Performance Evaluation of MYSQL and MONGODB using YCSB Benchmark Tool

Vishal Kumar Singh
18201687
Data Storage and Management
MSc in Cloud Computing

24th April 2020

Abstract

Nosql and sql databases have different ways of storing and retrieving data in a database, and have some unique capabilities of their own. Many companies still use Mysql databases for their data storage and management system, though Mongodb is considered a database for new cloud technologies and storage for the emerging companies. Mongodb is a popular and most used Nosql database system and Mysql is a popular sql database system. In this paper I will be comparing and evaluating the performance of Mysql and Mongodb, to analyse the databases and their functionalities. YCSB an open source benchmarking tool by Yahoo will be used to obtain the score and results of the test conducted. I have conducted the test on multiple workloads of the YCSB tool, from the benchmark results and evaluation Mongodb emerged to be the better performer in all the workloads.

Contents

1 Introduction

Database is a collection of organized information stored in a systematic way to easily update, manage and access the data in the database. There are two types of databases, namely SQL database and NOSQL database. SQL database is a relational database (RDBMS) and has structured data and are vertically scalable, some examples of sql databases are MYSQL, oracle database, sql server etc. NOSQL databases are non-relational databases and has unstructured data and are horizontally scalable, some examples of nosql databases are Mongodb, cassandra, Hbase etc. Although both of these databases have their own pros and cons, nosql databases seems to have gained popularity in the market as it stores document oriented and distributed data than structured data like sql. Lets further discuss about Mysql for sql database and Mongodb for Nosql database.

1.1 Mysql

Mysql is a relational database management system and it is an open-source software, it stores data in tables and in that rows and columns contain data in systematic and organized order. It was written in C and C++ and was released in may 1995, the latest version of mysql released is version 8.0 which is said to be two times faster than its previous version 5.7. Mysql is used by many organizations even today as it is fast and the best database system to store structured data, many companies still prefer storing data using mysql. Some of their main customers are google, facebook, ebay, netflix, linkedin, twitter and others. Mysql can easily be downloaded and installed on different platforms, i have installed mysql on linux-ubuntu by downloading the packages and setting up the environment on it.

1.2 Mongodb

Mongodb is a non-relational database management system and is more of a document oriented database and stores unstructured data comes under Nosql databases, it stores data documents and graphical information and other data. Mongodb was written in pyhton, javascript, C++ and was released in feb 2009, the latest version of mongodb is 4.2.5. Mongodb is also free to download and install on different platform. Mongodb is considered as the most popular database for modern applications and is used by many organizations, some of their main customers are facebook, google, adobe, sap and others. Mongodb is one of the best database for creating and visualizing graphs of the data stored in the database, it also has many distinctive features. I have installed mongodb on linux-ubuntu by downloading and installing and setting up the environment on it.

1.3 YCSB

YCSB is yahoo cloud serving benchmark tool, which is used to retrieve and obtain performance benchmarks of databases running on the cloud and other computer programs. YCSB is also an open source program which was developed by yahoo in 2010, it was developed to measure the performance and compare different computer programs and databases. Ycsb has five different workloads with different read and update loads namely, Workload A, B, C, D, E. YCSB is used by many industries to compare their database performances, it is mostly used to compare the performance of Nosql database systems. I will be using YCSB to compare Mysql and Mongodb for their overall performance in ubuntu cloud platform. I have installed YCSB in Ubuntu server running on openstack cloud platfrom.

2 Key Characteristics of Mysql and Mongodb

Mysql and Mongodb both have some distinctive features and characteristics which differentiate both of these databases management systems. However, mysql is a sql database system and mongodb is nosql database system, so there will be certain features which is offered by one and not the other.

2.1 Mysql Characteristics and Features

Mysql is a sql database and relational database, the characteristics of mysql are as follows:

- Mysql stores data and individual entries as rows in a table followed by section data as columns in a table.

- Mysql as the name suggests only stores structured data in the database and the input must follow certain syntax.
- Mysql has JOIN operation, which allows it to query across multiple tables in the database.
- It supports triggers to maintain the integrity of data in a database and it is automatically executed
- Mysql has built in support of replication, which allows data sharing to ensure consistency.
- Full text searching and indexing.
- Provides both transactional and non-transactional engines of storage.
- Mysql functions uses ACID paradigm, which stands for atomicity, consistency, isolation, durability.
- Supports complex transactions.
- Atomic data definition statements(ADDL), this statement combines the data dictionary updates, operations, and binary log writes into a single atomic transaction.

2.2 Mongodb Characteristics and Features

Mongodb is a nosql database and non-relational database, the characteristics of mongodb are as follows:

- Mongodb stores data as documents, as json like documents and these documents stored belong to a class or group as stored in collection.
- Mongodb stores unstructured data, it can store documents of different kinds and different structure in the same class or group as collection. It is dynamic.
- Mongodb supports documents and arrays of multidimensional data types.
- Mongodb allows indexing of documents with primary and secondary indices.
- It supports load balancing.
- Mongodb has Ad-hoc queries.
- MongoDB supports aggregation, which provides three ways to perform aggregation namely, map-reduce function, single-purpose methods, aggregation pipeline.
- It allows server side execution of JavaScript and can be directly sent to the database for execution.
- It supports data locality, auto-sharding and rich data model.

3 Databases Architecture

In this section, i will be detailing and describing the architectures of Mysql and Mongodb.

3.1 Mysql Architecture

Database architecture of Mysql has different layers and each of these layers has different components in it, as shown in figure 1.

