DATA STORAGE MANAGEMENT

MongoDB vs HBase

Project Guide: Michael Bradford

X18110088 (M.Sc. DATA ANALYTICS)

Project Requirements

Requirement A. Install and implement instances of your chosen data storage and management systems. You must install your instances on the NCI OpenStack Cloud Platform. (See Creating OpenStack Ubuntu VM for Project B document for instructions on how to set-up a virtual machine on OpenStack.)

Requirement B. Devise and implement a test strategy in order to perform a comparative analysis of the capabilities of each system in terms of the performance. You should record a set of appropriate metrics and perform a quantitative analysis for comparison purposes between the two chosen systems.

Requirement C. Additionally, for comparison purposes, research and present your findings of the capabilities of your chosen systems in one of the following four areas:

(i) Storage and retrieval of Structured, Semi-Structured and Unstructured Data
(ii) Scalability, Availability and Reliability
(iii) Transaction Management
(iv) Security
(v) Requirement D.

You must complete a technical report document including the following sections

(i) Abstract
(ii) Introduction
(iii) Key Characteristics of Chosen Data Storage Management Systems
(iv) Database Architectures [5 marks]
(v) Chosen Area from Requirement C section
(vi) Learning from Literature Survey
(vii) Performance Test Plan
(viii) Evaluation and Results

Contents

1. ABSTRACT:

NO-SQL consist of large number of the database that support recently developed applications. large amount of the data is created now a day's which is structured, unstructured, and semi-structure and it is not possible to handle such data through legacy relational database system. As these system are not developed by considering the data generated by recent applications also there is limited scope for the modification of the relational system. (mongoDB, nd) That is why the NoSQL database are emerge to handle such high volume of the unstructured data. There are different database under the NO-SQL such as mongoDB, Hbase. The purpose of this project is to analyze the performance of the these NO-SQL database through the benchmarking system YCSB. Hbase and Mongodb, hadoop are install over the open stack cloud platform. YCSB workloads are used to analyze the system latency and throughput for read and write operation. Test harness is the tool which is used for the automating the process of testing for two different databases. Finally latency and throughput results are compare to identify the system with better performance. Mongodb is much better than Hbase on every aspect of the YCSB test for the different workload

2. INTRODUCTION:

There are different reasons for building database such as size, accuracy, security, redundancy, importance, overwriting of the data. Its structured system which allows the data to in and then apply some rules on the data and these rules change according to the problems. (Gabry) These problems based on the scenario such as some time DATA size is Important and in other situation for the small data security is important. Data base management system is actually the software which interacts with end user, application to capture and analyse the data. DBMS manages the database and imposes rules on the database. There are different DBMS such as relational, Hierarchical, No-SQL, Object-oriented databases system. NO-SQL databases divided into four category first is Document Database example is MongoDB, Key-value database, Wide column database example is HBase and fourth is Graph or node base database Neo4j.Purpose of this project is to analyse and understand the capabilities of No-SQL database such as MongoDB and HBase in terms of performance with the help of Yahoo! Cloud Serving Benchmark (YCSB) . (Gabry)

REQUIRNMENT 1 : INSTALLATION OF MONGODB AND HBASE

```
hduser@x18110088-dsmprojb:/usr/local$ ls
bin                     lib
etc                     man
games                   mongodb
hadoop                  mongodb-linux-x86_64-ubuntu1604-3.2.10
hadoop-2.9.2            mongodb-linux-x86_64-ubuntu1604-3.2.10.tgz
hadoop-2.9.2.tar.gz     sbin
hbase                   share
hbase-1.4.8             src
hbase-1.4.8-bin.tar.gz  testharness
include                 testharness.tgz
```

Hbase is running on HDFS

```
hduser@x18110088-dsmprojb:/home/ubuntu/ycsb-0.11.0/output$ jps
22224 Jps
13250 NodeManager
10691 HRegionServer
13669 HQuorumPeer
12773 SecondaryNameNode
12504 DataNode
12936 ResourceManager
12313 NameNode
13740 HMaster
hduser@x18110088-dsmprojb:/home/ubuntu/ycsb-0.11.0/output$
```

Mongodb process

```
hduser@x18110088-dsmprojb:/home/ubuntu/ycsb-0.11.0/output$ ps
  PID TTY          TIME CMD
 1703 tty1     00:00:06 bash
10677 tty1     00:00:00 bash
10691 tty1     00:21:06 java
12936 tty1     00:14:49 java
13726 tty1     00:00:00 bash
13740 tty1     00:11:14 java
14208 tty1     00:15:57 mongod
22264 tty1     00:00:00 ps
```

FINAL RESULT DIRECTORY

```
hduser@x18110088-dsmprojb:/home/ubuntu/ycsb-0.11.0/output$ ls -ltr
total 4
drwxrwxrwx 2 hduser hadoop 4096 Dec 19 14:28 final_result_20181218
```

Instance Name: x18110088-DSMPROJB

Username:ubuntu

Password : dsm

3. CHARECTERSTICS OF DATA STORAGE AND MANAMENT SYSTEM

3.1 hbase

1. Linear and modular scalability. (HBASE)

   It is possible to increase Storage capacity and processing by increasing the region server to the HBase Cluster.

