**Performance and comparison of MySQL (Relational Database) and MongoDB (Non-Relational Database)**

**Ritesh Aswin Raam Sivagangai Gopinath**

**X19154160**

**16-08-2020**

**Abstract:**

The main objective of this paper is to compare two different database management systems (DBMS), namely MySQL and MongoDB and to find out which one is better. Relational and non-relational databases follow a different set of procedures, whereas relational databases such as MySQL represent their data in rows and tables and NoSQL databases such as MongoDB represent their data as collection of JSON documents. In this we conducted our benchmarking using YCSB (Yahoo Cloud Service Benchmark) tool which is an open source specification used by DBMS vendors to evaluate tool performance. The test was done on a cloud computing platform called Amazon Web Service by creating an EC2 instance. At the end, a list of tests has been conducted to compare the data latency, performance of the databases. As we go further, the features of MySQL and MongoDB such as scalability and Performance & Accessibility are compared.

**Introduction:**

Over the past few years , the usage of IOT-based application has increased in everyday life, leading to massive data generation and thousands of images, pictures are created, so there is a need for a database to store this information so that it can be accessed anywhere, anytime. So set of databases are considered, among the databases the oldest one is the relational databases and it is implemented using Structured query language(SQL) where the information is stored in tables and columns and data can be added or tables can be modified without affecting rest of the database. Some of the examples of relational databases are MySQL, PostgreSQL, IBM DB2 and Microsoft SQL server etc. Since RDBMS (Relational database management system) contains a single attribute functions and generation of data in applications have increased over the recent years, the people are moving from relational to NoSQL databases which otherwise called as non-relational databases. In this data is represented in JSON(JavaScript Object Notation) documents, it stores vast amounts of unstructured data, it is not necessary to use tabular format to access the data, and it is not necessary to define the type of data in advance when storing the data , which makes it easier to operate compared to relational databases. Some of NoSQL database examples include MongoDB, Cassandra, DocumentDB etc.

**Key Characteristics of MongoDB and MySQL:**

**MongoDB:**

**Some of the characteristics of MongoDB are,**

1. Easy writing of Queries: MongoDB is a document-oriented database or semi-structured data with no joins and transactions. So, it will be very much easier to write queries.
2. Indexing: To optimize the performance of a database by minimizing the disk access indexing is used in any attributes, secondary indexing is also available in MongoDB [10].
3. Flexibility: MongoDB represent the data in JSON documents and helps to analyse and size up the value of data very quickly and maps to native programming language types.
4. Easy to use: Same as RDMS (Relational database management system) it is also very easy to install, maintain and use.
5. Storage: With the help of the function GridFS, any number of files can be stored in the MongoDB and it is also used as a file system [10].
6. Replication: Master/Slave replication is supported by MongoDB and there is a server failure the application moves to standby using automatic failure.
7. Use of multiple storage engines: Many storage engines types such as WiredTiger, MMAPv1 storage engines are supported, and it can also support pluggable storage engine API which enables third parties to build storage engines for them.
8. MapReduce: MongoDB also supports MapReduce functions which changes large volumes of data into aggregated results. Which allows users to obtain the result for which SQL GROUP BY clause is utilized [11].
9. Load Balancing: MongoDB uses Sharding to distribute their data against multiple machines and it also supports high throughput operations and implement of large data loads and all the Shards work on a single collection [10].