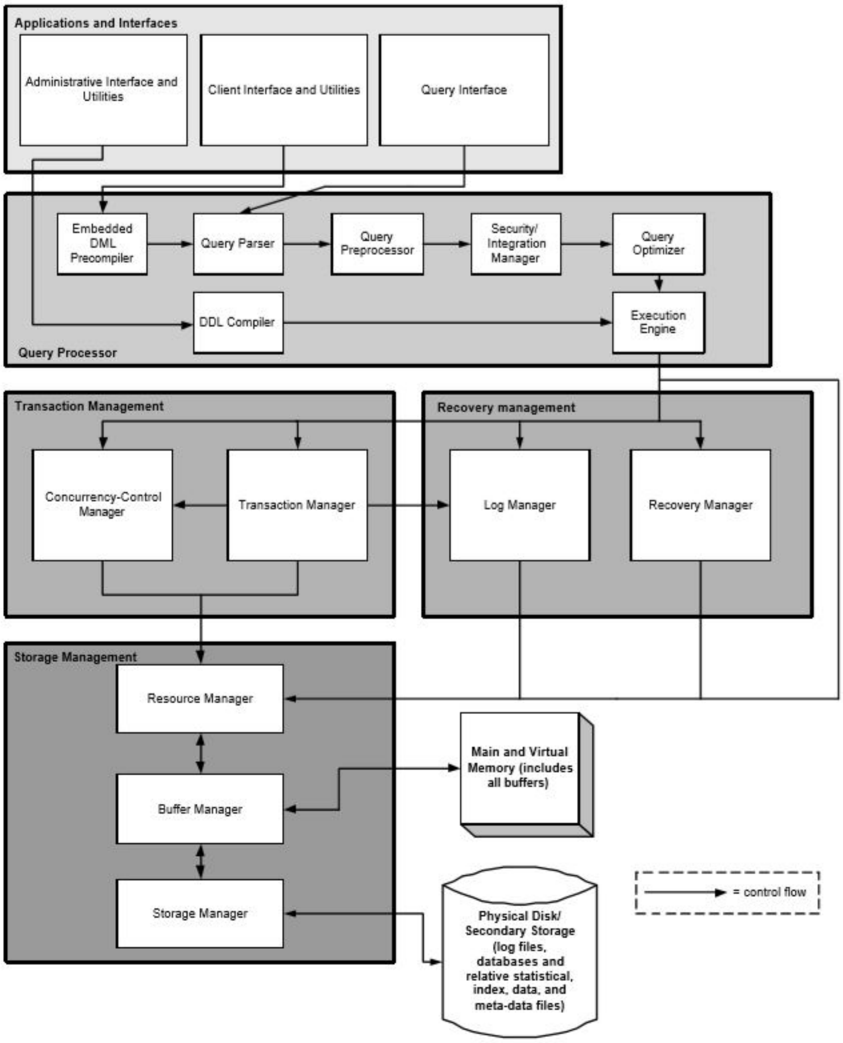

Applications and Interfaces
Administrative Interface and Utilities
Client Interface and Utilities
Query Interface
Embedded DML Precompiler
Query Parser
Query Preprocessor
Security/ Integration Manager
Query Optimizer
DDL Compiler
Execution Engine
Query Processor
Transaction Management
Recovery management
Concurrency-Control Manager
Transaction Manager
Log Manager
Recovery Manager
Storage Management
Resource Manager
Buffer Manager
Storage Manager
Main and Virtual Memory (includes all buffers)
Physical Disk/ Secondary Storage (log files, databases and relative statistical, index, data, and meta-data files)
= control flow

Figure 1: Mysql Detailed Conceptual Architecture.

The Application layer is responsible for the user and clients to interact with the database system of mysql. There are three main components in this layer, which describes the kind of users accessing the mysql database. The admin use administrator interface with admin rights to the database, the client interacts with client interface using programming languages. The query user uses query interface of mysql, that gives access to send statements and view results from the server.

The next layer is the logical layer also named as query processor, this layer is responsible for viewing or updating the data stored in the database. This layer has many components involved and each component has certain policies and tasks to perform, the output from the application layer becomes the input for these components. Compilers are responsible to convert the requests received into sql statements to have a direct interaction with the database, the query parser has to parse the query in a tree structure for the next components to understand easily. Preprocessor checks the syntax of the statements, security checks for the permissions to access and allows the statements to pass, optimizer is to make the execution process as fast as possible. Finally the execution engine executes the query received against the database.

Transaction manager is to make sure that the transaction is executed automatically and is logged, it also resolves any situations and deadlock issues that may occur. It is also responsible for assigning ROLLBACK and COMMIT sql commands. Recovery management has two components, log manager has to log and create entries of every executed operation in the mysql database. Recovery manager is to roll back the database to its last stable state. Storage manager is responsible to manage requests between secondary storage and buffer manager and make request to the physical disks, buffer manager creates buffer memory for use by updating and viewing data process. Resource manager manages the resources and receive requests from the execution engine and put them into table requests from the buffer manager.

3.2 Mongodb Architecture

Mongodb architecture in the core works and store documents in JSON style, it uses BSON as the document storage format which is an extented version of JSON (Binary JSON).

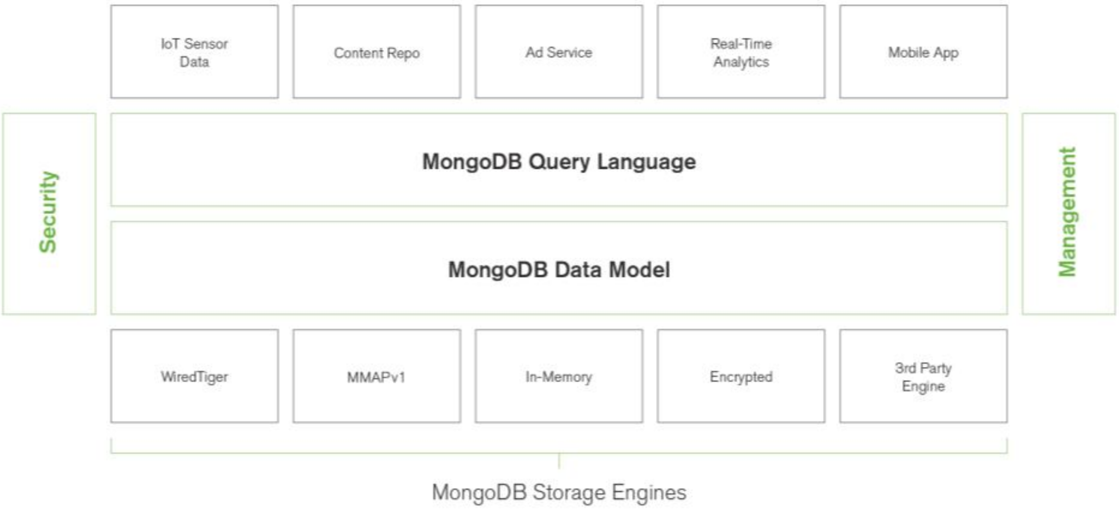


Figure 2: Mongodb Storage Architecture.

The storage engine of mongodb has many components involved, Mongodb data model and mongodb query language are the main components. These components handle the data and query received in the database and storage of the data. Security is for the security policies and management to manage the ongoing process in the database.

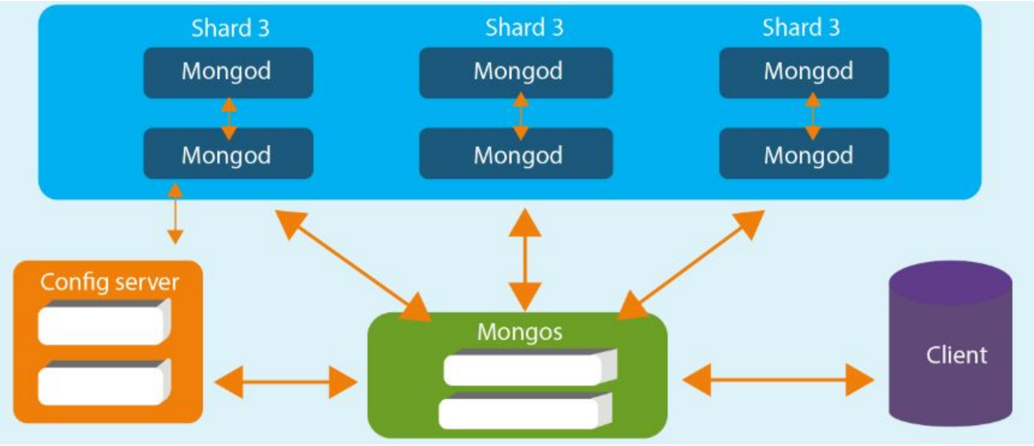


Figure 3: Mongodb Architecture.

Mongodb stores data in forms of documents unlike mysql in rows and tables, mongodb can store unstructured data and documents together in a database. Mongodb Collection is a group of documents similar to a table in sql databases, a single collection can have different kinds of documents all used for a similar purpose. Mongodb is a dynamic scheme database, so it can different schemes and can be changed anytime, the documents stored can be any graphical document or any format based document.

4 Requirement C Section Topics

4.1 High Availability and Performance

MYSQL

Mysql achives high availabilty and performance using group replication.

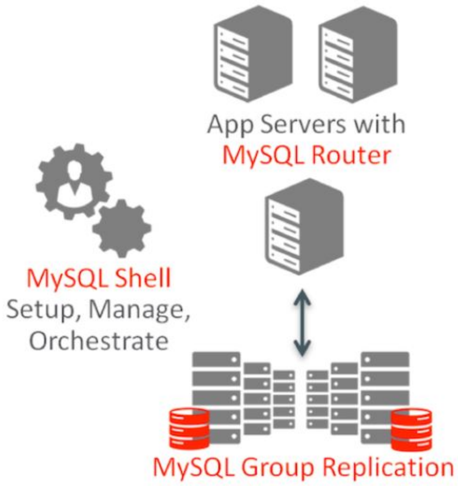


Figure 4: HA solution for MYSQL Database.

