

Performance Evolution of Wide Column Store NoSQL Database in non-distributed environment

¹Mrs.Rohini Ganesh Gaikwad, ²Dr.Amol C. Goje

¹Research Scholar, ²Research Guide

¹Neville Wadia Institute of Management Studies & Research,
Pune, Maharashtra, India

²Neville Wadia Institute of Management Studies & Research,
Pune, Maharashtra, India

Abstract

As the high volume of a variety of data generation creates the hurdle to process, store, and handle with traditional database systems. To deal with such mixed format data many organizations currently using NOSQL database technology along with RDBMS as well as using various analytical tools in support. Due to its scalability property, the evolution of NOSQL databases performed with a multimode cluster environment. Still, there are startup organizations that are seeking for use of NOSQL databases in their projects with minimum infrastructure. This paper describes about the wide column store database –Cassandra which is widely used by the industries due to its huge set of characteristics. This research paper details the performance evaluation of Cassandra on standalone infrastructure with three different datasets and with various client threads. The parameter for comparison consists of Throughput, Runtime, and Latency. Also at the end Statistical Analysis of results with ANOVA has been presented.

Keywords: *NOSQL, Column Store, Performace, Evaluation, Ycsb, Throughput, Latency*

1. Introduction

IDC forecasts growth in wearable devices from 2018 – 2019 in 3%; There were 28.3 million wearable devices sold in 2016 and estimates that 198 million will be sold in 2019. They predict that, between 2016 and 2022, IoT devices are expected to increase at a rate of 21 percent, driven by new use cases. In 2018, mobile phones are expected to be surpassed in numbers by IoT devices, which include connected cars, machines, meters, wearable, and other consumer electronics.

Much of that data is unstructured, in terms of Documents, photos, audio, videos, and other unstructured data can be difficult to search and analyze.

The IDG report found that from 2015 to 2016 14% grows unstructured data. The data generated from multiple sources like IoT, Internet, Mobile devices, Communication, Social Media, and sensor devices. That data obtained in multiple formats so-called semi-structured and

unstructured data. To manage this volume of a variety of data at real-time process new technologies emerge as NOSQL that is “Not Only SQL”, Hadoop Framework, and its various tools.

Nowadays there are more than 225 NOSQLdatabases are available. These databases are categorized into four main categories as below:

- 1) Document Store Database
- 2) Column Wide store Database
- 3) Key Store Database
- 4) Graph Store Database

The main focus of this paper is on the Wide Column store database –Cassandra which is widely used by many applications like social networking websites, banking, and finance, real-time data analytics, online retail, etc.

The remaining part of this paper is organized as follows. Section 2 Literature Review, Section 3 Cassandra summary, Section 4 Experimental Evaluation Section 5 Statistical Analysis, Finally Section 6 presents conclusions.

2. Literature Review

For conducting literature review various oapers,white papers collected from Research gate, Academia,IEEE explore etc in the duration of 2011 to 2019.

[1] Testing and evaluation are done with Mongoddb, HyperDb, and MySQL.[2][4] present the theoretical differences of NOSQL databases.

[2] Comparative analysis table of NOSQL databases DynamoDB, Riak , Voldermort, Tokyo Cabinet, CouchDB MongoDB, RavenDB, Cassandra, Hbase, Neo4j has been illustrated.

[3] presented the comparison of NOSQL databases out of which MongoDB, Redis, and OrientDB are databases optimized to perform read operations, whereas Colum Family databases, Cassandra and HBase, have a better performance during execution of updates.

[4] The analysis of MongoDB,Redis, Scalaris, Tarantool and OrientDB , Cassandra and HBase done and noted that Tarantool is the best database as it shows good execution times for all types of workloads.

[5] Comparison of Mongoddb,Cassandra and HBASE performed. This paper also demonstrated evaluation of cassandra for an industry specific use case and results are published.

[6] presented their research work on Cassandra, Mongoddb and Hbase. The comparison factors are classified as Node Capacity, Number of nodes and Replication.

[7] provides the use cases of Mongoddb,Hbase,Redis,Dynamo,Cassandra and Couchdb.