**MySQL:**

Some of the key characteristics possessed by MySQL are,

1. Ease of Management: The software is very easy to download and does not include any kind of complex operations and the tasks are scheduled using the event scheduler which makes easy to work.
2. Security of data: MySQL provides authentication features which make sure only registered or authorized person can only access the databases [9].
3. Easy to use: To access MySQL databases, only a few basic SQL information is needed and with few simple statements one can create and communicate with MySQL.
4. Open Source: It is a free software that can be accessed by anyone, it can be used in any platform anywhere, anytime.
5. Platform Independence: MySQL is a platform-independence framework that offers consistency and scalability for their applications to be distributed on different platforms.
6. Productivity: Stored procedures, triggers are used my MySQL that allows a developer to give an increased and efficient productivity [8].
7. High Flexibility: MySQL supports a large set of embedded application which is used to perform a specific set of functions. This makes MySQL more flexible.
8. Multitasking: MySQL is designed in such a way that it can run on any operating system like Windows, Linux, UNIX and it can run as a server on the same computer or another computer using local area network or internet.
9. Dual Password: The MySQL version 8 contains the default primary password and the secondary password, which is a simple way to switch to the new password.
10. GUI support: MySQL server is supported by integrated development environment called MySQL workbench where its main purpose is to provide an interface to work with databases more easily. It supports database modelling, server administration, SQL development and designing [8].

**Architecture of Databases:**

**MongoDB Architecture:**

MongoDB is open source data base that utilizes non-structured query language and document-oriented data model. MongoDB is considered one of the most efficient databases. In this there is no need to access tabular format to get the data, the architecture is built on documents and collections and documents to have different structure. The data represented in here are in a form of JSON documents, a highly elastic data model is followed and when you want to dynamically modify the schemas, there is no downtime.

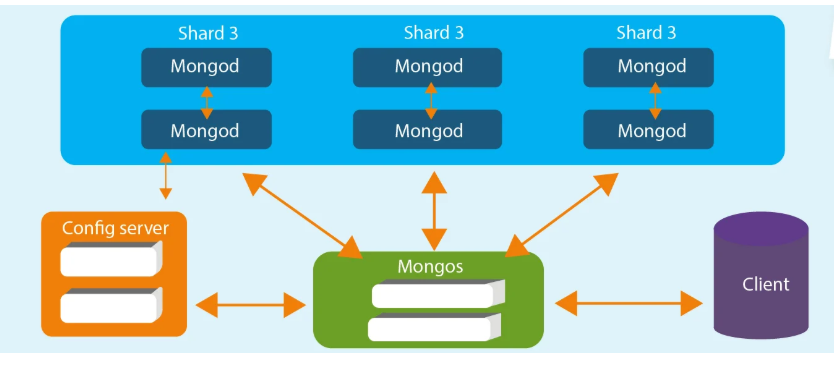


Figure 1: MongoDB Architecture [12]

Architecture of the MongoDB consist of 3 blocks Database, Collection and Document.

1. Database: It is a container which holds all the data stored in the MongoDB and is also called as a physical data container. Each database as its own files, and on a single MongoDB server there are multiple databases that exist.
2. Collection: Documents from a database group can be collected as collections, and are present in a single DB. There is no proposed plan or theory regarding collection. There will be different documents with different fields inside the collection, but the main objective of the collection is to share one common goal for one purpose.
3. Document: A set of key-value pairs forms a document and provides a dynamic schema. The advantage of having dynamic schemes is that a document must not have the same structure or fields in a single collection. A variety of data can also be found in the common fields.

This database is designed in a way to overcome the Scalability issues faces by traditional database systems and there will be no downtime when the application is being scaled. Since it possesses outstanding scalability, it makes data collection easy and provides continuous and automatic integration [12].

**MySQL Architecture:**

MySQL architecture varies significantly from other data servers, and this offers a fine range of services and is useful for many purposes. MySQL supports highly available redundant systems, OLTP (online transaction processing), embedded applications, content indexing and delivery software are there are much more services offered. Because MySQL is versatile it can be set up and run on a wide combination of devices. MySQL storage architecture follows a different set of rules by isolating query processing and other server tasks from data storage and retrieval, It has 2 separate engines, namely the storage and query processing engine, where the query and data are processed separately on those 2 engines and these benefits the user by letting them choose by what characteristics you want and what features and how your data will be stored.

