Comparison between MongoDB and HBase Data Storage Management Systems

Arth Vyas

X17170516

M.Sc. Data Analytics

1. Abstract

In the era of growing Data, predication is made that by 2020, 35 ZB data is going to be generated. There are plethora storage management systems available for storage and analyze. Due to advancement of technology new storage system have been generated which overshadow previous database system, there is big debate on selection of database usage based on the requirement of an organization. The objective of this paper is to evaluate the performance of two No-SQL databases called Hbase and MongoDB. Here Yahoo Cloud Serving Benchmark (YCSB) has been used as standard to test the performance and characteristics of both the database. The Results and interpretation of each test has been explained in this Paper. It has been seen that both the database have their own pros and cons. It is up to individual organization to select the database based on the requirement. The test plan, procedure and results are discussed briefly in this paper.

2. Introduction

2.1 Big Data

Big data is an evolving term that describes a large volume of structured, semi-structured and unstructured data that has the potential to be mined for information and used in machine learning projects and other advanced analytics applications [1]. It is often characterized by 3Vs because of extreme Volume of data, wide variety of data type and velocity of data to be processed. Now storage and retrieval of such data is done in NoSQL database which will be used in relational databases.

2.2 HBase

HBase is an open source project by Apache to handle big data storage. HBase is a column-oriented database built on top of Hadoop file system which is specially designed for huge tables. It is schema-less because it does not have concept of fixed column. Some features of HBase are integration with Hadoop as a source and a destination, automatic failure support, consistent read/writes, data replication across clusters and linearly scalable.

2.3 MongoDB

MongoDB is one of the most powerful NoSQL systems and databases in today's world. It is an open source database which uses document-oriented data modeling and non-structured query language. It does not deal with the rows and column like traditional

method of database management normally it consists of key value paris. It uses binary style of JSON style document format called BSON [2].

2.4 YCSB

Yahoo! Cloud Serving Benchmark aka (YCSB) is an open-source program suit for evaluating retrieval and performance test of NoSQL database management system as well as computers program. Here it has been used to compare HBase and MongoDB for validating the performance regarding throughput, Average latency and update latency. Generally, it is used for testing benchmarking of all the different databases performance.

3. Key Characteristics of HBase and MongoDB

3.1 HBase

a) Linearly Scalable.
b) Automatic failure support.
c) Consistent read and write.
d) Act as source or a destination while integration with Hadoop.
e) Data replication across clusters are available.
f) Client can easily access java API.

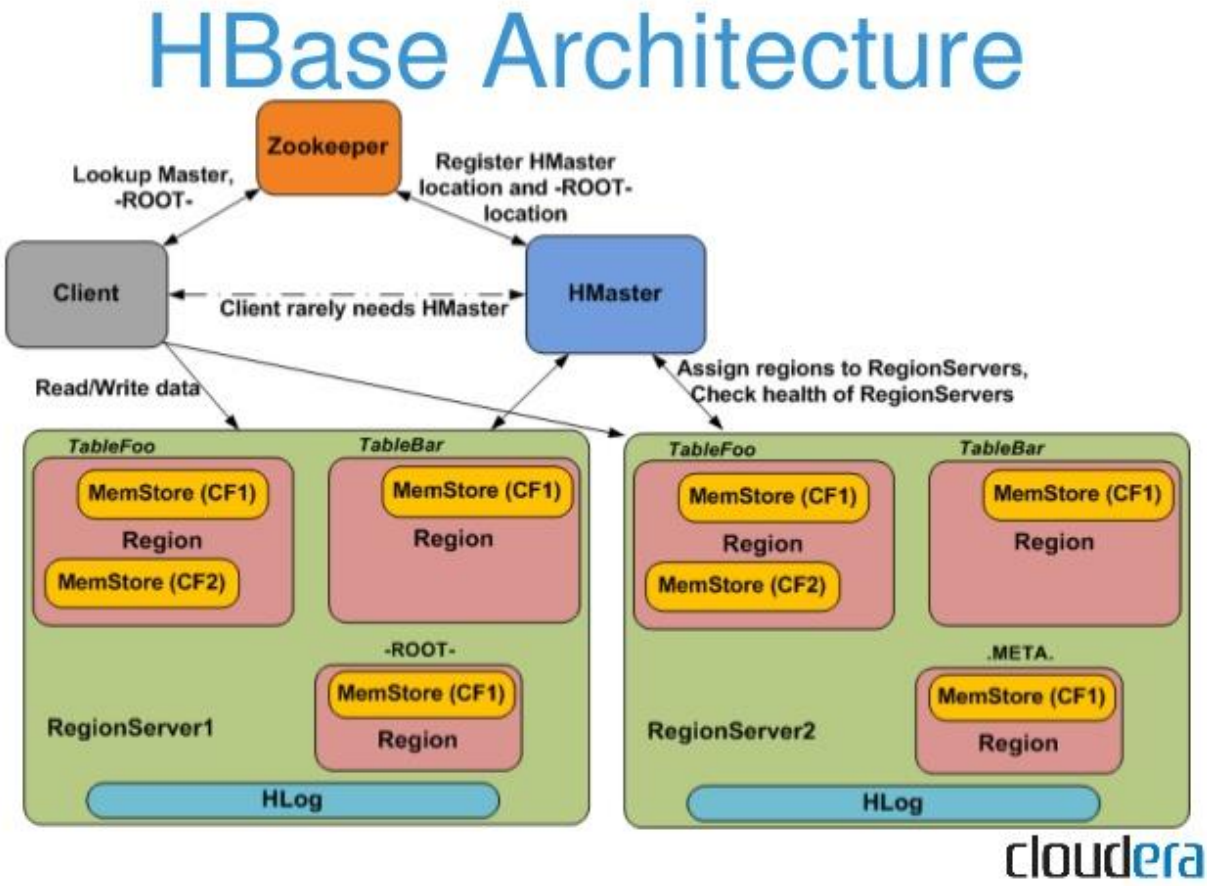
3.2 MongoDB

a) Queries: It supports ad-hoc and document-based queries
b) Index Support: It can index any field
c) Replication: It has self-healing shard to prevent downtime of database, it operates in master slave replication. It produces multiple copies of data by using native application.
d) Multiple server: It allows database to run in multiple server and data is duplicated to have backup in case of hardware failure.
e) Auto-Sharding: Sharding create multiple partition where data can process due to this it supports automatic load balancing feature.
f) MapReduce: it is an aggregation tool.
g) Failure handling: It create multiple duplicate data while processing so, every time when system get failure data is secure to retrieve from database.
h) GridFS: Irrespective of file size, it breaks file in small size and stores it in a separate document.
i) Schemaless: written in C++
j) Document-Oriented Storage: BSON format is used which is JSON like format.

4. Database Architectures

4.1 Below is the architecture of HBase:

HBase can run in multiple master setup but there will be only one active master at a time. There is facility of portioning the table in multiple region, each region can store multiple rows. [3]



There are mainly 3 important components of HBase architecture: HMaster, Region Server and Zookeeper.

a) HMaster: Load balancing in Hadoop Cluster will take place only when HMaster will assign region to region server of Hadoop. HMaster is responsible for monitoring Hadoop cluster, failover, DDL operations, Schema change request by client and administrative tasks (creating, updating and deleting tables).
b) Region Sever: The basic function of Region server is to read, write, update and delete request from clients. It run on HDFS Datanodes and consist of following components:-

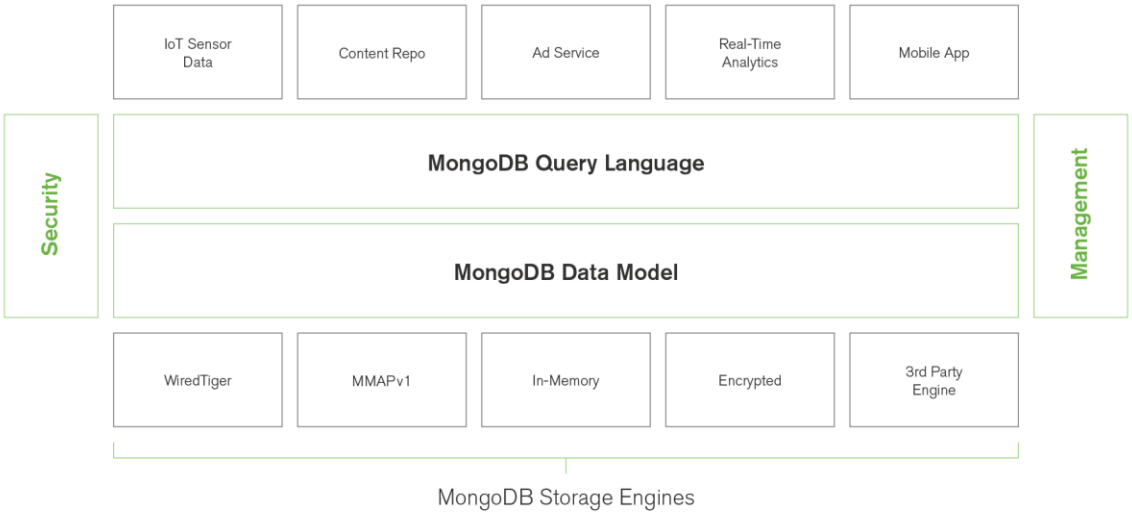
- Block Cache: It is read cache, read data is stored in the read cache.
- MemStore: It is write cache, it is present in each column family and data which has not been written yet is stored here.
- Write ahead log (WAL) store the data which is not yet given permanent location for storage.
- HFile is the file where actual storage takes place and rows are stored on disk as sorted key values.

