Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549626

AN ALGORITHM TO COMPARE PERFORMANCE OF MONGODB AND ORACLE DATABASES FOR BIG DATA APPLICATIONS

Jyoti Chaudhary*, Vaibhav Vyas
*1-Research Scholar, Department of Computer Science, BanasthaliVidyapith, Rajasthan, India, Jyotichaudhary1410@gmail.com
2-Associate Professor, Department of Computer Science, BanasthaliVidyapith, Rajasthan, India
***Corresponding author:**

Abstract:
Software developers have started to take NoSQL data storage solutions into consideration in context of the big data's demanding requirements. The performance of a NoSQL database in terms of speed of data access and processing, particularly response times to the most crucial CRUD activities, is one of the key factors to consider when choosing a NoSQL database for an application (CREATE, READ, UPDATE, DELETE). In this study, the behavior of two important databases-Oracle, a well-known SQL database, and MongoDB, a document-based NoSQL database-will be examined in terms of the complexity and effectiveness of CRUD operations, particularly in query operations. The primary goal of the study is to conduct a comparative examination of the effects that each unique database has on the efficiency of the application when processing CRUD queries. A case-study application for both of the databases that aims to model and simplify the operations of organizations that use massive data, this application is designed using Python. The findings demonstrate how both the databases perform for various data volumes. Based on these, a thorough analysis and a number of conclusions are offered to aid in the decision-making process for selecting an acceptable solution for use in big data applications.
Keywords: MongoDB, Oracle, SQL, NoSQL database, Performance, CRUD operation

1.Introduction

The digital human race is expanding rapidly and becoming more intricate in terms of volume, variety, and velocity. According to an IBM analysis, ninety-five percent of the world's information has been produced in the last few years, and production is still going toughwith a lookout of 2.5 quintillion bytes per data. Due to the advancement in technologies, including the internet of things, social networks, web technologies, the growth of cloud computing, as well as the increase in smart devices, huge data volumes are created each day from various sources. The need for highlyefficient database systems for storing and retrieving data is driven by the steadily increasing volume of data [8].

Currently, the quantity of data being produced daily by business applications and the web is too large for relational database management systems to handle their data. This has increased interest in more choices for RDBMSs. Relational database systems are significant technology that depends on SQL to provide ad-hoc querying capabilities and store substantial amounts of unstructured data (SQL). Relational databases like MYSQL, MS-Access, Oracle, MS SQL

Server, and Sybase are used to store, alter, and retrieve data [9]. The fundamental problem with unstructured data is determining an efficient storage mechanism. It must be able to effectively deal with such a large amount and variety of data [1].
In the endeavor to deal with this data flood, distributed systems have proven to be invaluable. Two key characteristics distinguish these systems, one is system scalability which means the primary database system must be able to accomplish and store a large volume of data while also allowing applications to effectively read it or access it and the second is the associated performance which should be highly fast with response time to client requests [2].

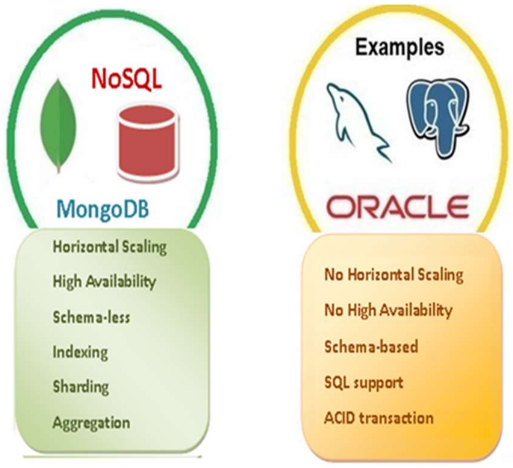

Fig. 1 Oracle and MongoDB database properties

MongoDB, a NoSQL database heeds key-value pair implementation that reinforces the single value abstraction JSON. MongoDB stores data in documents format that are called JSON documents with a wide range of structures. JSON can be viewed as a standard while transferring data between a server and a web application across a network that is in human-readable format across a network. MongoDB converts this JSON to BSON to store in binary format. BSON is authentic and efficient in speed and storage area. These BSON is stored in collections. This research considered two databases i.e. MongoDB, a NoSQL database, and Oracle, a relational database that will examine the performance in terms of time taken to execute Ad-hoc queries.