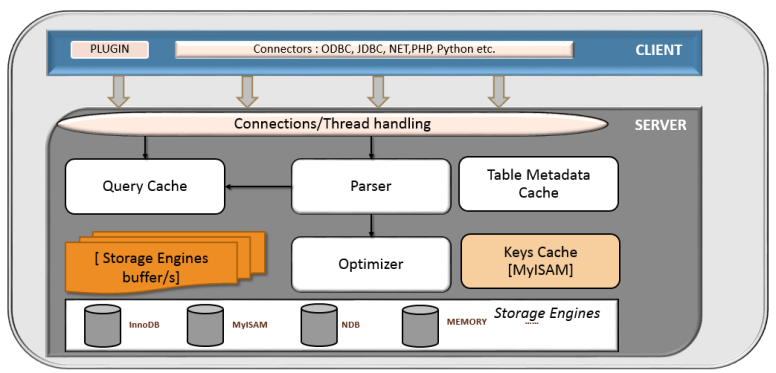


Figure 2: MySQL Architecture [13]

Architecture is classified into 3 layers,

The topmost layer is same as other servers or databases where the services carried out are like client server tools or They 're services that most client / server tools or servers need: handling of connections, security, authentication, and so on.

The second layer is the most important layer where important function of the database like optimization, analysis, caching, query parsing and built-in functions are carried out and all functions done in the storage is carried out in second layer.

The third and final layer is based on the storage layer where the storage engines are contained, this layer is responsible for data storage and recovery. The communication between is done through is storage engine API which hides the storage engine differences and makes them largely flexible on the query layer. In this, the API performs a set of basic operations such as starting the process and collecting the row that has the primary key, and these layers of storage will not interact with each other, they will only respond to server requests [14].

**Performance and availability:**

**Performance and availability of MongoDB:**

One of the main benefits of MongoDB over MySQL is the huge set of unstructured data where MySQL process only structured data. And it supports multiple insertion rate using a command called insertmany() in which multiple data can be inserted at a same time and the performance will also not be affected due to this. Since the data here is a collection of JSON documents, in this the data is stored as a collection of documents and it uses replica set which as a copy of same data which provides high set of availability and redundancy.

And if an admin wants to add some new fields to the data database it can be added or edited without affecting other documents. Since MongoDB comes with JSON support and the drivers are host to programming language it makes available for the developer who wants to communicate with API. It has an integrated tool called the MongoDB profiler that provides insight about the query and analyses all the queries running on the server [16]. Since most of the data is stored in the cloud and service-based, MongoDB is best suited due to its native-scale out architecture which is allowed by sharding, It is a process of data storage at very different locations and with the use of horizontal scaling more data can be written and read back as when data growth increases. It enables very fast access since it uses internet memory/cloud memory has its storage. MongoDB uses girdFS technology for retrieving images and videos and data, and it can store files that are larger than its document because of this the performance will be very high compared to MySQL [17].

**Performance and availability of MySQL:**

Performance of MySQL is comparatively low that of MongoDB, since MySQL is a traditional database that is still used in many companies due to its robust features that the MySQL possess rather than MongoDB. The small structured data in MySQL is addressed more efficiently compared to MongoDB, it also produces a very optimistic storage framework enables system administrators to configure a faultless productivity of MySQL database server.

It is modelled in a way that even the critical applications can be ensured ideal speed and distinctive memory caches for increased performance. MySQL provides a surveillance tool called a performance scheme which is used to monitor the operation of the SQL server at a low level and to investigate the server 's internal execution. In addition, the MySQL, which is used to balance high engine reliability and performance, follows the basic intent storage engine named InnoDB. It also supports for the ACID property (Atomicity, Consistency, Isolation, Durablity) and it is also stronger to table corruption which is unique compared to other databases [15].

**Scalability:**

**Scalability of MongoDB:**

MongoDB is considered due to its high performance since it performs horizontal scaling using sharding where it distributes the data across multiple machines which gives scalability for a large set of data. In takes a big amount of data and split those data into small data and store those in different servers whereas in MySQL it does only with the structured data and the data is stored in tables. We can adequately extract data if you request from the primary key, once the data can be requested by requesting all the servers if you do not have the primary key, MySQL does not support this feature. Load balancing of client requests is one of the scalable mechanisms which evenly send the request loads across the database. These approaches can be either client-based or server-based where in client based the client will be holding the requests and pics it using the round-robin fashion and in server-based it respond to the client requests and then load balances it. And in MongoDB it locks the whole database using the locking system called data object-level write locks. As the request load increases Bottlenecks in the processing path to read and write requests can quickly become scalability inhibitors [18].

