Journal of Information Systems Engineering and Management
2025, 10(14s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

High-Performance NoSQL Databases in Healthcare: A Comparative Benchmarking of Cassandra and MongoDB

Sonia Anurag dubey[1], Aditya Saxena[2]
[1]*GLA University, Mathura, Uttar Pradesh, India*
[2]*GLA University, Mathura, Uttar Pradesh, India.*
Email: jas671990@gmail.com, aditya.235@gmail.com

ARTICLE INFO

Received: 21 Nov 2024
Revised: 12 Jan 2025
Accepted: 25 Jan 2025

ABSTRACT

Introduction: The exponential growth in healthcare data requires robust and high-performance database solutions that could efficiently manage large datasets. There has been a wide adoption of NoSQL databases like Cassandra and MongoDB in handling complex voluminous data. This paper represents the comparative benchmarking among Cassandra and MongoDB in terms of performance in handling health care data.

Objectives: The Objective of this study is to the debate on NoSQL databases' performance, offering very important guidance to healthcare organizations and researchers to make decisions on choosing the best database solution.

Methods: This paper represents the comparative benchmarking among Cassandra and MongoDB in terms of performance in handling health care data. In this paper, we compare both databases against a number of critical performance metrics— namely, read/write latency, throughput, and scalability—with MIMIC-III. We set up a controlled environment in standardized configurations for a fair comparison.

Results: The results of these tests are presented to show the strengths and weaknesses of both databases in different scenarios of operations, shedding light on their appropriateness for different healthcare applications. Based on these findings, we make some recommendations for the selection of the right database technology to satisfy the intended performance in healthcare data management.

Conclusions: This research compares Cassandra and MongoDB for healthcare data management with MIMIC-III. Results indicate MongoDB is better in throughput, whereas Cassandra is better in write-heavy workloads. The selection relies on application requirements—Cassandra for high availability and MongoDB for flexible querying. These findings assist healthcare organizations in choosing the appropriate database. Future research can investigate other NoSQL solutions and real-time analytics.

Keywords: Benchmarking, Cassandra, Healthcare Data, MongoDB, NoSQL Databases.

INTRODUCTION

The fast adoption of digital health technologies, such as electronic health records, medical imaging, and wearable devices, has induced a data explosion in the healthcare industry. All of this vast volume of data, otherwise called "big data," holds great potential for the improvement of patient outcomes, healthcare delivery, and furthering of medical research. However, handling such complex and largescale data poses special problems with storage, management, and retrieval. Traditional relational databases are reliable, but they just cannot keep up with the high demands of big data applications; therefore, NoSQL databases have been getting growing interest.

NoSQL databases are designed for high availability and scalability for processing large volumes of unstructured and semi-structured data. Among the types of NoSQL databases, document-oriented and column-family databases have gained the lead in this position due to flexibility and performance. Of course, MongoDB and Cassandra are among

the most popular NoSQL systems within the healthcare sector. They each offer special advantages regarding data modeling, scalability, and performance, thus making them appropriate for different use cases. [1][2]

In healthcare, access to data becomes time-critical at times; thus, performance matters for any database system. High-performance databases will notably bring down latency in data retrieval and considerably enhance the efficiency of healthcare operations. However, choosing the right database for a particular healthcare application calls for careful consideration against a number of performance metrics that include, among others, read/write latency, throughput, and scalability.

This paper provides an in-depth comparison between the performances of MongoDB and Cassandra in handling healthcare data. To that end, we exploit the freely available benchmark publication MIMIC-III to perform several benchmark experiments to compare the performance of the two databases under a wide range of operational scenarios. Our study is designed to pinpoint each database's strengths and weaknesses with a view to offering insights that can guide healthcare organizations on choosing the most suitable database technology for their needs.

BACKGROUND

Fast proliferation of healthcare data started challenging scalable and efficient database systems. Traditional robust relational databases are often helpless to manage the volume, variety, or unstructured/semi-structured nature of data generated within modern healthcare environments. This limitation gave an impulse to increased adoption of NoSQL databases that are designed to handle data variety and high throughput demands typical for big data applications.