[8] authors conducted experiment on three NOSQL databases Mongoddb,Cassandra and Raik. Tests performed on application of Electronic Health Record (EHR) system.[17]

author evaluated MongoDB, Cassandra and Couchbase. MongoDB provides greater performance than Couchbase or Cassandra.

[9] Experimented with MongoDB, Cassandra and HBase. YCSB was used for tests. The focus of test was on horizontal scalability under different workload conditions with varying dataset sizes.

[10] used 5 virtual machines for testing Redis, MongoDB, Couchbase, Cassandra, HBase with 3 workloads. In this all three tests, Redis showed the best performance.

[11] MongoDB, Cassandra and Redis selected for experiments of integrity and availability. Result displayed that integrity of the data is affected, even in the presence of simple faults.

[12] provides comparison of in memory databases, MongoDB, Redis, Memcached, Cassandra and H2. Sorted result for Read-Cassandra, Redis, Memcached, MongoDB, H2. Cassandra provides efficient memory usage.

[13] presented the comparison and analysis of MongoDB and CouchDB for Twitter use case. For all four operation count, overall MongoDB performs well for these kind of applications.

[14] provides the comparison between MongoDB and Couchdb. The results suggests that MongoDB works better than CouchDB for CRUD operations

[15] provides an analysis and performance evaluation of Cassandra, Hbase, and MongoDB for E-Health clouds. Among the three databases,

Cassandra found to be best solution for E-Health clouds since provides higher throughput, HBase works well for complex read and write operations.

[16] Performs evaluation on 3 servers with MongoDB and Cassandra for 2 workloads. Conclusions approve that MongoDB performs better than Cassandra for workload B i.e mostly read workload, while Cassandra beat MongoDB for workload Q i.e update heavy workload.

[17] perform experiment on MongoDB, Couchbase, Cassandra, HBase. Different size of record sets considered as 1000, 5000, 10000, 50000.

Various work on performance evaluation have been done and yet many of the researcher working on different aspects of database performance. As the need for high performances arises the small industries also seeking for use of NOSQL database for their project work with small amount of infrastructure. This study focus on this issue and perform the research work in to this area of problem.

3. Database Configuration:

In column family data stores, data is arranged in the form of columns, and a set of columns produces the row. It uses the concept of "Keyspace". Create a Keyspace command contains strategy and replication factor as basic parameters. The keyspace contains all the column families, which contain rows, which

contain columns single row can contain n number of columns corresponding to it. The Column family represents the group of related data contents, and it can be accessed together. In column family stores a key identifies a row and a row can have multiple columns. In column family stores, all the rows do not need to have the same columns and a column can be added to a row at any time without affecting other values. It is designed for rows with many columns and can even handle millions of columns. Columns can be nested within columns called super columns. Further, it is simple and effective to use in a real-time scenario.

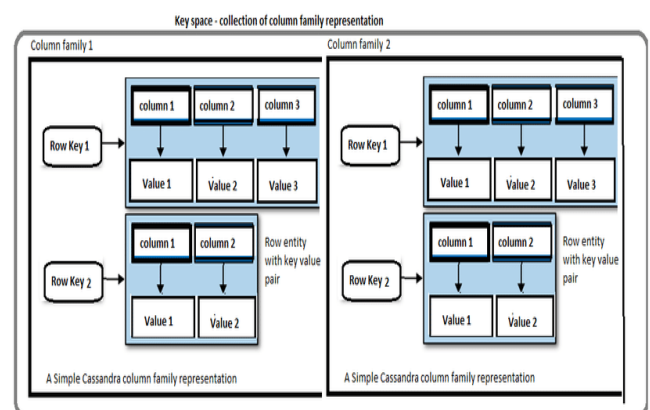
Apache Cassandra was developed by Apache Software Foundations and was released in 2008. It was developed using Java. It is based on Amazon's Dynamo model as well as Google's Big table. Because of that, it contains concepts of key-value stores and column stores.