2. Strictly consistent reads and writes. (HBASE)
3. Automatic and configurable sharding of tables (HBASE)
4. Regions are the part of the HBase Clustor which is use to stores the data and theses regions are automatically split and re-distributed as data is grow.
5. Automatic failover support between Region Servers.
6. HBase is on the top of HDFS and HDFS is internally distributed with help of multiple block allocation and replication, Also in the HBase automatic failover support is provided using RegionServer replication.
7. To process large volume of the data parallel processing is used with Mapreduce.
8. HBase support Java APIs so it is easy to use to the client.
9. HBase supports Block cache and Bloom Filters for processing the real-time queries.
10. HBase support both structured and semi-structured data.
11. concept of schema is not applicable in HBase as it defines only column family. (HBASE)

3.2 mongodb

1. MongoDB uses BSON format :

   MongoDB uses JSON as well as BSON format mongodb use this while storating the Document in the collection also it add support for the data type like date and Binary which is not supported by JSON. Because of the BSON mongoDB can internally index and map document properties. (jar)

1. MongoDB is Schema – Less

   MongoDb is is schema less so to save the object just have to serialise to JSON and send it to MongoDB.

2. MongoDB Indexing (jar)

   In the mongoDB each filed is having primary as well as secondary index and this is require to improve the Performance of the search for the query statement. (jar)

3. MongoDB Management Service (MMS)

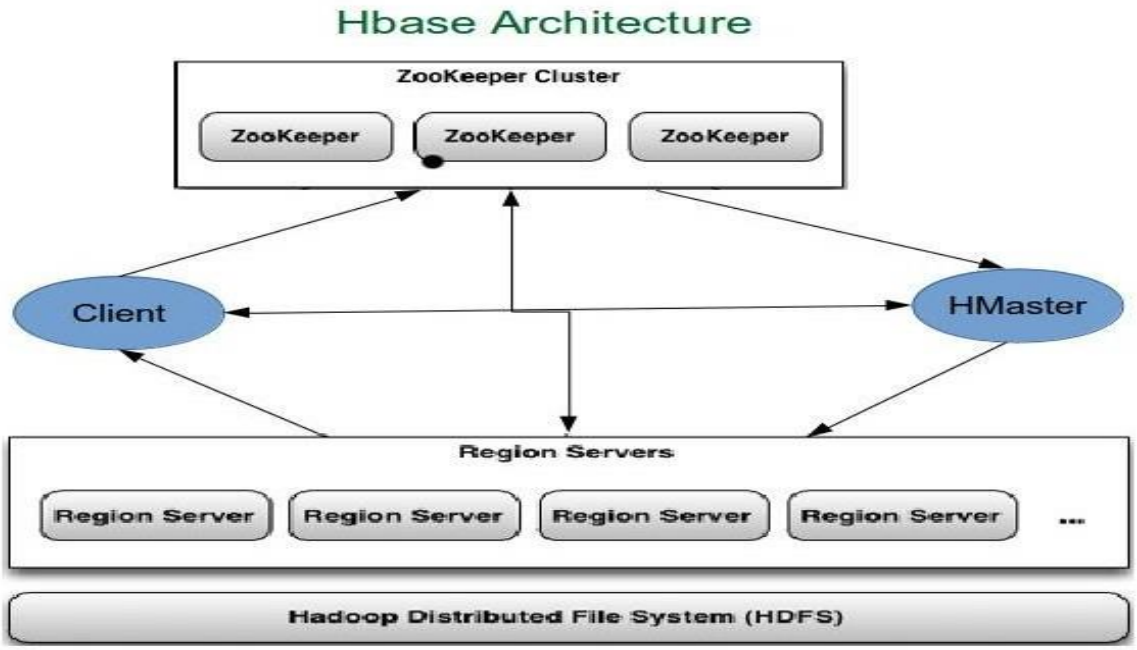
It is possible to track the activities against the database and machine and also support back up option for the data. Tool shows performance on the consol to help to optimise the deployment. Issues related to mongoDB instance can be track and resolved before affecting the instance. (jar)

4. Support Multiple Storage Engine
   Storage engine is used for storing the data into the Memory and MongoDB supports multiple storage Engine WiredTiger Storage Engine is used as a default engine. n-Memory Storage Engine and MMAPv1 Storage Engine. (jar)

4.DATABASE ARCHITECTURE

4.1 HBase :

Hbase implementation based on Google's Big table and it is build on the top of HDFS and Hadoop