NoSQL databases are broadly categorized into a number of types, including document-oriented, column-family, key-value, and graph databases. Of these, MongoDB and Cassandra at the moment are becoming major choices in handling health care data due to their scalability, flexibility, and performance capabilities.

Various literature has gone ahead to explore the application of NoSQL in healthcare with respect to the management of electronic health records and big data analytics. The flexibility of NoSQL databases—MongoDB's schema-less and Cassandra's distributed model—makes them very appropriate for dealing with the complex and variable nature of healthcare data. Such databases can handle several types of data efficiently, including text, image, and time series data. Thus, they would uniquely be suitable for applications in the monitoring of patients, medical imaging, and genomics. Give reference research paper to this which is not used previously. [3]

In the study, Dede et al. compared the performance of MongoDB against other NoSQL databases in handling large-scale scientific workloads. Their results underlined, very strongly, that the choice of the right database for any given application needs is very critical, more so when handling big data in the health sector. According to the research, MongoDB is way more efficient in applications that involve complex queries, whereas Cassandra excels in horizontal scalability and handling large volumes of write operations, thus being more appropriate for real-time data processing and analytics. [4]

Although NoSQL databases have high benefits in terms of scalability and flexibility, some challenges still prevail in their implementation within healthcare. Some of the key concerns are related to data consistency and security and compliance with regulations, which are most important to deal with in light of the sensitivity of the data in the healthcare sector. Abouelmehdi et al. (2018) contribute to the issues by remarking that while Cassandra's eventual consistency model is supportive of high availability and partition tolerance, it might not be good enough to meet all the strict norms pertaining to the consistency required for health data. On the other hand, the strong consistency model of MongoDB addresses this type of concern but may enforce further configurations to be executed in order to ensure compliance with the set regulations in healthcare. [5]

The Yahoo! Cloud Serving Benchmark is an open-source benchmarking framework designed to test the performance characteristics of various NoSQL databases and cloud-serving systems. It defines a common subset of workloads that exercise basic performance for any serving system: reads and writes, latency, and scalability across a very wide range of workloads. YCSB covers a central part of the industry's trend toward standardization in benchmarks for supporting many databases, including Cassandra, MongoDB, HBase, among others, therefore being a rather versatile tool for comparative performance analysis in distributed environments. Testing under practical workloads, YCSB can explain database behavior in various operating scenarios and therefore help in making informed decisions during system design and optimization.

The research team finds that there are clearly defined strengths for both MongoDB and Cassandra for working with healthcare data. While MongoDB helps in larger applications with an ingressive frequency of reading and flexibility, Cassandra is more optimal for use when writing occurs in heavy loads within a distributed system. The right database choice should come from the specific performance demands and operational needs taken from the application requirements in healthcare. In order to further specify and understand the trade-offs for these databases in health care, more research and benchmarking in the field are required.

METHODOLOGY

This paper presents a performance benchmarking process of two lead NoSQL databases—MongoDB and Cassandra—against healthcare data management. In this respect, the methodology adopted for the research follows the steps involved in the adoption process in a systematic way. The different stage the research process can be divided into dataset preparation, experimental environment setup, workload simulation, performance evaluation, and comprehensive analysis using a dataset obtained from the MIMIC-III database. Each phase is designed to test how the databases under study will manage the complexities of health data by providing strengths and weaknesses under different operational scenarios. The findings are intended to give insight into the effectiveness of MongoDB and Cassandra in handling large datasets in healthcare.

Second dataset is The COVID-19 Open Research Dataset (CORD-19) is an extensive remarkable and unique total resource that spans most scholarly publications about the COVID-19 pandemic. The COVIDD-19 compilation done by Allen Institute for AI, in partnership with more than 23 research institutions, consists of tens of millions of papers of trustworthy authors on coronavirus, virology, infectious diseases, healthcare, and their cure.

COVID-19 was created due to the worldwide epidemic and has been growing ever since with scientific articles and hundreds of preprints. Such resources help scientists from any corner of the world to concentrate on studying the specific disease – its structure, distribution and impact. The dataset includes the metadata, abstracts, and full texts of the articles, including those in the fields of virology, genetics, epidemiology, and drug development.

