PERFORMANCE EVALUATION

DE-CENTRALIZED P2P SYSTEM

ASSIGNMENT 3

DESIGNED BY:

SAMI AHMAD KHAN
A20352677
COURSE: CS 550 ADVANCE OPERATING SYSTEM

PERFORMANCE EVALUATION REPORT

The performance evaluation was made using the following steps:

- Firstly time assessment is done on various factors like; UPDATE, SEARCH and DOWNLOAD (OBTAIN) operations on Files by the clients. The following commands were used in this process:

  long lStartTime = System.currentTimeMillis();long lEndTime = System.currentTimeMillis();long difference = lEndTime - lStartTime; System.out.println("Elapsed milliseconds" + difference);

FIRST SET OF EXPERIMENTS WITH 10,000 FILES IN EACH CLIENT OF 1Kb SIZE:

A. ON LOCAL MACHINE

The following evaluation was first done on my Local machine's (Apple Macbook Pro) Terminal console:-

- 8 servers was made to run on 8 different host machines and all of them were connected in a Distributed P2P environment.
- Then 8 Clients were brought in, and each client was having **10,000 Files of 1 Kb**
- Then first one Client was performing all the operations and the time was noted.
- Then two clients were running concurrently and performing all the operations and the time was noted.
- Then four clients were running concurrently and performing all the operations and the time was noted.
- Then eight clients were running concurrently and performing all the operations and the time was noted.

The tests were conducted by using timers in the peer application. A timer was started in the code before the execution of each operation and stopped after the operation. The difference was calculated in seconds.
The performance test was conducted on the machines for the following operations,

- **UPDATE:** File names from individual clients were updated in their respective server's distributed hash table and the time was noted.

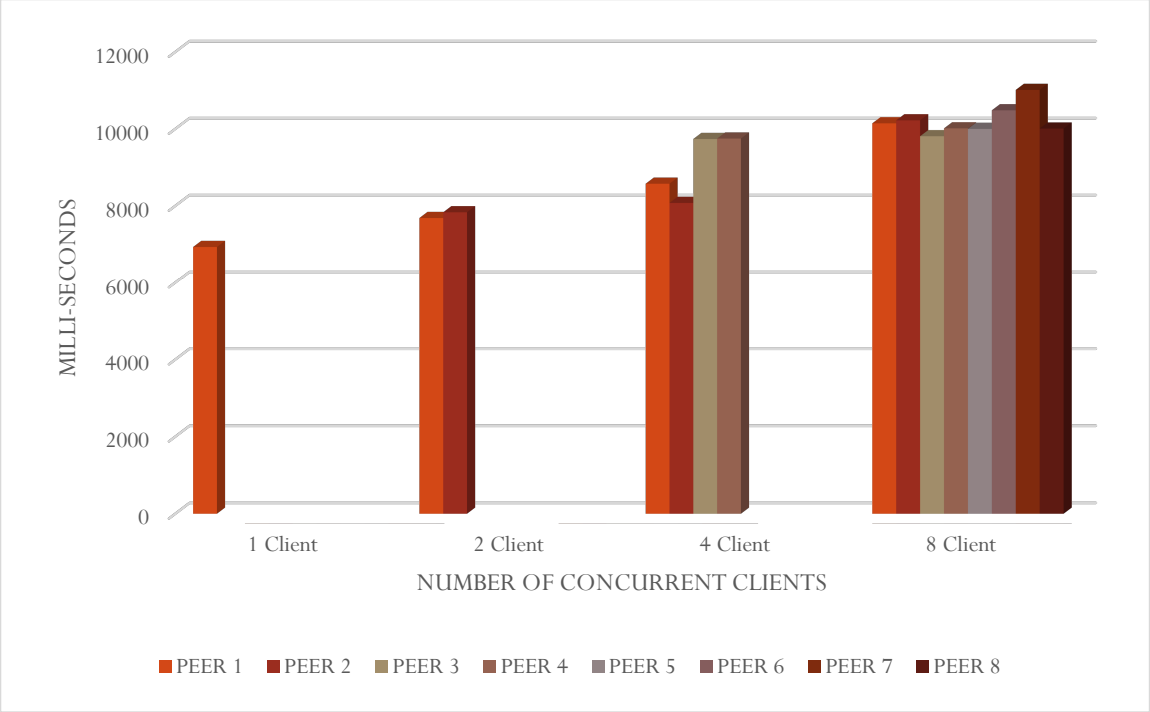
- **SEARCH**: A Client enters a File name it wants to download and it was searched in the distributed network using key.

- **DOWNLOAD/ OBTAIN**: The user selects the Peer it wants to download the file from and the time it takes to download is noted.