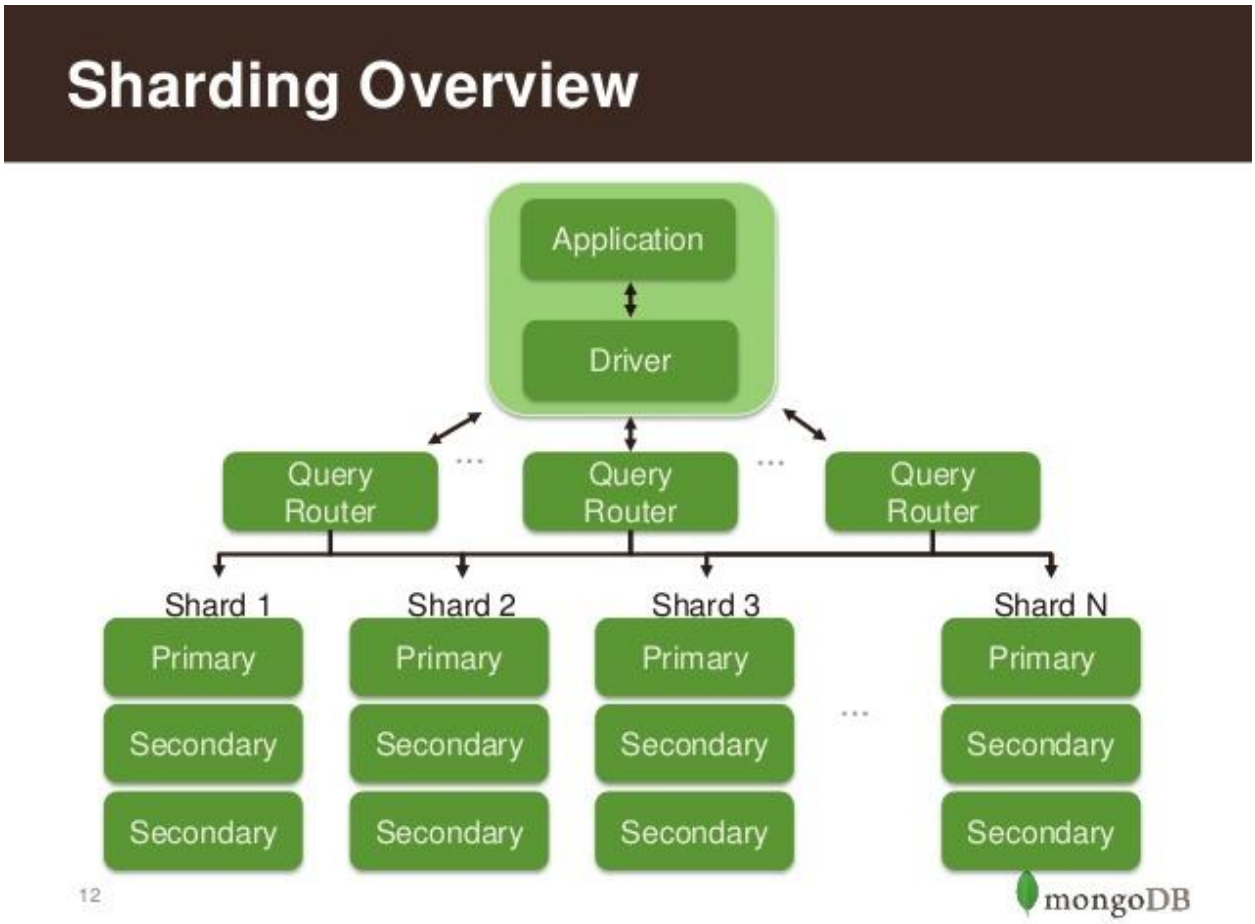
c) Zookeeper: HBase uses Zookeeper for few important tasks. Zookeeper work as a coordinator for region assignments and it records if any region gets crashes by

loading into server which are functioning. It provides distributed synchronization as well as maintain configuration information. If client want to connect HMaster and region server, they must access it through Zookeeper quorum. Whenever node failure occurs within HBase cluster, Zquoram will generate error and will take care of it by taking necessary actions. [3]

4.2 MongoDB

MongoDB architecture is the only one database maintaining the foundation of relational databases with No-SQL. It has a flexible architecture developer can use pluggable storage engine which reduce the complexity to run multiple databases. It can be mixed on same replica or sharded cluster. It leverages the user to use same mongoDB query language, data model, sacling, security and operational tool all of them are powered by different storage engines. [4]





Sharding method in MongoDB support deployment of the large data set and provide high through put by distributing workload across multiple machines.

The two methods for sharding are vertical and horizontal. In vertical method it focus on increase in the capacity of single server by using powerful CPU, addition of RAM or increasing space of storage space. In horizontal method it focusses on dividing workload in multiple servers by which increase the capacity as required. [5]

5. Security
   There are major components of security which must be taken care of while selecting or working with the database.
   - Authentication
   - Access Control
   - Secure configuration
   - Data Encryption
   - Auditing

Assessment Criteria	MongoDB	HBase
Authentication	Medium	Medium
Access Control	High	Medium
Secure Configuration	Medium	Low
Data Encryption	Medium	Low
Auditing	Low	Medium

Table1: Comparative Analysis of sharding Security in various Databases [6]

Authentication for both MongoDB and HBase are on medium level means. They can connect to only one type of authentication e.g. SSO, OpenID, SAML [6]

Access Control supported by MongoDB is high means it has dynamic access and UCON (usage control Mode). On the other hand HBase has Medium security which suggest it support RBAC, DAC, MAC, ABAC etc. [6]

Secure configuration for MongoDB is Medium and HBase is low. MongoDB permits minimum protocol for communication and strictly to registries and accounts. While HBase is on low security can access only default configurations.

Data Encryption is Medium for MongoDB and Low for HBase. So, MongoDB have security on data at rest and transport level. While, HBase have no data security.

Auditing MongoDB have no auditing mechanism can be said because it assessed on low level while HBase have some security level to Database and user profile hence assessed at Medium level.

6. Learning from Literature review:
   In this section a paper has been selected for the literature review, This paper is about Benchmarking Top NoSQL Databases. The test performance has been done on Apache Cassandra, Couchbase, HBase, and MongoDB these all are the top trending databases in NoSQL. The methodology, Configuration, and test results will be discussed and reviewed. End point performed a series of performance test on above mentioned database using YCSB benchmark. The overall conclusion made after the test was Cassandra was the leading NoSQL database in terms of throughput and latency whereas, HBase and Couchbase were next best and MongoDB stood last. Let us look into the finding in depth. [7]

   Test methodology and configuration [7]:
   - Amazon Web Service EC2 instances, to minimize the effect of AWS CU and I/O variability. Three tests were performed on three different days.
   - Test used i2.xlarge class instances for database node and c3.xlarge class of instance as client node.
   - Ubuntu 14.04 LTS, customized with Java 7 was used.
   - A script was used to start, terminate, configuration, calculation and issuing command to client to run test.

- Client software was a customized YCSB install, build from github.
- Cassandra software version 2.1.0 was used, first node was declared the seed_provider in Cassandra.yaml so that host could find each other and build cluster automatically.
- Couchbase software version 3.0.1 was used prerequisite setting like allocating RAM and configuration of cluster were made.
- HBase software was install with version 0.98.6-1 with latest Hadoop (2.6.0), all the xml files were read and necessary changes made to configure cluster. First node was declared as master and ran as both Hadoop namenode and Hmaster and ran Zookeeper process. All node started as datanode and pointing master node zookeepr for coordination.
- MongoDB software version 3.0 necessary changes for configuration of cluster were made and once the cluster was ready driver script connected to first node and enabled sharding.

