International Journal of Scientific Research and Engineering Development-– Volume 4 Issue 3, May -June 2021

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Comparative Study of MongoDB and Other Databases

Purvi G R ,Dr. K Badari Nath

*(Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, BengaluruEmail: purviraichur1919@gmail.com)

*(Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, BengaluruEmail: badarinath.kb@rvce.edu.in)

----------**********----------

Abstract:

Our world has reached its pinnacle of development . The rapid development of technology, gadgets, industrial standards, and other products has been encouraged by the explosive boom. Gadgets that generate massive amounts of data, all of which necessitate mechanism for managing and manipulating data that is absolutely necessary. The data collected from numerous input and output sources is used to provide a certain infrastructure, but it is also subject to damage if not handled properly, which can result in data loss.

To combat this tragedy, many methods that operate in tandem to prevent such tragedy are used, one of which is the NoSQL MongoDB model. With its prominent part of self sharding, MongoDB is a cross-stage, archive structured database that offers elite and simple adaptation, ensuring viable information across the spectrum. Parts of the database are sharded amongst several workers, increasing the limit and flexibility as needed. This component is in charge of appropriating knowledge in numerous hubs in order to increase plate space and effectively load adjustment enquiries.

A database is a logically structured collection of data. This data is extracted from computers and is usually saved. Databases are often divided into two types. Data is saved in a range of databases. MongoDB is compared to various SQL and No-SQL databases in this article. MongoDB is opposed to MySQL, Oracle, and Cassandra databases in terms of performance and other metrics.

Keywords **—Cassandra , MongoDB, MySQL, Oracle.**

----------**********----------

I. INTRODUCTION

Data is being generated, evaluated, communicated, and then used to make business decisions at a far quicker speed compared to the past. To give an idea about the amount of data generated, IBM calculated in 2013 that over 90% of the data which was generated by the worlds was produced is the years 2011 and 2012. Generally, Relational Database Management Systems (RDBMS) have been used to handle vast volumes of data created by enterprise applications and to satisfy the storage requirements by various industries. They also allow for effective organized data manipulation. As a result, the concept of NoSQL was developed as an easier and more effective solution to handle massive data. NoSQL stands for "Not Only SQL," and it expands on the traditional method to data storage and retrieval.

This article takes three papers into consideration and summarizes them. Each of the papers are a comparative study of MongoDB with other databases. Rows and columns which consist the table make up the relational databases. These columns in relational databases, contain the information for a given class and the rows in these databases, specify an instance of that data which is determined by the category. Examples of relational

databases are MySQL, IBM, Microsoft Examples of non-relational databases are ArangoDB, CouchBase, MongoDB, Adabas. MySQL and Oracle are relational databases. Cassandra and MongoDB databases are non-relational databases.
There are four different types of non-relational databases:

1. Key-Value: Among the various models of non-relational databases, this model is the most straightforward to deploy. The idea is to utilize a hash table and assign a unique key and pointer to each piece of data; data can only be searched using the key. Amazon simple DB and Oracle DB are two examples of key-value databases.
2. Column-oriented: This database type was created with the goal of processing large amounts of data distributed across several servers. To store data, they employ a dispersed, sparse, multilayer ordered map. The number of columns in each entry is configurable; there are still keys, but they point to many columns. Cassandra and HBase are examples of column-oriented database.
3. Graph-stored: This database design is depicted as a network of interactions between defined parts of graph topologies, such as edges and vertices, in order to explain data. In No-SQL querying the database depends on which model is chose , different from SQL, which provides high-level declarative query languages.. Neo4j and Giraph are examples.
4. Document-oriented databases: In relational databases data is stored and columns and rows, opposite to the databases which are document oriented in which data is stored in the documents, Traditionally, these documents have a JSON-like format . JSON stands for Java Script Object Notation. They offer a natural approach to handle the data which strongly matches object-oriented programming. In document databases, the idea of schema is adaptable: each document may include various attributes. Modeling unstructured and polymorphic data benefits from this versatility.Modeling unstructured and heterogeneous data benefits from this versatility. Document databases also enable for robust queries, allowing any mixture of attributes in the document to be merged for querying the information. Examples are MongoDB and CouchDB.