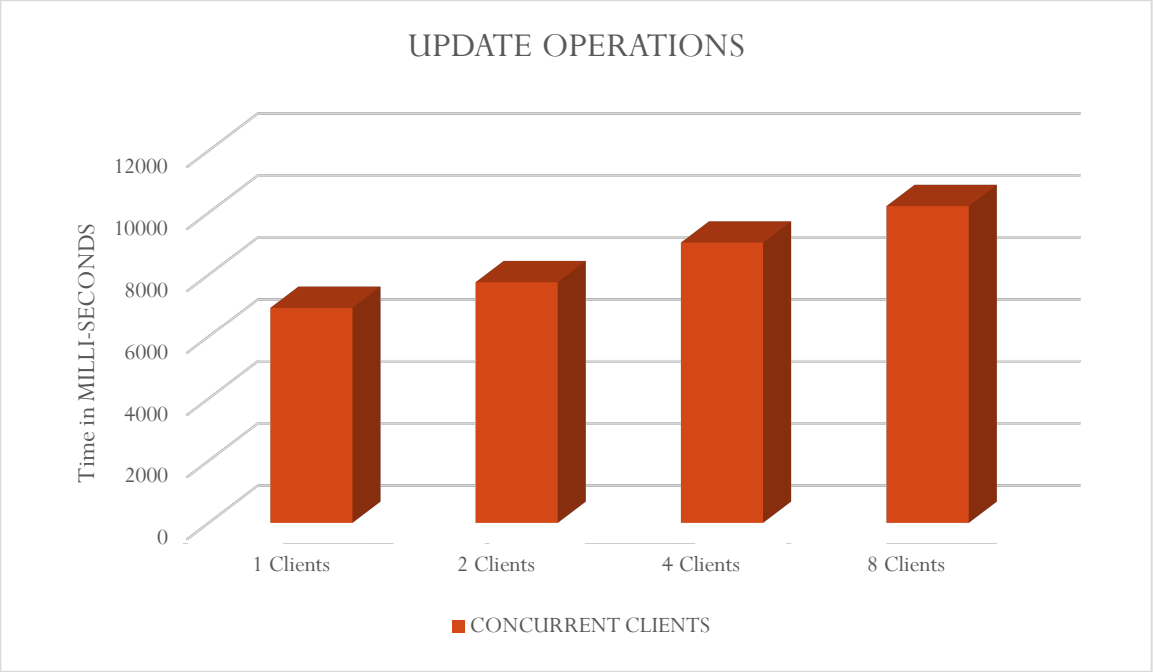
NOTE: The vertical Axis indicates the <u>Time in Seconds</u>
The horizontal Axis indicates <u>the number of Nodes</u> in the cluster.

1. **FOR UPDATING FILES**

a). Individual Values:

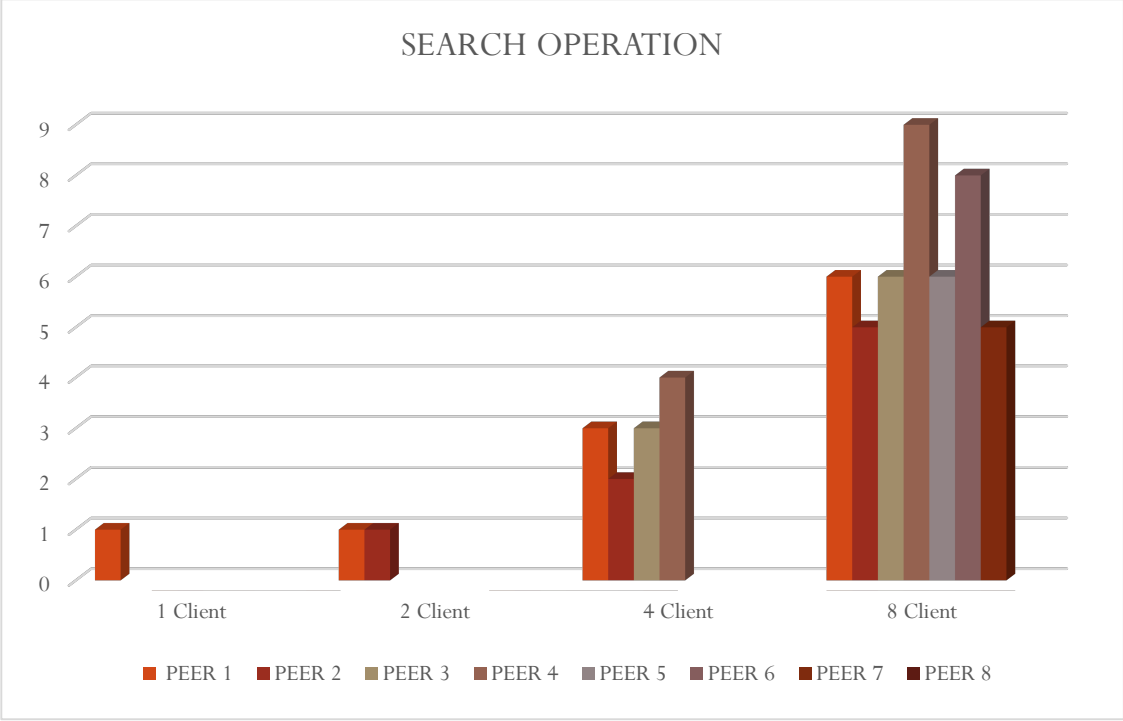


b). Average Values:

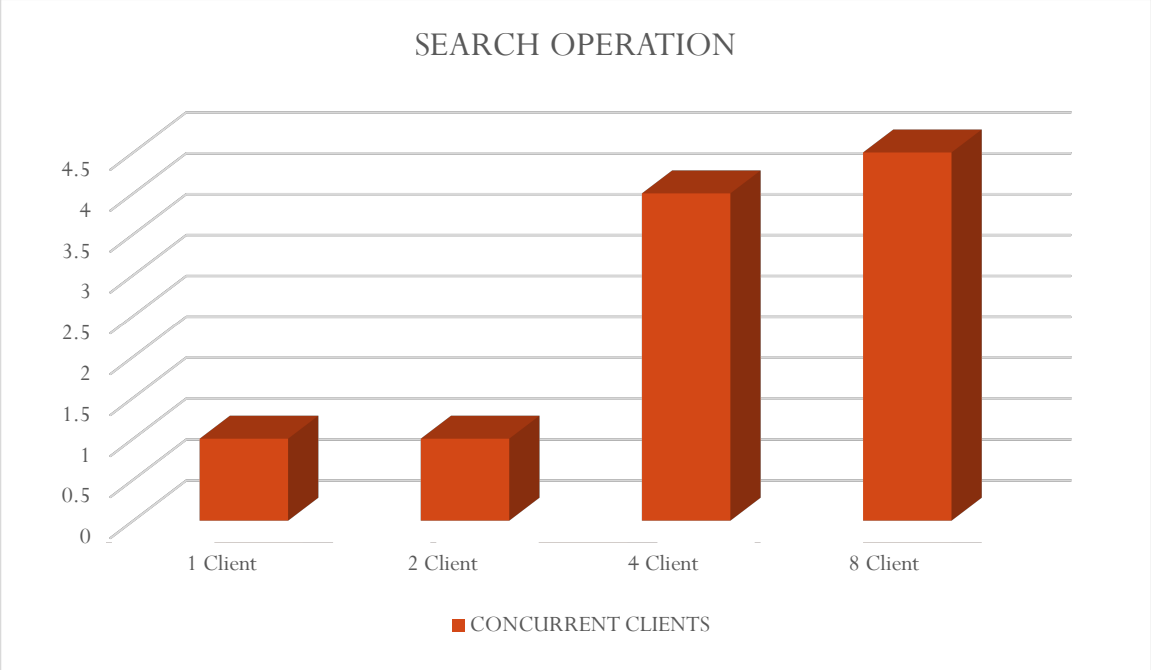


1. **FOR SEARCHING OF FILES**

   a). Individual Values:

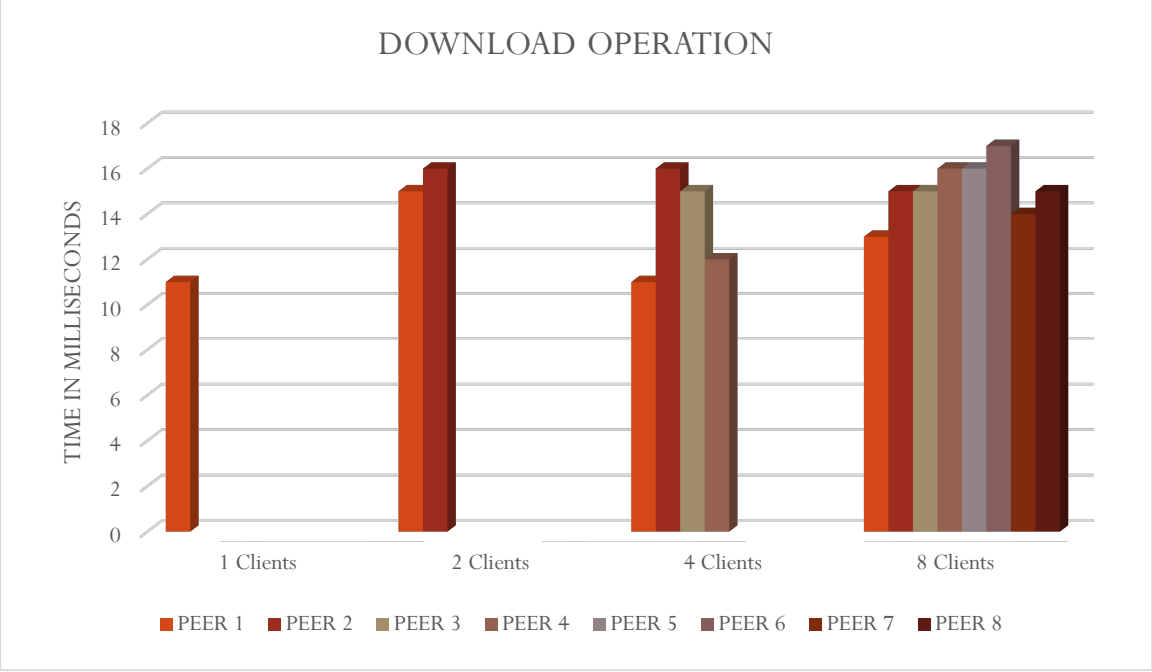


b). Average Values:

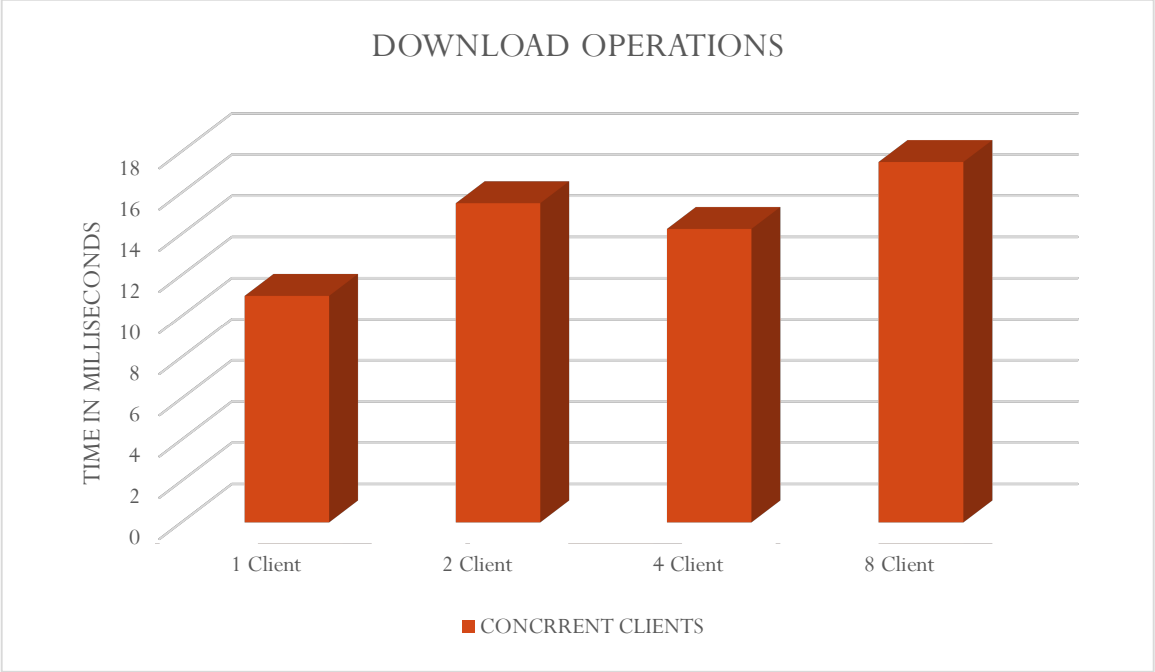


2. FOR DOWNLOADING OF FILES

a). Individual Values:



b). Average Values:



B. IN AMAZON AWS:

Amazon Web Services (AWS), is a collection of remote computing services, also called web services, that make up a cloud-computing platform offered by Amazon.com. These services operate from 11 geographical regions[2] across the world. The most central and well-known of these services arguably include Amazon Elastic Compute Cloud, also known as "EC2", and Amazon Simple Storage Service, also known as "S3".

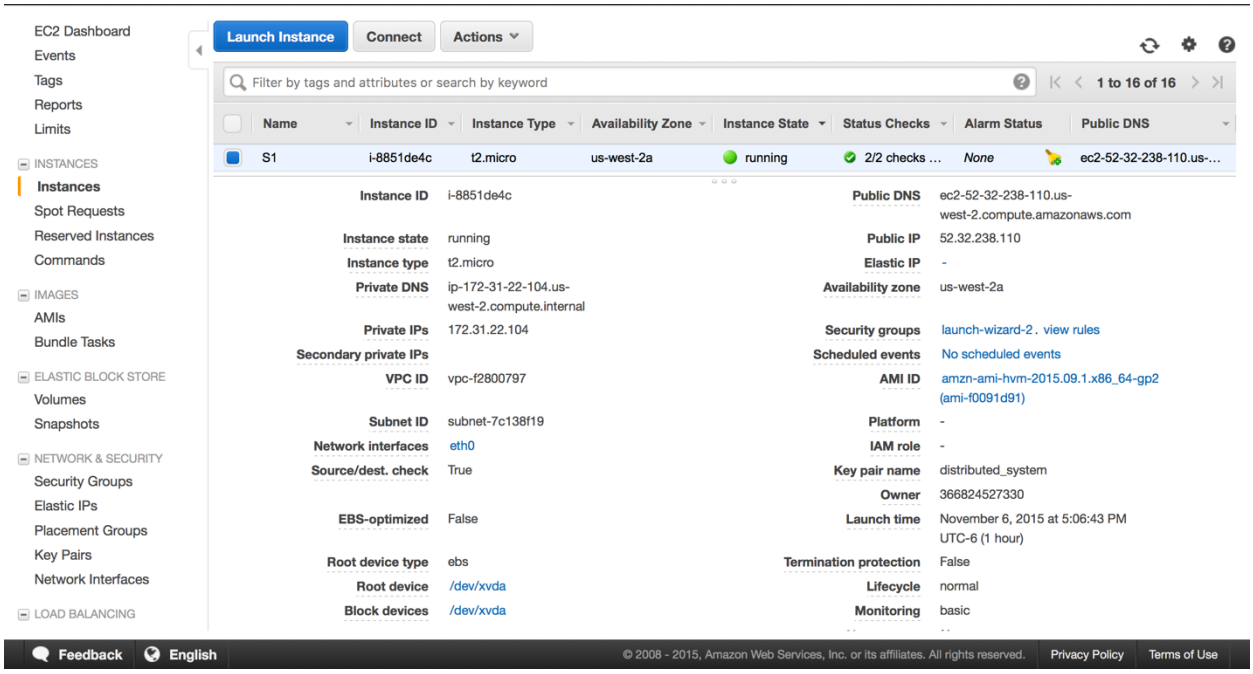
The following evaluation was now done on Amazon AWS using the following steps:-

SETTING UP AWS:

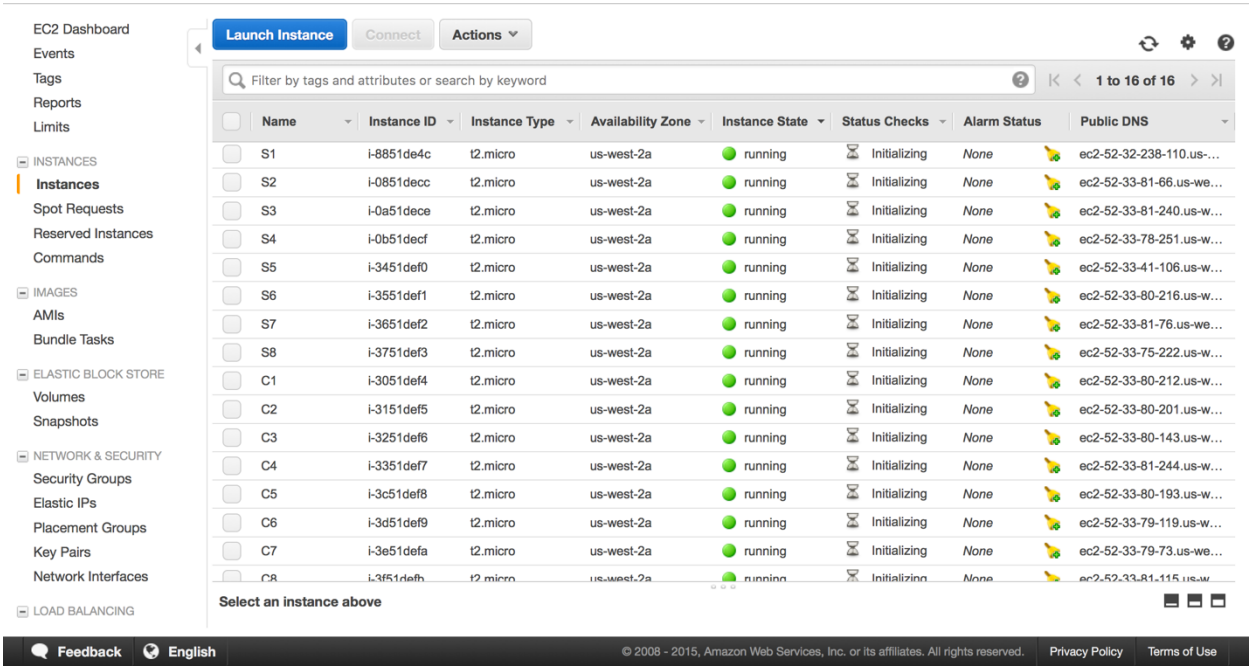
- 16 instances were created comprising of 8 instances of Servers and 8 instances of Clients. Following are the details along with the screenshots:

Amazon Machine Image(AMI): Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d9

Instance ID: i-8851de4c
Instance type: t2.micro



• Below image shows all the instances live and running in the AWS:

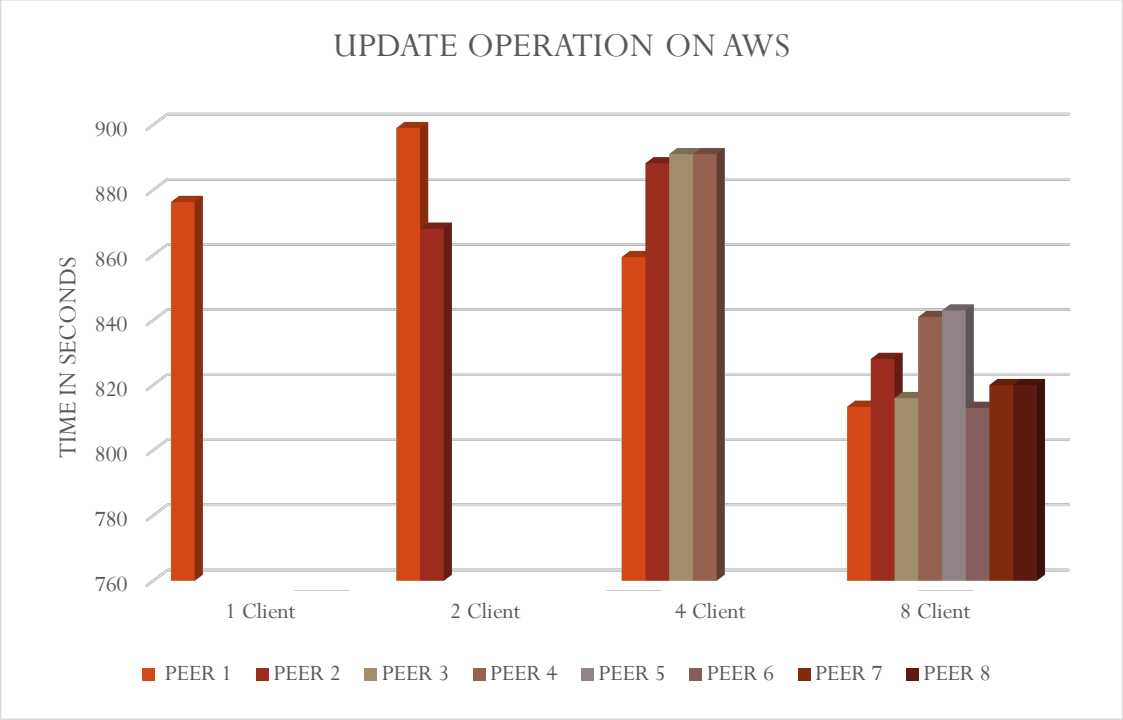


- After all the instances were created, FileZilla was used to transfer the P2P application and all the files to be tested on the instances.
- Now for the evaluation, 8 instances of Servers, having the *"Server.jar"* and *"config.properties"* files were created.
- Now other 8 instances were created having *"Client.jar"* and two folder namely *"UPLOAD"* and *"DOWNLOAD"* containing files that has to uploaded and receiving downloaded files respectively.
- Evaluation is done with 10,000 files each of 1Kb size on each peer through their UPLOAD folder. The evaluation is graphed on following three operations: UPDATE, SEARCH and DOWNLOAD.

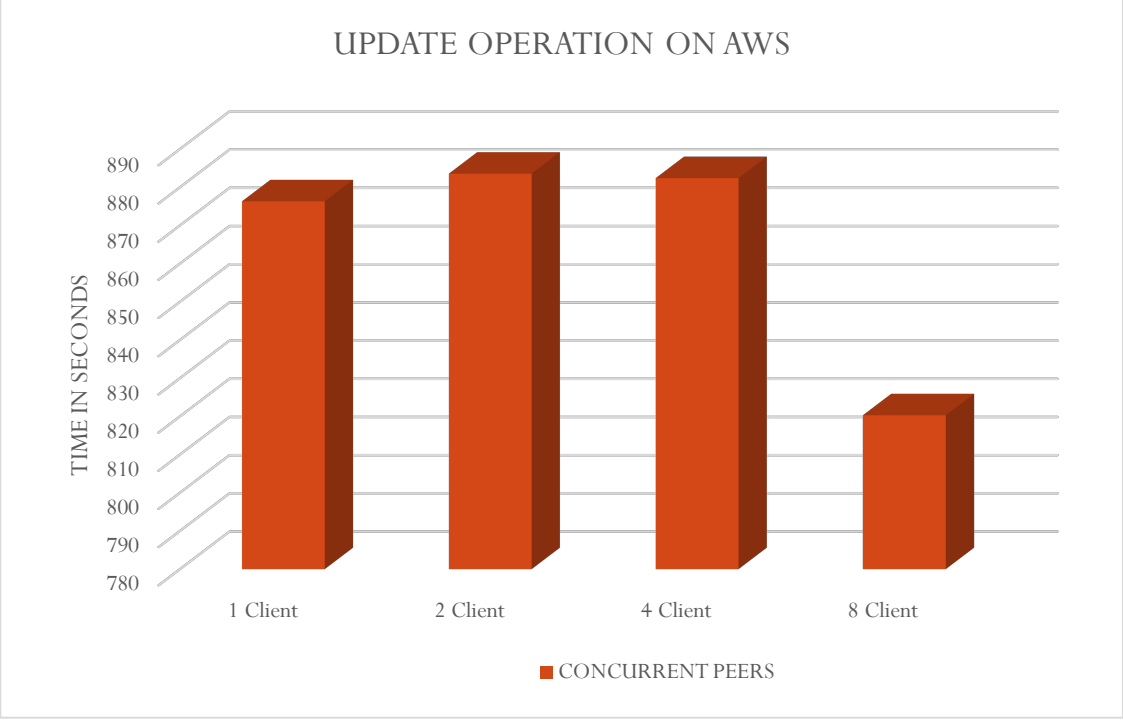
UPDATE OPERATION WITH 10,000 FILES:

Concurrent Running Clients	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8
1 Client	876.32 s	-	-	-	-	-	-	-
2 Clients	899.01 s	868.11 s	-	-	-	-	-	-
4 Clients	859.41 s	888.21 s	891 s	891.01 s	-	-	-	-
8 Clients	813.33 s	828.03 s	816 s	841.29 s	842.88 s	813.21 s	813.95 s	820.68 s

a). Individual Values:



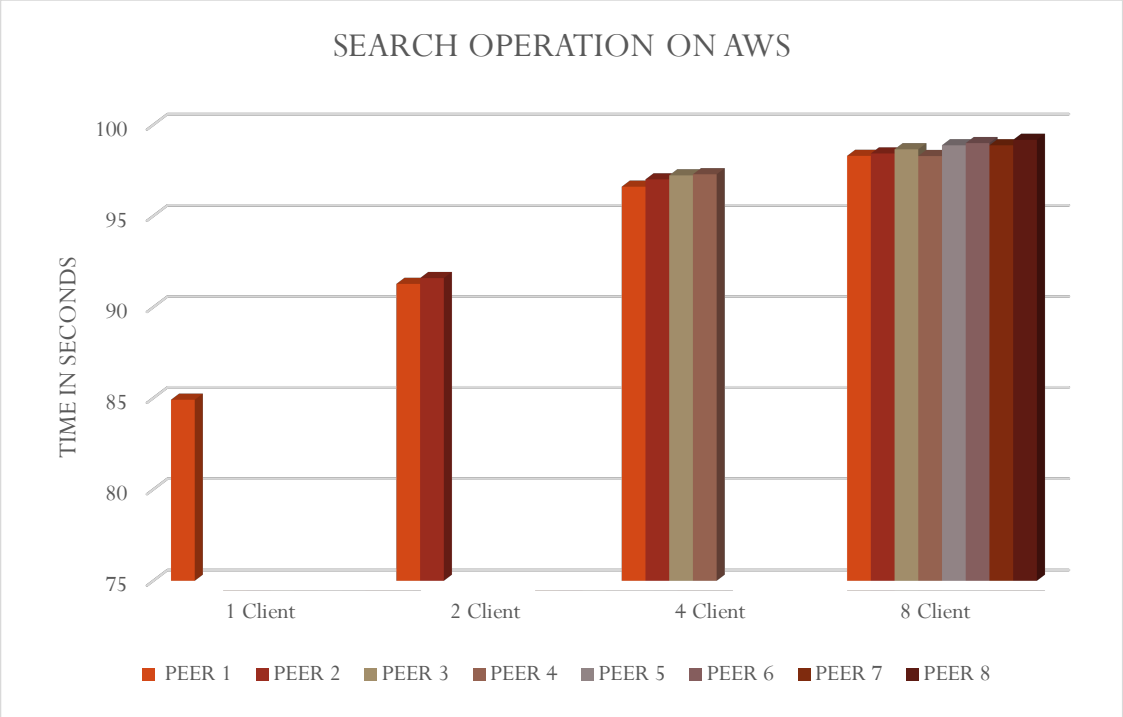
b). Average values:



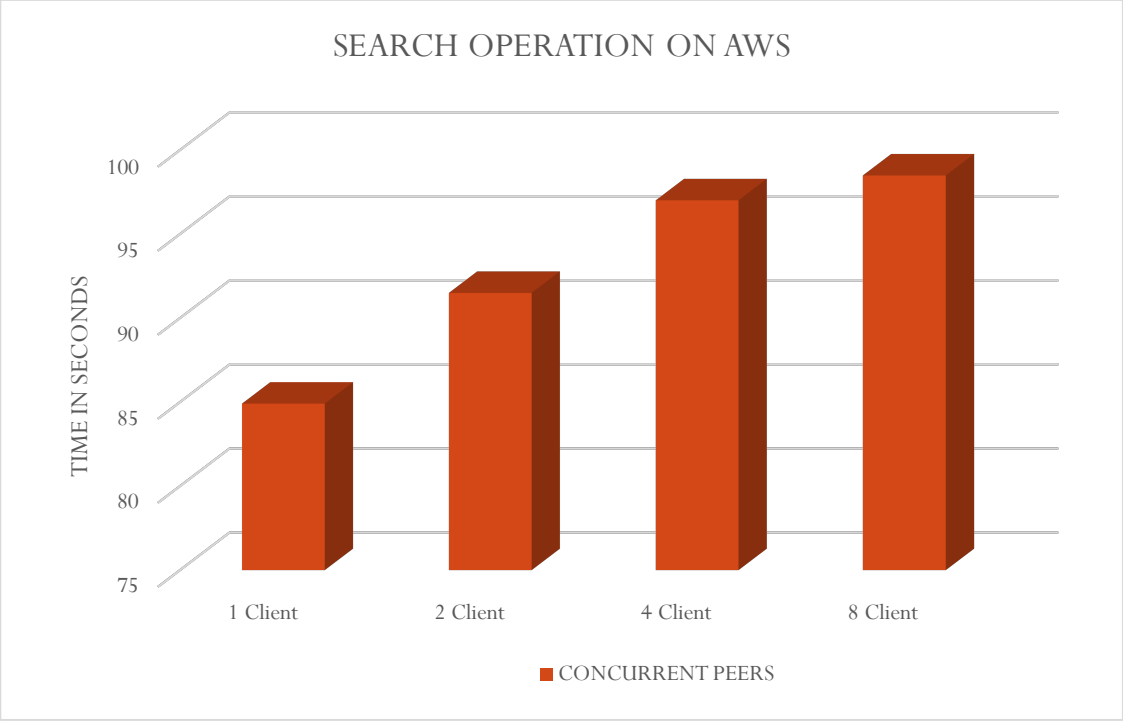
SEARCH OPERATION ON 10,000 FILES:

Concurrent Running Clients	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8
1 Client	84.91 s	-	-	-	-	-	-	-
2 Clients	91.28 s	91.60 s	-	-	-	-	-	-
4 Clients	96.61 s	97.01 s	97.23s	97.29 s	-	-	-	-
8 Clients	98.30 s	98.43 s	98.66 s	98.29 s	98.88 s	99 s	98.88 s	99.18 s