Mysql achieves high availability and performance using group replication. **Group replication** provides high availability with group management, conflict handling and detection, data consistency and database failure and detects any node failure, it does all these without any custom requests or intervention. It implements on both single primary mode and multi master mode. By using a powerful group communication system it, and implementing a paxos algorithm , the group automatically organizes and coordinates the consistency, replication and management which makes the Mysql database highly available and increases the performance.

Elasticity: with group replication, servers form a group by coordinating. The group reconfigure itself automatically to ensure that any new member joining is synced with the group. This process makes it efficient to scale database capacity down and up easily and as per required, hence providing elasticity.

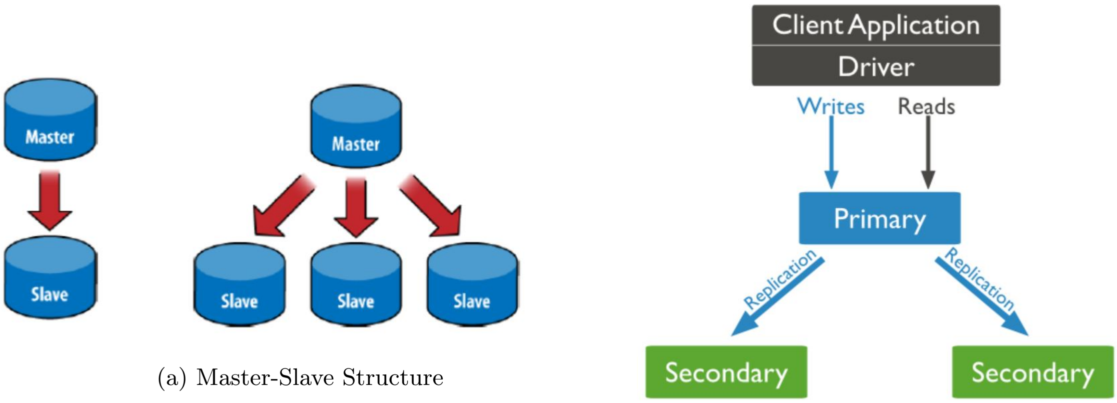
Failure Detection: The group replication has a failure detector, to find any databases or servers that has failed and report them and notify the members of the group to reconfigure.

Fault Tolerance: This works when one of the servers or databases fails, it automatically assigns a different database and the mysql database functions without any interruption and the risk of data being lost.

Monitoring and self healing: Group replication has monitoring capabilities which constantly monitors the nodes, cluster in the database and take actions if any issues arise. Self healing is sync the data when a database is removed and added or fails and starts again.

These are some of the features which makes Mysql server perform better and provide high availability.

MONGODB



(a) Master-Slave Structure

(b) Replica Set

Figure 5: HA solution for MONGODB Database

High availability and performance of mongodb cluster database has a few approaches. **Master-slave approach**: In this approach the master node runs the tasks and when it fails, the slave node comes and takes over the control and runs the task. When master node is back online it takes backs the control from slave node.

Dual-machine running approach: In this two master nodes runs the task and if any of the other fails, the other node takes the control.

Replica Set: This mechanism has two purposes, which provides high performance and availability.

- One is for failure recovery and reduce data redundancy, when one node fails or is down, it use the replica in place for recovery.
- The other purpose is to increase performance by splitting the read-write workload between the replicas from the primary node to decrease the workload pressure and get the work done faster.

Sharding Technology: This feature is new and very useful, when the data amount is very large, it divides and runs the data on different machines to reduce load on CPU and memory.

4.2 Security

MYSQL

Mysql has varoius security features and plugins provided, some of them are discussed below.

- **Authentication**: Authentication plugin is used by user or clients to login and connect to the mysql server, there are several protocols and plugins for authentication:
  - Native authentication
  - SHA-256 pluggable authentication
  - Caching SHA-2 authentication
  - Windows authentication
  - Peer credential authentication
- **Connection Control**: This controls and collects information about the incoming connection between the server and the client and delays the response of server as necessary.
- **Password Validation**: This defines policies for passwords and also measures the password strength, and also validates the password entered with the databases and gives access to the user.
- **Mysql Firewall**: This is an application level firewall, this allows the admin of the database to permit or deny statement execution by checking the white-lists of statements accepted previously. Firewall can run on detecting, protecting and recording mode to secure the connection.

- **Mysql Audit**: Mysql audit uses an open API to allow standard polices for monitoring, blocking connections, logging and activity of execution of quires. It keeps track of all these process to have a detailed report of the process involved and going on in the mysql server databases.

MONGODB

Mongodb has many security features to secure mongodb database and its components,some of the few features are listed below.
Authentication:

- Authentication is done by username and password, mongodb support multiple authentication mechanism.
- SCRAM-SHA-1.
- SCRAM-SHA-256.
- x.509 Certificates.

Authorization: Mongodb has few authorization methods involved, which allows different kinds of user with different privileges over the database.

- Role based access control
- Enable access control
- Managing users and roles

Encryption: Mongodb does client-side encryption, in which it can specifically encrypt data and sensitive information like ssn and encrypted fields are stored in the form of binary data. Mongodb supports two encryption mechanism:

- Explicit encryption
- Automatic encryption

These are some of the security features of mongodb. Mongodb is a secure and trusted, popular database management system.

5 Literature Survey

In this section i will review some journals and works done by authors in benchmarking mysql and mongodb databses for their performance in various aspects and fields.

In this article Aghi*et al.* [11] have compared Mysql and Mongodb databases with small, medium and large datasets. The article is well constructed with much detailed specifications, abstract and introduction clearly describes the work done in the article. The authors have compared mysql and mongodb for three operations in a database, insertion time, join time and retrieval time, similar work done in this article [13]. The authors claims are well supported with results using graphs and charts in the article. From the experiments and results we can see that mongodb has less insertion time than mysql and less join time as well for all three types of datasets. Retrieval time is almost similar for mysql and mongodb for small and medium datasets, but for large datasets mongodb outperforms mysql by a close mark. The authors have well documented the results and the conclusions is well described. To conclude by the works of the authors mongodb performs much better than mysql in the experiments conducted by the author.

B.F.Cooper*et al.* [12] have used YCSB benchmark tool to compare SQL and Nosql databases, the authors have selected Mysql, hbase, cassandra and pnuts to test the performance of these databases using YCSB tool, a similar approach have been conducted by the authors in this article[14]. The article is in a systematic order and well detailed descriptions about the tool and the databases. The authors have conducted test on different workloads of YCSB tool, the results are included in the journal using graphs. On workload A both cassandra and hbase perform better than mysql and pnuts, both the nosql databases performance are similar and both sql databases performance is also similar. However, nosql performs better than sql databases. On workload B also the results were the same. The authors claims are well supported by the results and evidence provided in the journal. Authors have concluded that nosql databases perform better and score higher in every benchmark test.

Patil *et al.* [13] have analysed the performance of mongodb and mysql databases based on the insertion and retrieval time. The authors have made many grammatical mistakes, one in the title of the journal. The journal have much detailed explanation about the databases and related works, but not much evidence and details on experimented methods. The test conducted are provided with the results using graphs, it clearly shows from the results that mongodb insertion time is much less than mysql insertion time. The authors have used a common and efficient way of concluding the article. The article does not have enough evidence to continue the experiments conducted, but the claims made by the authors are supported by the results and graphs provided in the article.

Dipina Damodaran *et al.* [15] have evaluated the performance of Mysql and Mongodb databases for their insertion and search operations execution time. The sections in the article are in a systematic order and well constructed, but the pictures included are not well placed in the article. In the test conducted by the authors, it is evident that mongodb takes much less time in both inserting and searching operations for all number of records, for larger number of records it performs exceptionally well than mysql database. The article has evidence and results using graphs which clearly shows the difference between the databases in execution time taken, but the method used to conduct the test are not well documented in the article. To conclude the authors have also found mongodb to be a better performing database than mysql database.