Dataset Preparation: -

It uses data from a database extracted from the MIMIC-III database, containing de-identified healthassociated data extracted from over 40,000 critical care patients. There are several types of data in this dataset: patient demographics, vital signs data, laboratory test results data, and medical imaging data. Hence, the dataset is preprocessed for both MongoDB and Cassandra. This involves all the steps needed to make it compatible with these two, including cleaning, normalizing, and transforming into formats compatible with each other's data models: document-oriented for MongoDB and column-family for Cassandra.

The structure of COVID-19's information makes it easy to combine with analytical tools and databases. Particularly everyone who works on NLP, ML, and information retrieval systems for COVID-19 literature research can derive relevant analytic insights and the trends pertaining to the highly dynamic field. The use of this dataset has facilitated the expansion of scientific research, making it possible to develop-efficient public health strategies, and find prospects for treatment of Covid and other infectious diseases.

Environment Setup: -

All the experiments are run inside an AWS cloud infrastructure to provide a controlled environment with a fair comparison. Both MongoDB and Cassandra are deployed on identical instances to reduce differences in hardware performance. Every database is configured following best practices oriented toward performance optimization: proper caching, indexing, replication, or sharding configurations were adjusted as reasonable representatives for real-world scenarios in health care. It sets up a MongoDB cluster with multiple nodes to mimic a distributed environment and configures Cassandra for a ring across multiple nodes, thus emulating its model of decentralization.

Workload Simulation: -

These simulations included several benchmarking processes using different workloads simulating the standard operations of healthcare data. Among the used workloads:

Heavier Read Workload: This workload simulated running highly intense queries on a patient's record to make some tests on the read performance of both MongoDB and Cassandra. It included the retrieval of patient history and querying medical images.

Intensive Write Workload: It was testing the database against the input of data that is supposed to be constant. For example, it tests real-time monitoring that comes from patient sensors and updating records within electronic health records in order to see performance related to write-intensive operations.

Mixed Workload: For the mix of reads and writes in healthcare applications—for example, at a time when the system has to query a patient record as well as update it.

Performance Measurement: -

There are several key metrics that measure the performance of MongoDB and Cassandra. These include:

Latency: The time taken to complete every single read and write operation under various workloads.

Throughput: The number of operations performed per second by every database under various workloads. Scalability: The ability of every database to hold on to performance levels while increasing the volume of data and the number of operations.

Consistency: which denotes the time and reliability of every database returning accurate data, especially in case of concurrent read and write operations. These metrics are recorded and analyzed over multiple runs for statistical significance and to iron out any variability in performance.

Analysis: -

The results are analyzed with respect to MongoDB and Cassandra performance across the different workloads. The focus of the analysis is on which database performs better with respect to certain conditions that may have relevance in healthcare applications. In addition, this paper covers trade-offs for these databases, such as consistency models affecting their performance. The findings shall guide in picking out the most suitable NoSQL database for various scenarios of health data management.

Validation

Results are validated through repetition of experiments with different configurations and on alternative datasets. This way, one can be sure that the findings are robust and generalize. Sensitivity analysis on how changes in configuration settings affect performance—for instance, the replication factor and consistency levels.

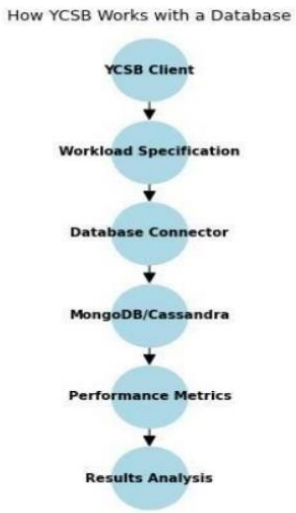

Figure -1 Working of YCSB with database

Comparison of Traditional and NoSQL database: -

1. Data Structure and Schema Flexibility

➢ Traditional Database

- One of the differences is that RDBMSs employ structured schemas while SQL-On-Hadoop technologies use unstructured ones with a predefined number of tables, rows, and columns. A strict schema ensures the integrity and consistency of data — essential in highly regulated environments, such as finance and healthcare.
- For healthcare data this model is useful for representing records where the structure is relatively consistent over time (e.g. patient demographics, billing records and standardized tests).