2. Literature Review

Paci 2022 [11] states application developers must often choose between SQL and NoSQL databases while creating their solutions. Relational or SQL databases are in use for a long period and are still in the Top-10 list of DB-engines. Meanwhile, NoSQL databases are also a part of the Top-10 list of DB-engine like MongoDB, a places database in the list. This article has proved the usage of SQL and NoSQL databases where NoSQL are still high if required for specific work like handling a large amount of data.
Antaset.al., 2022 [12] appraise different SQL and NoSQL databases to figure out the most suitable database to handle and mine COVID data. This evaluation has been done on different parameters like query runtime, memory usage, and size required for storing data. The databases included by the author in this study are Microsoft SQL, MongoDB, and Cassandra. Cross-validation classification tests were run on a table of around 3M data including COVID-19 assessments with patient symptoms. And, NoSQL database MongoDB has proved itself the best database to handle tests with a large volume of data.

Mukashevaet.al., 2021 [3] states that the research is a part of a larger investigation into diabetes solution and treatment. The system offers data storage and processing functionalities for both structured and unstructured data. The system may be controlled in both automated and manual modes. Unstructured data is imported into the established information system platform using the NoSQL booster tool in manual management mode. Data loading necessitates using the MongoDB tool in manual administration mode to create and load unstructured data. Thus, the author summarizes that new information technologies have the potential to enhance the quality of therapeutic services delivered in the healthcare system. As a result, the researcher's uniqueness and significance are confirmed. Big data technology solutions enable you to analyze and store heterogeneous data while also allowing you to resist the system in both modes automatic and manual. Data can be uploaded to theMongoDB database via the website or manually via the created information system for diabetes diagnosis.

Tan 2018 [4] describes MongoDB as a distributed file storage database that falls among relational and non-relational databases. MongoDB is a fine database to store documents. It primarily addresses the issue of enormous data storageto approach efficiency. MongoDB is a versatile NoSQL database system for video-intelligent large-data research. Implementation of computing, cloud provisioning of storage resources, ad-hoc networks for devices, self-management, uninterrupted operation, and flexible expansion as needed. Supports distributed search engines through the MongoDB distributed database, supporting large-scale data storage and analytics. The author concludes that the query language that MongoDB enables is quite strong, and its syntax is comparable to that of an object-oriented query language which virtually can do all of the operations that a single table query can perform in the relational database, and it also permits data indexing.200000 measured points can be stored by a collection of SQL real-time history database for several decades after layer-by-layeroptimization; however, the network deployed NoSQL real time history data that has no higher limit, and users may install easily to comprehend large data storage.

Tang & Fan, 2016 [5] highlights the BASE (Basic availability, Soft State, Eventual consistency) capabilities, NoSQL databases, and associated technologies have recently grown in popularity and are now extensively used in a variety of contexts. There are currently over 225 different types of NoSQL databases. However, individuals find it difficult to compare their performance and pick an acceptable database because there are so many and they are continually updated. This paper aims to gauge the performance of five NoSQL clusters (Redis, MongoDB, Couchbase, Cassandra, and Hbase) using YCSB. This helps to conclude that NoSQL databases are popular for their own specialties but when there is a need for efficiency and scalability document-orientedMongoDB database has proved the best to deal with thehuge amount of data.

3. MongoDB

With the distributed expansion in mind, MongoDB was developed initially. Using a document-oriented data model, data is automatically distributed among several servers. Automatically distributing the documents that handles the amount and load of data in the cluster, as a result of the developer's increased focus on programming rather than database expansion. When extra capacity is required, simply add the new node to the cluster and let the database handle the rest [16]. Since its release, MongoDB has steadily and securely increased in popularity, making it the most widely used type of NoSQL database that maintains documents in the JSON format and is document-based (built in C++). Each version is improved, and the flexible structure allows for frequent changes throughout development. This allows for automated scalability, great performance, and availability [23].