**Scalability of MySQL:**

MySQL provides a thread pool plugin which is some other thread-handling model which is used to increase the performance and to minimize the overhead in the server. This plugin introduced a thread pool which enlarges the server performance by managing the client establishment. And the MySQL clusters performs automatic data partitioning and there will be a separate write operation for each partition which ensures scalability. By SELECT MAX which is used to fetch the results from the database quickly by finding the maximum id correctly using this command. The read and write operations is being distributed across all nodes which provides high scalability and with the help pf Sharding horizontal scalability is done in which the database is divided into small parts which will be easy to manage since MySQL manages structured and small data well. MySQL cluster provides auto-sharding and shared nothing clustering which is introduced to produce increased throughput and achieve linear scalablity [19].

**Literature Review:**

A comprehensive analysis and then the comparison of document based NOSQL with SQL relational databases was done by Raif et al. [1] where the paper aimed to perform CRUD (Creating, Reading, Updating and Deleting) operations on MongoDB and MySQL as a representation of the data models to find out which databases is better. In this paper the authors discussed the benefits, limitations and performance of each model in terms of CRUD operations.

In this paper the authors made their analysis on 4 different ways to figure out which is better, in terms of data structure, transaction support, query language and security aspects where both systems showed a substantial improvement and there are some drawbacks faced by these systems and finally insert, read, update and delete operations is performed on 2 different NodeJS scripts, at the end the authors stated that the updated version on MongoDB shows a very promising results in terms of CRUD operations while working on large data sets. So, in terms of choosing databases MongoDB continues to dominate compared to MySQL.

A performance comparison of various queries and commands is performed by Benymol et al. on both relational database and NoSQL databases. The aim of this paper is to find out how NoSQL is better in performance compared to relational database.

In their research they did the comparison on commands and queries using operations like insert, update, simple select and select with condition with 2 different datasets of large workload , at their comparison they showed that MongoDB showed a better performance in terms of response time to complete the queries compared to that of relational database(MySQL). So it was found that scalability and performance of large datasets is better in MongoDB compared to that of MySQL where many organizations motive is to convert from conventional database systems to NoSQL databases [2].

Dipina et al. [3] performed a performance evaluation between relational and non-relational databases where MySQL and MongoDB are taken respectively on a hypermarket application. In the research Searching and Insertion operations are performed by using the large set of hyper-market application on both databases, as a result it showed that when loading a small number of records there is no much difference shown in terms of response time by the databases, but when workload is increased on both MongoDB and MySQL there was a significant time reduce in execution of large workloads in MongoDB compared to that of MySQL since MongoDB can handle diverse types of data and built an application faster and it can remove the complex object-relational mapping (ORM) layer and in MySQL each item can contain only a single attribute where all the data’s are stored in each single records where in non-relational database’s each information can be stored in a single record. At the end authors stated that MongoDB is better when the workload is increased when compared to MySQL.

Similarly, a comparative study is made by Cornelia et al. between relational and non-relational databases namely MongoDB and MySQL. In this paper the authors did their comparison by implementing population records. They performed 4 different operations Select, Insert, Update and Delete on the registered population records. And the operations were done on 1, 100, 500, 1000, 5000, 25000, 50000 and application is supported by C#.

At the end it showed that in terms of select operations MySQL is efficient and fast compared to MongoDB but in terms of update operation there was a difference on 2 seconds for 5000 records and it terms of 50000 records the difference went up to to 29 seconds which benefits the MongoDB where in terms of delete operations there was a maximum difference when doing 50000 records like it went until 200 milliseconds whereas there is no much of a difference in comparatively small records. At the end author stated that MySQL is better performed in terms of search operations, whereas MongoDB is preferred in terms of updating and deleting operations, so MySQL is only suitable for small and medium-sized applications whereas the low-cost storage of a large data block is a decay taken by many companies so they prefer NoSQL databases such as MongoDB [4].