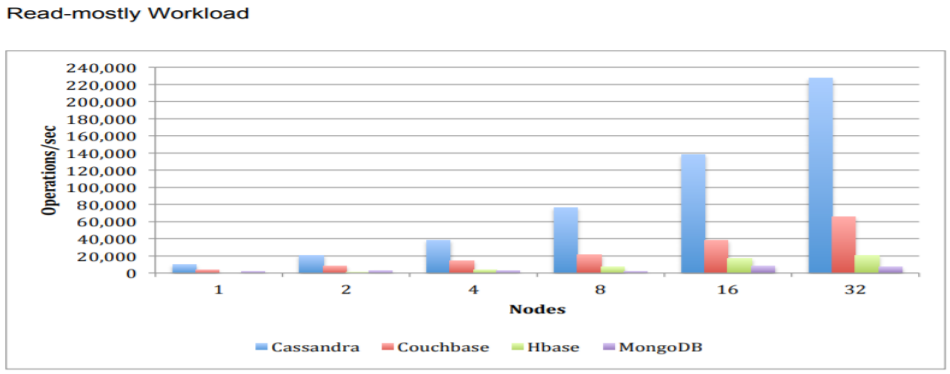
Workload Selection [7]: Major five workload were selected as mentioned below

- Read-mostly workload B: 95% read to 5% update ratio
- Read/write combination workload A: 50% read to 50% update ratio
- Read-modify-write workload F: 50% read to 50% ready-modify-write ratio
- Mixed operational and analytical: 60% read, 25% update, 10% insert, and 5% scan
- Insert-mostly combined with read: 90% insert to 10% read ratio
- Field length was set to 10 to produce 100 byte record.
- For scan 100 record were incorporated.
- Operations counts were 9 million for each workload.

Results collection and Aggregation [7]: All the output generated by databases were collected In terms of overall throughput, operations per seconds and average latency. Each respective value for three test were average and plot against each other to find the results.

A sample graph for throughput and average latency will be reviewed for learning here.

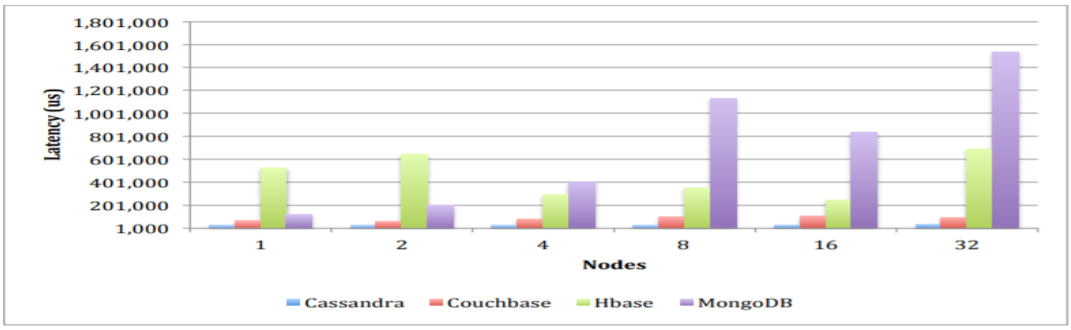
Throughput by workload: Database to perform well through of each database should be more. Let have a look on Read-mostly workload



Nodes	Cassandra	Couchbase	HBase	MongoDB
1	9,768.76	3,950.90	428.12	2,149.08
2	20,511.51	8,147.78	1,381.06	2,588.04
4	38,235.67	14,503.08	3,955.03	2,752.40
8	76,169.72	20,904.02	6,817.14	2,165.17
16	138,141.79	38,084.77	16,542.11	7,782.36
32	227,292.80	65,978.17	20,020.73	6,983.82

Here we can see the Cassandra is outperforming among all the other database.

Average Latency by workload: In case of latency for better performance latency value should be low let us see which database has performed well in latency test.



Nodes	Cassandra	Couchbase	HBase	MongoDB
1	26,312.23	66,166.99	522,835.62	119,139.25
2	25,170.28	62,683.11	645,326.42	197,806.93
4	26,816.47	82,561.51	291,440.74	407,118.75
8	26,973.70	98,514.50	350,383.54	1,126,244.08
16	30,918.79	109,061.54	248,318.31	835,206.29
32	36,322.71	97,200.37	689,000.88	1,540,875.22

It is evident from the graph above that Cassandra again has stood low in terms of latency as against other database.

Finally, after the test performance in several conditions with different workload and different database software. It is evident that Cassandra is undeniably won the race among other database in terms of throughput and average latency. It depends on need of an organization other database. Still all the databases are well known for their unique performance the only difference is the selection of database for which purpose. This review give us strong understanding about

how to performance test of different database of NoSQL and benchmarking each of them according to our need.

7. Test Plan and requirement:
   To perform the test on MongoDB and HBase following databases, systems and files are required.

   - Hadoop
   - HBase
   - MongoDB
   - Java Runtime Environment Open JDK8
   - Java SE Development Kit Open JDK8
   - Python
   - ssh
   - YCSB
   - MySQL
   - Instance on Openstack

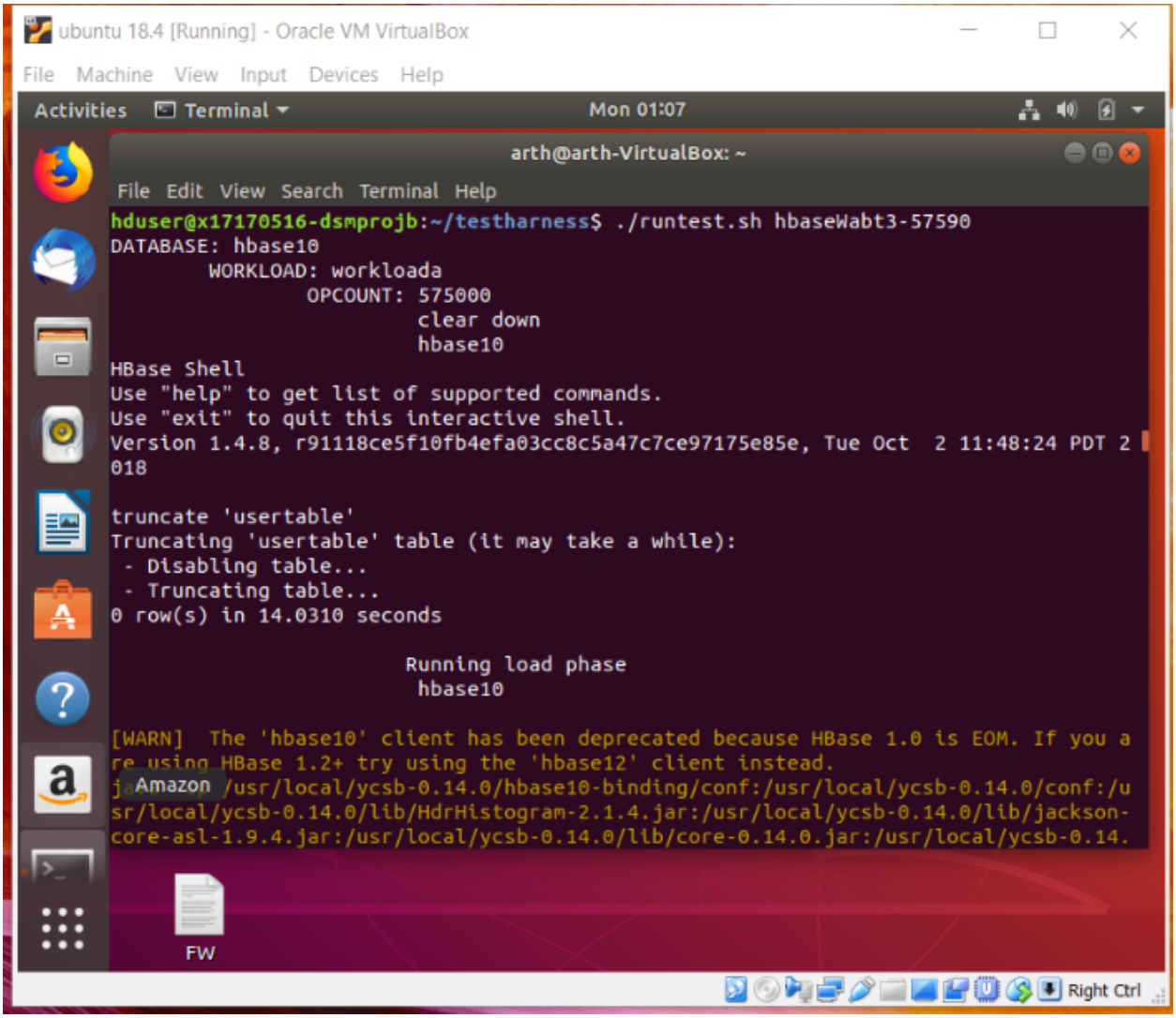
   The opcounts and workload has been test for HBase and MongoDB for three time so that we can take average value of overall test for each opcounts for both workload which will put us in strong position to evaluate the results of tests and better interpretation will be made between two database. The following five opcounts were selected for the tests are 500000, 525000, 550000, 575000 and 590000. These large number of opcounts will be a reasonable opcounts for workload to be test both the database.