II. MONGODB

MongoDB is an example of a document-oriented database system. It's difficult to store enormous amounts of data using traditional relational database management systems to manage unstructured data. Collections, not tables, are the most significant storage components of document databases like MongoDB. In MongoDB, these collections are made up of JSON and BSON files that are similar or different. Documents that are dependent on other documents or documents that are sub-documents. Collections are made up of documents that have a similar structure. It can be constructed anytime it is needed, with no predefined parameters. Documents containing instances within documents or instances within instances. Even lists or arrays of documents can be used in documents. MongoDB is an open source database.
One of MongoDB's main benefits over its competitors is its ability to scale up and down quickly. Furthermore, MongoDB is excellent for content management platforms as well as large data and associated technologies. In this article, we'll go through some of the most common MongoDB use cases from diverse sectors.
Following are a few use cases of MongoDB:

A. Single Page Application

MongoDB use case for HSBC:

For quite some time, HSBC, one of the world's leading financial institutions, has been focusing on the "Digital First" strategy. This campaign's major goal is to increase data accessibility, insights, and compatibility. In the previous five years, its data assets have more than doubled. The operational database of HSBC was stored in a relational

database. The breadth of data insights and data scalability have been constrained as a result. The database has grown significantly in size during the past five years, making it increasingly difficult to monitor.HSBC intended to develop a system that would allow its customers to obtain data on both basic and sophisticated terms, such as shares and bonds, as well as future trade projections, derivative trading, and so on. As a result, it teamed with MongoDB to meet these needs.

MongoDB use case for Expedia:

Expedia is a travel aggregation website run by an American travel corporation. These websites have extremely limited inventory because travel rates change from day to day. MongoDB is used to analyze large big data and store and in order to provide better use experience. Usually travelers plan their trips ahead of time. They also take into account a variety of possibilities. But, generally they miss about their earlier searches. Expedia devised a service called Scratchpad to address this issue. MongoDB is used to operate it. Personalizing the travel package according to the specific requirements while also automating note-taking process is facilitated by Scratchpad. This even lets you browse on your computer and saves the results to your phone or tablet.Scratchpad also gives you the best discounts based on your prior searches. To customize the delivery of the exact content, automate, filter a system must manage a lot of information and process a lot of information. Expedia uses MongoDB to help them do this functionality. Because MongoDB is a non-relational database, it's simple to customize all of the data with no difficulty and very fast. MongoDB was chosen because of its speed and scalability, as well as the ability to store information in a single view. Processing all of the data and delivering the findings in real time would have taken too long and been too sluggish.

B. IoT

MongoDB use case for Bosch:

Bosch is one of MongoDB's most intriguing use cases. Bosch , as among the leading companies is not far backward in using Internet of Things to change the way its organization functions. Significant investments in intelligent power tools and sophisticated connectivity were done by Bosch. The IoT has been a major driving force behind these initiatives. IoT isn't only about linked devices; it's also about providing real-time information to help businesses optimize their operations. Bosch uses MongoDB to offer the apps which work on IoT to its Bosch used MongoDB to help offer IoT-powered apps to its clients. Bosch aims to adopt IoT in many of its offerings as the IoT industry grows, and MongoDB is helping them accelerate this performance by adding high availability and scalability. With relational database systems, Bosch could've had a tough time executing and implementing the IoT applications.

C. Analytics

MongoDB use case for Chicago City:

The WindyGrid project was motivated by a sample model created by a single individual on a single system using MongoDB software, and it aimed to extract real-time information from Chicago government information. The WindyGrind project was among the nation's first in this industry, and it took only four months to complete.

It can not only handle information from seven million distinct bits of data, but it can also provide real-time insights on events. This initiative is seen as an important stage in making Chicago a better and more livable city. WindyGrid provides robbery, road construction, and public transportation reports, real-time traffic, among other things. Collecting each of these facts is not only time intensive, but relational databases make it nearly hard to perform and evaluate the results in a timely manner. However, using MongoDB, the City of Chicago was able to expand what was once just a sample, running on a single computer and using a single resource, into a massive operation. It evolved into a model capable of analyzing and forecasting activity throughout the whole region.