Further Seyyed et al. [5] performed the same comparison where how NoSQL databases are highly performed compared to that of relational databases. Main aim of this paper is to do the research on document-oriented MongoDB with SQL in terms of performance of common aggregated and non-aggregate queries where the same read, write, delete and select on the same e-commence data is performed in 2 different databases.

In evaluation of MongoDB the data where passed in a form of JSON files and in the SQL server java programming operations is used to run the different operations and they used JDBC to connect for connecting both databases to prevent the negative outcome of results, each operations is done like 100 times in some cases whereas 10 times in some cases. So, the authors stated at the end that MongoDB is scalable in the process of development and that data with a lot of complexity can also be inserted in the single field. MongoDB is better in terms of performance and response time compared to MySQL and the authors stated that they face trouble when the aggregate functions are done on non-key attributes.

A comparison is made by Mahmoud et al. [6] between 2 famous databases like MySQL from relational databases and MongoDB from non-relational databases by inserting a large amount of IoT (Internet of Things) data for evaluating its performance. And the authors also proposed a prediction model which chooses an appropriate database to store and manage a specific set of data.

Authors performed the evaluation in 3 parts, on the first part it compares the performance of the databases when loading a large IoT data and secondly it tells a decision of which database is better in saving resources and at last it compares both the databases and concludes which one is accurate. In the paper the authors explained what the impacts are by increasing workloads on latency and database size and the impacts of increasing the sensors for both insertion and selection operation. It was found that in terms of database size and latency MySQL performance is comparatively less than MongoDB. A prediction model, linear regression and non-linear regression is introduced to estimate the latency and to find the low latency-based data base. And it is found that non-linear is better compared to that of linear regression.

Another related work is done by Sowndarya and Suvitha where they did the comparison between MySQL and MongoDB, the aim of this paper is to cover the advantages and disadvantages of both the databases and their characteristics. In this the author followed ACID (Atomicity, consistency, isolation and durability) for SQL databases and CAP (Consistency, Availability and Partition Tolerance) theorem for MongoDB and the comparison, types and the query processing for both the databases were discussed. At the end of the paper the authors illustrated that that relational database will a perfect option for structured data, whereas for unstructured data or structured data with rapid growth then we can choose MongoDB over MySQL [7].

**Performance Test Plan:**

Here we perform the benchmarking on 2 databases namely MySQL for relational database and MongoDB for non-relational database using the benchmarking tool named YCSB (Yahoo Cloud Serving Benchmarking) where it utilizes set of workloads to check the performance & availability and scalability of MongoDB and MySQL.

The local machine used is,

* Processor: Intel® Core™ i5-9300 CPU @ 2.40 GHz
* Installed Ram: 8 GB
* OS: Windows 10
* System Type: 64-bit operating system, x64-based processor

The Virtual Machine used is,

* OS: Ubuntu 18.4(Bionic) (64-bit)
* Cloud Host: AWS EC2
* Instance ID: i-0e729e4f6a0c1a9c4
* Private IP: 172.31.65.88, Public IP= 100.25.160.60
* Flavor: t2.xlarge

The MongoDB version is 4.4.0 and SQL version is 14.14

**Test Strategies:**

The comparison between MongoDB and MySQL is done first creating an instance in the AWS and logging into the ubuntu machine using putty. After started running the instance MySQL server and client is installed and after that bench test is created and inside its user table is created to check the load. And in MongoDB since it posses in-built table creation functions, there is no need to create a table.

**Database Setup:**

This is the first step to run the test by logging into the AWS EC2 instance by typing the commands along with the associated IP address and all the installation as to be done in before to do the benchmarking. At first, the latest java version is installed after that I have installed MySQL which the permission given to read and insert it any time. And for running the script the test harness file is taken which is used to do the different set of operations and set of commands like runtest to run the operation and set of commands to suspend the program, fg command to run the suspended program again, the list of running programs can be listed using jobs command and the process is continued till be get the correct workload. Same as MySQL, after that MongoDB is also installed using the set of commands, and autostart is also added with system reboot using systemctl enable command, where in this no need to create a database it will be created automatically while runtime. After installing the latest version same as MySQL, the set of workloads are assigned by using the test harness file and the values are noted.