   Major two characteristics was aimed while test are mentioned below:

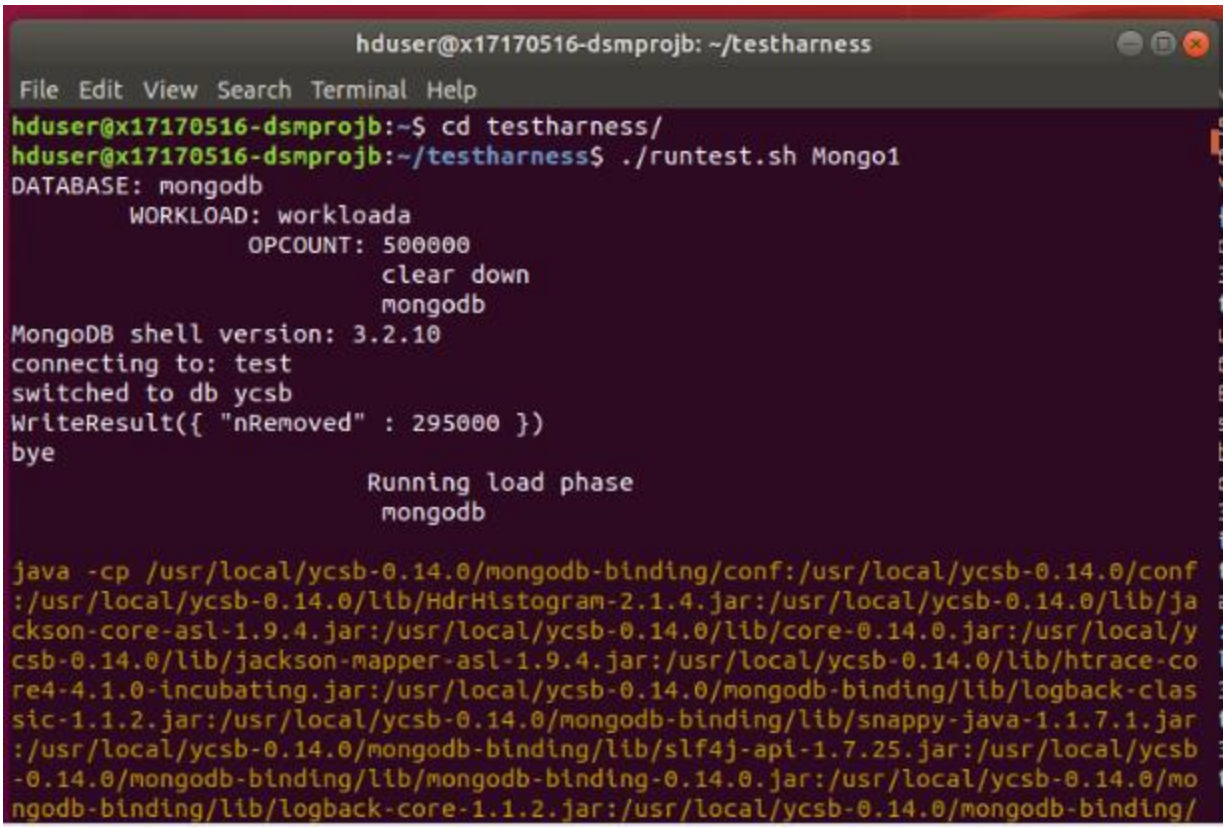
   - Average Read/Update Latency.
   - Throughput.

   For any database the above two component are very important to be outperformed, Which is the requirement of today's IT sector because of the increase the Raw data.

   Below is the screenshot of Running test on HBase.



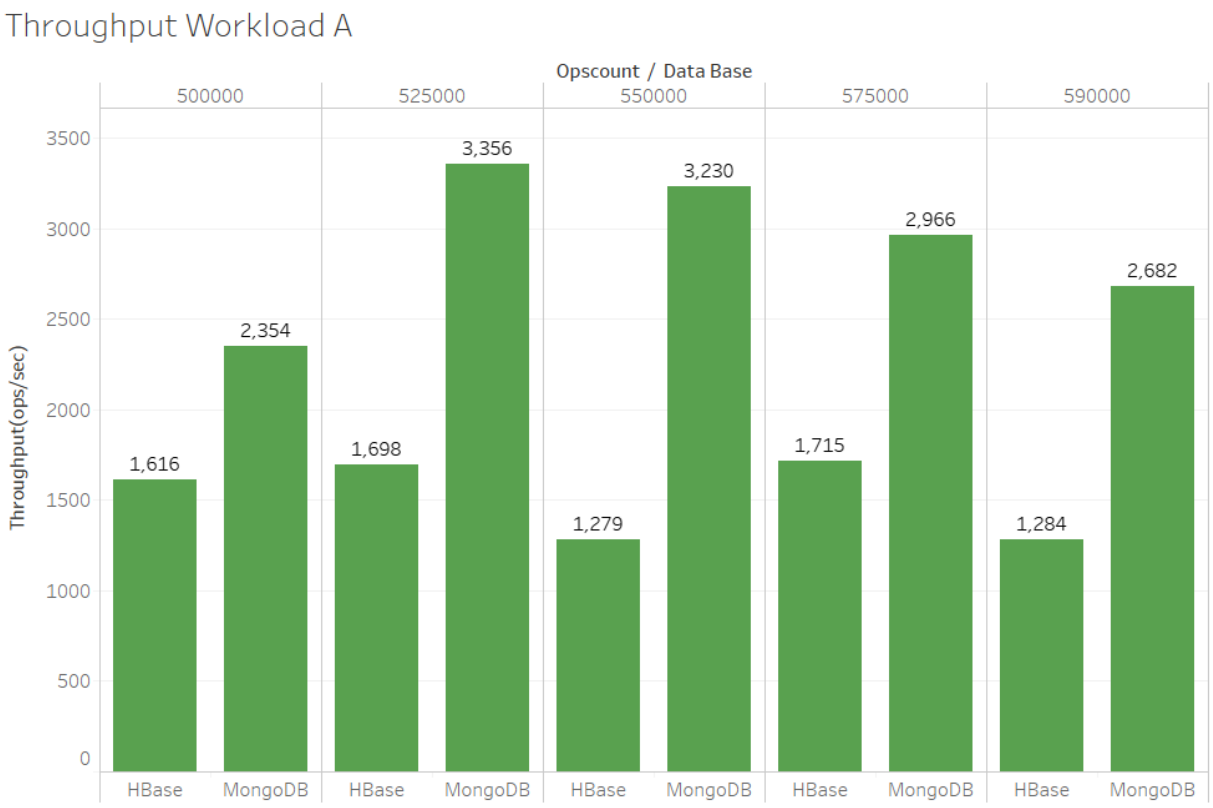
The next screenshot is of running test on MongoDB



8. Results and Interpretation:

Workload A: Operations Count Vs Throughput for HBase and MongoDB

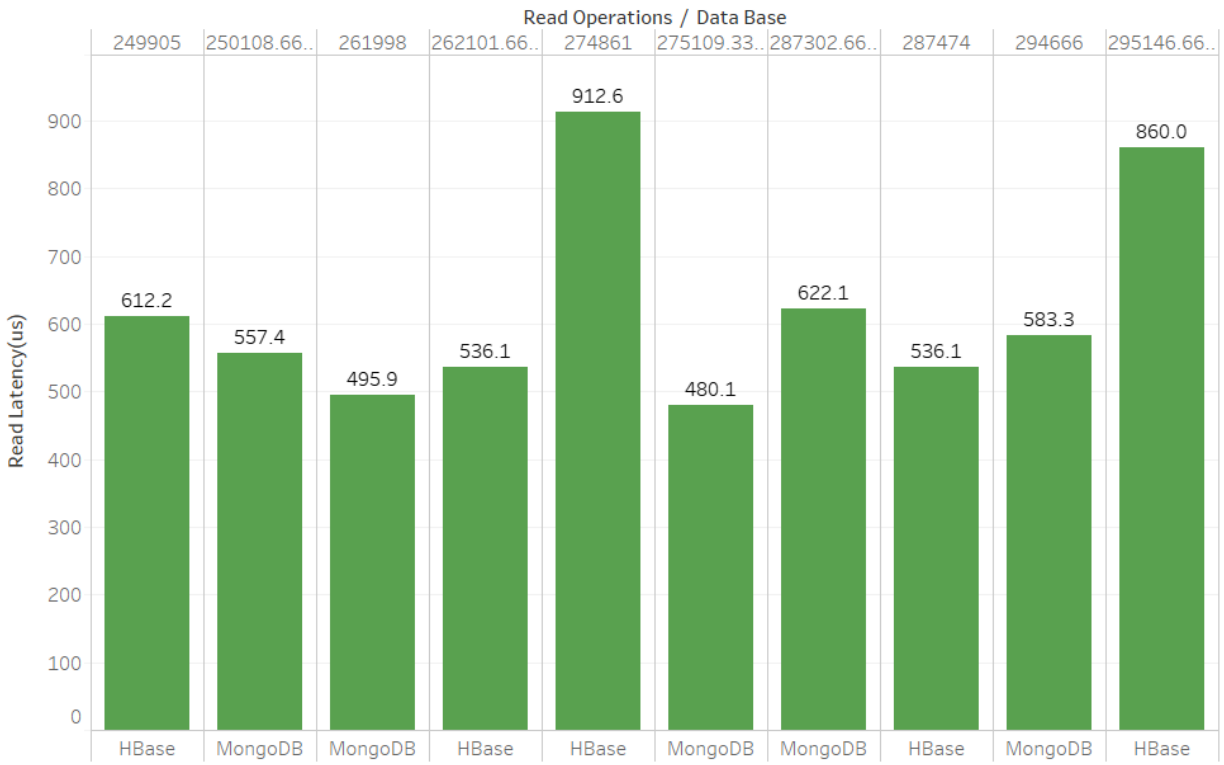
Data Base	Opscount	Throughput(ops/sec)
HBase	500000	1616
HBase	525000	1698
HBase	550000	1279
HBase	575000	1715
HBase	590000	1284
MongoDB	500000	2354
MongoDB	525000	3356
MongoDB	550000	3230
MongoDB	575000	2966
MongoDB	590000	2682



The above clearly summarized the difference of throughput for HBase and MongoDB. The trend and value of HBase is significantly low with respect to MongoDB. Performance of MongoDB is better on higher operations count too. So if the requirement is for high through than without doubt MongoDB can be preferred over HBase.