In this article Ahmed .M *et al.* [16] has analysed the performance of mysql database on fedora and ubuntu servers. The article sections are well constructed with detailed explanations, the abstract and introduction is well formed. In this journal the authors have recorded the values of CPU, RAM and read and write operations for different table sizes. From all the test conducted it was evident that mysql database performed better in ubuntu linux operating system than fedora operating system. The authors claims are well supported by the results and graphs included in the journal, it also has evidence to continue the experiments conducted. To conclude the author, ubuntu OS is better for running mysql server than fedora OS and for larger table sizes the difference is much noticed when compared to small table sizes. However, both these linux OS can run with limited computer specifications.

Nyati S.S *et al.* [17] have evaluated the performance of unstructured Nosql data over distributed framework, the authors have selected Mysql and Mongodb for their experiments. The article is well constructed with detailed descriptions, the abstract and introduction defines the work done in the article. The authors have tested these databases for their insertion and search operations execution time, the results are shown using graphs in the article. As always mongodb outperforms mysql database in both the tests conducted by a huge margin, with and without indexing. The authors claims are well supported by the results provided in the article and has evidence to reproduce and continue the experiments conducted. The conclusion details the results achieved by the tests and claims mongodb to be a better performer.

Conclusion: By reviewing these articles by various authors, we clearly understand and know that nosql databases are much better performer than sql databases. Mongodb database when compared with Mysql database, is always the better performer in all kinds of operations and test conducted by the authors. In my experiments i will also be evaluating the performance of Mongodb and Mysql database to further understand the experimented results, i will be using YCSB benchmarking tool to calculate the overall latency and average latency of read and update operations in both the databases and show the results. By conducting the experiments i will be able to better understand and explain the workings and better evaluate and analyse the performance of Mongodb and Mysql Database.

6 Performance Test Plan

I am using Ubuntu operating system running in openstack cloud platform, I have installed the following softwares/packages and tools in the ubuntu OS server:

- Java JDK

- Mysql 8.0
- Mongodb
- YCSB Tool
- python
- Testharness package

The performance of Mysql and Mongodb databases are evaluated using YCSB benchmark tool, I will be running the tests in two different workloads configurations of the YCSB tool those are, workload A update heavy (50% read and 50% update operations)and workload D read latest (95% read and 5% insert operations) and workload B read heavy (95% read and 5% update) and workload C read only (100% read) and workload E short ranges (95% scan and 5% insert). I will be running the test for the following numbers of operation counts:

- Workload A and Workload D
- 7500 opscounts
- 12500 opscounts
- 25000 opscounts
- 50000 opscounts
- 100000 opscounts

For Workloads B,C and Workload E i will be testing for the following operation counts, because operation counts above 25000 takes a lot of time(hours) to complete the tests.

- 5000 ospcounts
- 7500 opscounts
- 10000 opscounts
- 25000 opscounts

By running the test, the output will have number of benchmark scores and times taken for each operation. But i will be considering the average of these time taken for read and update operations and overall throughput of the two databases.

Overall Throughput ops/sec: Is the overall number of operations executed in a second, the higher the throughput means the better the database is performing.

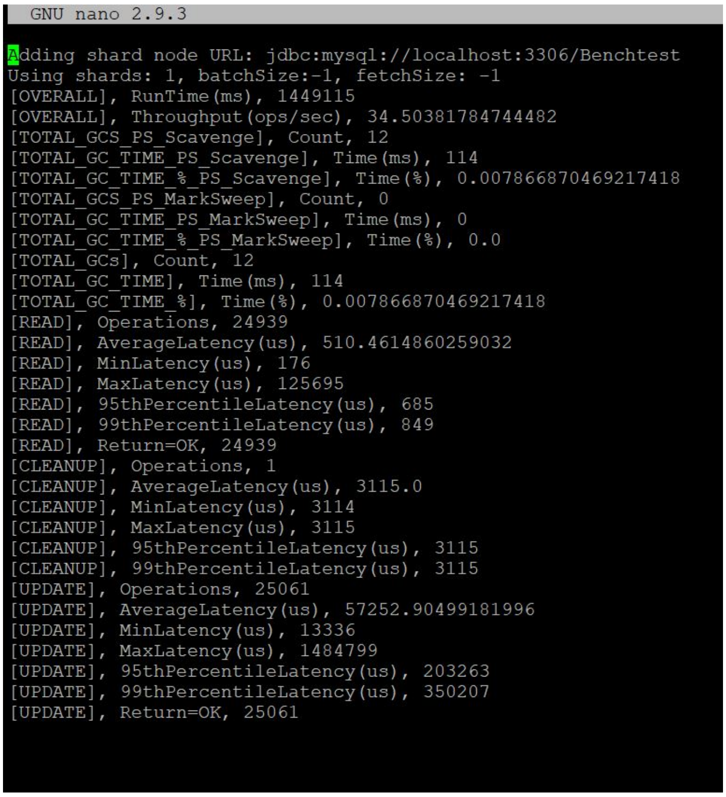
Average latency for read: This is the average latency of time in milliseconds to complete the read operation in the database, the lower the latency means the better performance.

Average latency for update: This is the average latency of time in milliseconds to complete the update operation in the database, the lower the latency means the better it is performing.

7 Evaluation and Results

7.1 Workload A

The first test is conducted for Workload A for different number of operation counts on mysql and mongodb databases.

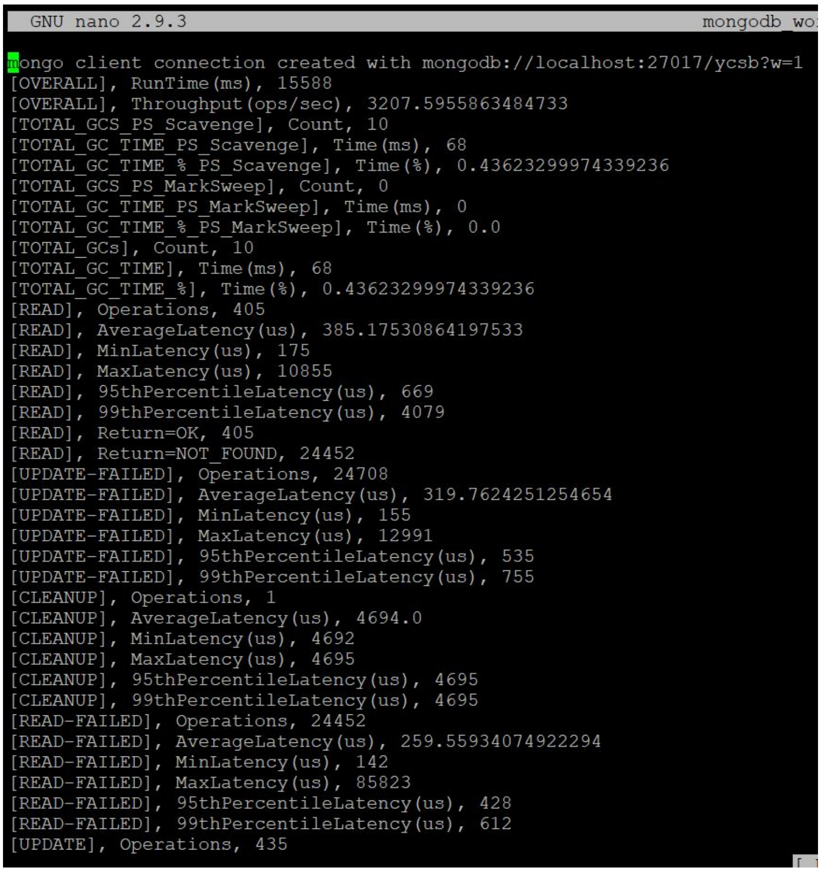


```
  GNU nano 2.9.3

Adding shard node URL: jdbc:mysql://localhost:3306/Benchtest
Using shards: 1, batchSize:-1, fetchSize: -1
[OVERALL], RunTime(ms), 1449115
[OVERALL], Throughput(ops/sec), 34.50381784744482
[TOTAL_GCS_PS_Scavenge], Count, 12
[TOTAL_GC_TIME_PS_Scavenge], Time(ms), 114
[TOTAL_GC_TIME_%_PS_Scavenge], Time(%), 0.007866870469217418
[TOTAL_GCS_PS_MarkSweep], Count, 0
[TOTAL_GC_TIME_PS_MarkSweep], Time(ms), 0
[TOTAL_GC_TIME_%_PS_MarkSweep], Time(%), 0.0
[TOTAL_GCs], Count, 12
[TOTAL_GC_TIME], Time(ms), 114
[TOTAL_GC_TIME_%], Time(%), 0.007866870469217418
[READ], Operations, 24939
[READ], AverageLatency(us), 510.4614860259032
[READ], MinLatency(us), 176
[READ], MaxLatency(us), 125695
[READ], 95thPercentileLatency(us), 685
[READ], 99thPercentileLatency(us), 849
[READ], Return=OK, 24939
[CLEANUP], Operations, 1
[CLEANUP], AverageLatency(us), 3115.0
[CLEANUP], MinLatency(us), 3114
[CLEANUP], MaxLatency(us), 3115
[CLEANUP], 95thPercentileLatency(us), 3115
[CLEANUP], 99thPercentileLatency(us), 3115
[UPDATE], Operations, 25061
[UPDATE], AverageLatency(us), 57252.90499181996
[UPDATE], MinLatency(us), 13336
[UPDATE], MaxLatency(us), 1484799
[UPDATE], 95thPercentileLatency(us), 203263
[UPDATE], 99thPercentileLatency(us), 350207
[UPDATE], Return=OK, 25061
```