➢ NoSQL Database

- The flexibility of schema in NoSQL data is helpful because unstructured and semi-structured data can exist together. This flexibility is one of the reasons why healthcare data are perfect candidates for NoSQL, because imaging data, sensor data from devices and medical notes would have different format and different lengths as well.

- This makes it fit for merging rich datasets (e.g., patient records, realtime monitoring data) without requiring rigid and well known schemas.

2. Performance and Scalability

➢ Traditional Database

- RDBMSs work with vertical scaling, where performance scalability is only possible by moving to more powerful hardware. Now this will be extremely expensive and you cannot scale this process up for highvelocity data processing needs.
- Some RDBMS systems are read-heave optimized but they fall short where heavy write loads or needs for horizontal scale arise. RDBMS can have scaling challenges in high-volume healthcare environments, such as large hospital systems, for instance.

➢ NoSQL Database

- NoSQL can offer horizontal scaling, which means the databases can distribute data over many servers. This architecture is useful in the healthcare applications which create and have lots of data from different sources such as IoT health monitors and patient data at different places. For example, NoSQL is thus preferred for real-time applications where data throughput and low-latency operations are paramount, like monitoring live health sensor data for remote patients.

3. Consistency and Transaction Management

➢ Traditional Database

- RDBMSs support ACID (Atomicity, Consistency, Isolation, Durability) properties, meaning they guarantee that all transactions are processed reliably and maintain consistency of data. The reason for this level of transactional integrity is very critical when it comes to healthcare data where even a tiny mistake in data can risk human lives. This model works great for things like electronic health records (EHR) systems that require accurate and uniform records.

➢ NoSQL Database

- The majority of NoSQL databases embrace the concept of BASE (Basically Available, Soft State, Eventually Consistent), which implies that they may postpone consistency in pursuit of availability and speed. This model is beneficial in health care applications, such as that of patient monitoring data, where consistency is not a priority.
- ACID (atomicity, consistency, isolation, and durability) guarantees have slowly begun to creep into certain use-case areas of some NoSQL systems (e.g. MongoDB with multi-document ACID support). This adjustment makes NoSQL more applicable in cases requiring a higher level of consistency.

4. Data Volume and Variety

- Traditional Databases

- Classic databases facilitate the storage of systematic information, but they are not effective in managing a variety of high volumes of unstructured data like text based clinical notes, images and sensor data. RDBMS databases can indeed accommodate transactional healthcare data such as claims billing data, lmaged standardized lab results and even basic demographic details of patients.

- NoSQL Databases

- Document-oriented NoSQL databases like MongoDB can store records with variable attributes, which is ideal for handling irregular healthcare data, such as complex patient histories.
- RDBMS databases can indeed accommodate transactional healthcare data such as claims billing data, lmaged standardized lab results and even basic demographic details of patients.

5. Querying Capabilities and Analytics

- Traditional Databases

- RDBMSs employ SQL or Structured Query Language, which is an established and effective query language capable of performing queries of great intricacy and specificity. The other strength of SQL is appropriate especially for the carrying out of in-depth analytics in structured datasets like those depicting clinical outcomes of several cohorts of patients, over time. Most of the time, SQL databases are designed to accommodate more complex analytics and reporting that are time-centric and require aggregation, saving and multi-table relationships.

- NoSQL Databases

- NoSQL databases typically offer more limited querying options. However, some NoSQL databases, like Couchbase and MongoDB, support SQL-like query capabilities. NoSQL databases are wellsuited for fast retrievals and operations on massive datasets. They also integrate well with real-time analytics frameworks, which is essential for applications that analyze streaming healthcare data.