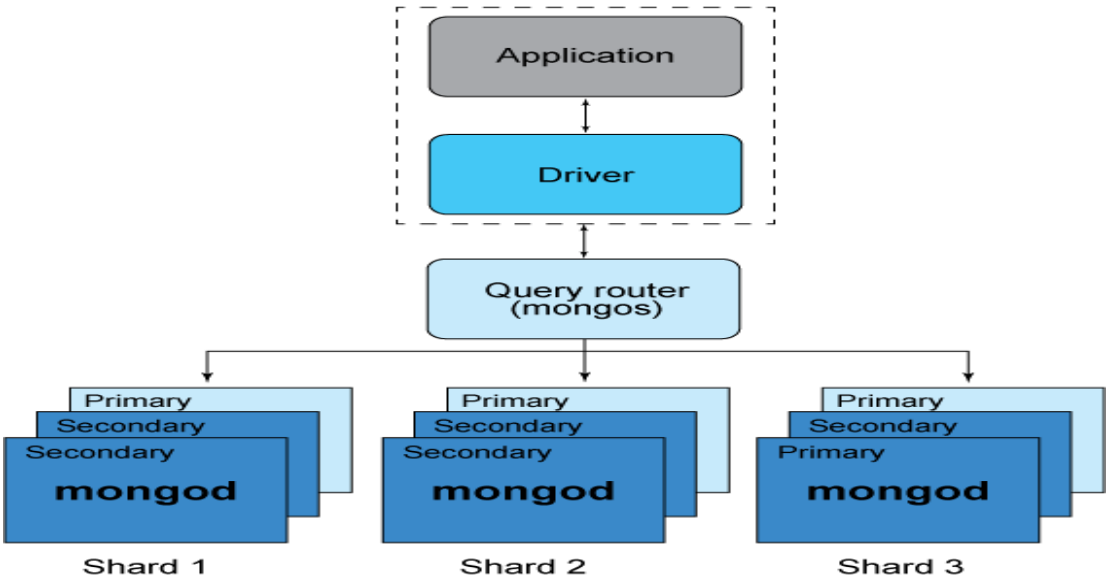
(HBase Overview)

1. Hbase follows Master Slave Architecture similar to that of HDFS and in this architecture HMaster is master node and region servers are slave node.Hbase is on the top of HDFS and provide fast update lookups and update for the large table.
2. Hbase conf/rigionserver file contains the information about the slave node. Hbase is highly dependent on zookeeper and manages the instance. By default Hbase is configure to use the zookeeper instance.ie connectivity maintain between user and database by zookeeper.

3. Hadoop file system APIs are used to get the data. Htable are important term as it contains the information about the regions and communicate with the region server to read and to write the key-value pairs. Information about the list, state and location of the server in .META table So .META table holds the list of all user specific regions. (HBase Overview)
4. For the few records Hbase is not so useful. Hbase handy for processing billions and trillions of the data. (HBase Overview)

4.2 Mongodb :

1. Mongodb is the No-SQL Database as it is uses Document Data Model. Name-value pair is used, Json is used to model the document in Hierarchical fashion
2. value can be anything like single value ,group ,array or the Documents. model provide fast queries Because of the data stored next to each other in the document rather than stored in the table and require join to retrieve the query result



(IBM-mongodb)

3. sharded environment is shown in the figure, Mongodb provides Auto sharing for the horizontal Scalability on the low cost commodity hardware or the cloud infrastructure.
4. if the security is turn off then client needs to turn on the security and then perform the operations like insert, update.
5. Because of this kind of architecture it is possible to address the issue like bottlenecks in RAM or disk I/O, Multiple copies of the Data are maintain by MongoDB same as that of Hbase called as the replica.
6. Manual interference is not required in the replica failover as the process is fully automated.
7. Mongodb is Consistent and one member act as the Primary and other act as secondary replica Member. Secondary Member is automatically selected if the Primary Member is failed. (IBM-mongodb)

5. SCALABILITY, AVAILABILITY AND RELIABILITY :

Scalability :The term scalability means the capability of the system to handle increasing amount of the data when processing power increases. System is Scalable only and only when it will increase workload and throughput with addition of new resources. Scalable system should be accusable while changing the workload and thereby maintaining continuous availability during the changes are made. There are Two type of the database Scalability 1.Vertical Scalability 2.Horizonal Scalability (Craig S. Mullins, 2018)

1. Vertical Scalability : Different operations are performed under the scalablity such as adding the new memory or the more powerful CPUs.This type of modification is called as scale Up and by removing the memory or replacing the CPU causes the degradation and called as Scale down. This hole process required to configure the system again and also there are limitation of the modifications such as memory can't exceed after the particular slot and need to move to the bigger box.this process require the application downtime and highly costly. (Craig S. Mullins, 2018)

2.Horizonal Scalability: There are two Different things such as scaling out and scaling in. adding the servers to the existing system is called as the scaling out and removing is scaling in. Scaling is limited with the legacy database servers. (Craig S. Mullins, 2018)