**Evaluation and Result:**

In this we did the test on MongoDB and MySQL with 4 different workload A, B, C and D using different operation counts 12500, 25000, 50000, 100000.

These are the sample workloads I have given for the below A, B, C, D

The workloads for A as follow,

recordcount=12500

operationcount=12500

workload=site.ycsb.workloads.CoreWorkload

readallfields=true

readproportion=0.5

updateproportion=0.5

scanproportion=0

insertproportion=0

requestdistribution=Zipfian

The workload for B,

recordcount=25000

operationcount=25000

workload=site.ycsb.workloads.CoreWorkload

readallfields=true

readproportion=0.95

updateproportion=0.05

scanproportion=0

insertproportion=0

requestdistribution=zipfian

The workload for C,

recordcount=50000

operationcount=50000

workload=site.ycsb.workloads.CoreWorkload

readallfields=true

readproportion=0.95

updateproportion=0.5

scanproportion=0

insertproportion=0

requestdistribution=Zipfian

The workload for D,

recordcount=100000

operationcount=100000

workload=site.ycsb.workloads.CoreWorkload

readallfields=true

readproportion=0.95

updateproportion=0.5

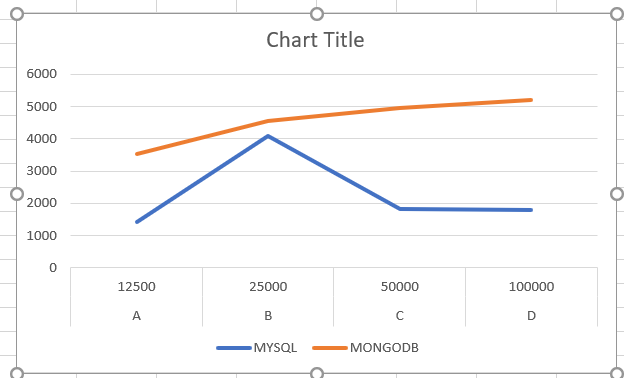
scanproportion=0

insertproportion=0

**Overall Throughput run operations comparison between MySQL and MongoDB:**

With the help of testharness script the set of the workloads A to D is run using set of following commands 12500, 25000, 50000,100000. It tells how both performed when analysed by set of workloads.

|  |  |  |  |
| --- | --- | --- | --- |
| **Workloads** | **Database** | **Total Operations** | **Throughputs** |
| A | MongoDB | 12500 | 3522.11891 |
| A | MySQL | 12500 | 1417.87659 |
| B | MongoDB | 25000 | 4553.73406 |
| B | MySQL | 25000 | 4084.29995 |
| C | MongoDB | 50000 | 4951.47554 |
| C | MySQL | 50000 | 1824.48458 |
| D | MongoDB | 100000 | 5216.21199 |
| D | MySQL | 100000 | 1782.49942 |



Overall in the chart tells us about the total throughput of operations, MongoDB shows throughput ranging from 3522 to 5216 and MySQL ranges from 1417 to 1782 where it clearly shows that MongoDB does give a very high throughput when compared to that of MySQL databases.

**Average Latency against the number of Read operation:**

In this the comparison is made between two databases like how much time taken to read the data when executed is compared.

|  |  |  |  |
| --- | --- | --- | --- |
| **Workloads** | **Database** | **Read Operations** | **Throughputs** |
| A | MongoDB | 12500 | 236.8092 |
| A | MySQL | 12500 | 161.3258 |
| B | MongoDB | 25000 | 193.456 |
| B | MySQL | 25000 | 143.7534 |
| C | MongoDB | 50000 | 179.3968 |
| C | MySQL | 50000 | 156.3274 |
| D | MongoDB | 100000 | 174.8196 |
| D | MySQL | 100000 | 156.9432 |