D. Single Page Application

MongoDB use case for urban outfitters:

Platforms for content handling are meant to offer tailored, customer-driver content with minimal effort. Urban Outfitters is a pioneer in apparel innovation, and it's leveraging the MongoDB platform to swiftly create and publish apps for its clients to provide a tailored experience and optimize their trip. MongoDB was picked for its rapidity, accessibility, and versatility in dealing with the unpredictable, periodic ups and downs of retail industry. Urban Outfitters' eCommerce app is unique in that it places a strong emphasis on user experience, and it has been a major success for the firm.In addition, many other online retail firms are getting on board and using non-relational databases such as MongoDB.

MongoDB use case for Forbes:

Forbes has been a global leader since 1917, and making the transition from conventional business media sources to digital ones was not simple. Forbes has never had a problem with content quality. It is one of the most reliable sources of information. The main difficulty, though, was the speed with which the material was provided. It resulted in several outages and poor page performance. Forbes chose to overhaul its systems in order to resolve these issues.It picked MongoDB as their content platform, and the effects were immediately apparent. Moving to MongoDB completely altered the architecture of Forbes' website, and for a fraction of the cost that Forbes was paying previously.

Furthermore, the full website revamp on customized CMS on MongoDB took only a few months, and the phone site was released in a month. In only one day, mobile traffic on Forbes' website increased from a modest 5% to 15% of total traffic. It has now comfortably surpassed the 50% level.

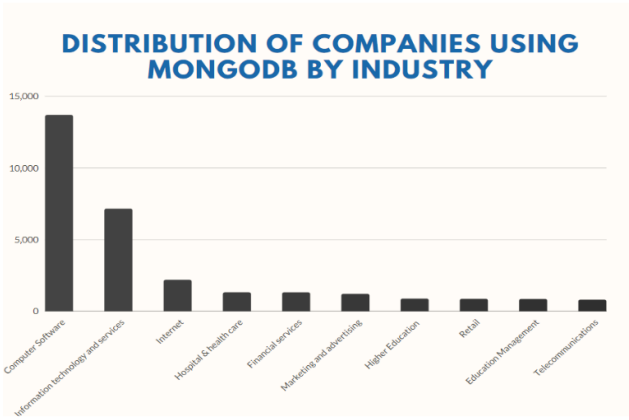

Fig. 1Grouping of companies using MongoDB by industry

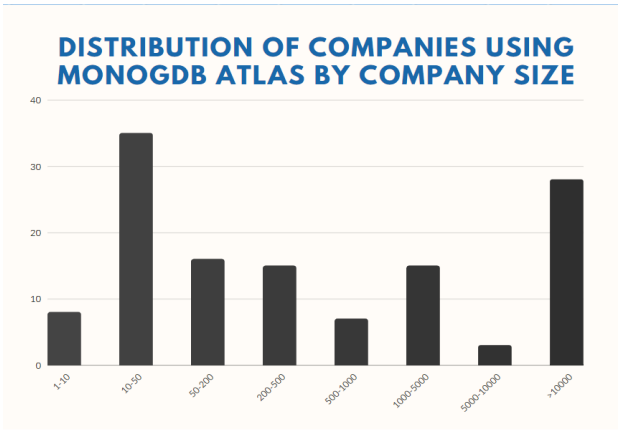

Fig. 2 Grouping of companies using MongoDB atlas by company size.

III. RELATIONAL DATABASES

A relational database organizes data into tables which can be linked—or related—based on data common to each. This property enables the user to retrieve the full table from the data which is present in one or more tables with a query.

Relational databases have the following characteristics:

- Tables containing values in columns inside rows must be used to store information in a relational database.
- The column name, ,table name, and information of the row's primary key must all be specified in

order for information in a column to be available.

- The database management system must provide systematic support for absent and unsuitable data, different from usual data and irrespective of data type.
- A live online catalogue must be supported by the database management system.
- Views must be updatable by the system.
- Update, insert, delete operations should be supported by the database management system.

IV. NOSQL DATABASES

The NoSQL databases have the following characteristics.

- Graph databases, document storage ,column-family storage, key-value storage, and even rows in tables are all forms that NoSQL systems collect and analyze information from.
- You may collect information from Nosql databases utilizing basic APIs that do not require joins.
- NoSQL databases may be stored on numerous CPUs while yet maintaining great performance.
- NoSQL databases are open source. Huge number of developers and users use NoSQL databases.
- NoSQL allows you to retrieve, manipulate and store information in a centralized place. Proponents of NoSQL have an open mind.
- They are distributed and horizontally scalable.