Read Operation Vs Read Average latency

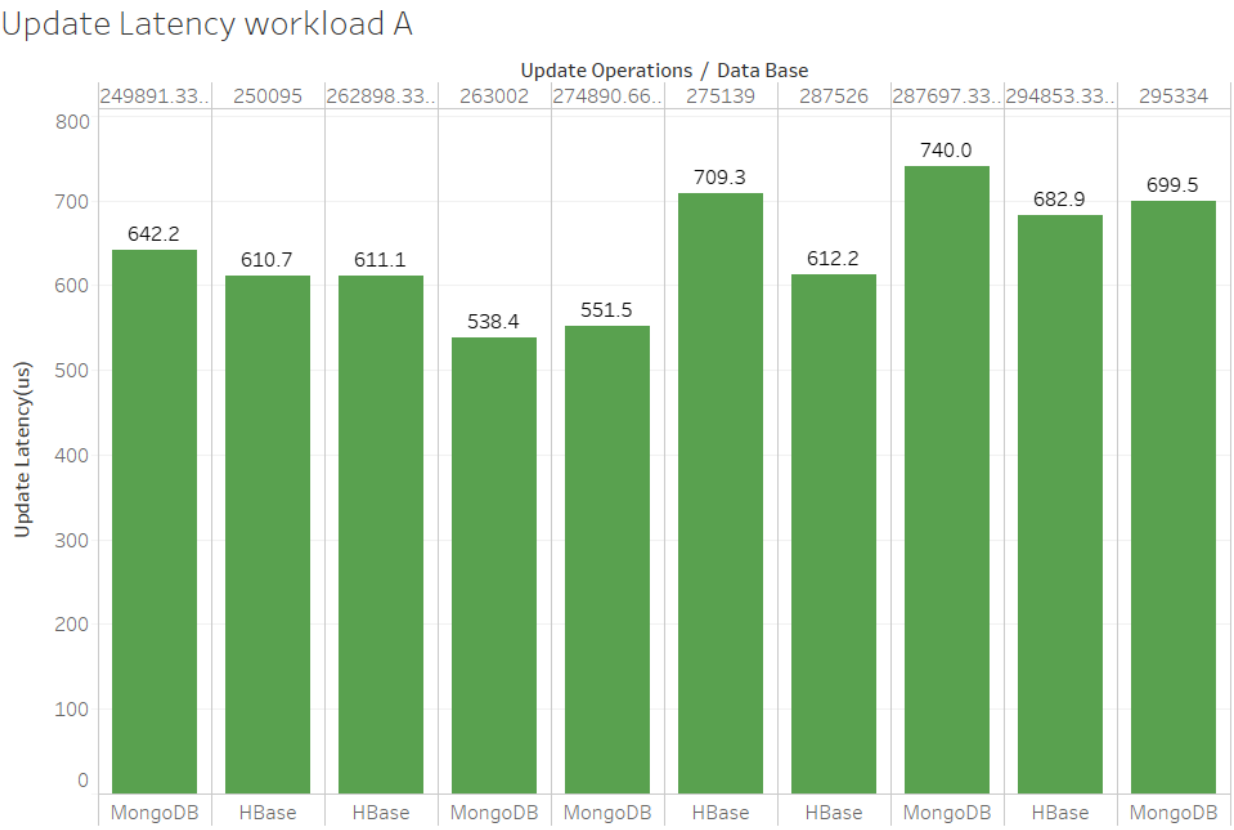
Data Base	Read Operations	Read Latency(us)
HBase	249905	612
HBase	262102	536
HBase	274861	913
HBase	287474	536
HBase	295147	860
MongoDB	250109	557
MongoDB	261998	496
MongoDB	275109	480
MongoDB	287303	622
MongoDB	294666	583



The value of Read latency will start increasing with increase in number of read operation count. In the above graph it can be interpreted that the value of MongoDB is increasing slowly with the increase in operation count whereas the value of HBase value is showing inverse trend of decrease in latency. The better performance is definite on low value so, we can say HBase will be perform better against high read operations as against MongoDB.

Update Operation Vs Update Latency

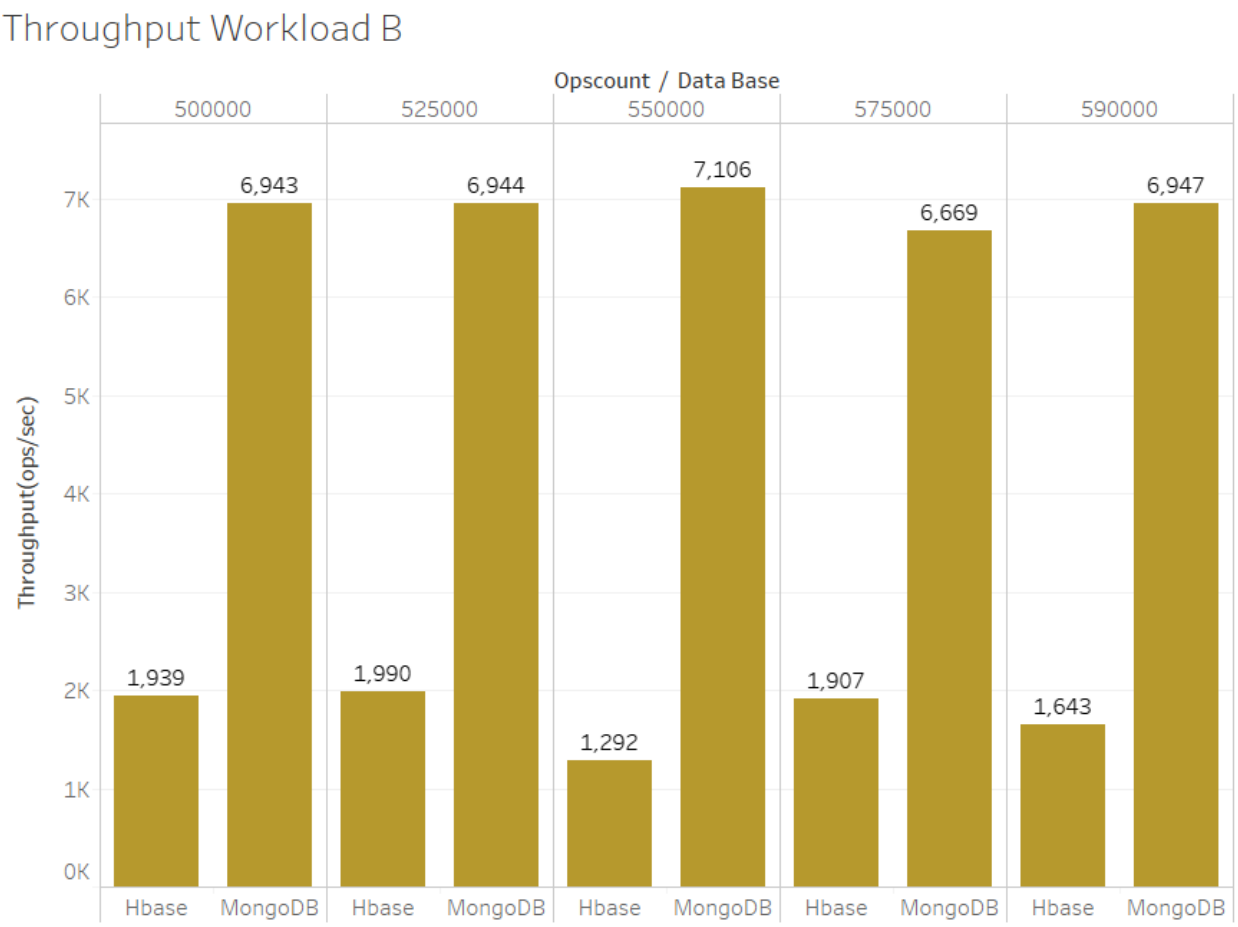
Data Base	Update Operations	Update Latency(us)
HBase	250095	611
HBase	262898	611
HBase	275139	709
HBase	287526	612
HBase	294853	683
MongoDB	249891	642
MongoDB	263002	538
MongoDB	274891	552
MongoDB	287697	740
MongoDB	295334	700



The above chart clearly represents that with rise of update operation value of MongoDB is increasing HBase is slightly on lower side which indicate for achieving good latency HBase is preferred over MongoDB.