3.1 Characteristics of MongoDB:

1. Queries: Rich document-based queries and Ad-hoc queries are simple and have good reliability.
2. Document-Oriented: Unlike RDBMS, MongoDB stores all data in documents rather than tables. These documents are encoded in Javascript Oriented Notation (JSON) like format which is called BSON (Binary JSON), which helps in easy storage [6].
3. Replication: It makes use of a master-slave mechanism for replication and failure. When the data is kept in a single database, it is susceptible to several of problems, such as hardware failure, server crashes, and service interruptions. It would be extremely difficult to retrieve your data if something happened. You may prevent these risks by setting up extra servers for backup and disaster recovery. Data availability and stability are significantly increased when the same data is horizontally scaled over several servers that store the same data (or shards of the same data). Of course, replication helps with load balancing. When several users access the same data, the burden may be split evenly amongst servers [7].
4. Sharding: Data is sharded when it is distributed across numerous computers. MongoDB offers an automated load-balancing function because of sharding [6].Scaling a huge online application with millions of daily visitors is practically difficult without sharding. Sharding in MongoDB, like replication via replication sets, enables significantly better horizontal scalability. Horizontal scaling refers to the fact that each shard in a cluster stores a subset of the dataset in question, thereby acting as a separate database. The collection of dispersed server shards creates a single comprehensive database that is significantly more suited to addressing the demands of a popular, expanding zero downtime [7].
5. High Scalability: MongoDBcan scale as the workload grows. Horizontal scaling is supported by MongoDB. Scales well due to sharding, allowing for increased performance under all circumstances [6].
6. Indexing:The purpose of indexes is to increase the speed and performance of searches. MongoDBoffers a wide range of directories and features, including language-oriented sort orders, to oblige complicated dataset access patterns. MongoDB indices may be generated on the fly to support real-time, dynamic query patterns and application needs [7].
7. Map reducing: It is compatible with map-reduced tools. Map-reduce is a type of data processing that collects and aggregates enormous amounts of data [6].

4. Oracle

One of the most trustworthy and popular relational database management systems, Oracle was created by Oracle Corporation. Businesses all across the world utilize it because it offers excellent performance, security, and scalability. Oracle is most frequently used by businesses to store data, transaction processing, and business analytics. Oracle database management can be performed with a range of tools [18].Numerous features are included in terms of usefulness, performance, and scalability. Because of this feature, Oracle DBMS is ideally suited for business applications that demand highly sophisticated scalability and stability. Oracle developed PL/SQL, a Procedural Language (PL) that extends SQL, in 1991 to provide a development-immersed environment that could benefit from the benefits provided by SQL. Data encapsulation, exception handling, information hiding, and object orientation are all aspects of PL/SQL, a standard data access language for Oracle relational databases [19].

Data is physically stored in the form of data files and logically stored in table spaces by the Oracle RDBMS. At the physical level, data files are made up of one or more data blocks, with different data files having different block sizes. Data dictionaries, indexes, and clusters are all characteristics of Oracle whereas, after version 10g, grid computing capabilities were implemented, allowing instance applications to utilize the CPU resources of a different grid node [20].

1. Scalability: The scalability of an Oracle database depends on features like portability and clustering of real applications. Data concurrency and consistency, which Oracle takes into account, must be under-organized in a multiuser database [21].
2. Schema:The Oracle database schema is a grouping of logical data structures, also known as schema objects. A database user is the owner of the schema that shares the same identity as the user. Schema objects are user-erected structures that refer to database data directly. The most significant schema objects supported by the database are tables and indexes [22].
3. Indexing: An index is an elective structure connected to a table or table cluster that can occasionally boost data access. You can sometimes extract a small group of randomly dispersed data from a table by intensifying an index on one or more of its columns. Indexes are one of many strategies for minimizing disc I/O [22].

5. Method and Implementation

MongoDB database differsfrom Oracle in various features and a wider range of terminology. In order to compare both databases an application is created for every database which aims to process respective queries to draw a strong outline of the databases which are being compared considering a few parameters like features, performance, scalability, etc. however, this study also compares two databases which are MongoDB and Oracle. Both the databases are best among their categories, like MongoDB ranks higher if question document-oriented NoSQL databases and Oracle is one of the famous SQL databases. These databases are compared on the basis of their performance in CRUD operation which also includes the COUNT function as a part performance evaluation query. The application here is created using PYTHON that communicates both databases on an 8GB RAM 64-bit operating system. The dataset for this comparison is taken from the kaggle.com website. To process this comparison a generalized algorithm is mentioned next with followed up steps:

Algorithm 1: to compare MongoDB NoSQL database and Oracle Relational Database

```
1.  Import important libraries
    Import time library
2.  set connection using:
    Con = database_connector.connect()
    Cur = con.cursor()
3.  create database
    Create a table for oracle and a collection for MongoDB
4.  Insert data into respective table and collection
5.  Query execution logic for both databases:
    a.for oracle database
    Try

    {
    t1_start = process_process_time_ns()
            cur.execute(query statement)
            if(query = condition)
                    counter+ = 1
            else
                    exit
            con.commit()
    t1_stop = process_time_ns()
    }
    Except
            Set err to e.args
    b. for MongoDB database
    Try
    {
     t1_start = process_time_ns()
            col.query(data)
            if(query = condition)
                    counter+ = 1
            else
                    exit
    t1_stop = process_time_ns()
    }
6.  Repeat the 4th step until all operations create, retrieve, update, delete and count perform.
```

6. Time Comparison and Performance Analysis

The comparative study of different databases is analyzed on parameters like features, terminologies, and CRUD operation to extract the best databases which will deal with all kinds of data types for current industrial use cases which was earlier compared in many pieces of research by Wadhwa&Kaur, 2017, Arauja et.al., 2021, Aghi et.al., 2015, Seo et.al., 2017, and Ceresnak&Kvet, 2019.

Some tests have been performed in this study and calculated how much-selected databases have taken to perform various actions to make the best comparison between the search engines. Test cases were performed for theOracle databasewhich has been chosen among all relational databases and the Mongo databasewhich has been chosen among document-orientedNoSQL databases. The test has been performed on different datasets taken from the Kaggle.com website. The workload of 5000 (5k), 50000 (50k), and 500000(500k) datasets respectively have been analyzed. Same datasets were used to perform the operations in a single instance and not in a cluster.

6.1.Performance evaluation for Insert operation

In the first test, we generated objects that have inserted into the database. We took a set of 5K inserts and compare the time taken by the databases and then a set of 50K has been inserted

and this continues for 500K data inserted by using insert_one() or insert_many() in MongoDB and equivalent operation in Oracle is done using CREATE or INSERT. A collection is implicitely created using syntax syntax db.collection.insert_one()
db.collection.insert_many(). To do the same in Oracle, we first need to create a schema in tabular format and lately the insertion can occur using syntax INSERT INTO TABLE_NAME (attributes).

Table.1 Time comparison for Insert Operation

INSERT OPERATION (time in milliseconds)		
Data Size	**Oracle**	**MongoDB**
5K	4250	578.125
50K	60718.75	3906.25
500K	365609.4	18937.5

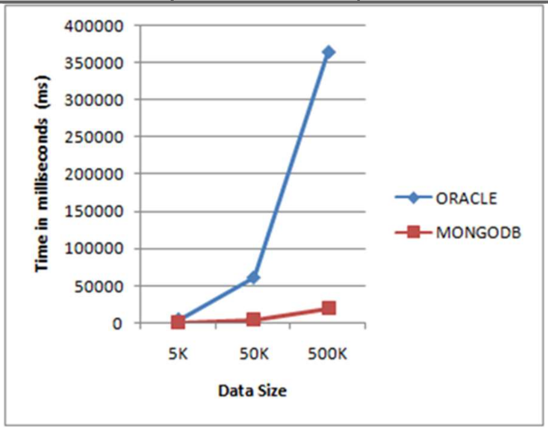

Fig.2Performance analysis for Insert Operation

The table 1 shows that putting a lot of data into MongoDB is more effective. On the other side, inserting records took far too long. When dealing with enormous datasets, Oracle Database behaves rudely by taking longer to complete operations since there are more records.

6.2. Performance analysis for Retrieval operation

The retrieval operation performs on the same datasets that have been inserted during the insert operation. The equivalent syntax for both databases has been used to execute the query. In MongoDB we stored comments in an array of BSON documents, and keeping in mind that the relation between articles and comments is one to many, we actually do one projection operation using the query. Aggregation functions can be used to retrieve data like count, distinct, or aggregate. MongoDB provides two functions when to retrieve data find() and findOne() which is similar to SELECT in SQL database. The equivalent syntaxes for MongoDB and Oracle db.collection.find() and SELECT * FROM TABLE_NAME.