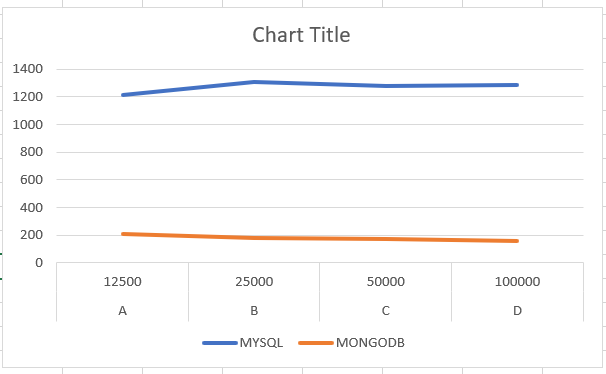
The bar diagram states that there only small difference between MySQL and MongoDB databases, in my execution it stated that MySQL comparatively have small runtime that of MongoDB, when MySQL is executed the time respond to read the set of data is low in my case.

MySQL gives high performance here even when a large set of workloads is benchmarked. In my case MySQL performed well compared to that of MongoDB which is interesting to watch.

**Average Latency against the number of Insert operation:**

The insert operation for various workloads is implemented and tested, in this below graph it clearly stated that performance of MongoDB is high compared to that of MySQL, like when a data is inserted, MongoDB performs the data within a matter of seconds while the MySQL takes more time like there was a drastic difference. MongoDB is comparatively having a high latency at 20000 operation count and a least count of about 160000, in case of MySQL the latency is extremely high when compared to MongoDB.

|  |  |  |  |
| --- | --- | --- | --- |
| Workloads | Database | Read Operations | Throughputs |
| A | MongoDB | 12500 | 207.66776 |
| A | MySQL | 12500 | 1214.99232 |
| B | MongoDB | 25000 | 181.3304 |
| B | MySQL | 25000 | 1309.36336 |
| C | MongoDB | 50000 | 171.14192 |
| C | MySQL | 50000 | 1279.22062 |
| D | MongoDB | 100000 | 160.05075 |
| D | MySQL | 100000 | 1288.96509 |



**Average Latency against the number of Update operation:**

The update operation for various workloads is done and checked and in the below graph it clearly shown that the time taken by the MongoDB in case to update or edit a file is comparatively low as compared to that of MongoDB. Even though there was a sudden increase in throughputs while reading the operations 25000 in MongoDB but then again, the throughputs came back to normal, but MongoDB latency stays close to constant, for MySQL the throughputs kept on increasing.

|  |  |  |  |
| --- | --- | --- | --- |
| Workloads | Database | Read Operations | Throughputs |
| A | MongoDB | 12500 | 250.692808 |
| A | MySQL | 12500 | 1197.03219 |
| B | MongoDB | 25000 | 289.016614 |
| B | MySQL | 25000 | 1891.21528 |
| C | MongoDB | 50000 | 209.930099 |
| C | MySQL | 50000 | 1273.25547 |
| D | MongoDB | 100000 | 202.417033 |
| D | MySQL | 100000 | 1312.26572 |

**Conclusion:**

At the end we were able to do the bench marking using the 2 different relational and non-relational databases namely MySQL and MongoDB with the help of the bench-marking tool YCSB run on the amazon EC2 instance. Set of benchmarking is done and 4 workloads is run against each system and the following Average latency of read and write, insert and throughput value is checked for both data bases. Each has its own set of advantages and flaws as well, at the end of the paper and comparing all the test MongoDB is quite outstanding in all the performance compared to that of MySQL, since MySQL was significant in terms of read operation compared to that of Mongo, it runs structure data in a correct way, even though due to many advancements the people are preferring over Mongo due to its robust features and ability to handle the large amount of unstructured data which is impossible by the traditional MySQL systems. As previously concluded and many studies made MongoDB outperformed MySQL in many ways. After the 4 set of operations tested on 4 set of workloads it has proved that the Performance of MongoDB is suitable to large set of data whereas MySQL is suited to process a limited amount of data. Since due to this nowadays many organizations are moving to NoSQL databases due to its high level of scalability and performance.