Hbase

Scalability: In the Hbase data storage capacity can be increase or scale out by adding the extra node. Ideally this should be done by Database with the no downtime and minimum overhead. Minimum data movement is expected when the system is scale up. Hbase supports the hashing which is design to minimise the data movement during this process. Data rebalancing is automatic in this case. (HBase Scalability Features) A node can be add to the existing system with the command **$ ./bin/hbase-daemon.sh start regionserver** and ssh can start the daemons after the adding new hostname *conf/regionservers*. Now regin server dose not contain the data regions and regions can be move to the region server if the balancer is enable and this will cause to shutdown the region also this will affect the latency in adverse manner for the small/medium cluster. (node.management)

Availability: Hbase uses different technique to keep database alive such as partition of the data automatically and distributes the data on the different nodes so failure of the node causes loss of the data but other nodes are still working and in available status. Also HBase is implement on the top of the HDFS, this allows to store the data in 3 different copies on different machine and available to every node of the cluster. This way Hbase redistribute the data from the failed node to the node that are sill running. This will allow 99.9% availability for the Hbase. (Apache Availability)

Reliability: When the Hbase is configured properly with region server, backup Master Server and proper zookeeper then it is highly reliable. We can consider Hbase as fault tolerance But in some case such as failure of the all the master servers at once and also cluster failure during the cascading. so we can say that HBase is not fault tolerance. but

consider the situation where we are retrieving data through query and region server failure occur and the server was managing the key where we were querying and the good thing is that data will be recover without manual intervention if the system is configured properly.So is Hbase is Tolerant of network failure (HBase Architecture)

MongoDB

Scalability: through sharding MongoDB perform the scaling operation and also distributes data among the machines to get the high Throughput operation for the huge amount of the dataset.the biggest advantage of the Sharding is that, it allows Administrator to increase capacity on the fly. In mongodb scaling is divided into three different part such as Cluster Scale, Performance Scale and Data scale. (MongoDB at Scale)

Availability: Replica set is maintain by MongDB, data enter, rack, servers are use to store the multiple copies of the data. This is useful to avoid the downtime and all this process is automatic. Primary member and secondary member are maintain by replica set and at time time of primary member failure secondary member elected as the primary member and this way MongoDB maintain the availability. (MongoDB avilablity)

Reliability: Mongodb is highly reliable system which uses Replica set to maintain the copies of the data and give hardware failover capabilities. Recovery interval is possible by taking backup on regular interval. And also sharding is available so we can say that Mongodb is highly reliable system problem is not with the Reliability it may also associated with the process and recovery plan. (mongodb reliablity)

7. PERFORMANCE TEST STRATEGY

1. Data base management system use for the test
    1. HBase 1.4.8
    2. Mongo for linux 3.2.10
2. Benchmarking system use for the test
    1. YCSB.11
3. Hadoop version used for the project
    1. Hadoop -2.9.2

4. Specification of the machine
    1. Intel(R) Core(TM) I3-3120M CPU @2.50GHz
    2. Installed RAM : 10GB
    3. System type: 64bin operating system
    4. Open stack instance -Name : x18110088-DSMPROJB
    5. Memory Specification-40Gb Hard disk and 4gbRAM.

    6. Hadoop was installed on the instance as Hbase is on the top of the hadoop Distributed file system.
5. **Test harness**

1. Test harness is tool is used which consist of two main part test execution engine and test script repository objective of the test is to automation of the testing process ,execution of the test suit for the test cases finally generation of the Associated test report. Runtest.sh script contains the functions and these functions are use to featch the information for the different file to get the information such as databases which in this case is mongodb and Hbase and the count for which the performance testing is to be performed. Truncate the table before performing the operation. (Test harness, nd)

6. Workload chosen :
    1. YCSB is used for the Benchmarking .ycsb consist of total 6 default Workload
    2. These workload are important to obtain benchmark number for the different system and this will allow to understand system performance.
    3. Two type of the workload has been chosen to perform the database test. (Core YCSB properties, nd)

    4. **Workload A** :
       Update Heavy Workload.
       This workload contain 50/50 read and write operation
       Read /Update ratio 50/50
       Read proportion = 0.5
       Update proportion = 0.5

    5. **Workload D**: Read latest workload
       In this workload only new records are inserted and recently inserted record is most popular