Workload B Operations counts Vs Throughput

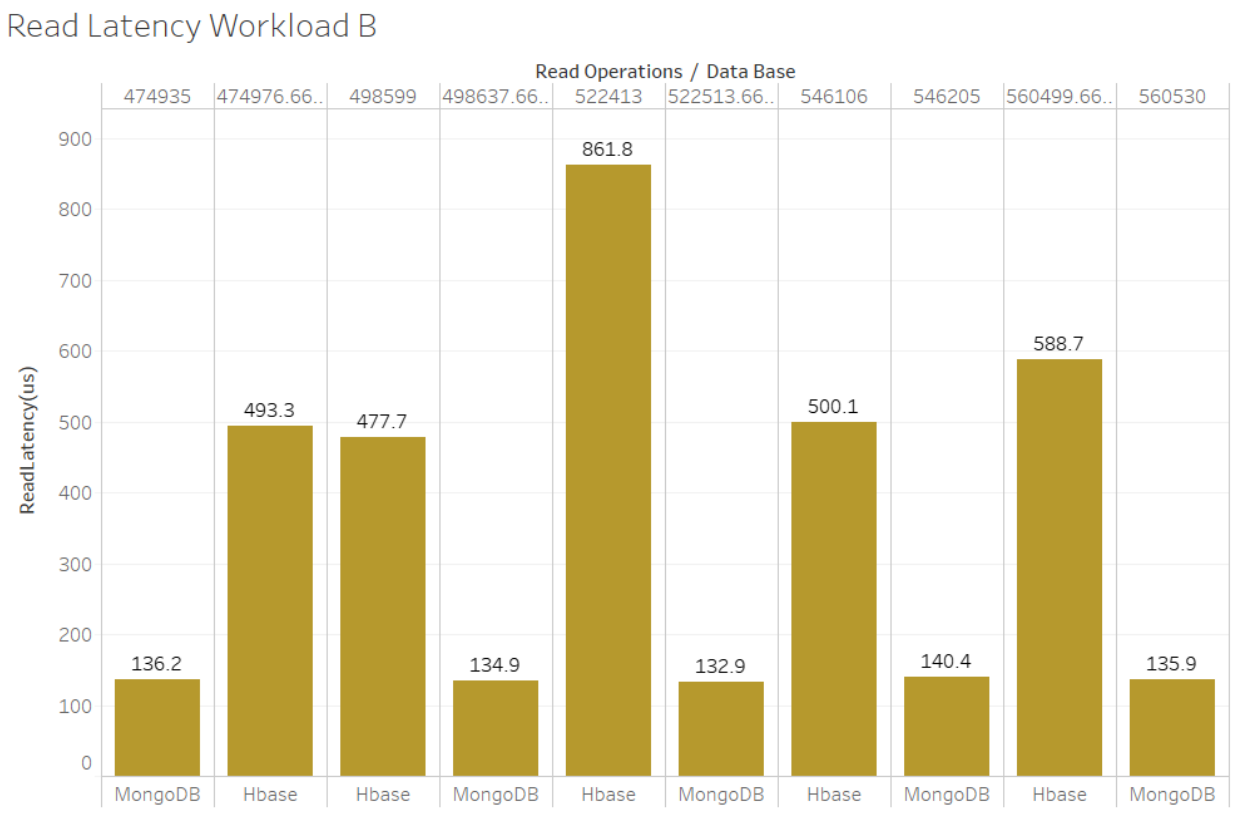
Data Base	Opscount	Throughput(ops/sec)
Hbase	500000	1939
Hbase	525000	1990
Hbase	550000	1292
Hbase	575000	1907
Hbase	590000	1643
MongoDB	500000	6943
MongoDB	525000	6944
MongoDB	550000	7106
MongoDB	575000	6669
MongoDB	590000	6947



It is evident from the above graph that throughput of MongoDB is far better than HBase. So it advise that MongoDB will be the first choice whenever the better throughput requirement is there.

Read Operation Vs Read Latency

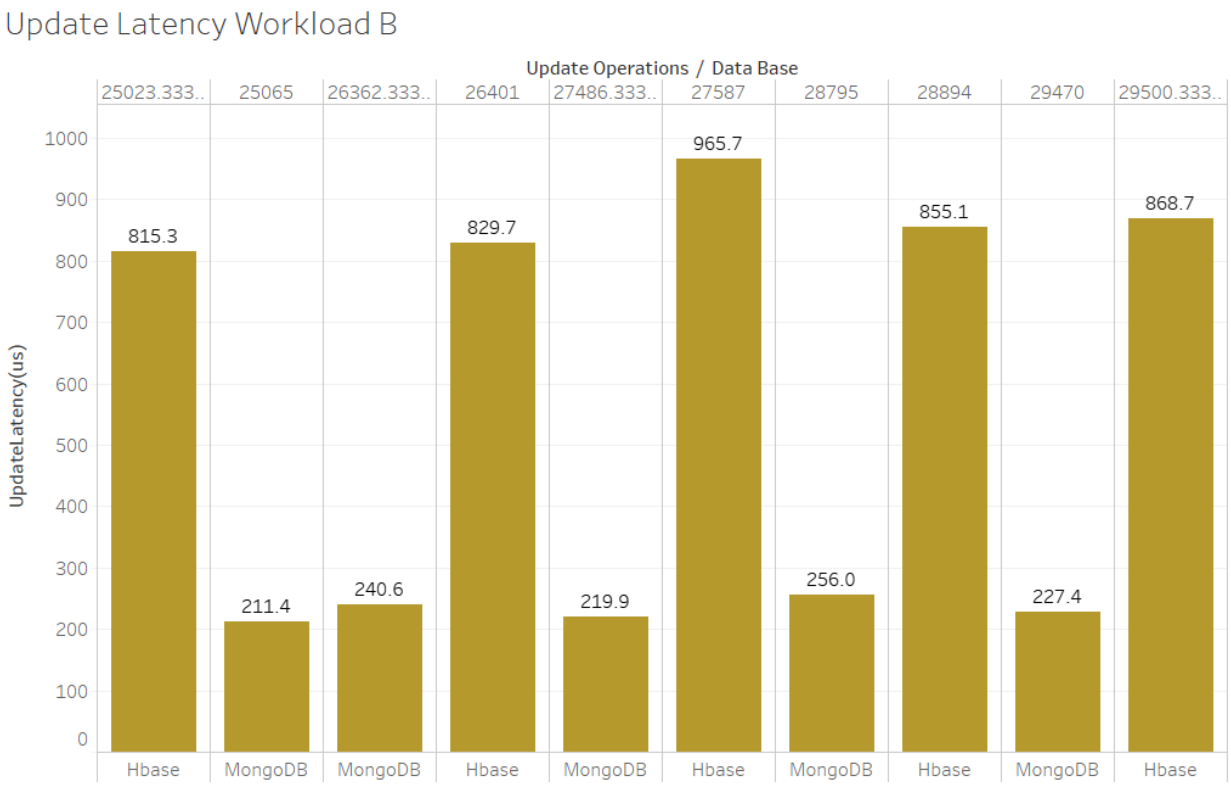
Data Base	Read Operations	ReadLatency(us)
Hbase	474977	493
Hbase	498599	478
Hbase	522413	862
Hbase	546106	500
Hbase	560500	589
MongoDB	474935	136
MongoDB	498638	135
MongoDB	522514	133
MongoDB	546205	140
MongoDB	560530	136



The above graph shows that the read latency for MongoDB is very low when test on workload B against HBase. For better performance the latency should be low. In this scenario MongoDB can be preferred over HBase.