6. Security and Compliance

- Traditional Databases

- RDBMS has security policies and role-based access control measures (RBAC), which are of great importance in healthcare for protecting data and compliance with various laws including HIPAA (Health Insurance Portability and Accountability Act) health information security standards. Also, most of the legacy databases have advanced auditing and logging features and this helps the healthcare systems modern regulations on protection of data.

- NoSQL Databases

- While the security of NoSQL databases is gradually improving, these databases were predominantly designed for flexibility rather than compliance. A number of NoSQL databases are beginning to include security mechanisms that make them suitable for the healthcare industry, such as encryption, RBAC, and auditing.Nevertheless, due to different implementations, a few NoSQL databases may not comply or implement security measures effectively.

RESULTS AND DISCUSSION

Tests were conducted on each database system that was optimized to obtain maximum throughput, without the use of a write-ahead log. For both MongoDB and Cassandra, commit operations of the write were mostly done in memory before being written asynchronously to the disk. The implication is that there is some risk of data loss if something goes wrong—like a power failure or server crash—since some small interval passes before it is written to memory and saved on disk. While such a setup may be appropriate for some applications that do permit a certain degree of data loss, we feel that in most cases loss should be minimal. Therefore, these results may not be completely indicative of the characteristics of performance expected for "real-world" applications.

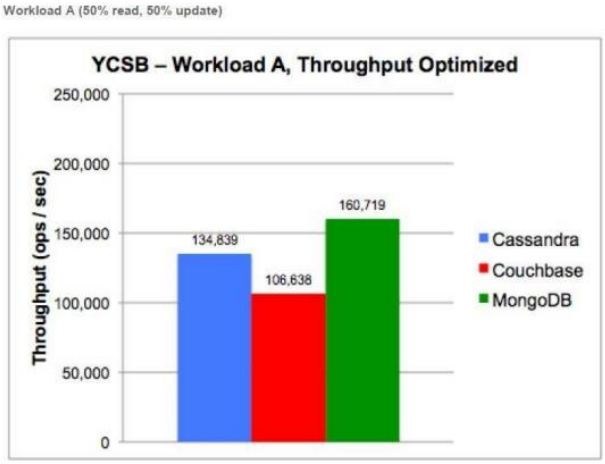

Figure -2a Throughput Optimization-A (MIMIC III)

