ANALYSIS OF CASSANDRA CLUSTERS

BY
SAIKRISHNA JALIPARTHY

Introduction:

In this project we have learned about the Apache Cassandra database which is a key-value store database. The main aim of the project is to get hands-on experience with Cassandra cluster with the Docker containers. As the evaluation metrics we consider measuring of read, write latency using the Cassandra stress tool and as another metric which is clock skew.

Description:

a) **Apache Cassandra:** Apache Cassandra is a distributed database which is linear-scalable designed for handling enormous amount of data across multiple servers with-out compromising the performance.

Features of Apache Cassandra:

- **a.** ***Fault-Tolerant:*** It is fault tolerant system because data is automatically replicated to specified number of nodes, So, that if any one of the nodes is inactive data can be retrieved from replicated nodes.
- **b.** ***Elastic Scalability:*** As the Cassandra is highly scalable system it allows to add more number of nodes as needed so that this helps to accommodate more clients and more data as per the client requirement. So, by adding the new machines both the read and write throughput increase linearly.
- ***c.*** ***Durability:*** Cassandra is a durable NoSQL database, So, that it doesn't lose data even when the entire data-center is down.
- ***d.*** ***Flexible and dynamic data model:*** Cassandra supports all types of datatypes which include the structured, semi structured and unstructured. It will dynamically replicate the changes to the data with fast read and. Writes.

b) **Docker:** Docker is an open source tool container management service. The main aim of Docker is to build, ship, run which means that engineers can develop applications then ship the applications into the containers so that it helps to deploy it anywhere.

Difference between VM and Docker: Technically virtual-machines are built on top of the host operating system by allocating amount of space and memory to it. So, if we add more and more number of VM's in-accordance to the requirement it takes considerable amount of space and memory of the host operating system. Due to this system may encounter into slow-boot situation which eventually decrease the system performance. Moreover, Web-service providers will also charge more. So, Docker eliminates all of the above by simply sharing the OS kernel across all the containers that are running on the multiple process above host OS.

Topology 1:

Topology 1 consists of a total of five nodes out of which four are Cassandra nodes which are present in four different docker containers and one ntp node is in separate docker container. This ntp node is in synchronous with other nodes. All nodes are connected to docker network which again connected to Internet.

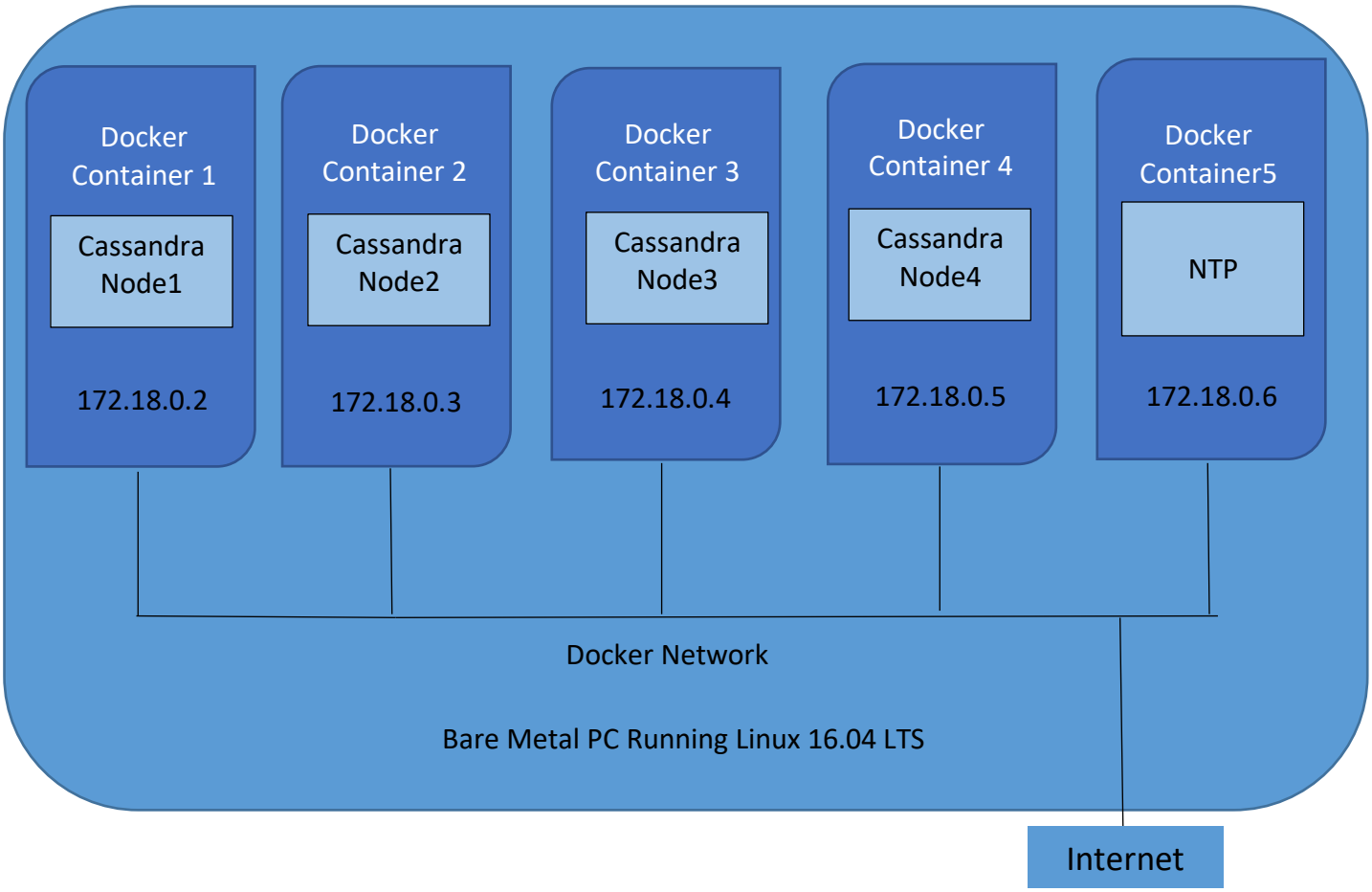


Fig. 1.1 Topology

Corresponding measurements of Read and Write Latency is measured using the Cassandra-stress command with respective read and write arguments, Clock Skew was measured using the provided server and client scripts.

Measurements of Latency:

a) Graph of Write Latency:

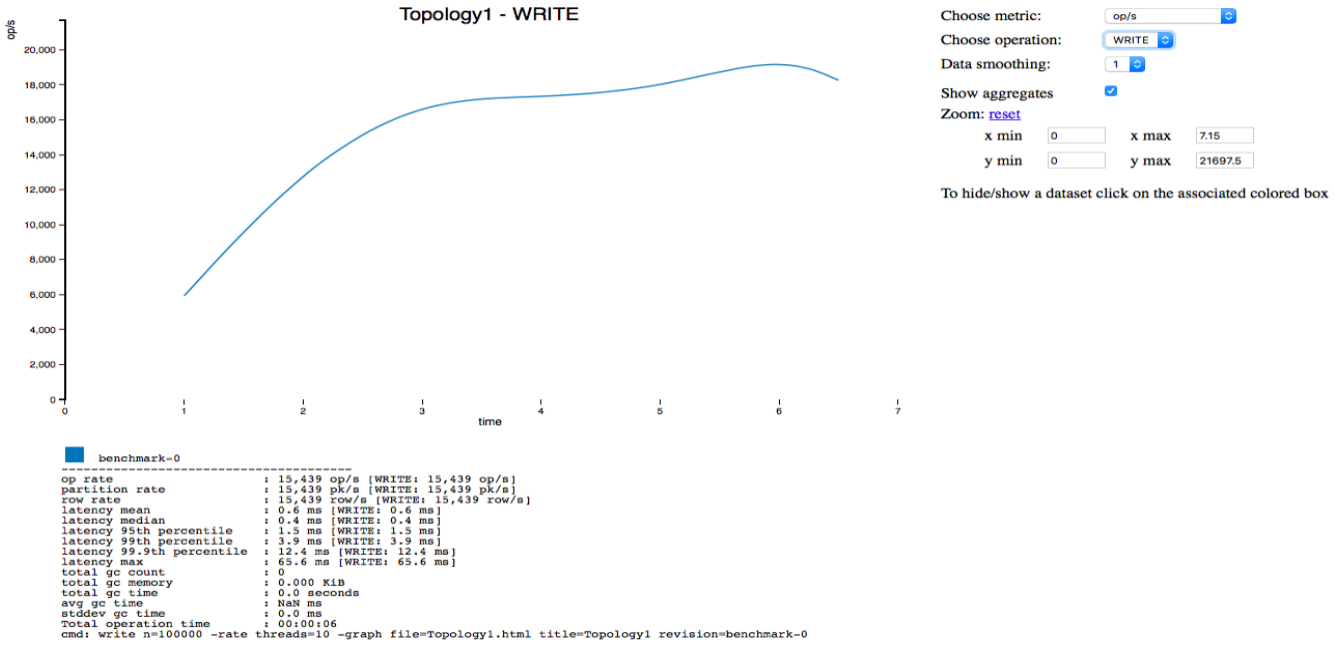


Fig. 1.2 Write Latency

Fig 1.2 Shows the write Latency graph for Topology 1

By observing the above, we can state that the write Latency of 99 percentile for write operation is 3.9 ms which means the write operation is taking of 3.9 ms to perform its operations.

b) Graph of Read Latency:

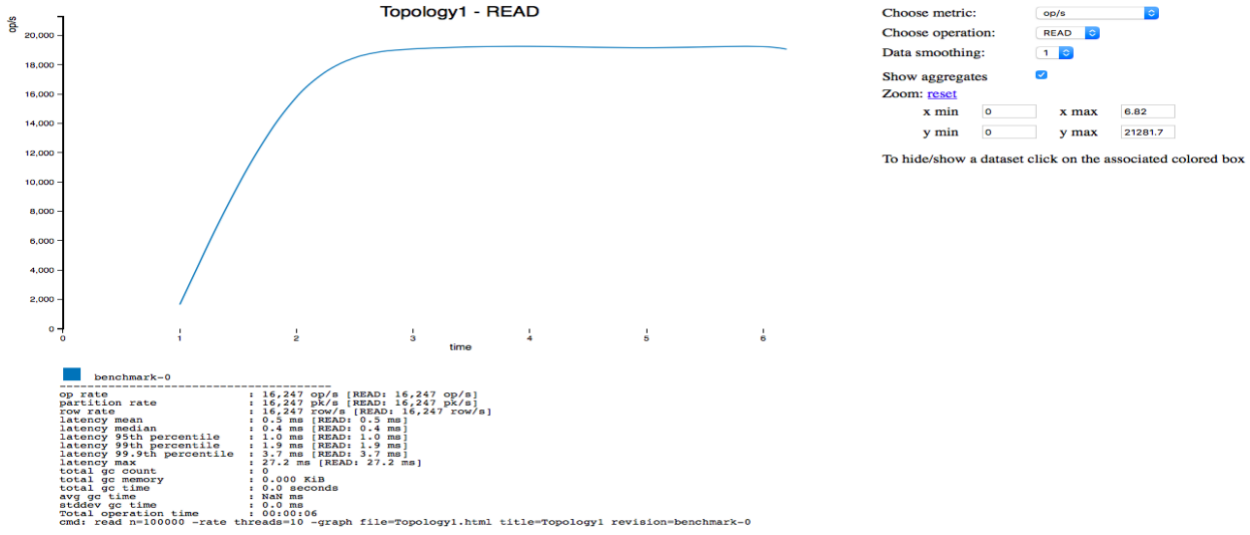


Fig. 1.3 Read Latency

Fig 1.3 Shows the read Latency graph for Topology 1

By observing the above, we can state that the read Latency of 99 percentile for read operation is 1.9 ms which means the read operation is taking of 1.9 ms to perform its operations.

When we compare the read and write 99 percentile latencies read latency is less when compared to write latency i.e the amount of time it is taking for read to perform it's operations is less when compared to write operations.

In addition to the latency 99 th percentile for read and write we can observe latency mean, latency 95th percentile and latency max and 99.99th percentile. We mainly interested in these percentiles because we want to observe the trend as the operations proceed further.

c) Mixed Latency:

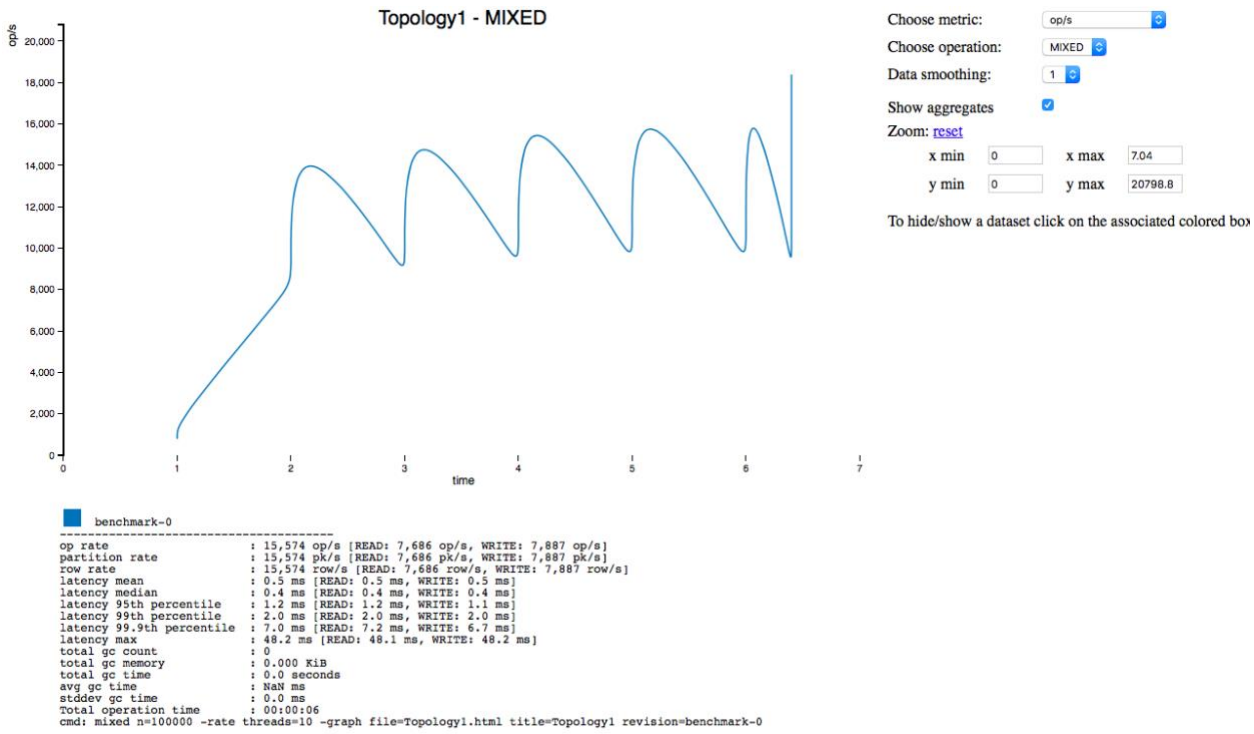


Fig. 1.4 Mixed Latency

Fig 1.4 Shows Mixed Latency graph for Topology 1

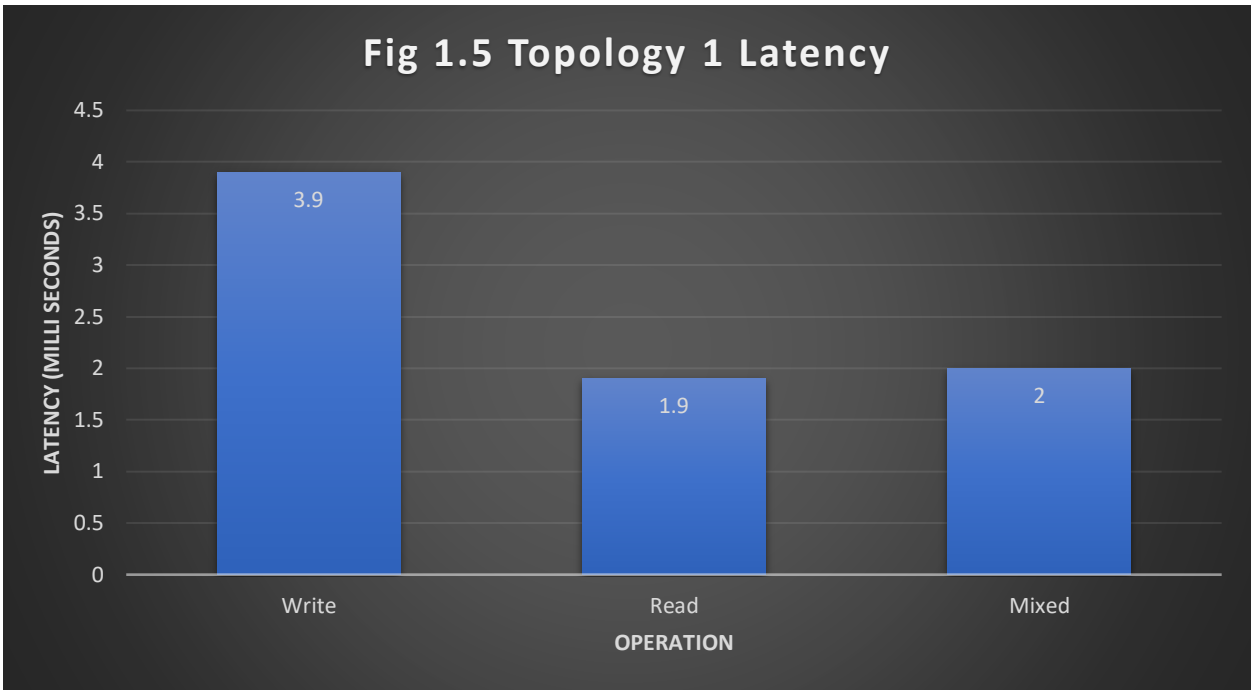


Fig 1.5 Shows Latencies graph for Topology 1

b) Measurements of Clock Skew:

a) Round Trip delay:

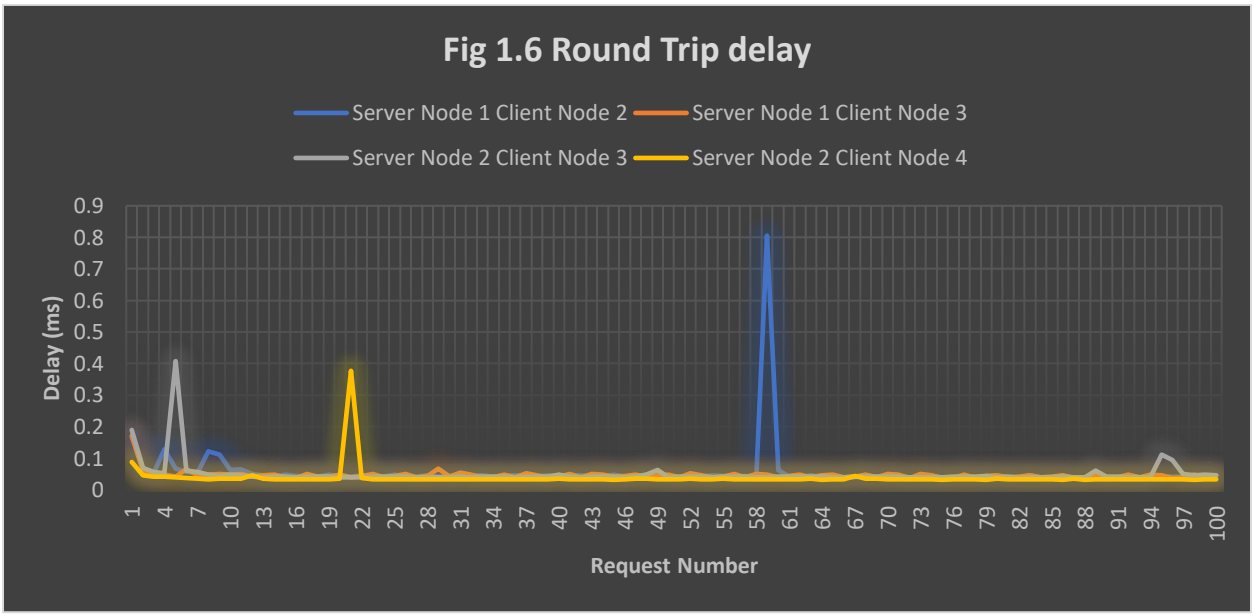


Fig 1.6 Shows the Round-Trip Delay graph for Topology 1

b) Time Offset:

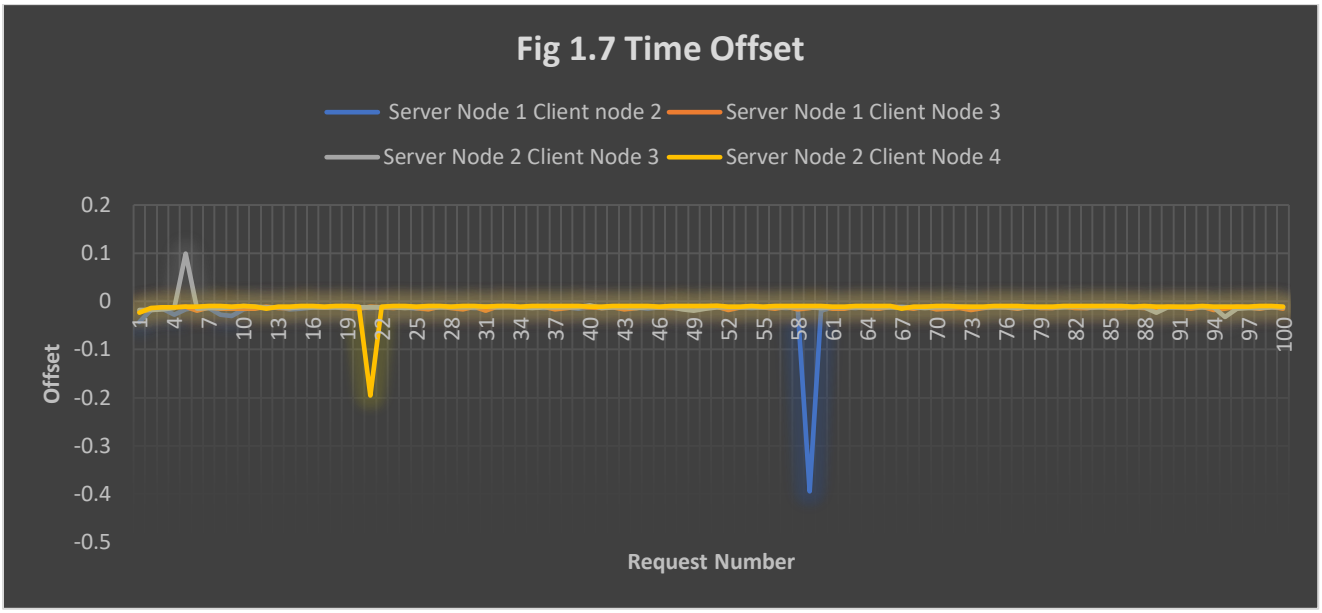


Fig 1.7 Shows the Time offset graph for Topology 1

Topology 2:

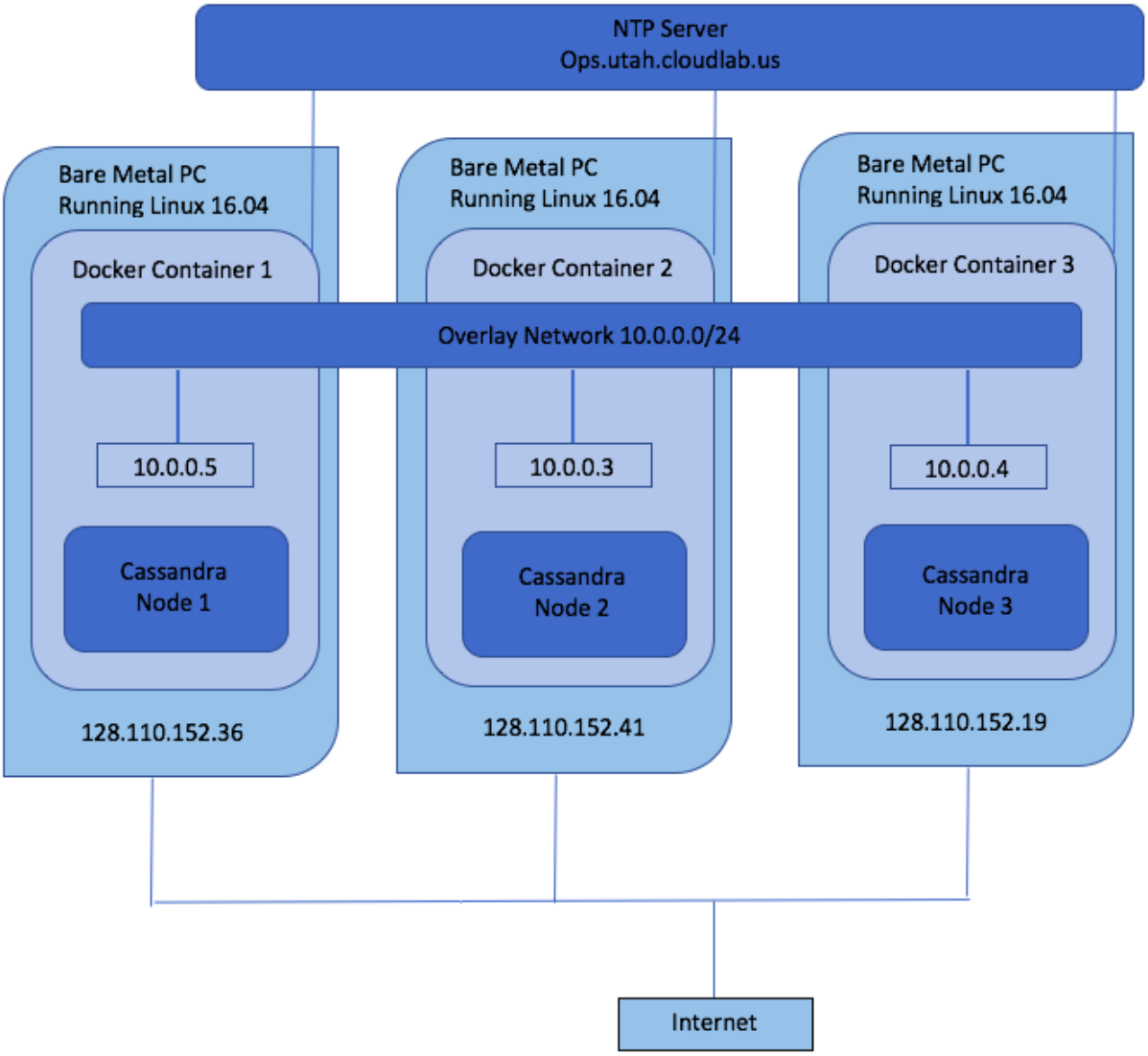


Fig. 2.1 Topology

Topology 2 consists of 3 different Virtual Machines which are Bare Metal PC Running Linux 16.04 LTS each are in same Utah location. Inside each of the three Virtual Machines docker container was installed and inside each docker container corresponding Cassandra node was installed and running. In order to Connect all the three different containers an Overlay network was created in docker Swarm mode.

In order to synchronize the time all, the docker containers are connected to the NTP Server ops.utah.cloudlab.us and we verified the synchronization by checking the ping. We Connect to this ops.utah.clouldlab.us NTP Server because we were unable to implement our own NTP Server

in the docker because it is not allowing us to add/edit the configuration file particularly SYS_TIME operation.

Corresponding measurements of Read and Write Latency is measured using the Cassandra-stress command with respective read and write arguments, Clock Skew was measured using the provided server and client scripts.

Measurements of Latency:

a) Graph of Write Latency:

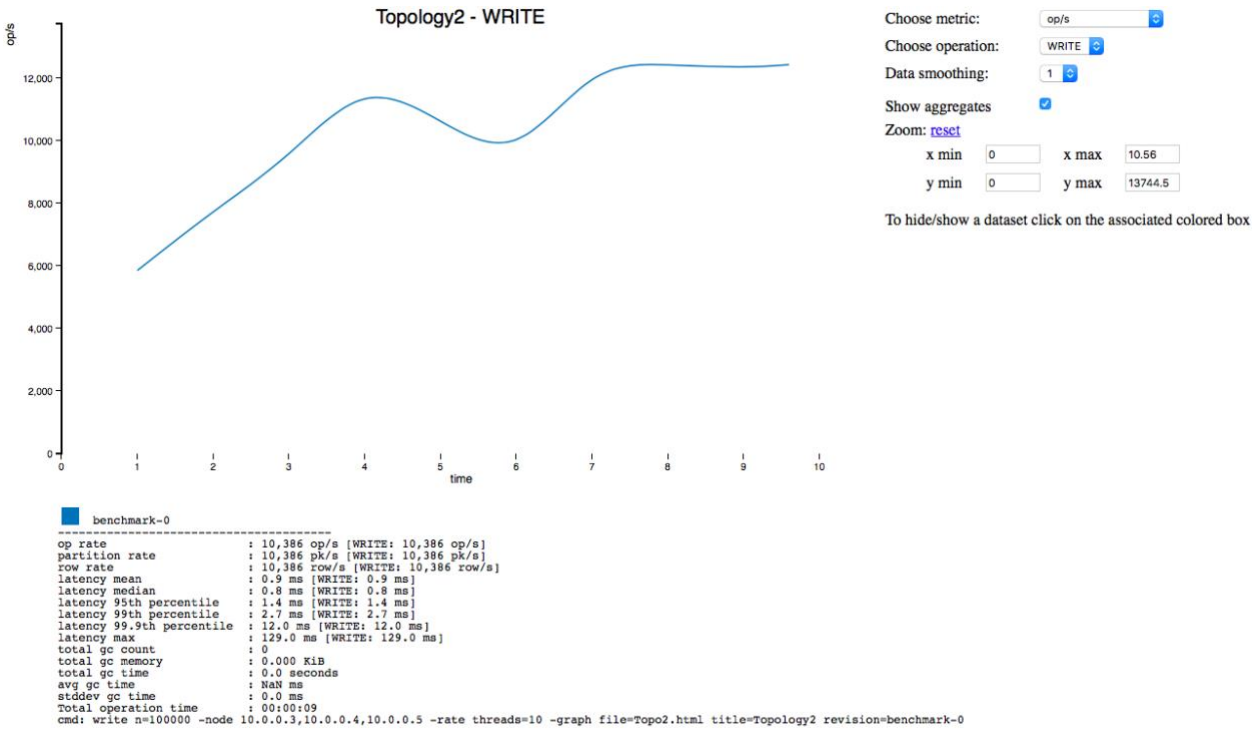
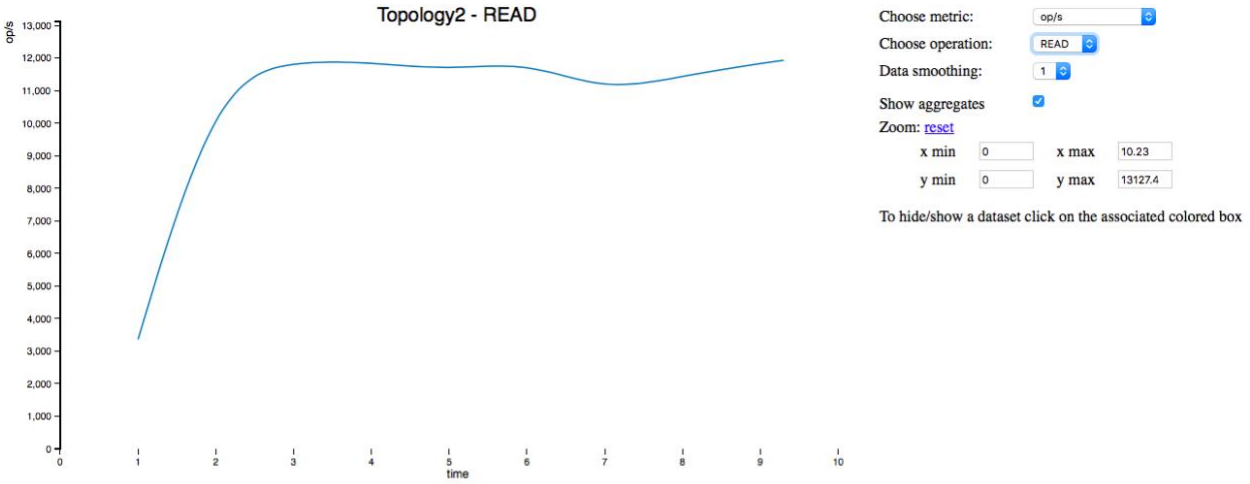


Fig. 2.2 Write Latency

Fig 2.2 Shows the Write Latency Latency graph for Topology 2

By observing the above, we can state that the write Latency of 99 percentile for write operation is 2.7 ms which means the write operation is taking of 2.7 ms to perform its operations.

b) Graph of Read Latency:




```
benchmark-0
----------------------------------------
op rate                   : 10,734 op/s [READ: 10,734 op/s]
partition rate            : 10,734 pk/s [READ: 10,734 pk/s]
row rate                  : 10,734 row/s [READ: 10,734 row/s]
latency mean              : 0.8 ms [READ: 0.8 ms]
latency median            : 0.8 ms [READ: 0.8 ms]
latency 95th percentile   : 1.2 ms [READ: 1.2 ms]
latency 99th percentile   : 1.5 ms [READ: 1.5 ms]
latency 99.9th percentile : 6.2 ms [READ: 6.2 ms]
latency max               : 74.6 ms [READ: 74.6 ms]
total gc count            : 0
total gc memory           : 0.000 KiB
total gc time             : 0.0 seconds
avg gc time               : NaN ms
stddev gc time            : 0.0 ms
Total operation time      : 00:00:09
cmd: read n=100000 -node 10.0.0.3,10.0.0.4,10.0.0.5 -rate threads=10 -graph file=Topo2.html title=Topology2 revision=benchmark-0
```

Fig. 2.3 Read Latency

Fig 2.3 Shows the read Latency graph for Topology 2

By observing the above, we can state that the read Latency of 99 percentile for read operation is 1.5 ms which means the read operation is taking of 1.5 ms to perform its operations.

When we compare the read and write 99 percentile latencies read latency is less when compared to write latency i.e the amount of time it is taking for read to perform it's operations is less when compared to write operation of data.

In addition to the latency 99 th percentile for read and write we can observe latency mean, latency 95th percentile and latency max and 99.99th percentile. We mainly interested in these percentiles because we want to observe the trend as the operations proceed further.

Mixed Latency:

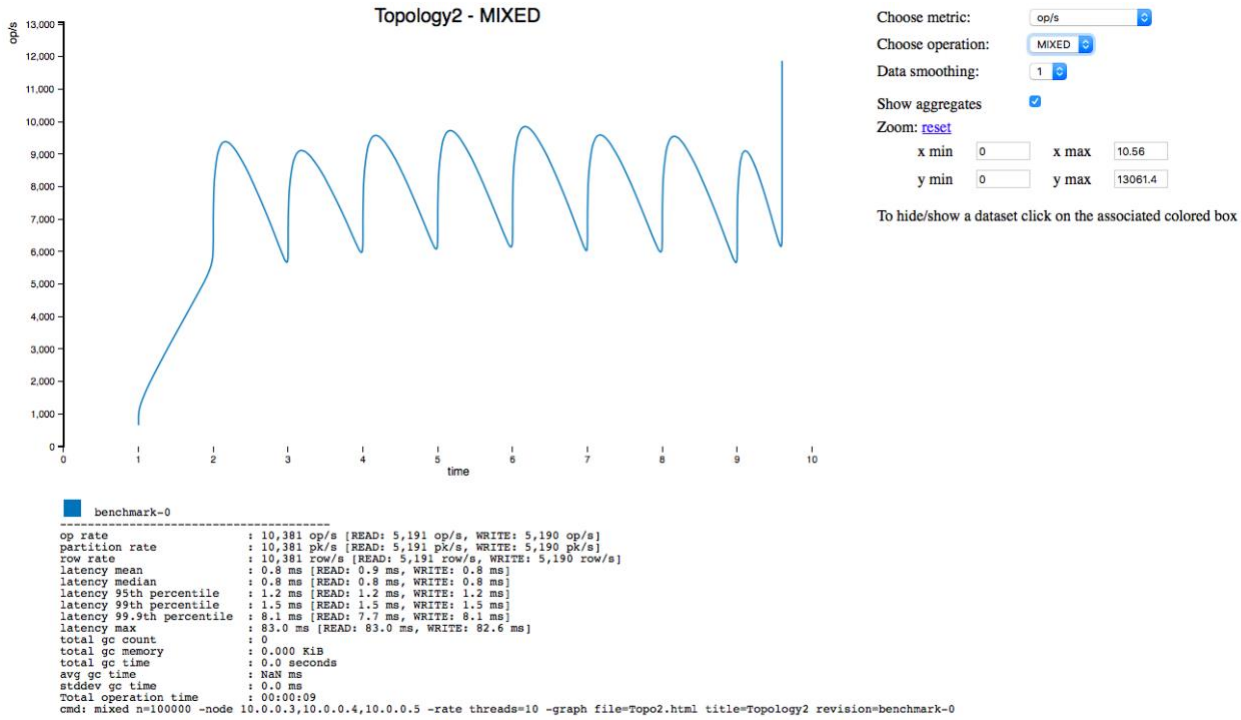


Fig. 2.4 Mixed Latency

Fig 2.4 Shows the mixed Latency graph for Topology 2

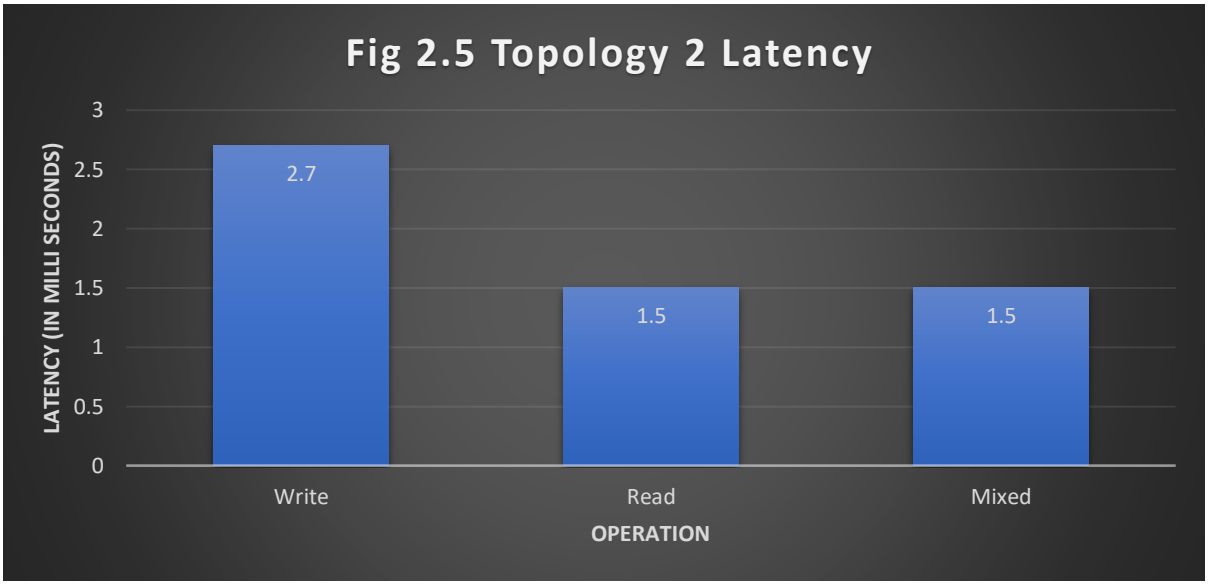


Fig 2.5 Shows the Latencies graph for Topology 2

Measurements of Clock Skew:

a) Round Trip delay:

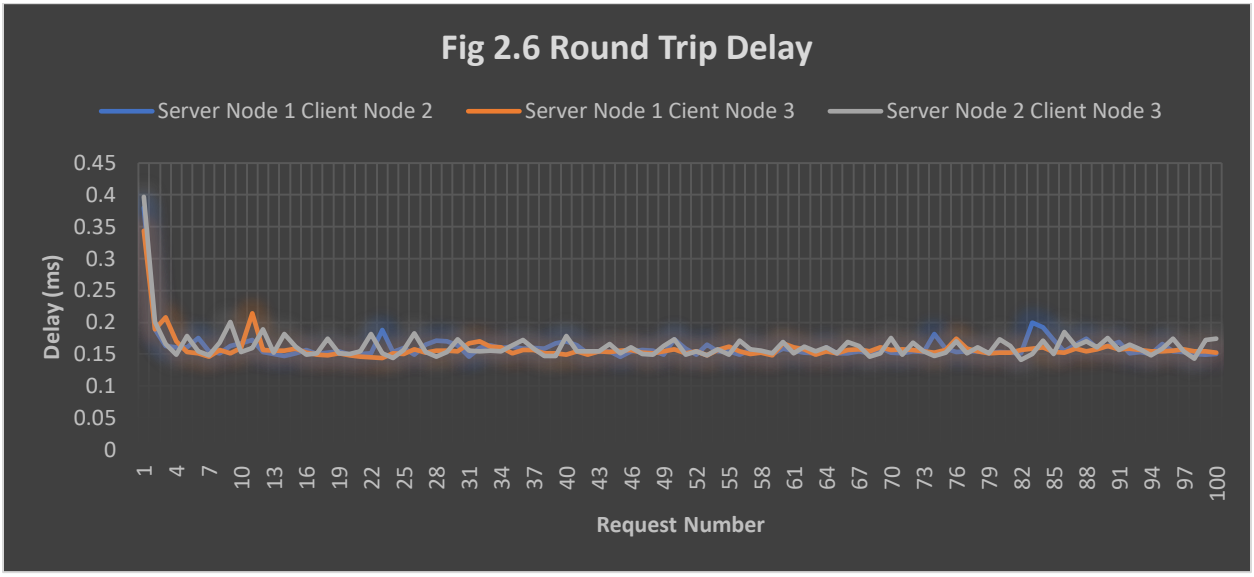


Fig 2.6 Shows the Round-trip delay graph for Topology 2

b) Time Offset:

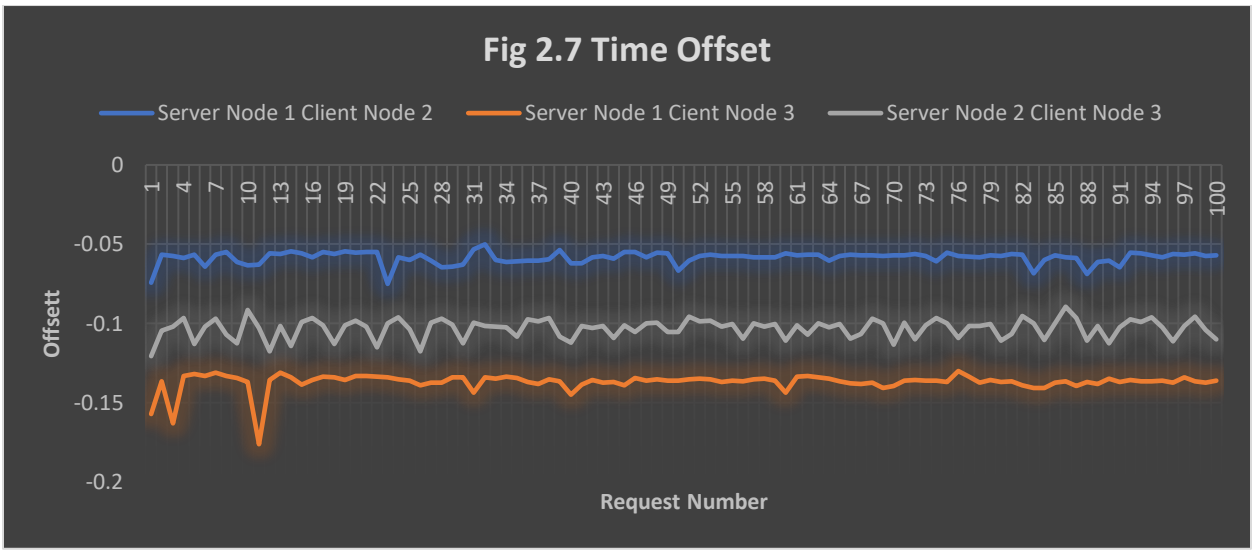


Fig 2.7 Shows the Time Offset graph for Topology 2

Topology 3:

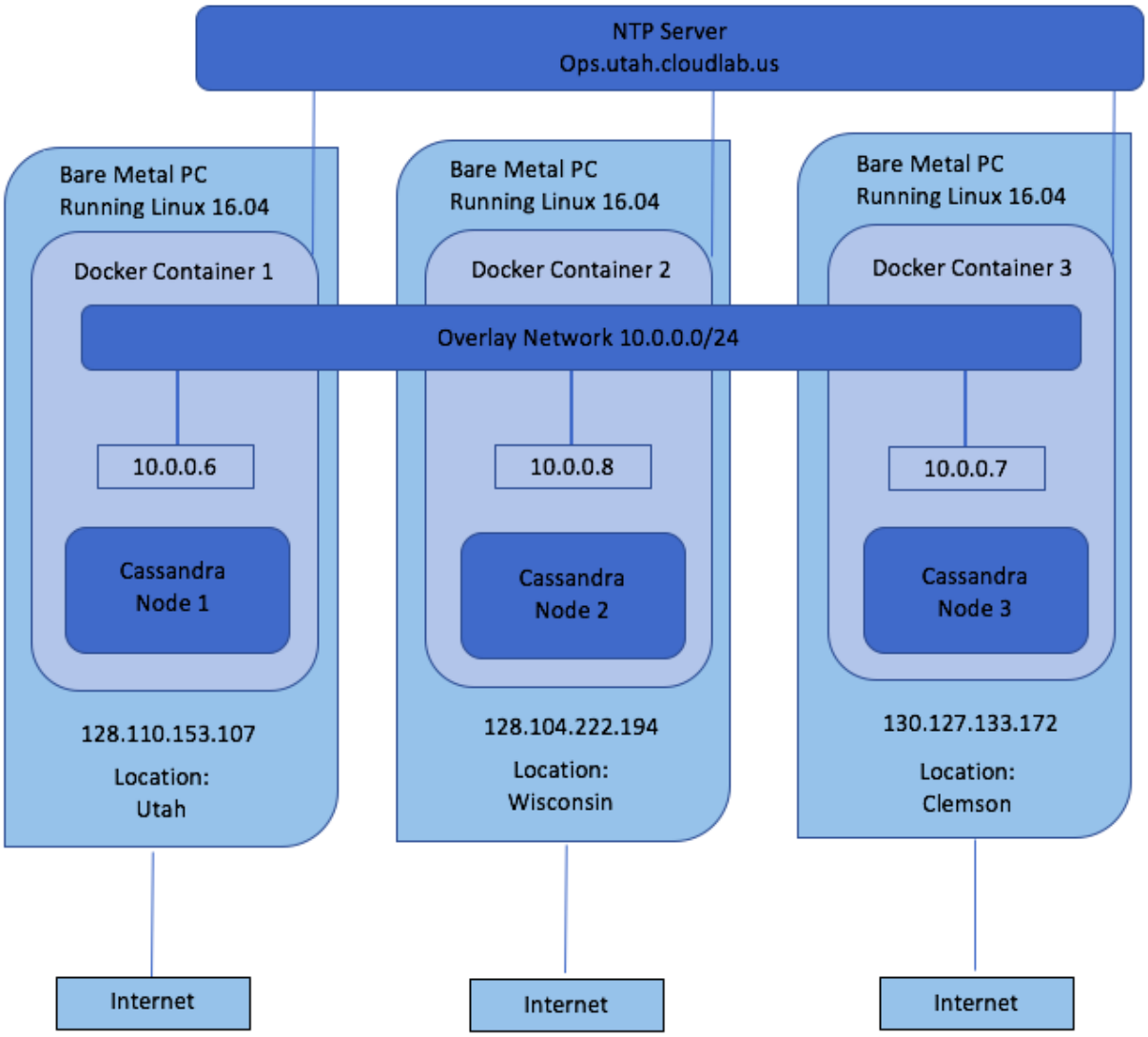


Fig. 3.1 Topology

Topology 3 consists of 3 different Virtual Machines which are Bare Metal PC Running Linux 16.04 LTS located in Utah, Wisconsin, Clemson and each are connected to its own internet. Inside each of the three Virtual Machines docker container was installed and inside each docker container corresponding Cassandra node was installed and running. In order to Connect all the three different containers an Overlay network was created in docker Swarm mode.

In order to synchronize the time all, the docker containers are connected to the NTP Server ops.utah.cloudlab.us and we verified the synchronization by checking the ping. We Connect to this ops.utah.clouldlab.us NTP Server because we were unable to implement our own NTP Server

in the docker because it is not allowing us to add/edit the configuration file particularly SYS_TIME operation.

Corresponding measurements of Read and Write Latency is measured using the Cassandra-stress command with respective read and write arguments, Clock Skew was measured using the provided server and client scripts.