    6. Record Count : Total numer of the recorde loaded to perform the test.

    7. Read Proportion : What proportion of the operation should be read.

    8. Update Proportion :What proportion of the operation should be update. (Core YCSB properties, nd)

8. EVALUATIONS AND RESULTS

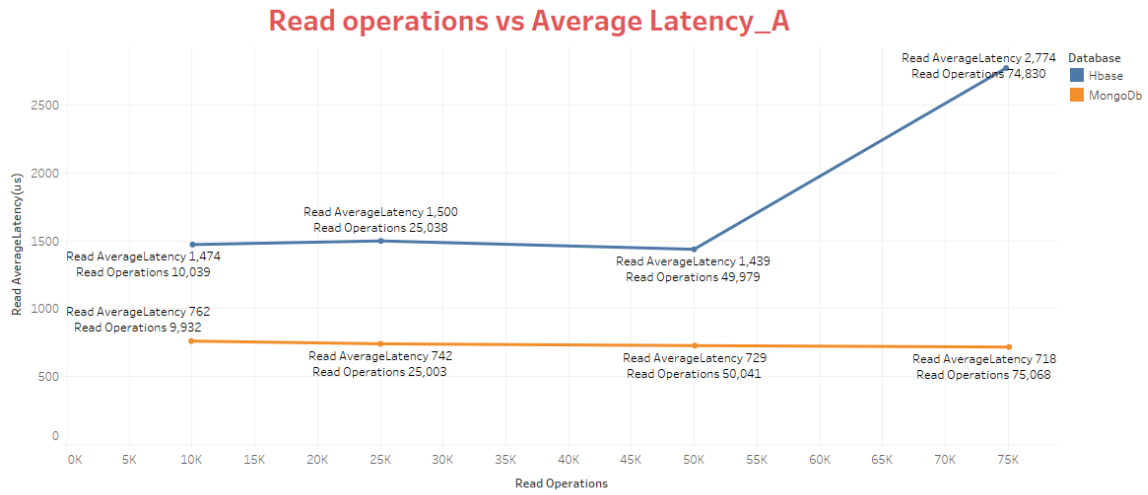
Below performance test are evaluate with help of graphical representation over Tableau

WORKLOAD A

Read operations vs Average Latency

The graph is the line graph which shows average latency in us Vs read Operations .As we can see from Graph average latency of the mongodb is less as compare to the that of the HBase and it is constant as read count increases for the different operation count, where as latency of the hbase is constant till read count is nearly 49,979 and increase rapidly with speed of 2X

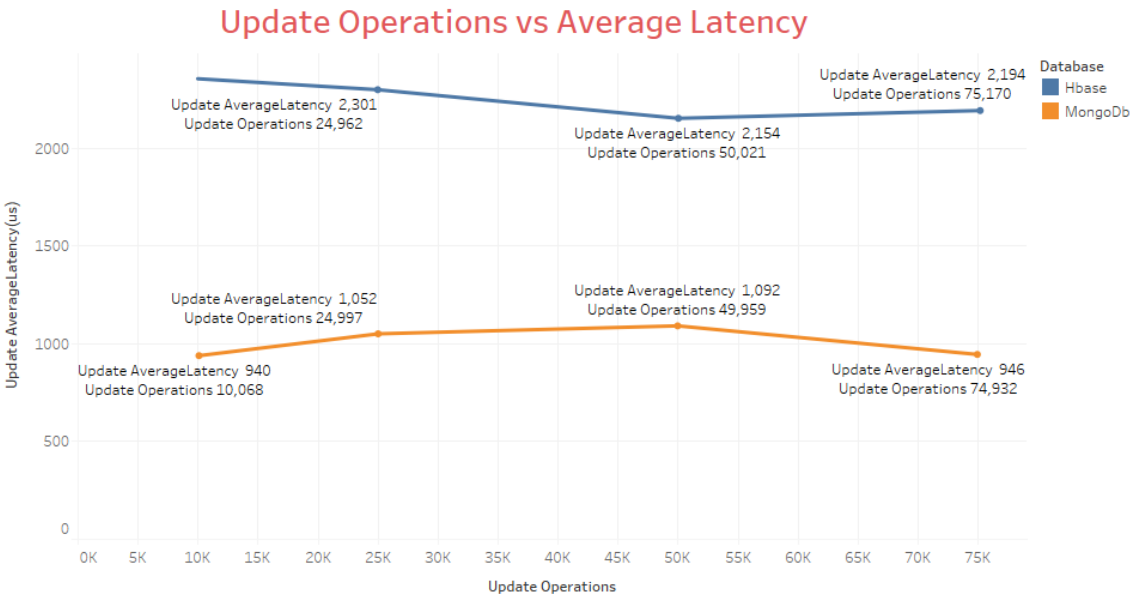
as the read operations increase to 74,830.This analysis shows that for the read operations performance of mongodb is Higher than Hbase as read operations increases.



Read Operations vs. Read AverageLatency(us). Color shows details about Database. The marks are labeled by Read AverageLatency(us) and Read Operations.