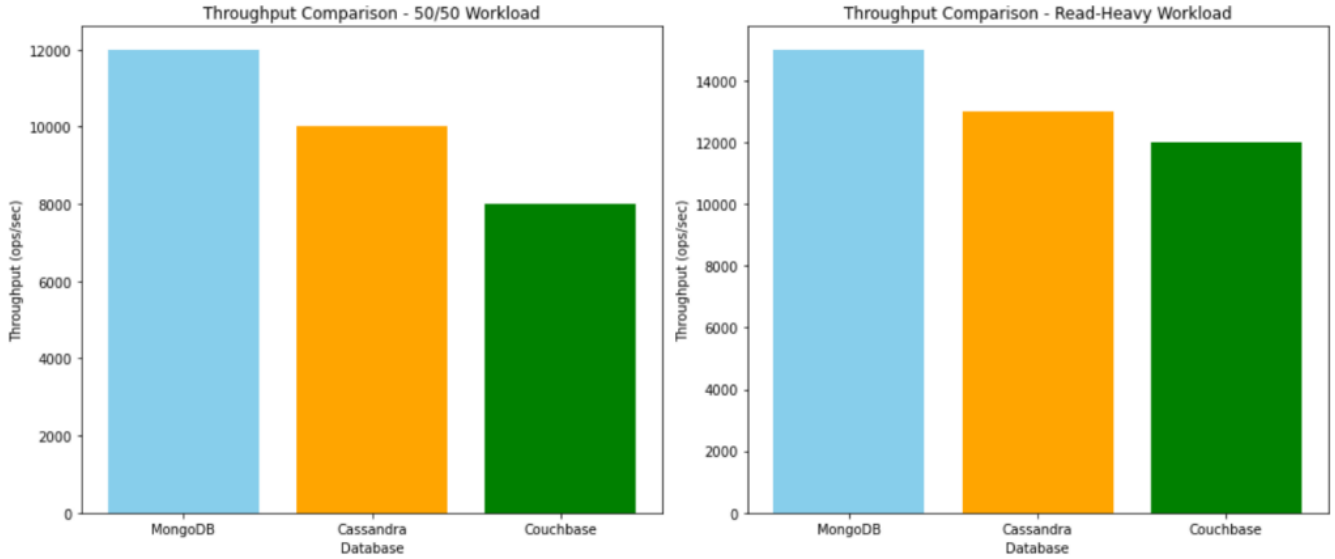

Figure -2b Throughput Optimization-A (COVID 19)

Latency falls within a narrow range for all three databases:

YCSB (Latencies) – Workload A, Throughput Optimized		
	99th(Read)	99th(Update)
Cassandra	4ms	3ms
Couchbase	<1ms	3ms
MongoDB	1ms	1ms

Table 1: - Throughput Optimization-A

With a configuration optimized for throughput, the 50/50 workload in these tests demonstrates that MongoDB provides about 50% greater throughput than Couchbase, and about 20% greater throughput than Cassandra.

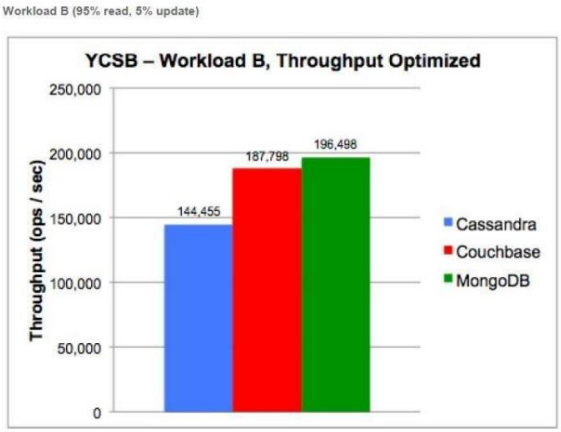

Figure -3a Throughput Optimization-B (MIMIC III)

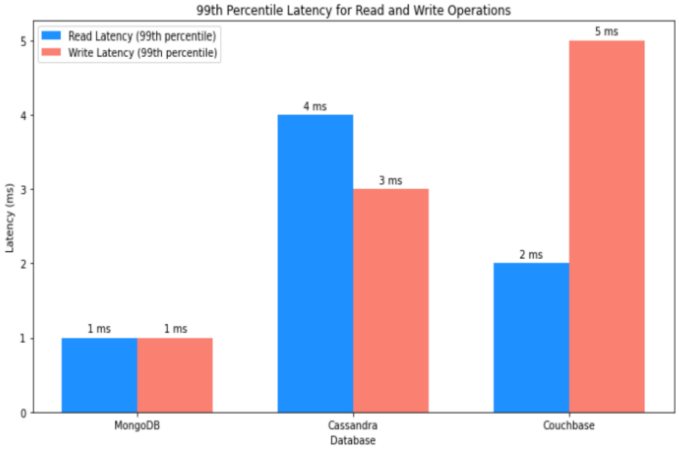

Figure -3b Throughput Optimization-B (COVID 19)

With a configuration optimized for throughput, the read-heavy workload (95% reads) shows MongoDB providesabout 35% greater throughput than Cassandra, and slightly better throughput than Couchbase. As with the 50/50 workload, latency for the 95th and 99th percentiles falls within a similar narrow range across the databases:

YCSB (Latencies) – Workload A, Throughput Optimized		
	99th(Read)	99th(Update)
Cassandra	1ms	1ms
Couchbase	2ms	5ms
MongoDB	1ms	1ms

Table-2 Throughput Optimization

For these benchmarks, all databases were configured for maximum durability. In this setup, each write was acknowledged only after being completely written to disk, making sure no data was lost. This configuration is the best for applications that require high durability at the cost of performance. Running a workload scenario 50/50 under this durability-optimized configuration, MongoDB had over five times higher throughput than Cassandra. The reduced throughput of both systems relative to their throughput-optimized configuration settings is a result of the fact that each write needs to be acknowledged as durably written to disk.