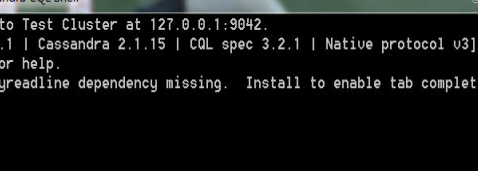
Cassandra system is a distributed database system which was composed of lots of database nodes. In a Cassandra cluster, for achieving scalability more nodes can be added. Cassandra also supports rich data structure and powerful query language.

Cassandra is being used by Adobe, Digg, eBay, Twitter, etc. It can be used for a variety of applications like social networking websites, banking, and finance, real-time data analytics, online retail, etc. [4] [11] [13][26] [28][29]

FEATURES OF CASSANDRA:

- 1) It has a dynamic schema.
- 2) Cassandra has a peer-to-peer distribution model,
- 3) Cassandra datasets are partitioned horizontally by consistent hashing
- 4) support range queries.
- 5) High scalability: a single point of failure does not affect the whole cluster, and it supports linear expansion.
- 6) Cassandra is often communicated as being an eventually consistent data store.
- 7) Cassandra uses its own *CQL Cassandra query language* to interact with its column family data model.
- 8) Cassandra offers atomicity at the column family level, it does not guarantee isolation and no locks.
- 9) It offers a feature like high availability, partition tolerance, persistence.
- 10) For intra communication, Cassandra uses a gossip protocol so that each node can have state information about other nodes.

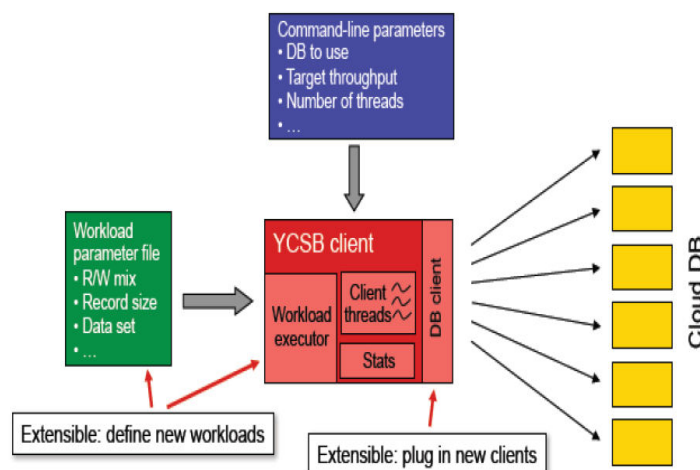




```
Select Cassandra CQL Shell
Connected to Test Cluster at 127.0.0.1:9042.
[cqqlsh 5.0.1 | Cassandra 2.1.15 | CQL spec 3.2.1 | Native protocol v3]
Use HELP for help.
WARNING: pyreadline dependency missing. Install to enable tab completion.
cqqlsh>
```

The above figure displays the Cassandra interface through which CQL commands can run.

Author used an existing tool provided by Yahoo, called the YCSB, to execute these benchmarks. A important design goal of this tool is extensibility as it can be used to benchmark new cloud database systems. Author have used this tool to measure the performance of Cassandra. This tool is available under an open source license. It has ready adapters for different NoSQL Databases. YCSB tool allows benchmarking multiple systems and comparing them by creating “workloads”. Using this tool, one can install multiple systems on the same hardware configuration, and run the same workloads against each system. The architecture of YCSB is as shown in figure 2



5. Experimental Evaluation :

For experimenting standalone machine with Windows 10, 500GB HDD, 8 GB RAM with Intel Core i5 have been used. Three different datasets have to be generated for conducting the test as 0.1 Million, 0.3 Million, and 0.5 Million record size. And Five different workload namely Workload A(50/50), Workload B (95/5), Workload C(100/0), Workload W(5/95), and Workload H(0/100). Experiments performed in two phases as the Load Phase and Run Phase. The Run Phase evaluated with different 11 client threads.