V. MONGODB AND MYSQL

Satyadhyan Chickerur et al, does a comparative study of MongoDB with MySQL database. The paper includes comparison of the syntaxes used in both the databases for various operations like insert, select, update and delete. The paper includes information on migration of data from MySQL to MongoDB. Migration includes two steps which are:

- CSV files to retrieve information from MySQL :
  A number of steps are involved in collecting information from MySQL to csv files. The input is a MySQL table, and the output is a csv file for the specified MySQL table.
- Uploading the csv files containing the collected information to MongoDB:
  A series of procedures are also involved in uploading the collected information to MongoDB in the form of csv files. A csv file prepared in the preceding phase is used as input. The result is a MongoDB collection with migrated data.

Queries were run to compare the performance of MySQL and MongoDB with respect to update, select, insert, and delete operations. The following tables show the times taken by both the databases to perform the operations.

The following table shows the comparison of time taken by MySQL and MongoDB to perform insert operations for different number of records.

TABLE I

INSERTION OPERATION MYSQL VS MONGODB

Number of tuples	MySQL (sec)	MongoDB(sec)
15000	1200	1050
20000	1600	1400
25000	2000	1750
30000	2400	2100

TABLE II

SELECT OPERATION MYSQL VS MONGODB

Number of tuples	MySQL (sec)	MongoDB(sec)
15000	1200	1050
20000	1600	1400
25000	2000	1750
30000	2400	2100

The following table shows the comparison of time taken by MySQL and MongoDB to perform update operations for different number of records. MongoDB takes less than one hundredth second, which is hard to replicate. MongoDB's updating procedure is lightning fast.

TABLEIII
UPDATE OPERATION MYSQL VS MONGODB

Number of tuples	MySQL (sec)	MongoDB(sec)
15000	150	-
20000	200	-
25000	250	-
30000	300	-

The following table shows the comparison of time taken by MySQL and MongoDB to perform delete operations for different number of records.

TABLE IV
DELETE OPERATION MYSQL VS MONGODB

Number of tuples	MySQL (sec)	MongoDB(sec)
15000	600	450
20000	800	600
25000	1000	750
30000	1200	900

VI. MONGODB AND ORACLE

Alexandru Boicea et al, does a comparative study of MongoDB and Oracle database. The contrasts between MongoDB database and Oracle database are highlighted in the study. For different actions, the article compares and contrasts the syntax of MongoDB and Oracle databases. Time was recorded to compare the performance of MongoDB and Oracle for various operations. . The following tables show the times taken by both the databases to perform the operations.

The following table shows the comparison of time taken by Oracle and MongoDB to perform insert operations for different number of records.

TABLE V
INSERTION OPERATION ORACLE VS MONGODB

Number of tuples	Oracle (msec)	MongoDB(msec)
10	31	800
10000	8750	681
100000	83287	4350
1000000	882078	57871

The following table shows the comparison of time taken by Oracle and MongoDB to perform update operations for different number of records.

TABLE VI
UPDATE OPERATION ORACLE VS MONGODB

Number of tuples	Oracle (msec)	MongoDB(msec)
10	453	1
10000	94	1
100000	1343	2
1000000	27782	3

The following table shows the comparison of time taken by Oracle and MongoDB to perform delete operations for different number of records.

TABLE VII
DELETE OPERATION ORACLE VS MONGODB

Number of tuples	Oracle (msec)	MongoDB(msec)
10	94	1
10000	94	1
100000	1234	1
1000000	38079	1

VII. MONGODB AND CASSANDRA

Dr. K. Anusha at el, is a big data analytics comparison between MongoDB vs Cassandra. The paper includes differences between the two databases, which are:

- Data framework: Cassandra is C-store database whereas MongoDB is a document

store. Cassandra saves information in tables with columns and rows.

- Scalability: Cassandra features several master nodes that will be used to incorporate information to the nodes, allowing it to grow better than MongoDB, which only has one master node that takes data. The output is utilized for all the other nodes.

- Query Language: Cassandra has its own query language called Cassandra Query Language, but MongoDB doesn't have one and every query is organized as JSON type objects.