Update Operations vs Average Latency

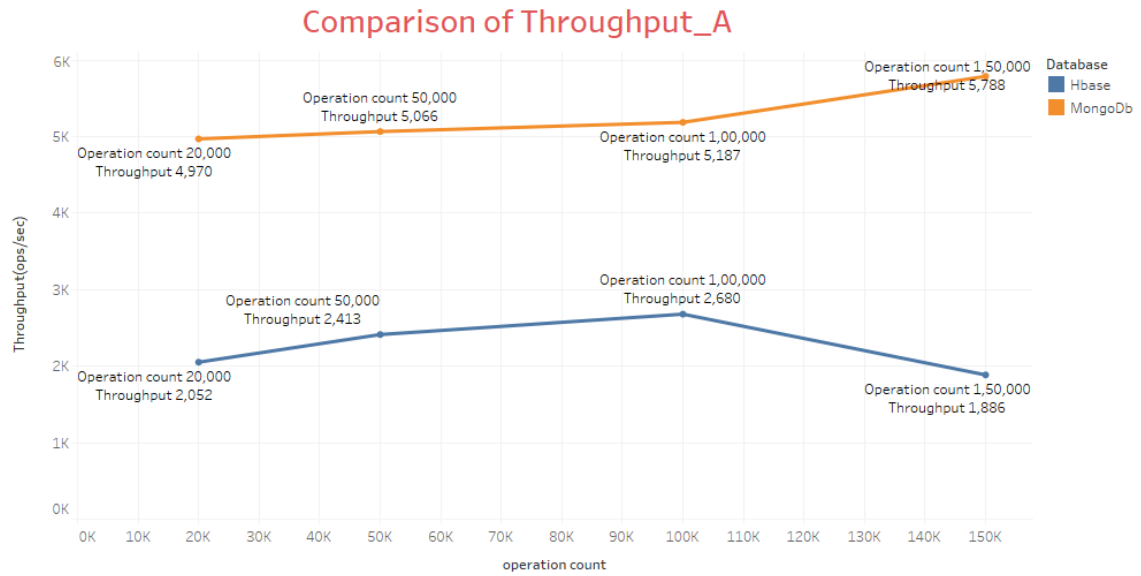
This is used to analyze the performance of the system when update operation is performed. Avg update latency is on Y-axis in us and update operation on x axis.From the graph avg update latency of the mongodb is better than that of hbase for the same number of update records. Update latency of the mongodb is increases with increase in the update operations for the particular period and again reduces to its original state. Inversely with Hbase for the low update operation count latency is maximum and reduces as update operation increases.



Update Operations vs. Update AverageLatency(us). Color shows details about Database. The marks are labeled by Update AverageLatency(us) and Update Operations.

Comparison of Throughput

Again for the workload A throughput of the MongoDB is better than throughput of the Hbase system. As we can see from the graph for the mongodb through put is low for the low operation count and slowly increases as operation count increase till 100000 operations than decreases suddenly as operation count increased to 150000 so we can say that throughput is decrease with increase in the workload. Similarly for the Hbase Throughput is low for the lower number of operations and increase as the operation increase so both Hbase and Mongodb used when the operation count is high. Among these two Mongodb is having better throughput.

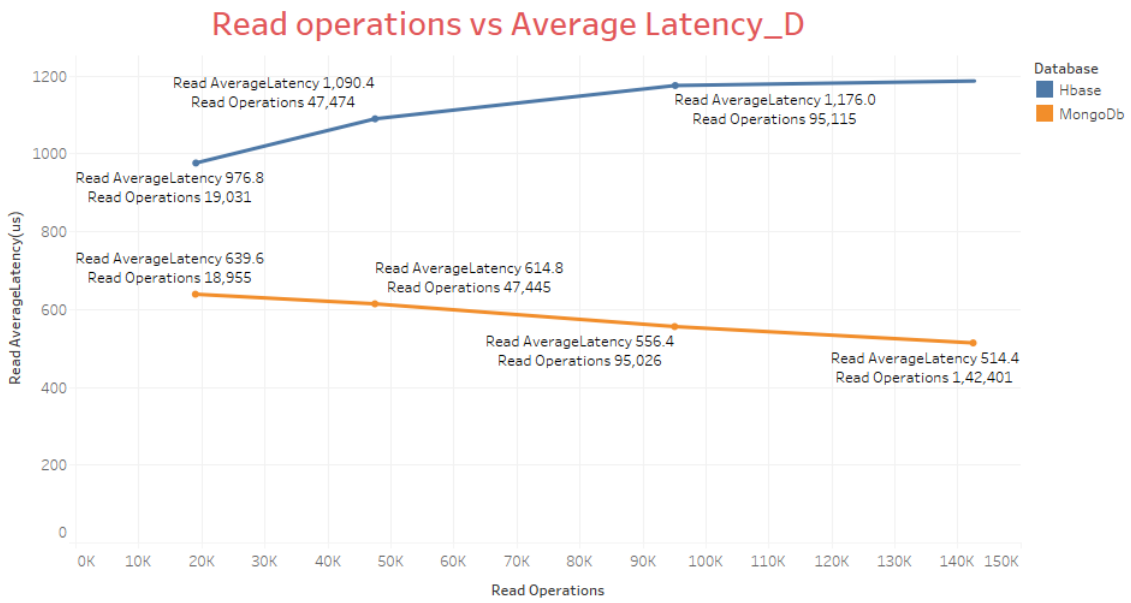


Operation count vs. Throughput(ops/sec). Color shows details about Database. The marks are labeled by operation count and Throughput(ops/sec). The data is filtered on Throughput(ops/sec) and operation count. The Throughput(ops/sec) filter ranges from 1885.725061 to 5787.930236. The operation count filter ranges from 20000 to 150000.