a) Threadcount Vs throughput

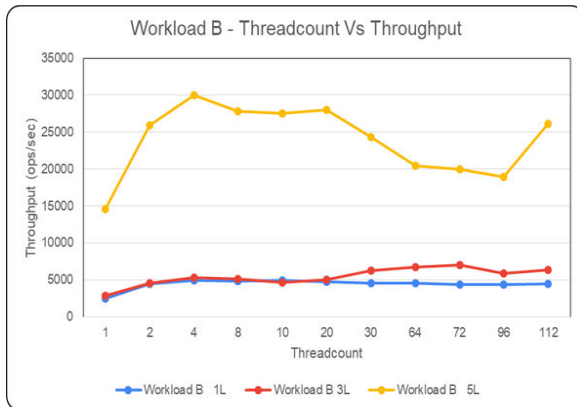


Fig 3: Workload A- Thread Count Vs Throughput

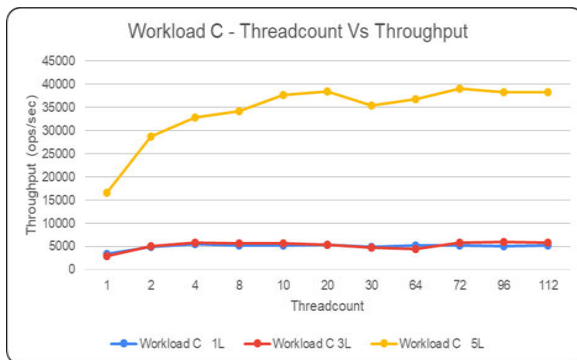


Fig 4: Workload B- Thread Count Vs Throughput

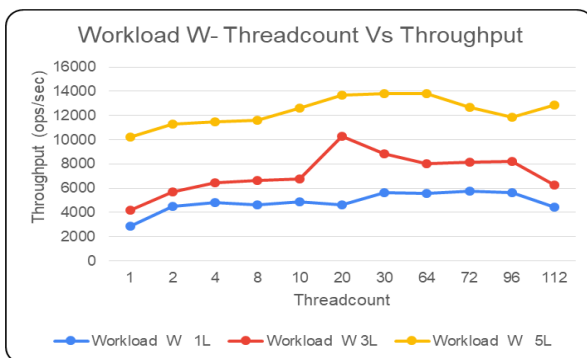


Fig 5: Workload W- Thread Count Vs Throughput

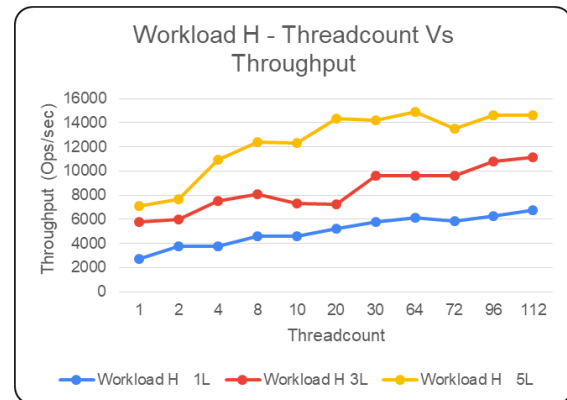


Fig 6 : Workload H- Thread Count Vs Throughput

For Read workloads, in Fig 3, Workload A the throughput for the large dataset is 2X times more than 0.1 million as well 0.3 Million dataset and in Fig 4, Workload B the throughput for the large dataset is 7X times more than 0.1 million as well 4X times more than the throughput of 0.3 Million dataset Comparison shows that the throughput values for 0.5 Million dataset are higher than the throughput values for 0.1 million dataset and 0.3 million dataset.

In fig 5, For Workload C, the throughput for 0.5 Million dataset is 8X times more than throughput for 0.1 million as well 7X times more than for 0.3 Million datasets.

In fig 6, For Workload W, For write intense workload values show that the throughput for 0.5 Million dataset is 2.5X times more than throughput for 0.1 million and throughput of 0.3 million is 2X times more than for 0.1 Million datasets.

Similarly, for Workload H, the maximum throughput for 0.1 million dataset is 6768.70 ops/sec; for 0.3 Million dataset maximum throughput is 11130.45 ops/sec. And for 0.5 Million dataset maximum throughput is 14919.56 ops/sec for thread count 64. These values show that the throughput for 0.5 Million dataset is 2.2X times more than throughput for 0.1 million and throughput of 0.3 million is 1.7X times more than for 0.1 Million datasets.