Figure 6: Workload A Mysql.

For Mongodb for the same number of counts testes in mysql database.



```
  GNU nano 2.9.3                                              mongodb_wo

mongo client connection created with mongodb://localhost:27017/ycsb?w=1
[OVERALL], RunTime(ms), 15588
[OVERALL], Throughput(ops/sec), 3207.5955863484733
[TOTAL_GCS_PS_Scavenge], Count, 10
[TOTAL_GC_TIME_PS_Scavenge], Time(ms), 68
[TOTAL_GC_TIME_%_PS_Scavenge], Time(%), 0.43623299974339236
[TOTAL_GCS_PS_MarkSweep], Count, 0
[TOTAL_GC_TIME_PS_MarkSweep], Time(ms), 0
[TOTAL_GC_TIME_%_PS_MarkSweep], Time(%), 0.0
[TOTAL_GCs], Count, 10
[TOTAL_GC_TIME], Time(ms), 68
[TOTAL_GC_TIME_%], Time(%), 0.43623299974339236
[READ], Operations, 405
[READ], AverageLatency(us), 385.17530864197533
[READ], MinLatency(us), 175
[READ], MaxLatency(us), 10855
[READ], 95thPercentileLatency(us), 669
[READ], 99thPercentileLatency(us), 4079
[READ], Return=OK, 405
[READ], Return=NOT_FOUND, 24452
[UPDATE-FAILED], Operations, 24708
[UPDATE-FAILED], AverageLatency(us), 319.7624251254654
[UPDATE-FAILED], MinLatency(us), 155
[UPDATE-FAILED], MaxLatency(us), 12991
[UPDATE-FAILED], 95thPercentileLatency(us), 535
[UPDATE-FAILED], 99thPercentileLatency(us), 755
[CLEANUP], Operations, 1
[CLEANUP], AverageLatency(us), 4694.0
[CLEANUP], MinLatency(us), 4692
[CLEANUP], MaxLatency(us), 4695
[CLEANUP], 95thPercentileLatency(us), 4695
[CLEANUP], 99thPercentileLatency(us), 4695
[READ-FAILED], Operations, 24452
[READ-FAILED], AverageLatency(us), 259.55934074922294
[READ-FAILED], MinLatency(us), 142
[READ-FAILED], MaxLatency(us), 85823
[READ-FAILED], 95thPercentileLatency(us), 428
[READ-FAILED], 99thPercentileLatency(us), 612
[UPDATE], Operations, 435
```

Figure 7: Workload A Mongodb.

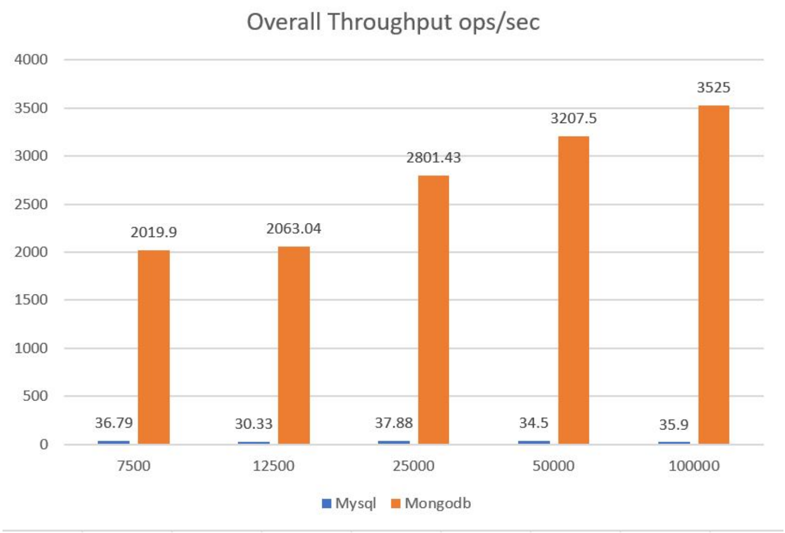
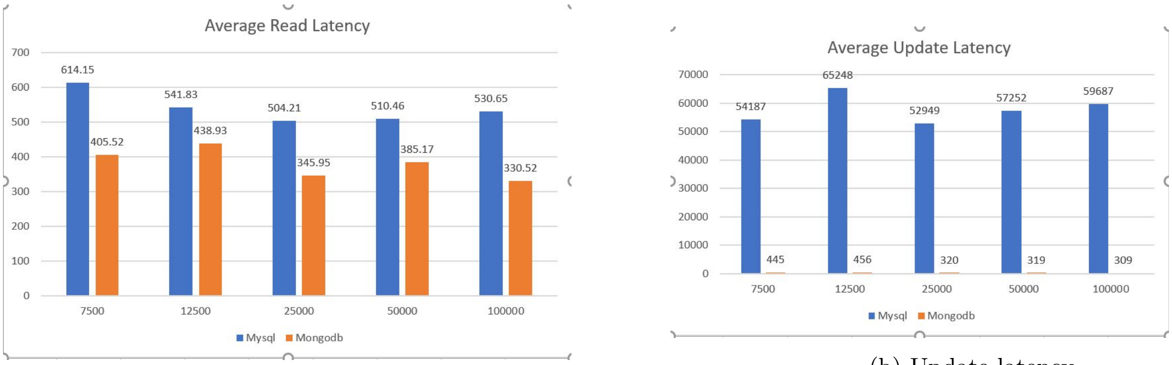


Figure 8: Workload A Overall Throughput ops/sec.

From the above graph we can see that the overall throughput of operation per second for workload A is increasing with increase in number of operation counts in Mongodb and in mysql it remains the same, it proves that mongodb executes much more number of operations per second than mysql. Hence mongodb database performs better and has better throughput than mysql.