Table.2 Time comparison for Retrieve Operation

RETRIEVAL OPERATION (time in milliseconds)		
Data Size	**Oracle**	**MongoDB**
5K	484.375	171.875
50K	3140.625	3093.75
500K	10828.13	5453.125

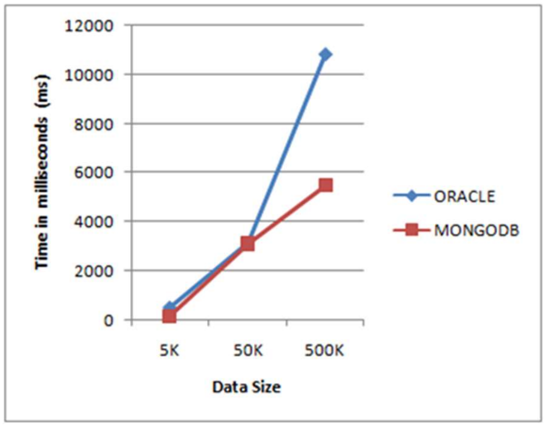

Fig. 3Performanceanalysis for Retrieve Operation

Table.2 and Fig.3 show the time comparison and performance of retrieval operation on MongoDB and Oracle Databases.It can be inferred from Fig.3 that MongoDB is found to perform much better than Oracle while performing the retrieving operations for huge volumes of data.

6.3. Performance analysis for Update operation

Here again, the updating statements have been executed for the same datasets depending on the condition that has applied to the number of records to retrieve all the records asked in the query. The equivalent syntax for MongoDB and Oracle used to update queries db.collection.update_one(), db.collection.update_many() and UPDATE TABLE_NAME SET value:1 WHERE value:2.

With the $addToSet operator, MongoDB enables updating nested documents. Unsettlement is a Boolean parameter. If the update() method returns true, any existing document that satisfies the query selection criteria will be updated; if none do, a new document will be added. If the query criterion matches many documents and the Boolean multi parameter's value is set to true, all the documents will be updated. This is another parameter that may be used with the update statement. It will only update one document if its value is set to false.

Table.3 Time comparison for Update Operation

UPDATE OPERATION (time in milliseconds)		
Data Size	**Oracle**	**MongoDB**
5K	15.625	0.25
50K	15.625	0.325
500K	15.625	0.387

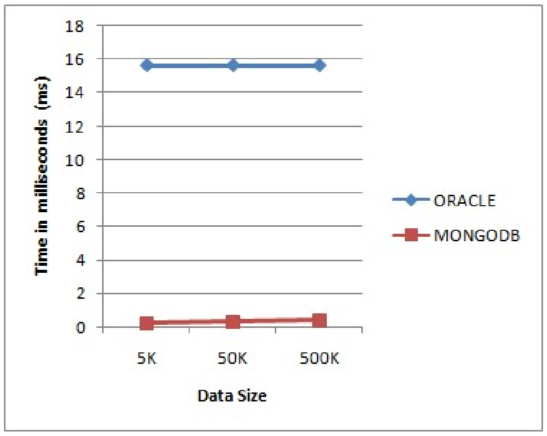

Fig. 4Performance analysis for Update Operation

The time taken by the MongoDB is still very efficient than Oracle engine. As Table.3 and Fig.4 shows the time and comparison analysis of the update operation reflects lower performance by the Oracle database. Updating a lot of data into MongoDB is more effective as it takes very less time to execute every query with the any size of data. On the other side, updating records took far too long in Oracle database.

6.4. Performance analysis for Delete operation

We employed the same type of test concerning to removing the objects from the database. The first 5K records have been deleted, then, 50K and 500K. The collections made throughout the previous tests were used for this test as well. As in Oracle database, we use the DELETE() method to remove all the data from the collection. The mongo shell command db.collection.drop will be used to remove a collection from the database (). T equivalent syntax for both database will be used db.collection.delete_one(), db.collection.delete_many(), and DELETE FROM TABLE_NAME WHERE value:?

Table.4 Time comparison for Delete Operation

DELETE OPERATION (time in milliseconds)		
Data Size	**Oracle**	**MongoDB**
5K	31.25	15.625
50K	31.25	15.625
500K	31.25	15.625

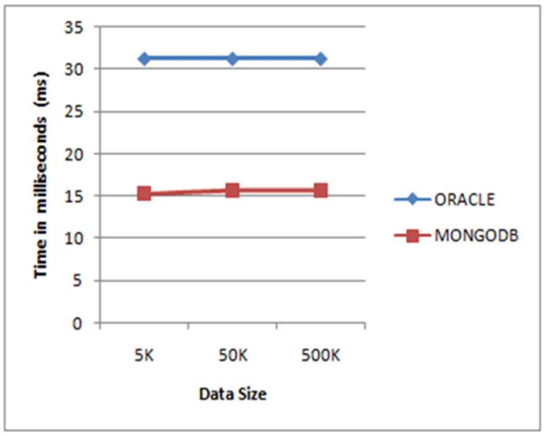

Fig. 5Performance analysis for Delete Operation

Below Table.4 and Fig.5 reflect the results produced after the statement execution. The time taken by Oracle is consistent but still very high compared to the MongoDB database. Thus, for delete operation, MongoDB is an efficient database.