It has been noted that with write intense workload Cassandra provides better throughput results.

b) Workload Vs Runtime

The following table gives the result of runtime in Sec for workload A-H for three datasets namely 0.1M, 0.3M and 0.5M

Table 1 : Runtime for Workload A-H

Workload	0.1 Million	0.3 Million	0.5 Million
A	28.812	427.345	152.48
B	39.452	91.101	25.12
C	27.422	92.483	34.231
W	36.066	187.32	500.327
H	33.886	178.12	1029.35

From the above figure it has been observed that , For workload B and C cassandra took less runtime whereas for Workload W and H which are leads to the write operation, Cassandra took more execution time as data size goes increases.

c) Operation Count Vs Latency

Here for latency measurement Average latency of Read and Average latency of Write considered for comparison. Out of 11 threads latency considered for first thread only.

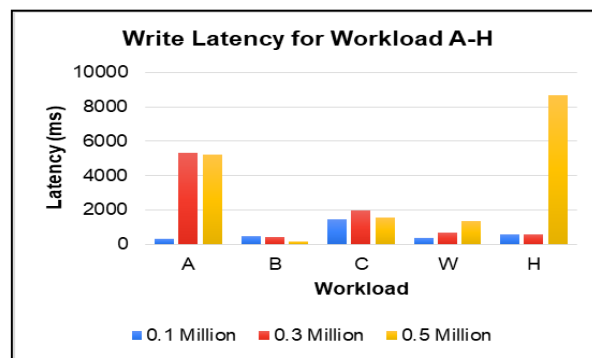


Fig 9: Operation Count Vs Read Latency

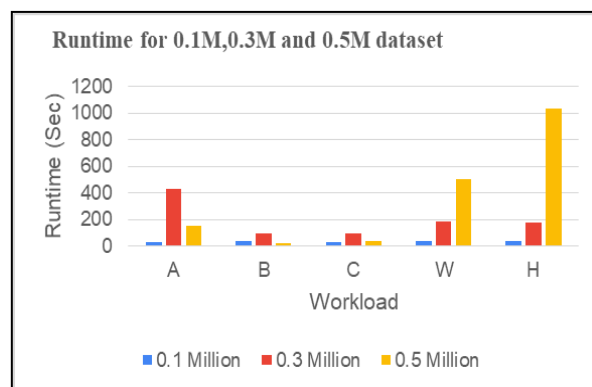


Fig 10: Operation Count Vs Write Latency

Above figure 9 represent Read Latency and figure 10 represent Write latency for workload A-H. In case of read-intense workload the Read latency and Write latency is quite linear but for Write intense workload the write latency is lesser than read latency.

6. Statistical Analysis

The Two Way ANOVA performed on the experiment results in values of throughput. The result of the ANOVA test is presented below:

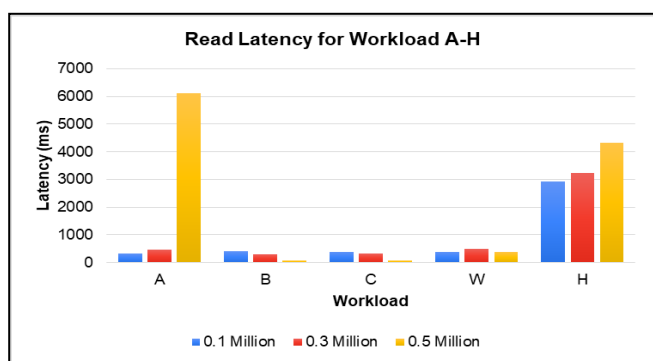


Fig 8: Workload Vs Runtime (sec)

Table: TWO WAY ANOVA DATABASE-CASSANDRA -THREADCOUNT-DATASET

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Sample	1.97E+09	4	4.93E+08	75.27185	7E-35	2.431965
Columns	6.08E+09	2	3.04E+09	464.2017	5.6E-65	3.056366
Interaction	3.89E+09	8	4.86E+08	74.24312	2.71E-48	2.000625
Within	9.83E+08	150	6552695			
Total	1.29E+10	164				

From the said ANOVA table. We find that the differences concerning an Increase in a data size are significant at a 5% level as the calculated value of F i.e **464.2017** which is more than its critical value **3.05636**. Here p-value is 5.6E-65 which is less than 0.05, it means differences between means must have the strongest statistical significance. Here We can say that the increase in a dataset size has a significant difference in Cassandra database throughput.