(a) read latency

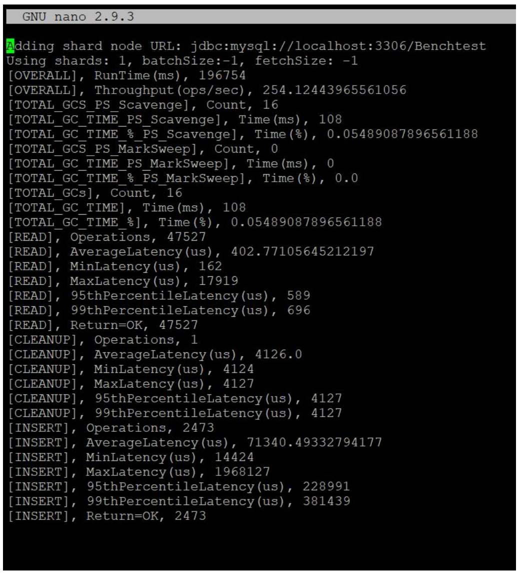
(b) Update latency

Figure 9: Workload A Average latency

From the graphs we can see that the average read and update latency decrease with increase in number of operation counts in mongodb and in mysql the read latency decreases but is still higher than mongodb. The update latency in mysql is also higher and keeps fluctuating with increasing operation counts. Mongodb has much lesser read and update latency and hence is the better performer here as well.

7.2 Workload D

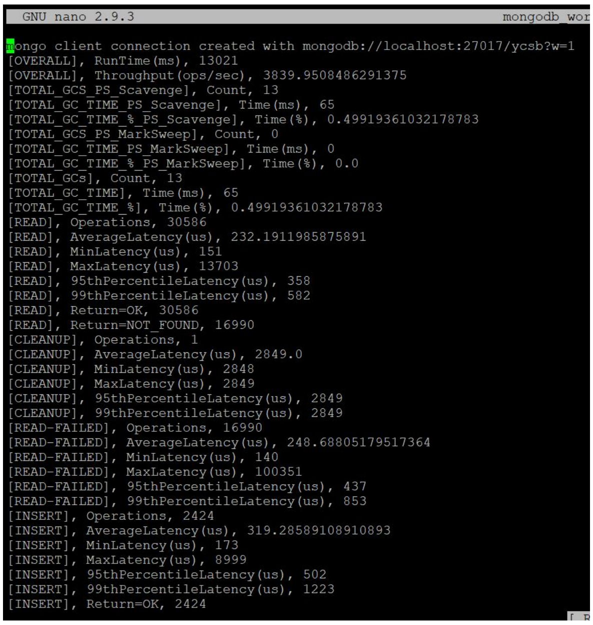
Test conducted for workload D on both mysql and mongodb databases.



```
  GNU nano 2.9.3

Adding shard node URL: jdbc:mysql://localhost:3306/Benchtest
Using shards: 1, batchSize:-1, fetchSize: -1
[OVERALL], RunTime(ms), 196754
[OVERALL], Throughput(ops/sec), 254.12443965561056
[TOTAL_GCS_PS_Scavenge], Count, 16
[TOTAL_GC_TIME_PS_Scavenge], Time(ms), 108
[TOTAL_GC_TIME_%_PS_Scavenge], Time(%), 0.05489087896561188
[TOTAL_GCS_PS_MarkSweep], Count, 0
[TOTAL_GC_TIME_PS_MarkSweep], Time(ms), 0
[TOTAL_GC_TIME_%_PS_MarkSweep], Time(%), 0.0
[TOTAL_GCs], Count, 16
[TOTAL_GC_TIME], Time(ms), 108
[TOTAL_GC_TIME_%], Time(%), 0.05489087896561188
[READ], Operations, 47527
[READ], AverageLatency(us), 402.77105645212197
[READ], MinLatency(us), 162
[READ], MaxLatency(us), 17919
[READ], 95thPercentileLatency(us), 589
[READ], 99thPercentileLatency(us), 696
[READ], Return=OK, 47527
[CLEANUP], Operations, 1
[CLEANUP], AverageLatency(us), 4126.0
[CLEANUP], MinLatency(us), 4124
[CLEANUP], MaxLatency(us), 4127
[CLEANUP], 95thPercentileLatency(us), 4127
[CLEANUP], 99thPercentileLatency(us), 4127
[INSERT], Operations, 2473
[INSERT], AverageLatency(us), 71340.49332794177
[INSERT], MinLatency(us), 14424
[INSERT], MaxLatency(us), 1968127
[INSERT], 95thPercentileLatency(us), 228991
[INSERT], 99thPercentileLatency(us), 381439
[INSERT], Return=OK, 2473
```

(a) Mysql



```
  GNU nano 2.9.3                                                    mongodb_wor

mongo client connection created with mongodb://localhost:27017/ycsb?w=1
[OVERALL], RunTime(ms), 13021
[OVERALL], Throughput(ops/sec), 3839.9508486291375
[TOTAL_GCS_PS_Scavenge], Count, 13
[TOTAL_GC_TIME_PS_Scavenge], Time(ms), 65
[TOTAL_GC_TIME_%_PS_Scavenge], Time(%), 0.49919361032178783
[TOTAL_GCS_PS_MarkSweep], Count, 0
[TOTAL_GC_TIME_PS_MarkSweep], Time(ms), 0
[TOTAL_GC_TIME_%_PS_MarkSweep], Time(%), 0.0
[TOTAL_GCs], Count, 13
[TOTAL_GC_TIME], Time(ms), 65
[TOTAL_GC_TIME_%], Time(%), 0.49919361032178783
[READ], Operations, 30586
[READ], AverageLatency(us), 232.1911985875891
[READ], MinLatency(us), 151
[READ], MaxLatency(us), 13703
[READ], 95thPercentileLatency(us), 358
[READ], 99thPercentileLatency(us), 582
[READ], Return=OK, 30586
[READ], Return=NOT_FOUND, 16990
[CLEANUP], Operations, 1
[CLEANUP], AverageLatency(us), 2849.0
[CLEANUP], MinLatency(us), 2848
[CLEANUP], MaxLatency(us), 2849
[CLEANUP], 95thPercentileLatency(us), 2849
[CLEANUP], 99thPercentileLatency(us), 2849
[READ-FAILED], Operations, 16990
[READ-FAILED], AverageLatency(us), 248.68805179517364
[READ-FAILED], MinLatency(us), 140
[READ-FAILED], MaxLatency(us), 100351
[READ-FAILED], 95thPercentileLatency(us), 437
[READ-FAILED], 99thPercentileLatency(us), 853
[INSERT], Operations, 2424
[INSERT], AverageLatency(us), 319.28589108910893
[INSERT], MinLatency(us), 173
[INSERT], MaxLatency(us), 8999
[INSERT], 95thPercentileLatency(us), 502
[INSERT], 99thPercentileLatency(us), 1223
[INSERT], Return=OK, 2424
```