**References**

1. Deari, Raif & Zenuni, Xhemal & Ajdari, Jaumin & Ismaili, Florie & Raufi, Bujar. (2018). Analysis And Comparision of Document-Based Databases with Relational Databases: MongoDB vs MySQL. 1-4. 10.1109/InfoTech.2018.8510719.
2. Jose, B. and Abraham, S. (2020). Performance analysis of NoSQL and relational databases with MongoDB and MySQL. Materials Today: Proceedings, 24, pp.2036–2043.
3. B, Dipina & Salim, Shirin & mariam varghese, Surekha. (2016). Performance Evaluation of MySQL and MongoDB Databases. International Journal on Cybernetics & Informatics. 5. 387-394. 10.5121/ijci.2016.5241.
4. Győrödi, Cornelia & Gyorodi, Robert & Pecherle, George & Olah, Andrada. (2015). A comparative study: MongoDB vs. MySQL. 1-6. 10.1109/EMES.2015.7158433.
5. Aboutorabi, Seyyed & Rezapour, Mehdi & Moradi, Milad & Ghadiri, Nasser. (2015). Performance evaluation of SQL and MongoDB databases for big e-commerce data. 10.1109/CSICSSE.2015.7369245.
6. M. M. Eyada, W. Saber, M. M. El Genidy and F. Amer, "Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments," in IEEE Access, vol. 8, pp. 110656-110668, 2020, doi: 10.1109/ACCESS.2020.3002164.
7. S. Palanisamy and P. SuvithaVani, "A survey on RDBMS and NoSQL Databases MySQL vs MongoDB," 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2020, pp. 1-7, doi: 10.1109/ICCCI48352.2020.9104047.
8. www.javatpoint.com. (2011). MySQL Features - javatpoint. [online] Available at: <https://www.javatpoint.com/mysql-features>.
9. Edureka Community. (n.d.). What are the features of MySQL? [online] Available at: <https://www.edureka.co/community/32994/what-are-the-features-of-mysql>.
10. G. Zhao, W. Huang, S. Liang and Y. Tang, "Modeling MongoDB with Relational Model," 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies, Xi'an, 2013, pp. 115-121, doi: 10.1109/EIDWT.2013.25.
11. Singh, S. (2019). A comparative study of NoSQL databases: MongoDB and HBase
12. Swetha Doddamani (2019). What is MongoDB? [online] Intellipaat Blog. Available at: <https://intellipaat.com/blog/what-is-mongodb>.
13. Lalit’s Blog. (2016). MySQL Architecture and Components. [online] Available at: https://lalitvc.wordpress.com/2016/11/03/mysql-architecture-and-components/ [Accessed 16 Aug. 2020].
14. Oreilly.com. (2019). O’Reilly Media - Fire Outage. [online] Available at: <https://www.oreilly.com/library/view/high-performance-mysql/9780596101718/ch01.html>.
15. Datamation.com. (2016). 8 Major Advantages of Using MySQL. [online] Available at: <https://www.datamation.com/storage/8-major-advantages-of-using-mysql.html>.
16. Hackr.io. (n.d.). MongoDB vs MySQL: Difference Which You Need to Know [Updated]. [online] Available at: <https://hackr.io/blog/mongodb-vs-mysql>.
17. DataFlair Team (2018). Advantages of MongoDB | Disadvantages of MongoDB - DataFlair. [online] DataFlair. Available at: <https://data-flair.training/blogs/advantages-of-mongodb/>.
18. quabase.sei.cmu.edu. (n.d.). MongoDB Scalability Features - QuABaseBD - Quality Architecture at Scale for Big Data. [online] Available at: <https://quabase.sei.cmu.edu/mediawiki/index.php/MongoDB_Scalability_Features>.
19. www.devx.com. (n.d.). Top 5 MySQL Scalability Tips. [online] Available at: <http://www.devx.com/dbzone/top-5-mysql-scalability-tips.html>.