Workload B

Read operations vs Average Latency

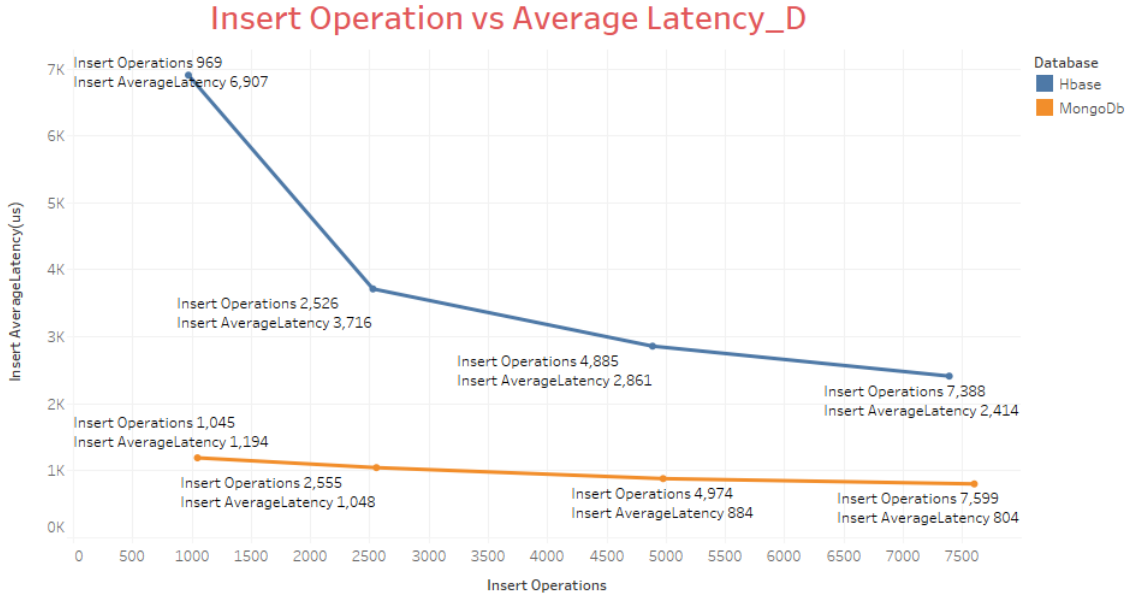
Chart shows Read operation on x-axis and read average latency on the y-axis. From the chart it is clear that read average latency of the Mongodb is better than that of the Hbase in the read operation. For the MongoDB average latency decrease with increase in the read operations where as latency of the Hbase increase with increase in the read operation. So we can conclude that the mongodb is better as compare to that of the Hbase Database.



Read Operations vs. Read AverageLatency(us). Color shows details about Database. The marks are labeled by Read AverageLatency(us) and Read Operations.

Insert Operations vs Average Latency

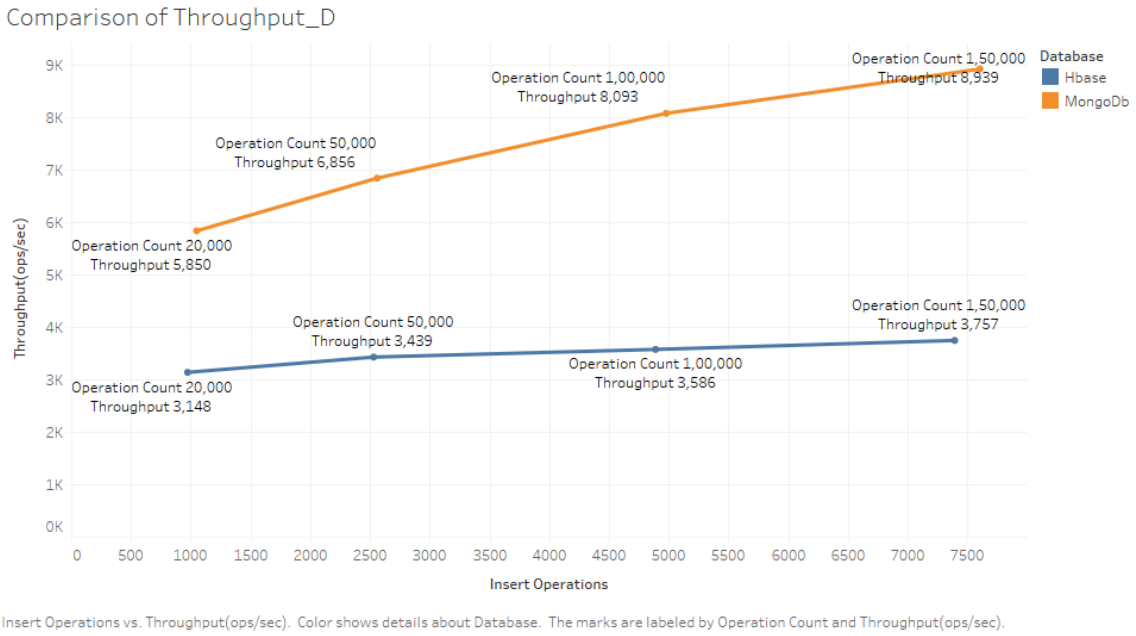
Analysis is for comparing the insert average latency with the insert operation for the database hbase and MongoDB. performance of MongoDB is better than that of Hbase database. In workload D very few records are insert records and in this case average latency of the Hbase is maximum at the beginning and decrease with the increase in the insert operations, similar for the mongodb database average latency is decrease with increase in the insert operations. so we can conclude from the graph that Mongodb is having better insert average latency than Hbase database .