(b) Mongodb

Figure 10: Workload D Screenshots

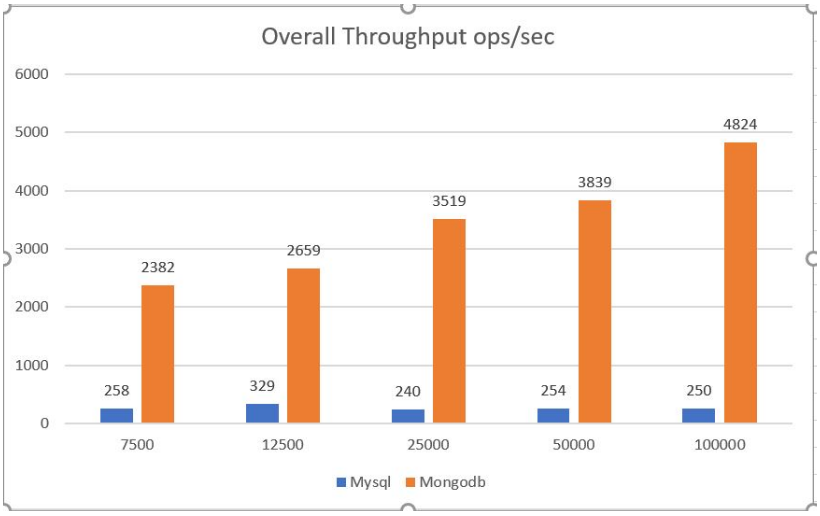
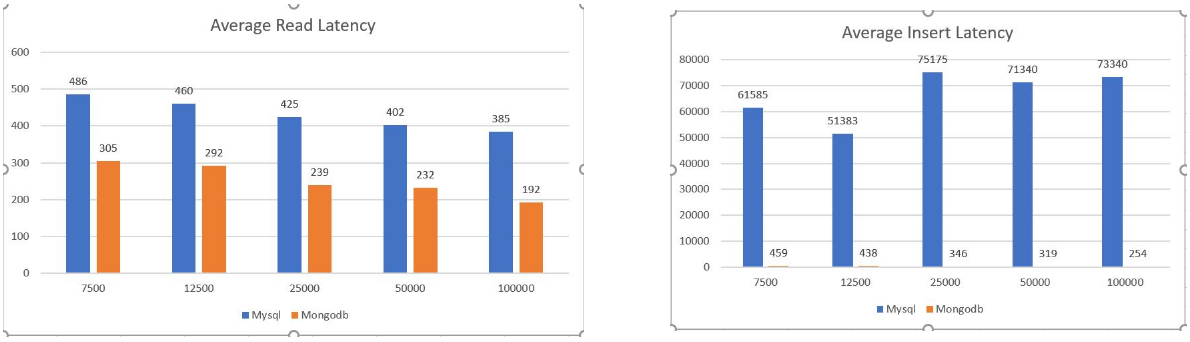


Figure 11: Workload D Overall Throughput ops/sec.

From the above graph we can see that the overall throughput of operation per second for workload D is increasing with increase in number of operation counts in Mongodb and in mysql it decreases, it proves that mongodb executes much more number of operations per second than mysql. Hence mongodb database performs better and has better throughput than mysql.



(a) read latency

(b) Insert latency

Figure 12: Workload D Average latency

From the graphs we can see that the average read latency decrease with increase in number of operation counts in mongodb and in mysql the read latency decreases but is still higher than mongodb. The insert latency in mysql is also higher and keeps fluctuating with increasing operation counts and for mongodb also it is not constant. Mongodb has much lesser read and update latency and hence is the better performer here as well.

7.3 Workload B

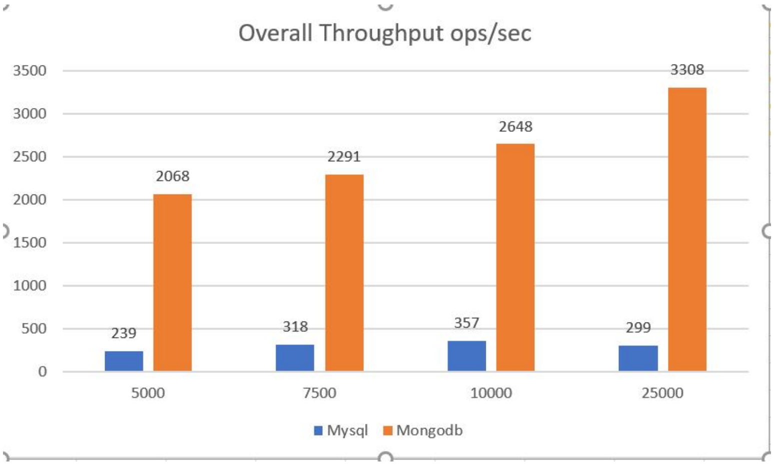
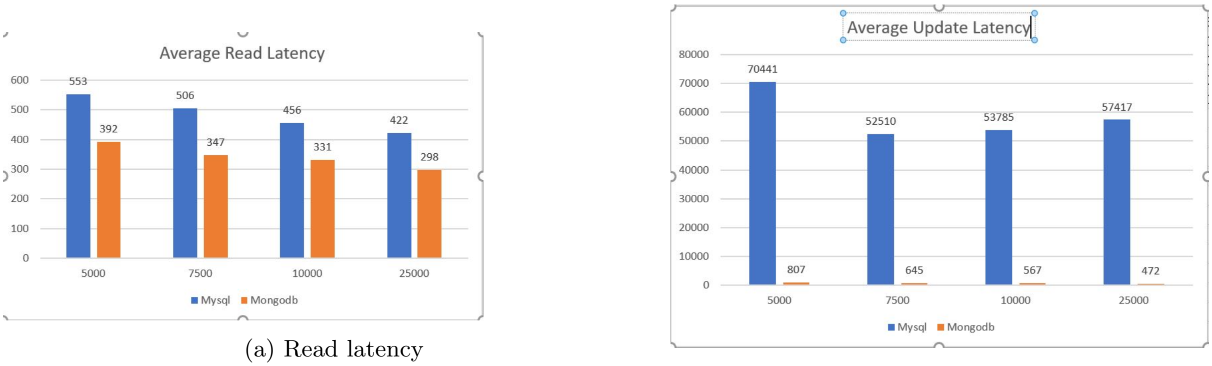


Figure 13: Workload B Overall Throughput ops/sec.

From the above graph we can see that the overall throughput of operation per second for workload B is increasing with increase in number of operation counts in Mongodb and in mysql it remains constant, it proves that mongodb executes much more number of operations per second than mysql. Hence mongodb database performs better and has better throughput than mysql.



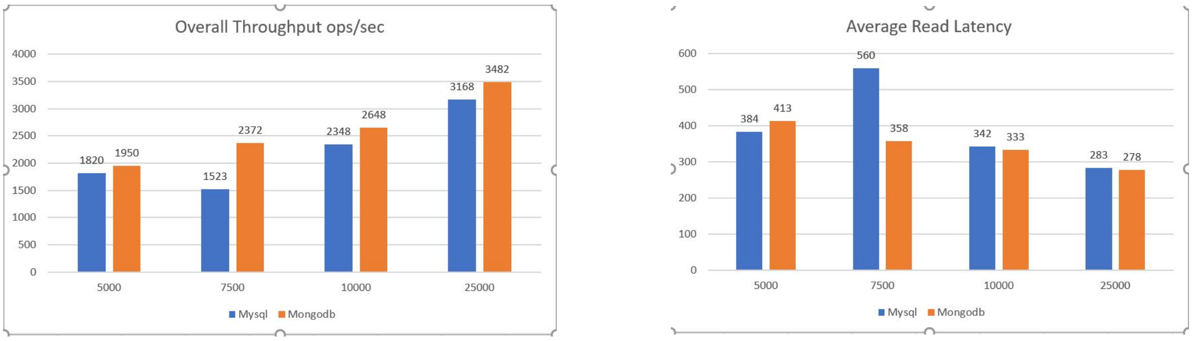
(a) Read latency

(b) Update latency

Figure 14: Workload B Average latency

From the graphs we can see that the average read latency decrease with increase in number of operation counts in mongodb and in mysql also the read latency decreases but is still higher than mongodb. The update latency in mysql is also higher and keeps fluctuating with increasing operation counts and for mongodb it decreases. Mongodb has much lesser read and insert latency and hence is the better performer here as well.

7.4 Workload C



(a) Workload B Overall Throughput ops/sec

(b) Read latency

Figure 15: Workload C Average latency

From the above graph we can see that the overall throughput of operation per second for workload C is increasing with increase in number of operation counts in Mongodb and in mysql also it increases, but for mongodb scores are higher, it proves that mongodb executes much more number of operations per second than mysql. Hence mongodb database performs better and has better throughput than mysql.