6.5. Performanceanalysis for Count Function

The Count Operation is the requirement to count the number of records left after all the query execution on both database engines. This operates using fetchall() and count_documents() statements.

Table.5 Time comparison for Count Function

COUNT OPERATION (time in milliseconds)		
Data Size	Oracle	MongoDB
5K	31.25	15.625
50K	359.375	15.625
500K	5531.25	15.625

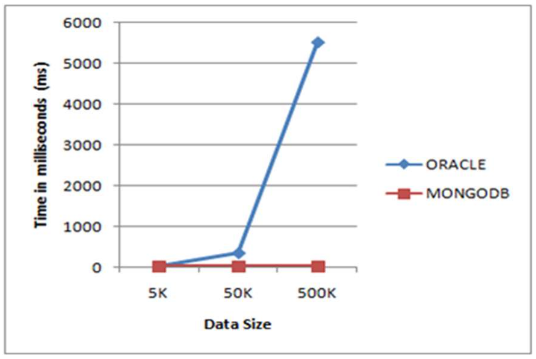

Fig. 6Performance analysis for CountFunction

Table.5 and Fig.6 shows the result of both database engines which concluded MongoDB is much better and more efficient compared to Oracle database.

7. Discussion

From the above graphs and results the performance of MongoDB hits higher in major queries, therefore, MongoDB gives developers flexibility while creating new software. Using import-export tools, it is simple to deploy and copy databases from one server to another. The database management system MongoDB operates more quickly. This is the option to select if you want a straightforward database that responds quickly.The map-reduce function can be used to combine data for reporting. When using map-reduce, you can combine fields, summarize the results, or create any other configuration you like. It moves quickly and is adaptable. Being open source, you can create plugins to make it simpler to use. It is a program that is always evolving, and the open-source community has many helpful hints. On the other hand, you should stick with the traditional Oracle Database if you need a more complicated database with relationships between tables and a fixed structure. Although it moves more slowly, it is a trustworthy database and serves as the foundation for more intricately structured databases.

8. Conclusion

Non-relational databases are a popular choice for data storage because of their expanding use of statistics. Several academics have addressed the issues, making data collection more efficient despite the fact the demand for information retrieval is continuously rising [24][10]. MongoDB is an extremely adaptable, schema-less database that can be used in a distributed

structure. By the use of Sharding, MongoDB can scale horizontally. Several shards can be created from a collection of data. Moreover, load balancing is built into MongoDB; data is copied to keep the system up and running in the event of a failure. This fact is invisible from the perspective of CRUD operations; data manipulation will be the same, and no additional query techniques are required to query a remote MongoDB server. A distributed system shows off indexes' full potential. They mostly aid in the quick performance of read queries. In MongoDB, master-slave replication is supported. The amount of slave servers a master server has has no impact on the CRUD activities from their point of view. When employing CRUD operations, a schema-less architecture is advantageous since they are simpler to grasp at first look and easier to develop. A quicker database management system is MongoDB. This is a basic database that responds quite quickly is the solution you ought to make. MongoDB abandoned up on achieving scalability and much greater performance.
A weaker concurrency paradigm called as BASE (Basically Available, Soft State, Eventual consistency) is implemented for ACID (Atomicity, Consistency, Isolation, Durability) transactions. This indicates that updates finally spread to every node in timely fashion.
In conclusion, MongoDB is the best option if a developer wants to create a web application that is quick and adaptable. A traditional relational database is the best option if the application developer's primary focus is the relationship between data and having a normalised database that employs ACID transactions.