Insert Operations vs. Insert AverageLatency(us). Color shows details about Database. The marks are labeled by Insert Operations and Insert AverageLatency(us).

Throughput Comparison

Graph show the throughput on the y-axis and operation count on x- axis from the graph performance of the database Hbase and Mongodb analyze through the throughput vs operation count graph. Performance of Mongodb is better than that of Hbase as the operation count increase the operation/sec handling is increase for the mongodb .Similarly for the hbase throuput increase with increase in the number of operation count. But for the same operation count throughput of the MOngodb is better than that of Hbase database.



Insert Operations vs. Throughput(ops/sec). Color shows details about Database. The marks are labeled by Operation Count and Throughput(ops/sec).

9. CONCLUSION AND DISCUSSION:

There are different types of database management systems software's available in the markets such as My-SQL ,oracle,db2,sqllite, performance testing is most important ,actually its ability to detect the flaws when we apply high volume of the data to the system and this is helpful to test the application in the different way such as used to identify the behaviour of the application when how high volume of the data is applied also this test is useful to check wheather scaling is works as per the expectation or not. (DEV9, 2017)Mongodb and Hbase are two No-SQL databases and YCSB is used for the benchmarking purpose and number of records generated are ranges from the 20000 to 150000 and these are uses for performance testing of the database. There are different region such as Memory leak, THREADING PROBLEMS, VALUE SIZE ERRORS, SHARED STATE PROBLEMS where performance test may find the hidden problems below are the results of performance test which compare the latency and through put of the Hbase database and Mongodb database. In every aspect of the YCSB Benchmarking MOngodb is better than Hbase for different workload having the different opcount.

Test result (workload A)

1) Update latency of the mongodb database is in much less than that of the Hbase for the same number of workload and the value average latency of mongodb is 946 us and hbase 2194us for the 75000 update operation

2) Read latency of the mongodb database is minimum than Hbase. the value of mongodb is 718us and for Hbase it is 2774us for the maximum read operation count

3) Throughput is the most important parameter which shows the total number of operations count by the system for the particular operation count, again throughput of mongodb is much higher compare to Throughput of the Hbase.

Test result (workload D)

1) For the workload D insert operation is performed and the average insert latency is calculated. For the mongodb and hbase again performance of mongodb is better than that of the Hbase database.

2) Read latency is less for mongodb than hbase. so we can conclude that mongodb is more better than Hbase.

3)Throughput of the mongodb is much higher than hbase and so the so the Mongodb capacity of handling operation count per second is greater than that of hbase .

Among to No-sql databases performance of mongodb is better than that of Hbase as read latency, update latency and insert latency of mongodb are less and throughput is more than HBase.

References

Apache HBase High Availability at the Next Level. (n.d.). Retrieved from https://hortonworks.com/blog/apache-hbase-high-availability-next-level/

Core YCSB properties. (n.d.). Retrieved from https://github.com/brianfrankcooper/YCSB/wiki/Core-Properties

Craig S. Mullins, p. &. (2018, 2 9). *What do we mean by Database Scalability?* Retrieved from http://www.nuodb.com/techblog/what-do-we-mean-database-scalability

flair, D. (n.d.). Retrieved from https://data-flair.training/blogs/features-of-hbase/

Gabry, O. E. (n.d.). Retrieved from https://medium.com/omarelgabrys-blog/database-introduction-part-1-4844fada1fb0

HBase Architecture: ZooKeeper and HBase Reliability. (n.d.). Retrieved from https://www.mindstick.com/Articles/12062/hbase-architecture-zookeeper-and-hbase-reliability-part-5

HBase Overview. (n.d.). Retrieved from http://hadooptutorial.info/hbase-overview/

HBase Scalability Features. (n.d.). Retrieved from https://quabase.sei.cmu.edu/mediawiki/index.php/HBase_Scalability_Features

HBASE, A. (n.d.). Retrieved from https://hbase.apache.org/

How does MongoDB ensure avilablity? (n.d.). Retrieved from https://www.mongodb.com/faq?jmp=footer

jar, T. (n.d.). *What Are The Key Features Of MongoDB?* Retrieved from https://www.tutorialsjar.com/key-features-of-mongodb/

MongoDB at Scale. (n.d.). Retrieved from https://www.mongodb.com/mongodb-scale

mongoDB. (nd). Retrieved from https://www.mongodb.com/nosql-explained

node.management. (n.d.). Retrieved from http://hbase.apache.org/book.html#node.management

Test harness. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Test_harness