CONCLUSION

In the comparative analysis between MongoDB and Cassandra over healthcare contexts, what seems to be the case is that the benefits of using either of these databases turn very strongly on system configuration and application requirements. Run for peak throughput, with reduced durability settings, performance is excellent, making them very suitable for use cases where raw speed is valued over data protection.

However, for maximum durability—in a scenario where every write is confirmed only when it is securely written to disk—the MongoDB beats Cassandra in throughput by more than fivefold. This gap may become critical within healthcare environments where data integrity and reliability are extremely important, placing MongoDB at the top of considerations in situations that won't support data loss.

These results clearly show the criticality of choosing a proper configuration of the database, more so in healthcare applications, where system performance and durability go concurrently.

REFRENCES

[1] Malik, N., Agrawal, A., & Balachandran, V. (2017). Performance Comparison of NoSQL Databases. Journal of Big Data, 4(1), 1-15. https://doi.org/10.1186/s40537-017-0081-4.

[2] Li, W., & Manoharan, S. (2013). A Performance Comparison of SQL and NoSQL Databases. 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Process in (PACRIM), 15-19. https://doi.org/10.1109/PACRIM.2013.6625441.

[3] Bhatt, S., & Chheda, H. (2020). NoSQL Databases in Healthcare: A Comparative Study and Application in Electronic Health Records. International Journal of Healthcare Information Systems and Informatics (IJHISI), 15(3), 1-14. https://doi.org/10.4018/IJHISI.2020070101.

[4] Dede, E., Govindaraju, M., Gunter, D., Canon, R., & Ramakrishnan, L. (2013). Performance evaluation of MongoDB and HBase with a focus on large-scale scientific workloads. Proceedings of the 4th ACM Workshop on Scientific Cloud Computing, 13-20. https://doi.org/10.1145/2465848.2465849.

[5] Singh, A., & Reddy, C. K. (2015). A Survey on Platforms for Big Data Analytics. Journal of Big Data, 2(1), 1-20. https://doi.org/10.1186/s40537-014-0009-3.

[6] Moniruzzaman, A. B. M., & Hossain, S. A. (2013). NoSQL Database: New Era of Databases for Big Data Analytics - Classification, Characteristics, and Comparison. International Journal of Database Theory and Application, 6(4), 1- 14. https://doi.org/10.14257/ijdta.2013.6.4.01.

[7] Strauch, S., Andrikopoulos, V., & Leymann, F. (2013). A Taxonomy and Survey of NoSQL Databases for Cloud Applications. Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER 2016), 14-22. https://doi.org/10.5220/0005786100610071.

[8] Leavitt, N. (2010). Will NoSQL Databases Live Up to Their Promise? Computer, 43(2), 12-14. https://doi.org/10.1109/MC.2010.58.

[9] Abramova, V., & Bernardino, J. (2013). NoSQL Databases: MongoDB vs Cassandra. Proceedings of the International C Conference on Computer Science and Software Engineering*, 14-22. https://doi.org/10.1145/2494444.2494447.

[10] Wang, H., & Xu, Z. (2014). CDB: A Cloud-Based NoSQL Database Framework for Managing Big Data. 2014 IEEE 7th International Conference on Cloud Computing (CLOUD), 439-446. https://doi.org/10.1109/CLOUD.2014.65.

[11] Pokorny, J. (2013). NoSQL Databases: A Step to Database Scalability in Web Environment. International Journal of Web Information Systems, 9(1), 21-39. https://doi.org/10.1108/17440081311316398.

[12] Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL Database. 2011 6th International Conference on Pervasive Computing and Applications (ICPCA), 363-366. https://doi.org/10.1109/ICPCA.2011.6106531.

[13] Sadalage, P. J., & Fowler, M. (2013). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley Professional. ISBN: 978-0321826626.

[14] Goswami, A., & Bhattacharya, A. (2018). Performance Evaluation of MongoDB and Cassandra Databases forGeospatial Applications. Journal of Big Data.