From the graphs we can see that the average read latency decrease with increase in number of operation counts in mongodb and in mysql also the read latency decreases but is still higher than mongodb, though both the database scores are closer and similar. Mongodb has a little lesser read latency and hence is the better performer here as well.

7.5 Workload E

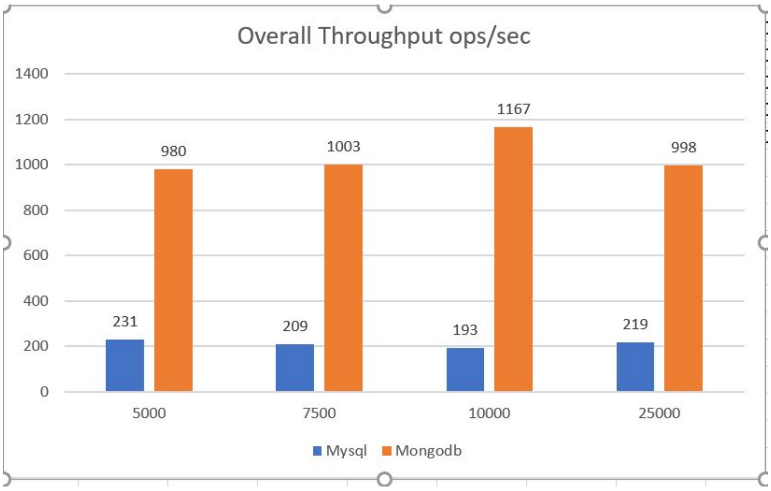
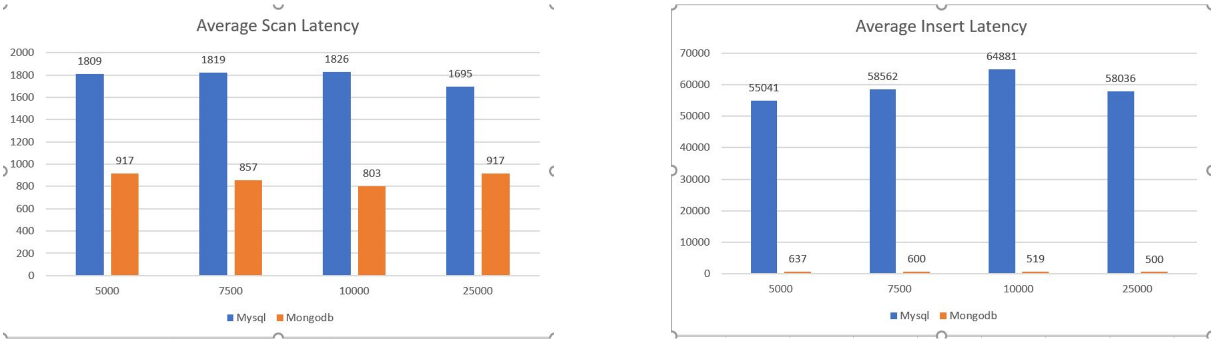


Figure 16: Workload E Overall Throughput ops/sec.

From the above graph we can see that the overall throughput of operation per second for workload D remains constant with increase in number of operation counts in Mongodb and in mysql , but the scores of mongodb is higher, it proves that mongodb executes much more number of operations per second than mysql. Hence mongodb database performs better and has better throughput than mysql.



(a) Workload B Overall Throughput ops/sec

(b) Read latency

Figure 17: Workload D Average latency

From the graphs we can see that the average scan and insert latency for both the databases remains constant with little difference with increase in operation counts. However, mongodb scores are better than mysql and has less latency, hence the better performer.

8 Conclusion and Discussion

Sql anf Nosql databases have difference in the way they store and retrieve data, Mysql a sql database and Mongodb a Nosql database were selected to evaluate and analyse the performance of these databases. YCSB benchmark tool was used to run test and obtain scores for the databases performance, three aspects were considered, overall throughput ops/sec, average read latency and average update latency. The tests were run on five different workloads of the YCSB tool. In all the workloads mongodb performed better than mysql and in some cases it performed exceptionally well. Mongodb has better overall throughput, it executes more number of operations per second than mysql in all the workload A, B, C, D, E and the execution time was also minimum. Whereas fro mysql the execution time was much more and the time consuming.

No doubt Nosql database performs better for large datasets and can perform more operation in less time and handle larger data faster than sql databases. However, mysql does performs the task given and executes them, but is time consuming for larger datasets and more operations counts. In this paper, the tests and experiments conducted proved that Mongodb database is better and faster performing database than mysql. From the comparisons made in this paper, Mongodb stands out and emerged the better performer.

9 References

1. "MySQL," Wikipedia, 20-Apr-2020. [Online]. Available: https://en.wikipedia.org/wiki/MySQL. [Accessed: 23-Apr-2020].

2. "MongoDB," Wikipedia, 23-Apr-2020. [Online]. Available: https://en.wikipedia.org/wiki/MongoDB. [Accessed: 23-Apr-2020].

3. "YCSB," Wikipedia, 21-Feb-2020. [Online]. Available: https://en.wikipedia.org/wiki/YCSB. [Accessed: 23-Apr-2020].

4. Ronstrom, M. and Thalmann, L., 2004. MySQL cluster architecture overview. MySQL Technical White Paper, 8.

5. R., Chin, A., Kassam, F., Roszko, A. and Holt, R., 2002. Mysql conceptual architecture. Technical report, University of Water-100.

6. "Architecture Guide," MongoDB. [Online]. Available: https://www.mongodb.com/collateral /mongodb-architecture-guide. [Accessed: 23-Apr-2020].

7. Cloud, "High-availability MongoDB Cluster Configuration Solutions," Medium, 02-Mar-2018. [Online]. Available:https://medium.com/@Alibaba_Cloud/high-availability-mongodb-cluster-configuration-solutions-465cc82cd0bc. [Accessed: 23-Apr-2020].

8. "MySQL Enterprise High Availability," MySQL. [Online]. Available: https://www.mysql.com /products/enterprise/high_availability.html. [Accessed: 23-Apr-2020].

9. "Security¶," Security - MongoDB Manual. [Online]. Available: https://docs.mongodb.com/manual/security/. [Accessed: 23-Apr-2020].

10. "Chapter 6 Security Components and Plugins," MySQL. [Online]. Available:https://dev.mysql.com/ doc/mysql-security-excerpt/8.0/en/security-plugins.html. [Accessed: 23-Apr-2020].

11. Aghi, R., Mehta, S., Chauhan, R., Chaudhary, S. and Bohra, N., 2015. A comprehensive comparison of SQL and MongoDB databases. International Journal of Scientific and Research Publications, 5(2), pp.1-3.

12. B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with YCSB," Proceedings of the 1st ACM symposium on Cloud computing - SoCC 10, 2010.

13. Patil, M.M., Hanni, A., Tejeshwar, C.H. and Patil, P., 2017, February. A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing—Sharding in MongoDB and its advantages. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 325-330). IEEE.

14. Jogi, V.D. and Sinha, A., 2016, March. Performance evaluation of MySQL, Cassandra and HBase for heavy write operation. In 2016 3rd International Conference on Recent Advances in Information Technology (RAIT) (pp. 586-590). IEEE.

15. Dipina Damodaran, B., Salim, S. and Vargese, S.M., 2016. Performance evaluation of MySQL and MongoDB databases. International Journal on Cybernetics and Informatics (IJCI) Vol, 5.

16. Ahmed, M., Uddin, M.M., Azad, M.S. and Haseeb, S., 2010, April. MySQL performance analysis on a limited resource server: Fedora vs. Ubuntu Linux. In Proceedings of the 2010 Spring Simulation Multiconference (pp. 1-7).

17. Nyati, S.S., Pawar, S. and Ingle, R., 2013, August. Performance evaluation of unstructured NoSQL data over distributed framework. In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1623-1627). IEEE.