB and My-SQL databases on the two websites. American Journal of Software Engineering and Applications, 4(2), 35.

[5] Abramova, V., & Bernardino, J. (2013, July). NoSQL databases: MongoDB vs Cassandra. In Proceedings of the International C* Conference on computer science and software engineering (pp. 14-22). ACM

[6] Brewer, E. A. (2000, July). Towards robust distributed systems. In PODC (Vol. 7).

[7] Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. Acm Sigact News, 33(2), 51-59.

[8] Wada, H., Fekete, A., Zhao, L., Lee, K., & Liu, A. (2011, January). Data Consistency Properties and the Trade-offs in Commercial Cloud Storage: the Consumers' Perspective. In CIDR (Vol. 11, pp. 134-143).

[9] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., & Gruber, R. E. (2008). Big table: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 26(2), 4.

[10] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., ... & Vogels, W. (2007). Dynamo: amazon's highly available key-value store. ACM SIGOPS Operating systems review, 41(6), 205-220.

[11] Nayak, A., Poriya, A., & Poojary, D. (2013). Type of NOSQL databases and its comparison with relational databases. International Journal of Applied Information Systems, 5(4), 16-19.

[12] Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M., & Bernardino, J. (2015). Choosing the right NoSQL database for the job: a quality attribute evaluation. Journal of Big Data, 2(1), 18.

[13] Sahafizadeh, E., & Nematbakhsh, M. A. (2015). A Survey on Security Issues in Big Data and NoSQL. Advances in Computer Science: an International Journal, 4(4), 68-72.

[14] Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2013). Comparison and classification of NoSQL databases for big data. International Journal of Database Theory and Application, 6(4.2013).

[15] Agrawal, D., Das, S., & El Abbadi, A. (2011, March). Big data and cloud computing: current state and future opportunities. In Proceedings of the 14th International Conference on Extending Database Technology (pp. 530-533). ACM.

[16] Grolinger, K., Higashino, W. A., Tiwari, A., & Capretz, M. A. (2013). Data management in cloud environments: NoSQL and NewSQL data stores. Journal of Cloud Computing: Advances, Systems, and Applications, 2(1), 22.

[17] Kulshrestha, S., & Sachdeva, S. (2014, August). Performance comparison for data storage-Db4o and MySQL databases. In Contemporary Computing (IC3), 2014 Seventh International Conference on (pp. 166-170). IEEE.

[18] Sandholm, T., & Lee, D. (2014). Notes on Cloud computing principles. Journal of Cloud Computing, 3(1), 21.