Update operation Vs Update Latency

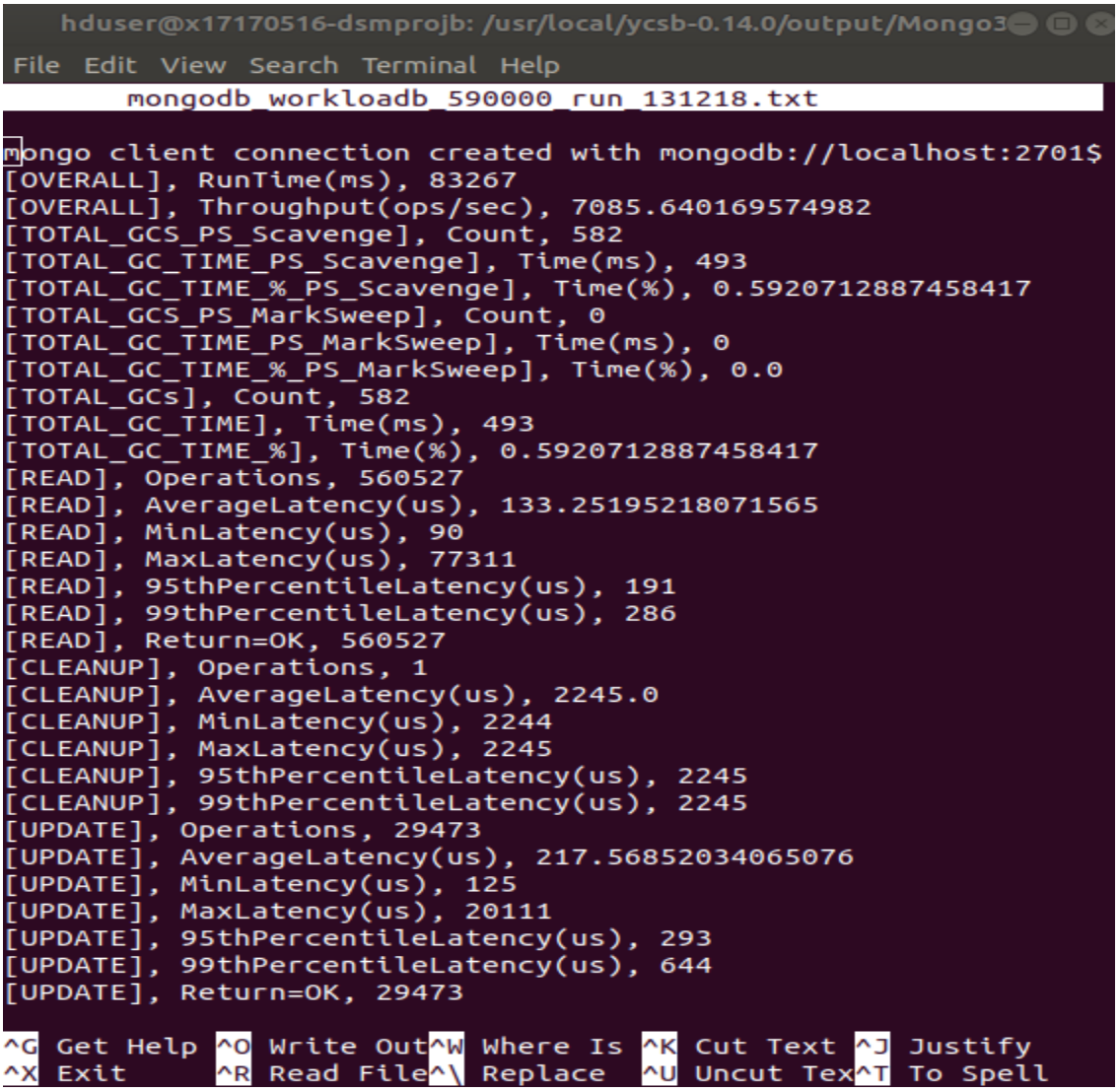
Data Base	Update Operations	UpdateLatency(us)
Hbase	25023	815
Hbase	26401	830
Hbase	27587	966
Hbase	28894	855
Hbase	29500	869
MongoDB	25065	211
MongoDB	26362	241
MongoDB	27486	220
MongoDB	28795	256
MongoDB	29470	227



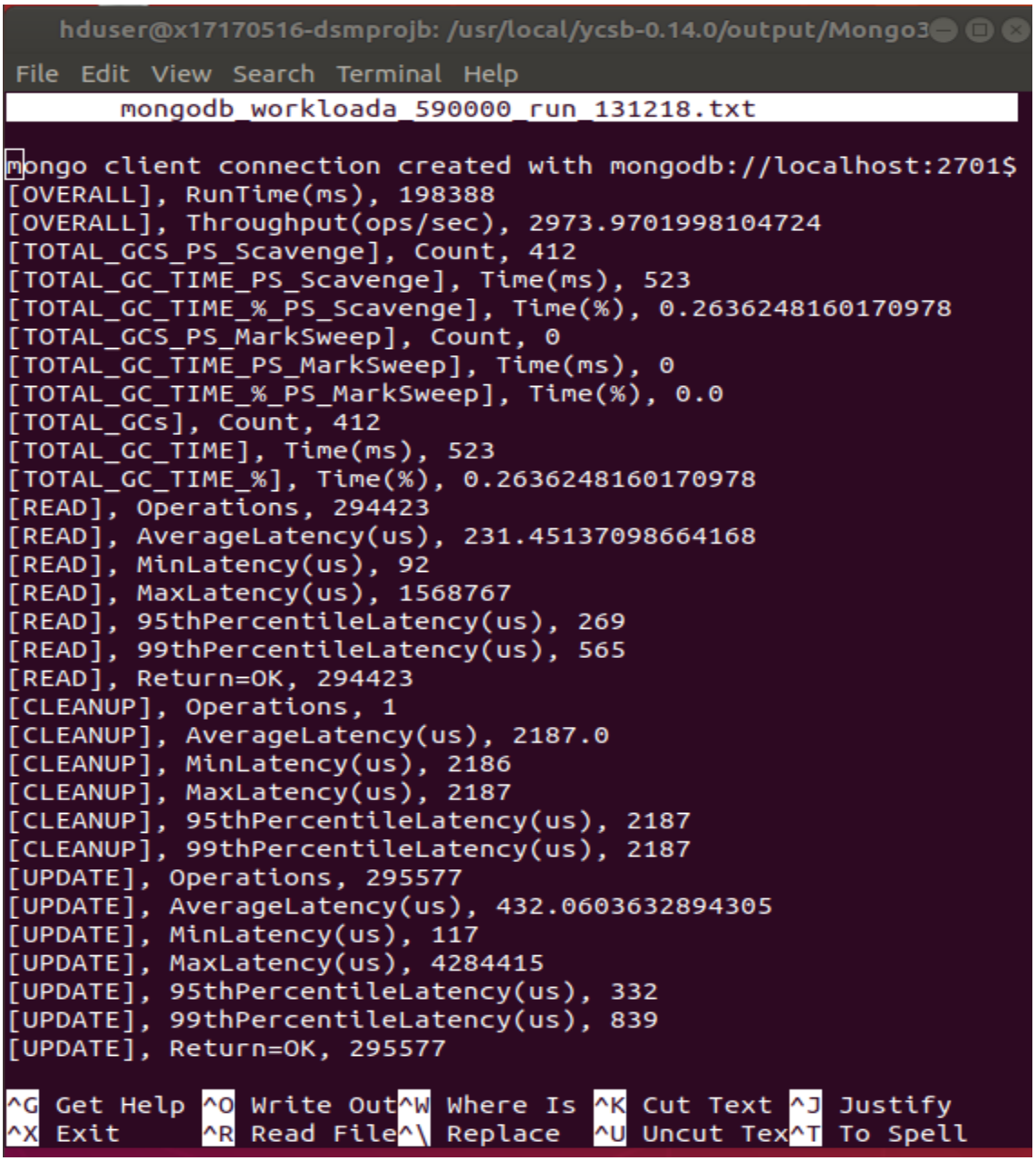
The scenario where the requirement will be like workload B in which % for read and % for write is available then MongoDB will be the best option with low latency over HBase.

Below are the screen shots of the value obtained in YCSB for HBase and MongoDB for operation count of 590000.