- Aggregation: There is no aggregating mechanism in Cassandra. As a result, it aggregates data using other technologies such as Hadoop, Apache Spark, and others, whereas MongoDB has its own designed aggregation architecture. MongoDB recovers data using an ELT technique and turns items into aggregated results with this capability.
- Architecture: Cassandra is a highly available distributed peer-to-peer design, whereas MongoDB has a master-slave design with poorer fault tolerance.

- CAP theorem: Cassandra follows Partial Tolerance and Availability(PA) whereas MongoDB achieves Consistency and Partition(CP).

- Reads/Writes performance: Reads are efficient in MongoDB when compared to Cassandra. And as mentioned above MongoDB has only one master node. The storage capacity of that lone master node makes its ability is hard to write any new information into the database.

VIII. CONCLUSION

Time comparison for insert, update, select and delete operations is done for MongoDB and MySQL. The findings are promising for a variety of processes that might be used in big data applications requiring large databases. MongoDB performs better when compared to MySQL for the operations performed.

During the development phase, MongoDB enables flexibility. It has support for horizontal scaling built in. Using export and import tools, it is simple to implement and copy databases from one server to another. You may store sophisticated information in a centralized field — for example a reference, an array or an object. The key distinction between the two databases is that Oracle Database contains table relationships. There can be one-to-one, one-to-many, or many-to-many relationships. You may use these relations to link tables and run complicated queries. The replication is the biggest issue with the Oracle Database. You won't be able to clone the database as simply as you can with MongoDB. You have tools that can achieve that, but they aren't very quick. In compared to MongoDB, it is a lot slower database. MongoDB is a fast and versatile database that you can rely on. You may rely on the standard option, Oracle Database, if database speed isn't your primary issue and you need relations between tables and collections.

MongoDB and Cassandra both have their own set of benefits and drawbacks.The database we should use is determined by our relevance. In case of enhanced accessibility, Cassandra has the high ground. We can write to a cluster even if nodes fail because of the highly dispersed design. In many use situations, Cassandra's record for rapid write and read efficiency, as well as delivering correct linear scale performance in a scale-out , masterless design puts it ahead of its NoSQL database competitors.MongoDB, on the other side, is ideal to store data which is not structured. It offers a schema-less design that allows for efficient caching and reporting. Quick caching and having to log activities are essential streaming and real-time

applications. Because it offers secondary indexes, MongoDB is also useful for reducing query times. If you plan to expand your data operations quickly, Cassandra is a better choice.

Overall, Cassandra outperforms MongoDB for tasks involving large amounts of data which is not structured because it can handle several master nodes in a group. But MongoDB is excellent for workloads involving large amounts of data which not structured.As a result, depending on the volume of information to be handled and the project deliverables, they each have their own share of benefits.

REFERENCES

[1] Satyadhyan Chickerur, Anoop Goudar, Ankita Kinnerkar, "Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications",8th International Conference on Advanced Software Engineering & Its Applications,2015

[2] Alexandru Boicea, Florin Radulescu, Laura Ioana Agapin," MongoDB vs Oracle - database comparison", ResearchGate conference paper, 2012

[3] Dr. K. Anusha, Nichenametla Rajesh, Dr. M. Kavitha, N. Ravinder," Comparative Study of MongoDB vs Cassandra in big data analytics", Fifth International Conference on Computing Methodologies and Communication, 2021

[4] Mayur M Patil, Akkamahadevi Hanni, CH Tejeshwar, Priyadarshini Patil, " A qualitative analysis of the performance of MongoDB vs MySQL Database based on insertion and retrieval operations using a web/android application to explore Load Balancing – Sharding in MongoDB and its advantages", International conference on I-SMAC, 2017.

[5] Some Pros & Cons of Relational Databases | Toolbox Tech. Retrieved May 31, 2021, from https://www.toolbox.com/tech/enterprise-software/blogs/some-pros-cons-of-relational-databases-050108/.

[6] MongoDB: The most popular database for modern apps. Retrieved May 31, 2021, from https://www.mongodb.com/

[7] Bruce, Debra. "Top 6 MongoDB Use Cases in Single view, Analytics, IoT, & CMS." Knowledgenile,ww.knowledgenile.com/blogs/mongodb-use-ases/#:~:text=One%20of%20the%20interesting%20use%20ases%20of%20MongoDB. Accessed 6 June 2021.