References

1. Jose, B., & Abraham, S. (2017, July). Exploring the merits of nosql: A study based on mongodb. In 2017 International Conference on Networks & Advances in Computational Technologies (NetACT) (pp. 266-271). IEEE.
2. Makris, A., Tserpes, K., Spiliopoulos, G., Zissis, D., &Anagnostopoulos, D. (2021). Correction to: MongoDBVsPostgreSQL: a comparative study on performance aspects. GeoInformatica, 25(1), 241-242.
3. Mukasheva, A., Yedilkhan, D., &Zimin, I. (2021, April). Uploading Unstructured Data to MONGODB Using the NoSQLBooster Tool. In 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST) (pp. 1-4). IEEE.
4. Tan, Q. (2018, October). Application of MongoDB Technology in NoSQL Database in Video Intelligent Big Data Analysis. In 8th International Conference on Management and Computer Science (ICMCS 2018) (pp. 104-108). Atlantis Press.
5. Tang, E., & Fan, Y. (2016, November). Performance comparison between five NoSQL databases. In 2016 7th International Conference on Cloud Computing and Big Data (CCBD) (pp. 105-109). IEEE.
6. Patel, N. (2019). Performance Evaluation of NoSQL Databases: HBase and MongoDB.
7. Retrieved July 17, 2022 fromhttps://www.mongodb.com/what-is-mongodb/features
8. Lieponienė, J. (2020). Recent trends in database technology. Baltic Journal of Modern Computing, 8(4), 551-559.
9. Faraj, A., Rashid, B., &Shareef, T. (2014). Comparative study of relational and non-relations database performances using Oracle and MongoDB systems. International Journal of Computer Engineering and Technology (IJCET), 5(11), 11-22.

10. Čerešňák, R., Matiaško, K., &Dudáš, A. (2021, March). Improvement of Data Searching in MongoDB with the Use of Oracle Database. In 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD) (pp. 1388-1393). IEEE.
11. Paci, H. (2022). SQL vsNoSQL databases from developer point of view. Industry 4.0, 7(3), 95-97.
12. Antas, J., Rocha Silva, R., & Bernardino, J. (2022). Assessment of SQL and NoSQL Systems to Store and Mine COVID-19 Data. Computers, 11(2), 29.
13. Wadhwa, M., &Kaur, E. A. Review of performance of various Big Databases. International Journal on Recent and Innovation Trends in Computing and Communication, 5(6), 179-182.
14. Araujo, J. M. A., de Moura, A. C. E., da Silva, S. L. B., Holanda, M., de Oliveira Ribeiro, E., & da Silva, G. L. (2021, June). Comparative Performance Analysis of NoSQL Cassandra and MongoDB Databases. In 2021 16th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.
15. Aghi, R., Mehta, S., Chauhan, R., Chaudhary, S., &Bohra, N. (2015). A comprehensive comparison of SQL and MongoDB databases. International Journal of Scientific and Research Publications, 5(2), 1-3.
16. Seo, J. Y., Lee, D. W., & Lee, H. M. (2017). Performance comparison of CRUD operations in IoT based Big Data computing. International Journal on Advanced Science Engineering Information Technology, 7(5), 1765-1770.
17. Čerešňák, R., &Kvet, M. (2019). Comparison of query performance in relational a non-relation databases. Transportation Research Procedia, 40, 170-177.
18. Ilić, M., Kopanja, L., Zlatković, D., Trajković, M., &Ćurguz, D. (2021, June). Microsoft SQL Server and Oracle: Comparative performance analysis. Book of proceedings of the 7th International conference Knowledge management and informatics, Vrnjačka Banja.
19. Almeida, F., Silva, P., &Araújo, F. (2019). Performance analysis and optimization techniques for oracle relational databases. Cybernetics and Information Technologies, 19(2), 117-132.
20. Bassil, Y. (2012). A comparative study on the performance of the Top DBMS systems. arXiv preprint arXiv:1205.2889.
21. What is Oracle (n.d), EDUCBA, Retrieved July, 2022, from https://www.educba.com/what-is-oracle/.
22. Database Concepts (n.d), ORACLE, Retrieved July 29, 2022, from https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT001.
23. Győrödi, C. A., Dumşe-Burescu, D. V., Zmaranda, D. R., &Győrödi, R. Ş. (2022). A Comparative Study of MongoDB and Document-Based MySQL for Big Data Application Data Management. Big Data and Cognitive Computing, 6(2), 49.
24. Chaudhary, J., Vyas, V., &Jha, C. K. (2022). Qualitative Analysis of SQL and NoSQL Database with an Emphasis on Performance. In IOT with Smart Systems: Proceedings of ICTIS 2022, Volume 2 (pp. 155-165). Singapore: Springer Nature Singapore.