7. Conclusion

NOSQL databases have gained the most popularity in the industries. Now even small IT organizations are also seeking the use of Big Data technologies like NOSQL databases, Hadoop, and its tool.

This paper analyzes the performance of Cassandra on a standalone Machine with limited infrastructure. The experiments conducted with three different data set as 0.1 Million, 0.3 Million, and 0.5 Million. The run phase executes with 11 client threads. Later the machine was not responding in a good way. Five different workloads have been considered for the analysis as workload

A(50% Read -50 Write), Workload B (95% Read -5% Write), Workload C (100% Read - 0% write), Workload W(5% read -95% Write) and Workload H (0% Read -100% Write). In the case of Throughput, the result noted that the increase in a dataset size has a significant difference in Cassandra database throughput.

In the case of Runtime, Cassandra took less execution time as data size goes increases for the write-intensive workload. In the last case of latency measurement, Cassandra shows consistent performance for Workload B and C of Read latency and Write Latency. But for Workload H shows less write latency. Measurement of Latency and Runtime considered for a single thread. Results may vary if these values considered for maximum throughput thread or average values. Overall Cassandra works well for a standalone machine with a certain number of threads provided that enough hardware requirements.

Acknowledgments

I would like to Thanks to all authors, blog writers whose written material, research papers, White papers helped me a lot to carry out my research work.

References:

1	Kai Orend, Prof. Florian Matthes, " Analysis and Classification of NoSQL Databases and Evaluation of their Ability to Replace an Object-relational Persistence Layer " Master's Thesis, Software Engineering for Business Information
2	R. Cartell. Scalable sql and nosql data stores. SIGMOD Record, 39(4):12–27, 2010.
3	B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with ycsb". In SoCC, pages 143–154, 2010.
4	Jing Han, Haihong E, Guan Le and Jian Du, "Survey on NoSQL database," 2011 6th International Conference on Pervasive Computing and Applications, Port Elizabeth, 2011, pp. 363-366.
5	Alexandru Boicea, Florin Radulescu, Laura Ioana Agapin, "MongoDB vs Oracle - database comparison" http://www.researchgate.net/publication/261040647 January 2012 DOI: 10.1109/EIDWT.2012.32
6	Datastax "Benchmarking Top NoSQL Databases A Performance Comparison for Architects and IT Managers" White Paper BY DATASTAX CORPORATION FEBRUARY 2013
7	Abramova, Veronika & Bernardino, Jorge. (2013). NoSQL databases: MongoDB vs cassandra. Proceedings of the International C* Conference on Computer Science and Software Engineering. 14-22. 10.1145/2494444.2494447.
8	Biswajeet Sethi et al., "A Study of NoSQL Database", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 3 Issue 4, April - 2014 p.1131-1135
9	Abramova, Veronika et al. "Which NOSQL Database? A performance Overview" Open Journal Of Databases (OJDB) – ISSN 2199-3459 Volume 1, Issue 2
10	Abramova, Veronika et al. , " EXPERIMENTAL EVALUATION OF NOSQL DATABASES " International Journal of Database Management Systems (IJDMs) Vol.6, No.3, June 2014
11	Manoj V, "COMPARATIVE STUDY OF NOSQL DOCUMENT, COLUMN STORE DATABASES AND EVALUATION OF CASSANDRA", International Journal of Database Management Systems (IJDMs) Vol.6, No.4, August 2014 DOI : 10.5121/ijdm.2014.6402 11
12	Yusuf Abubakar et al., "Performance evaluation of NOSQL system using YCSB in resource Austere Environment", International Journal of Applied Information System (IJ AIS)-ISSN:2249-0868 Foundation of Computer Science FCS, New-York, USA Sept 2014
13	Gandini A., Gribaudo M., Knottenbelt W.J., Osman R., Piazzolla P. (2014) Performance Evaluation of NoSQL Databases. In: Horváth A., Wolter K. (eds) Computer Performance Engineering. EPEW 2014. Lecture Notes in Computer Science, vol 8721. Springer, Cham
14	P. P. Srivastava, S. Goyal and A. Kumar, "Analysis of various NoSql database," 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Noida, 2015, pp. 539-544.
15	Y. Gu, S. Shen, J. Wang and J. Kim, "Application of NoSQL database MongoDB," 2015 IEEE International Conference on Consumer Electronics - Taiwan, Taipei, 2015, pp. 158-159.
16	J. Klein, I. Gorton, N. Ernst, P. Donohoe, K. Pham and C. Matser, "Application-Specific Evaluation of No SQL Databases," 2015 IEEE International Congress on Big Data, New York, NY, 2015, pp. 526-534.
17	Francesca Krihely "HIGH PERFORMANCE BENCHMARKING: MongoDB and NoSQL Systems" WHITE PAPER
18	B. Hou, K. Qian, L. Li, Y. Shi, L. Tao and J. Liu, "MongoDB NoSQL Injection Analysis and Detection," 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), Beijing, 2016, pp. 75-78.
19	S. N. Swaminathan and R. Elmasri, "Quantitative Analysis of Scalable NoSQL Databases," 2016 IEEE International Congress on Big Data (BigData Congress), San Francisco, CA, 2016, pp. 323-326.
20	E. Tang and Y. Fan, "Performance Comparison between Five NoSQL Databases," 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, 2016, pp. 105-109.
21	L. Ventura and N. Antunes, "Experimental Assessment of NoSQL Databases Dependability," 2016 12th European Dependable Computing Conference (EDCC), Gothenburg, 2016, pp. 161-168.
22	Kabakus, A.T., Kara, R." A performance evaluation of in-memory databases" . Journal of King Saud University – Computer and Information Sciences (2016), http://dx.doi.org/10.1016/j.jksuci.2016.06.007
23	K. B. S. Kumar, Srividya and S. Mohanavalli, "A performance comparison of document oriented NoSQL databases," 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, 2017, pp. 1-6.
24	Shyama M Nair, Rinu Ro, Dr Surekha Mariam Varghese, "Performance Evaluation of MongoDB and CouchDB Databases", National Level Technical Conference on Advanced Computing Technologies- n'CACT'17 2017 IJSRSET Volume 3 Issue 7 Print ISSN: 2395-1990 Online ISSN : 2394-4099
25	M.P.Gopinath, G.S. Tamilzharasi, S.L.Aarthy and R.Mohanasundram, " An Analysis and Performance Evaluation of NOSQL Databases for Efficient Data Management in E-Health Clouds", International Journal of Pure and Applied Mathematics Volume 117 No. 21 2017, 177-197 ISSN: 1311-8080 (printed version); ISSN: 1314-3395
26	CAMELIA-FLORINA ANDOR AND BAZIL P [^] ARV, " NOSQL DATABASE PERFORMANCE BENCHMARKING - A CASE STUDY", INFORMATICA, Volume LXIII, Number 1, 2018 DOI: 10.24193/subbi.2018.1.06
27	Shivani, "An Empirical study on performance Evaluation of NoSql Databases ", International Journal of Electronics Engineering (ISSN: 0973-7383) Volume 10 • Issue 1 pp. 235-244 Jan 2018-June 2018
28	Ameya Nayak et al., "Type of NOSQL Databases and its Comparison with Relational Databases", International Journal of Applied Information Systems (IJ AIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 5– No.4, March 2013 www.ijais.org
29	B. G. Tudorica and C. Bucur, "A comparison between several NoSQL databases with comments and notes," 2011 RoEduNet International Conference 10th Edition: Networking in Education and Research, Iasi, 2011, pp. 1-5.