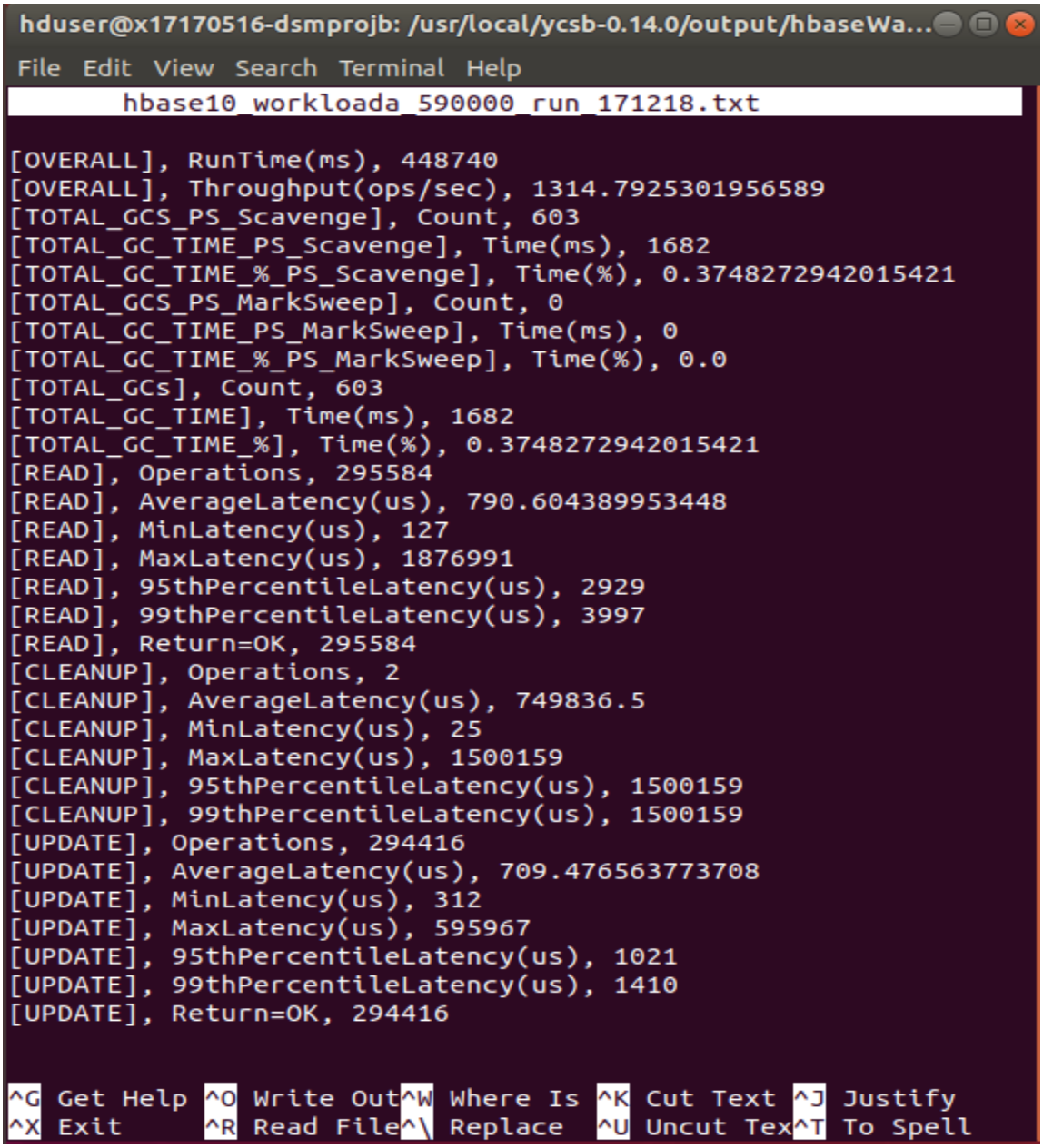
Workload B for MongoDB



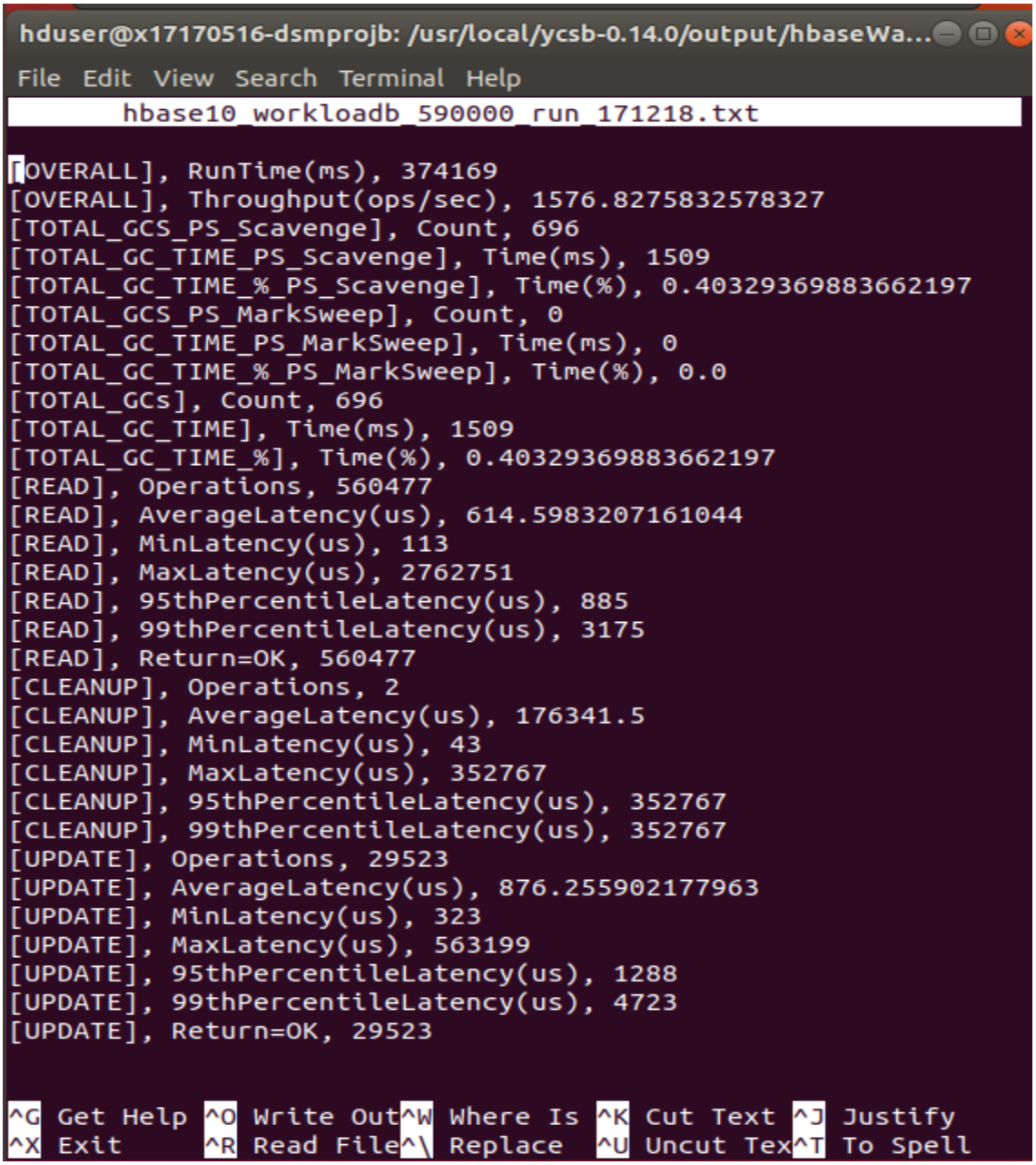
Workload A for MongoDB



Workload A for HBase



Workload B for HBase



9. Conclusion and Discussion:

All the above test done for different operation count performed for workload A and B has given fair enough idea about the characteristics of MongoDB and HBase performance. So, for any organization looking for NQL database between MongoDB and HBase. The findings done in this report will assist in selection of appropriate database according to the requirement. The same has been proved and compared with the technical paper on Cassandra, Couchbase, HBase and MongoDB. All the analysis and report fetch from research paper with respect to HBase and MongoDB can be compare and validate by the

actual practical result came after the completion of this test. The research paper have stated that in the league of NOSQL database, Cassandra is leading followed by Couchbase and HBase then MongoDB. The similar pattern we have got in the above test too. As per the tremendous increase of data in today's world all the IT giants want to develop, intercept and understand the raw data into useful form and use for the company growth as well as development. Let us summarize the findings of HBase and MongoDB which has been performed several times to get the accurate value for analyzing the best database based on requirement and give idea about the throughput, read latency and update latency. There is other workload too for carrying out mix read write, insert, mostly read, and so on. Here we have analyzed only latency and throughput which are main component in the view of IT sector. After analyzing all the results, it is clear that MongoDB is suitable for small operation counts where in return it gives the fantastic throughput in comparison of HBase. So, without any ambiguity or doubt if requirement is for achieving better throughput MongoDB is best solution. On the other hand, HBase take over the MongoDB in terms of Read and write latency. It has been observed that with the increase in operation count for read and write the performance of MongoDB get little bit shaky and whereas HBase shows improvement and provide low value then MongoDB. Latency is basically waiting time before start of process so, the less the latency is faster the job will be handled. In this case HBase latency give much more benefit than MongoDB the same can be verified in the discussion in research section HBase is better when it comes to latency. The graph is the evidence of that, in each section of latency HBase has taken lead over MongoDB presented in research paper and same has been derived through our test for workload A and workload B. First the test was done between 100,000 Operation count to 300,000 operation count but for more clarity on the performance of MongoDB and HBase the above mentioned test between 500,000 to ~600,000 operation counts. Hence, we can conclude that MongoDB will not fail your expectation regarding through but can disappoint you if it is Opted for read and update latency. However in future the test with still more operation count above 1 million with three database like Cassandra, MongoDB and HBase could be really a very good option to perform test and analyze the results and compare with each other for selecting the better database for NoSQL.

10. Reference:

https://searchdatamanagement.techtarget.com/definition/big-data

https://www.tutorialspoint.com/hbase/hbase_overview.htm

https://intellipaat.com/blog/what-is-mongodb/

https://www.dezyre.com/article/overview-of-hbase-architecture-and-its-components/295

https://www.ibmbpnetwork.com/linux-blog/mongodb-architecture

https://docs.mongodb.com/manual/sharding/

https://www.datastax.com/wp-content/themes/datastax-2014-08/files/NoSQL_Benchmarks_EndPoint.pdf