Measurements of Latency:

a) Graph of Write Latency:

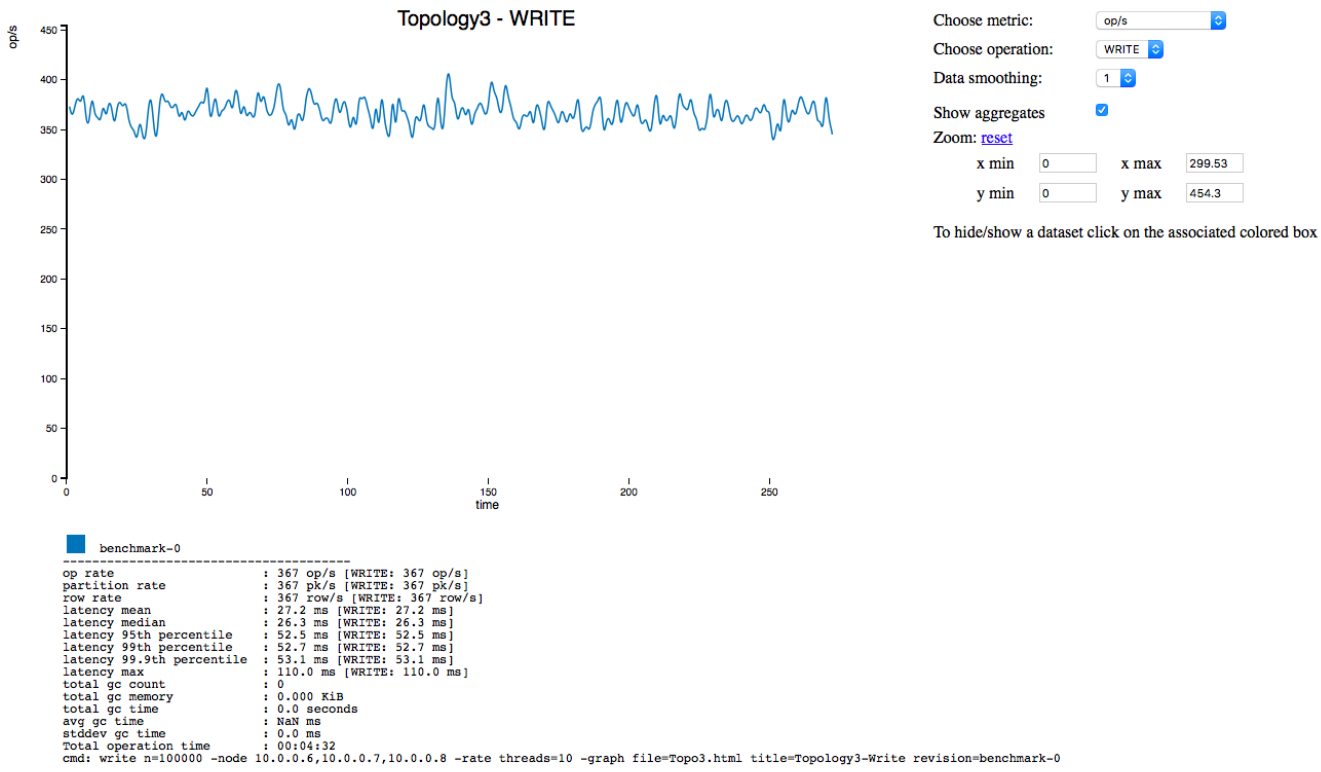


Fig. 3.2 Write Latency

Fig 3.2 Shows the write Latency graph for Topology 3

By observing the above, we can state that the write Latency of 99 percentile for write operation is 52.7 ms which means the write operation is taking of 52.7 ms to perform its operations.

b) Graph of Read Latency:

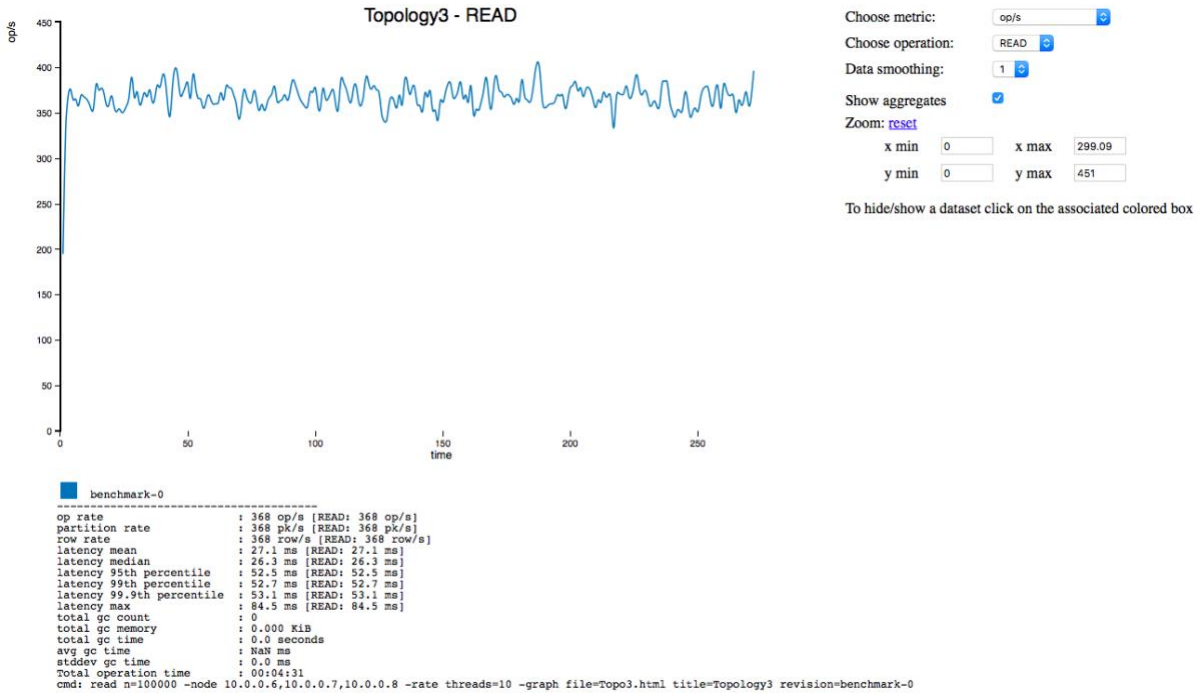


Fig. 3.3 Write Latency

Fig 3.3 Shows the read Latency graph for Topology 3

By observing the above, we can state that the read Latency of 99 percentile for read operation is 52.7 ms which means the read operation is taking of 52.7 ms to perform its operations.

When we compare the read and write 99 percentile latencies read latency and write latency are i.e the amount of time it is taking for read to perform it's operations are equal when compared to write operations of data.

In addition to the latency 99 th percentile for read and write we can observe latency mean, latency 95th percentile and latency max and 99.99th percentile. We mainly interested in these percentiles because we want to observe the trend as the operations proceed further.

c) Mixed Latency:

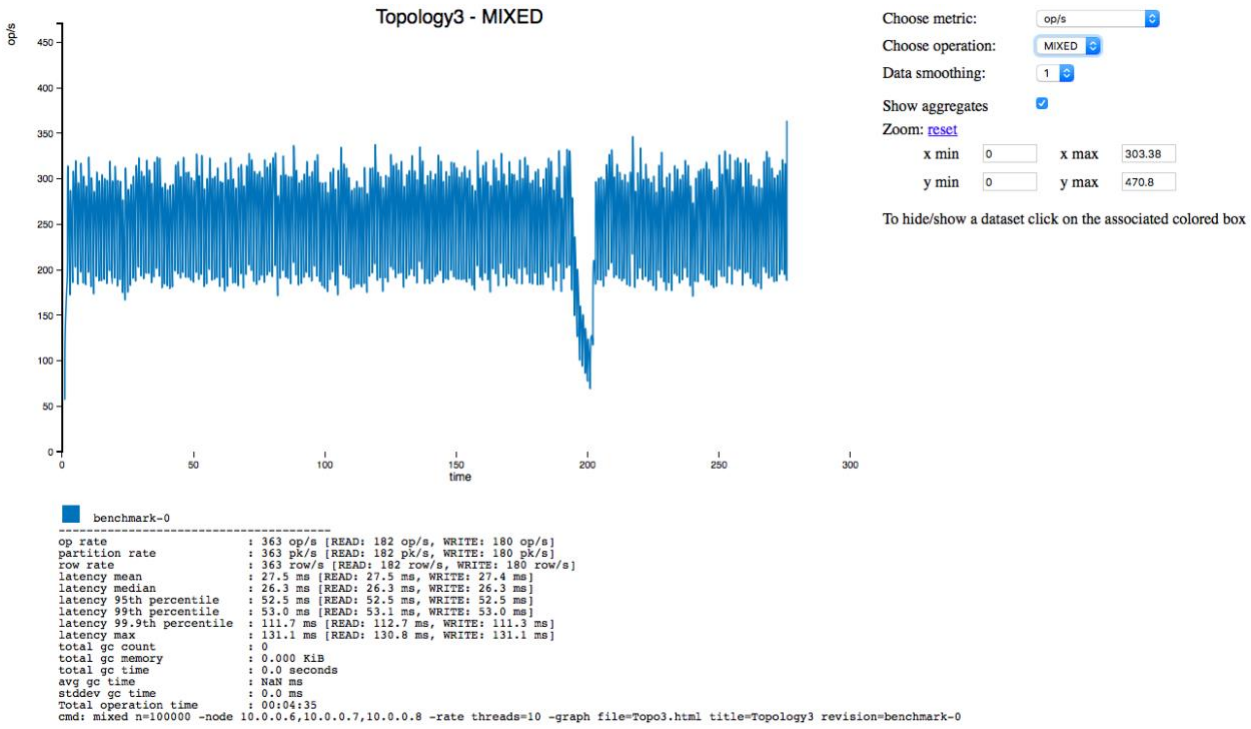


Fig 3.4 Mixed Latency

Fig 3.4 Shows the Mixed Latency graph for Topology 3

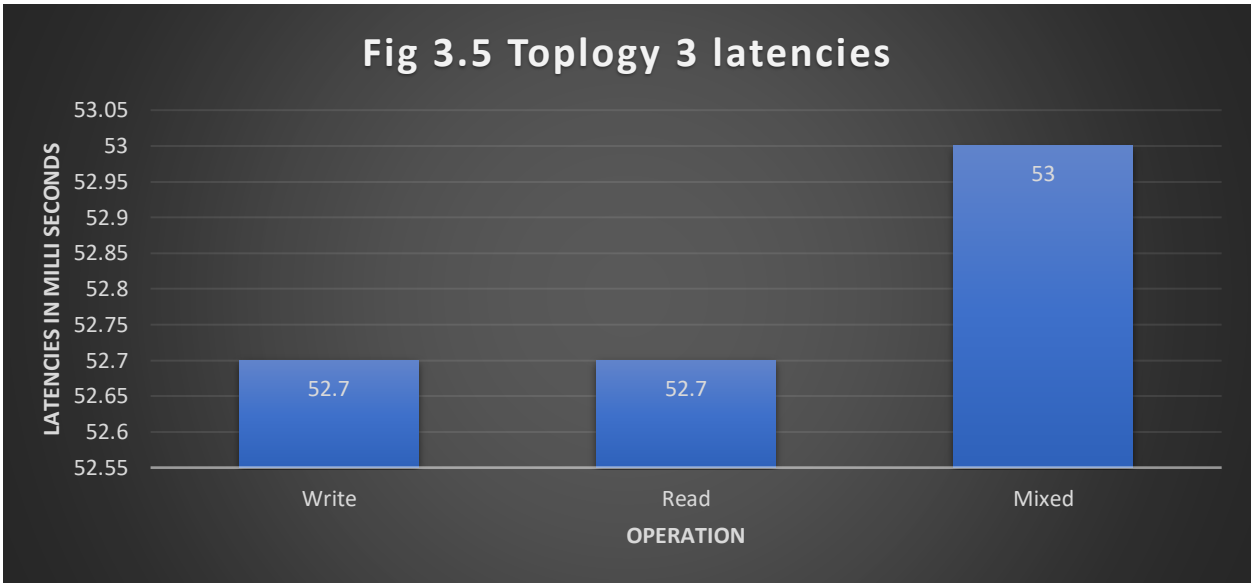


Fig 3.5 Shows the Latencies graph for Topology

Measurements of Clock Skew:

a) Round Trip delay:

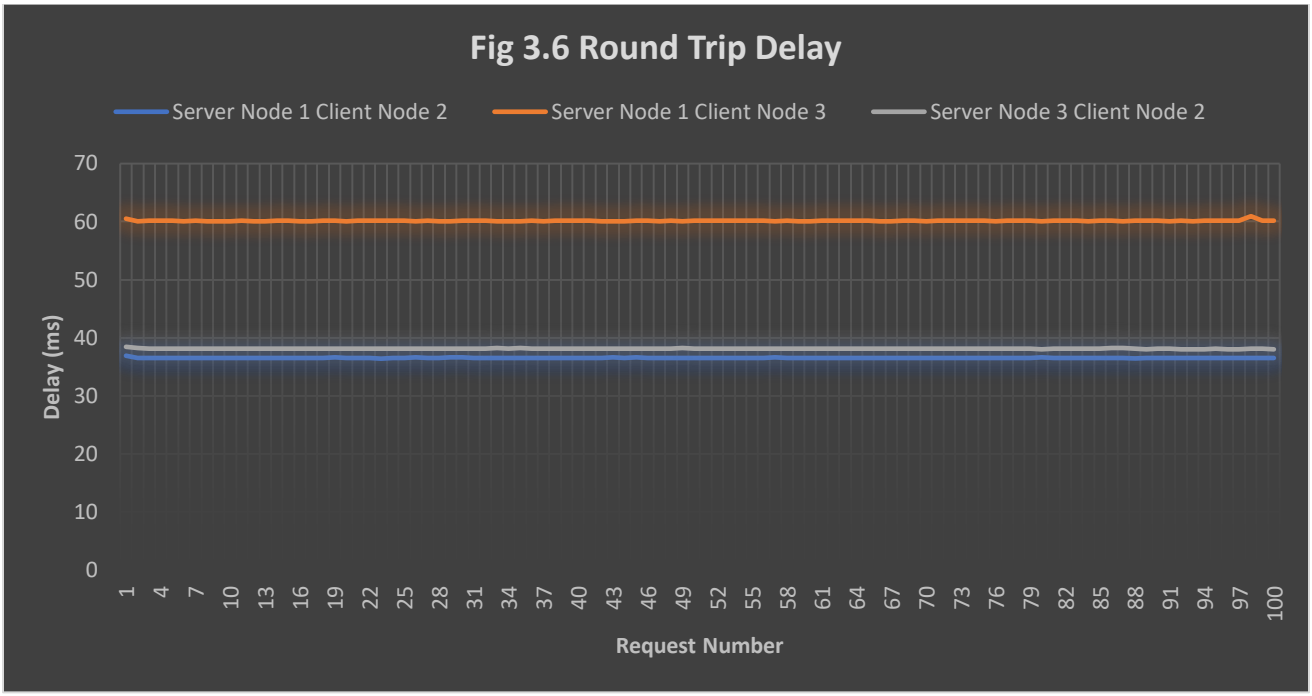


Fig 3.6 Shows the Round-trip delay graph for Topology 3

b) Time Offset:

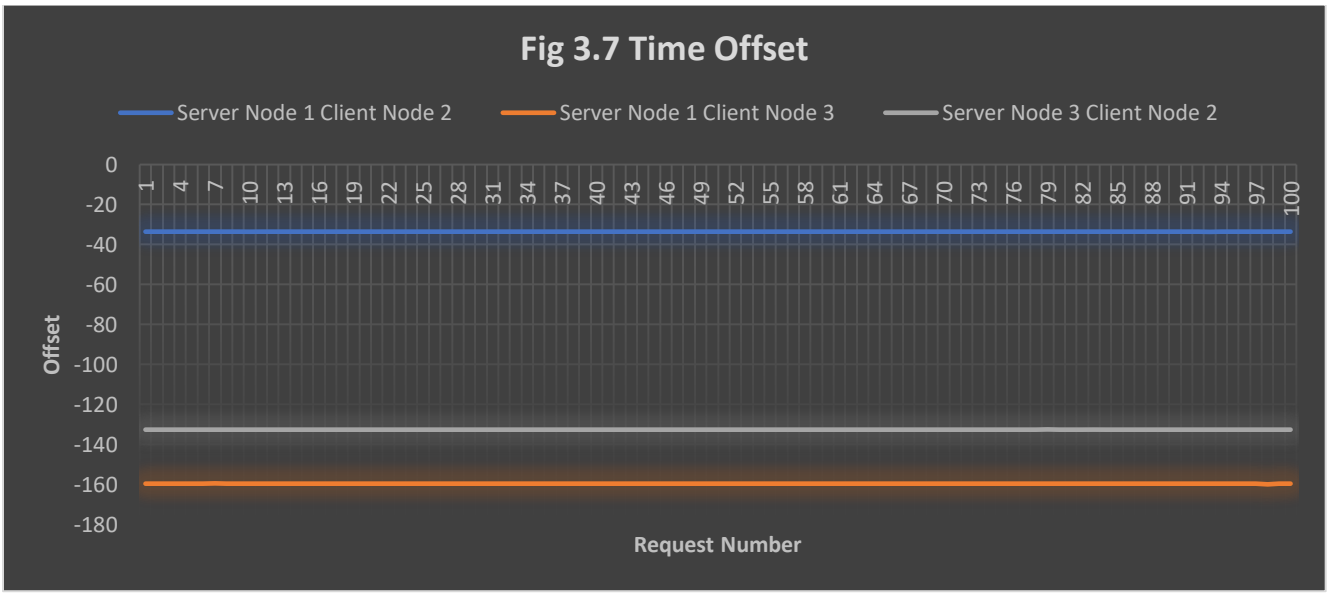
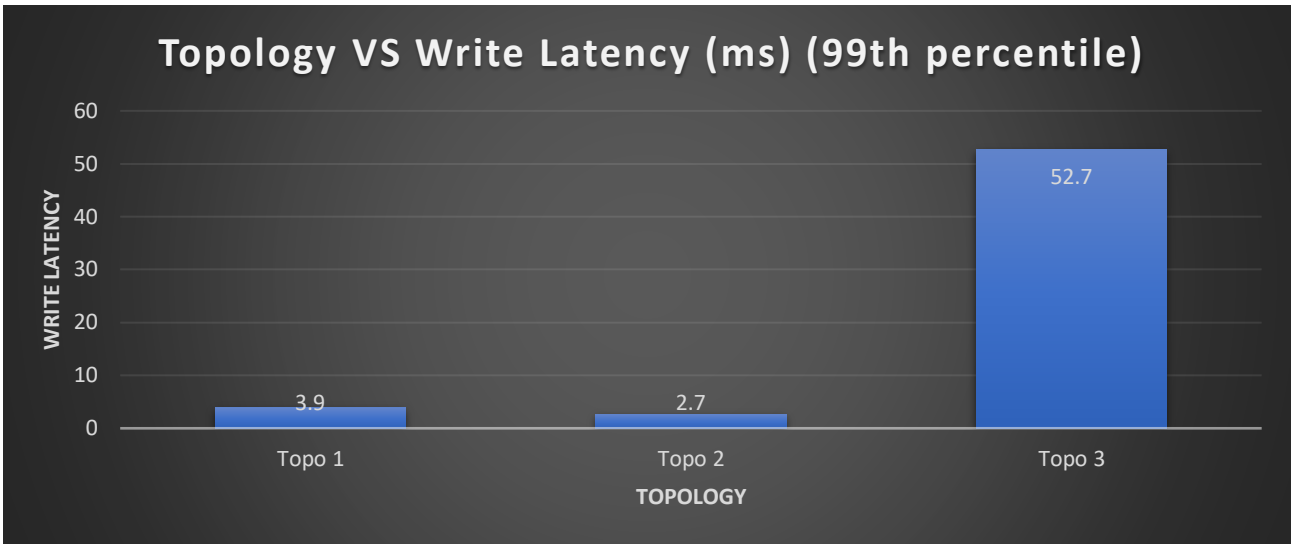


Fig 3.7 Shows the Time Offset graph for Topology 3

Comparison and Conclusion:

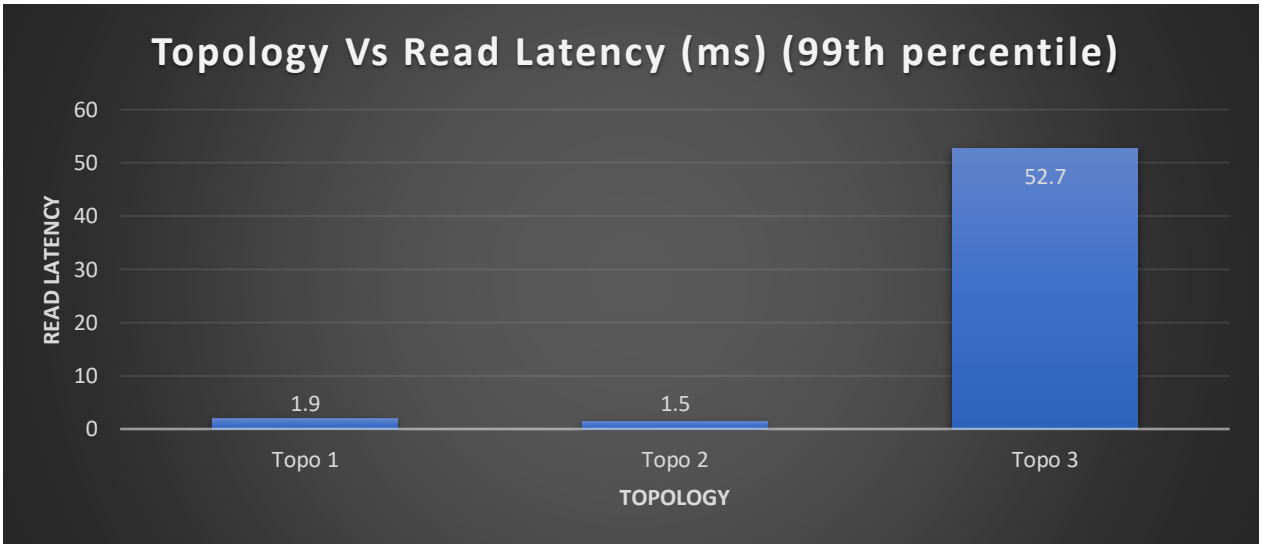
1) *Latency Comparisons:*

a) *Topology vs Write Latency:*



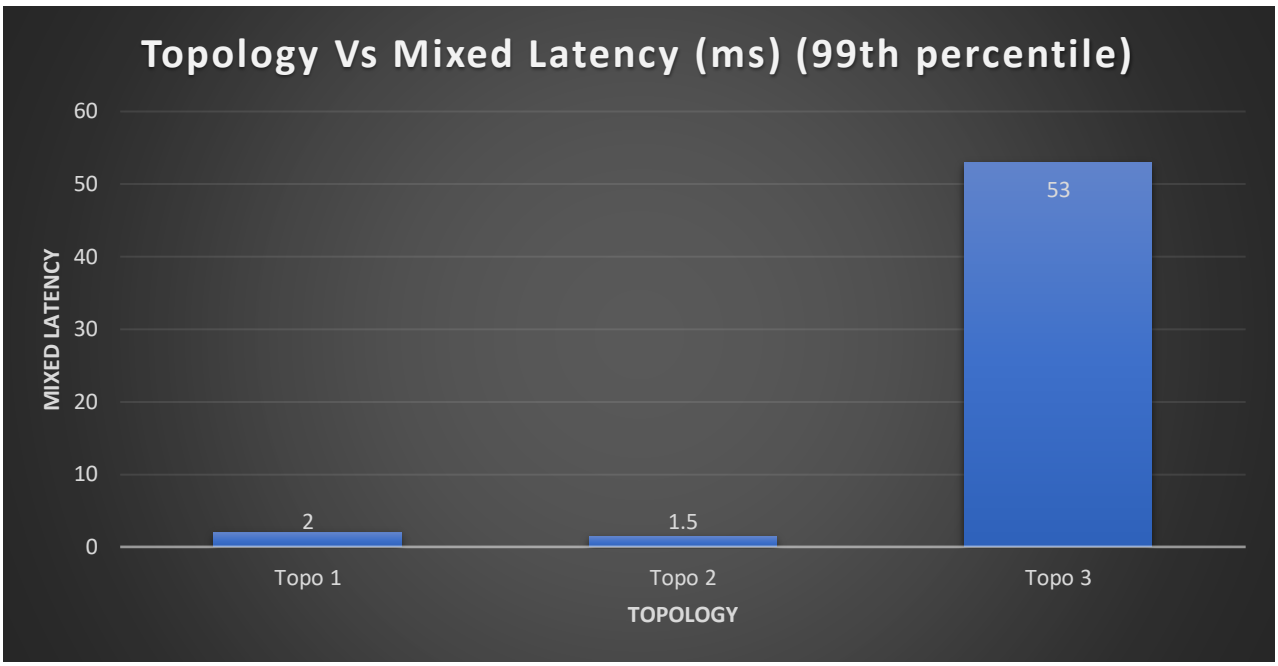
When we consider the write Latency Topology 3 has higher when compared to remaining topologies this is due to because the distance between Topology 3 nodes were significantly larger distance when compared to other two topology nodes distances. The time difference between write operations for topology 3 was more when compared to other nodes.

b) *Topology vs Read Latency:*



When we consider the read Latency Topology 3 has higher when compared to remaining topologies this is due to because the distance between Topology 3 nodes were significantly larger distance when compared to other two topology nodes distances. The time difference between read operations for topology 3 was more when compared to other nodes.

c) Mixed Latency:



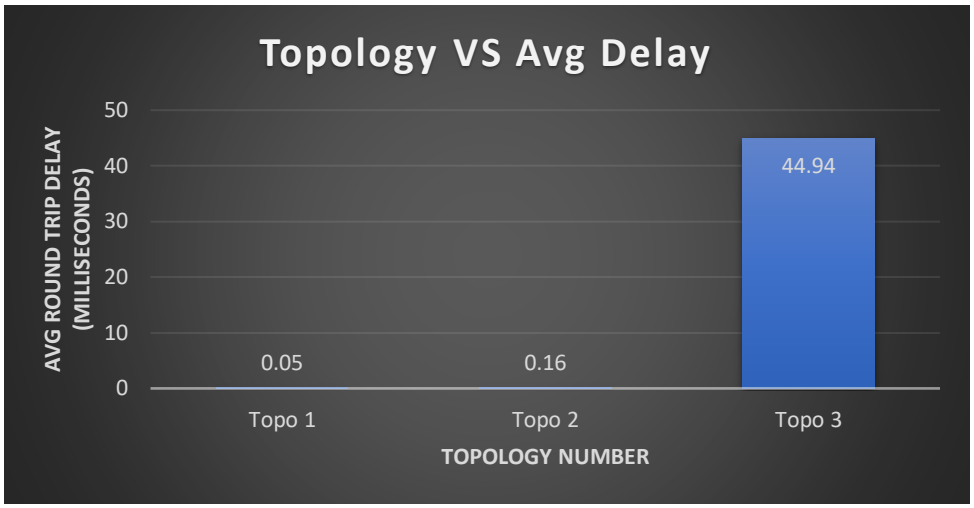
When we consider the Mixed Latency (Half will be reads and other half will be writes) Topology 3 has higher when compared to remaining topologies this is due to because the distance between Topology 3 nodes were significantly larger distance when compared to other two topology nodes distances.

So, when we consider as a whole Latency operation (include all types of Latencies) Topology 3 has higher latency than the other two topologies because the distance between Topology 3 nodes were significantly larger distance when compared to other two topology nodes. Distances.

When we consider we observe in topology 2 latency was more when compared to topology 1. This is because topology 1 has 4 nodes whereas topology 2 has 3 nodes.

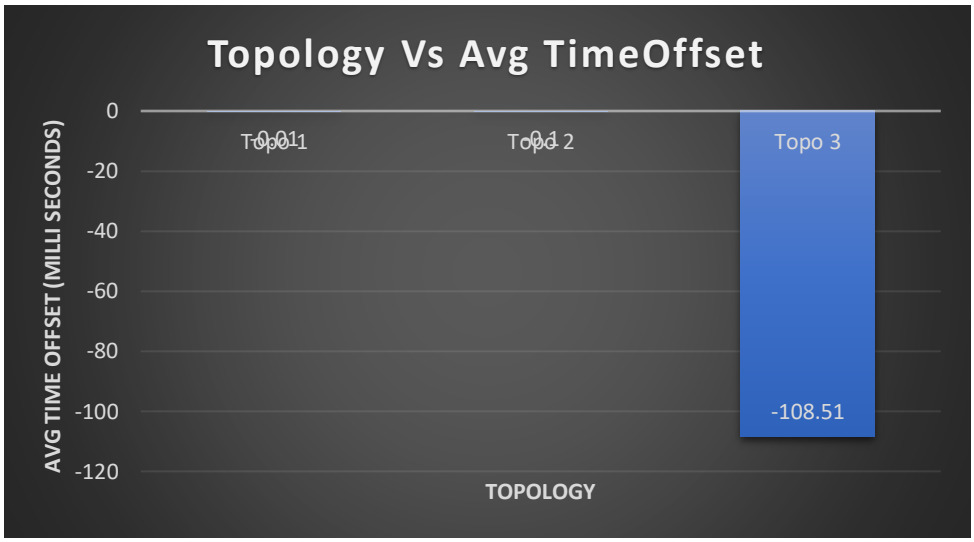
2) Clock Skew Comparisons:

a) Average Delay:



When we consider the Average Delay Topology 3 has higher when compared to remaining topologies this is due to because the distance between Topology 3 nodes were significantly larger distance when compared to other two topology nodes distances. So, in topology 3 the time taken to send/receive data between nodes will be more.

b) Average Time Offset:



When we consider the Average Time offset (considering magnitude) Topology 3 has higher when compared to remaining topologies this is due to because the distance between Topology 3 nodes were significantly larger distance when compared to other two topology nodes distances.

Conclusion:

So, When, we observe both the latency and clock skew measurements. Topology 3 has more value in measurements rather than different topologies because topology 3 nodes are geographically longer than the other topologies nodes.