a). Individual values



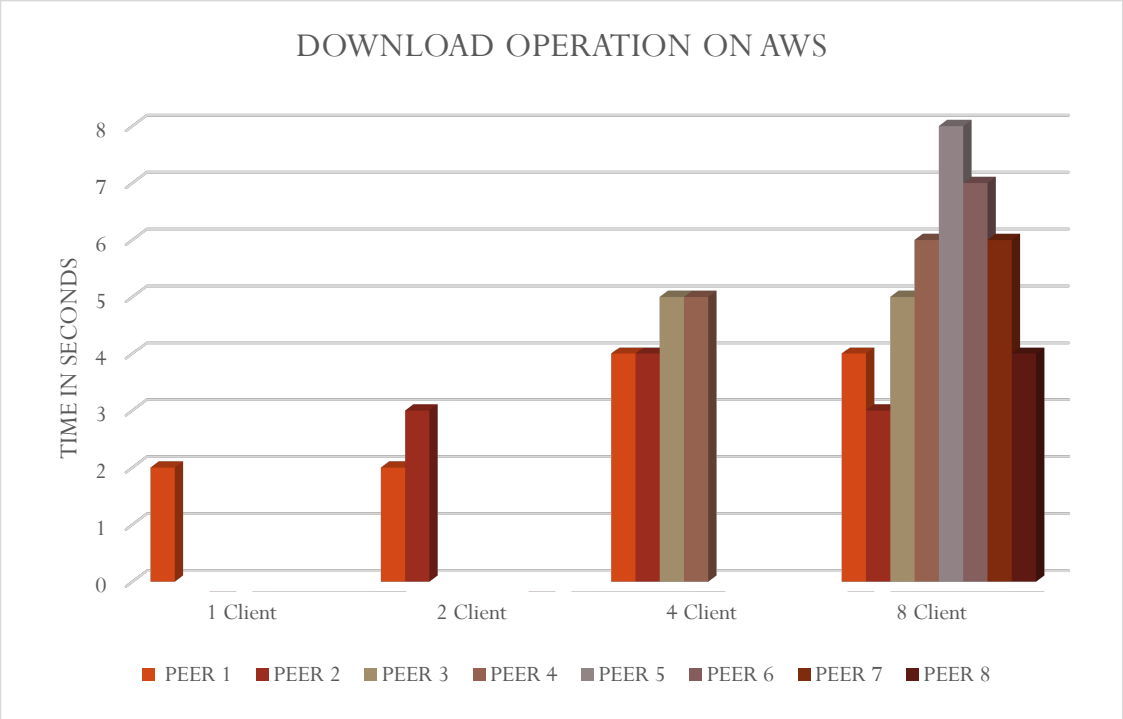
b). Average Values:



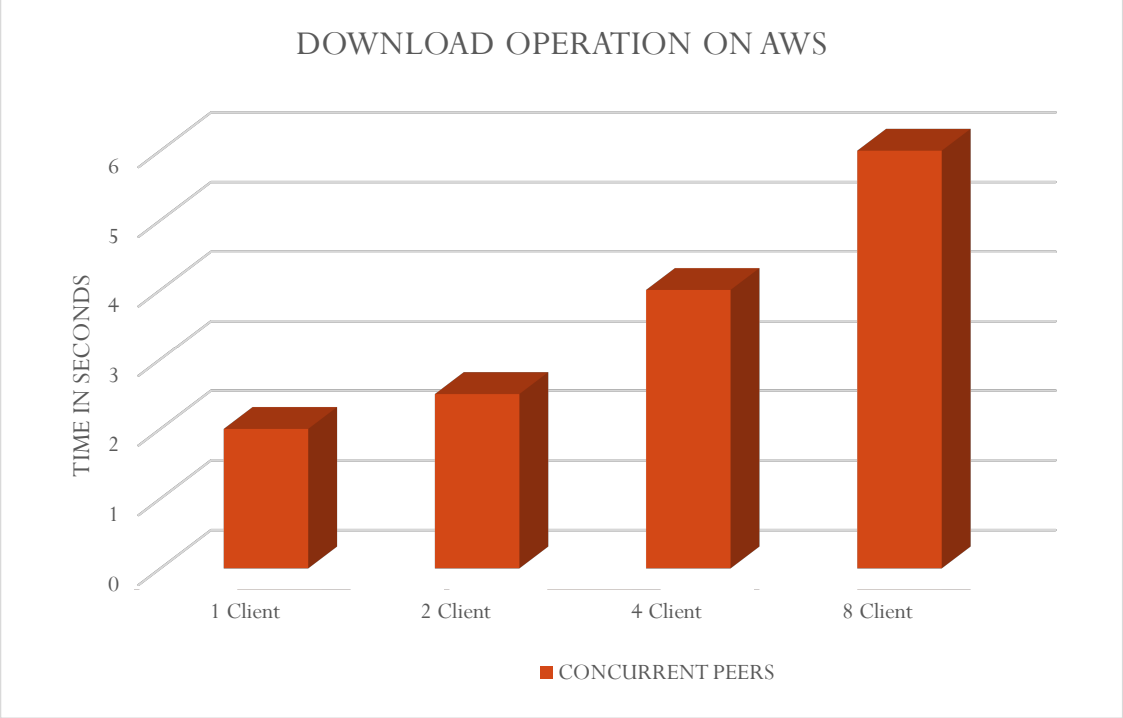
DOWNLOAD OPERATION OF A FILE AMONG 10,000 FILES:

Concurrent Running Clients	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8
1 Client	2 ms	-	-	-	-	-	-	-
2 Clients	2 ms	3 ms	-	-	-	-	-	-
4 Clients	4 ms	4 ms	5 ms	5 ms	-	-	-	-
8 Clients	4 ms	3 ms	5 ms	6 ms	8 ms	7 ms	6 ms	4 ms

a). Individual Values:



b). Average Values:



SECOND SET OF EXPERIMENTS WITH FILES OF VARYING SIZES AND COMPARISION WITH PROJECT ASSIGNMENT 1:

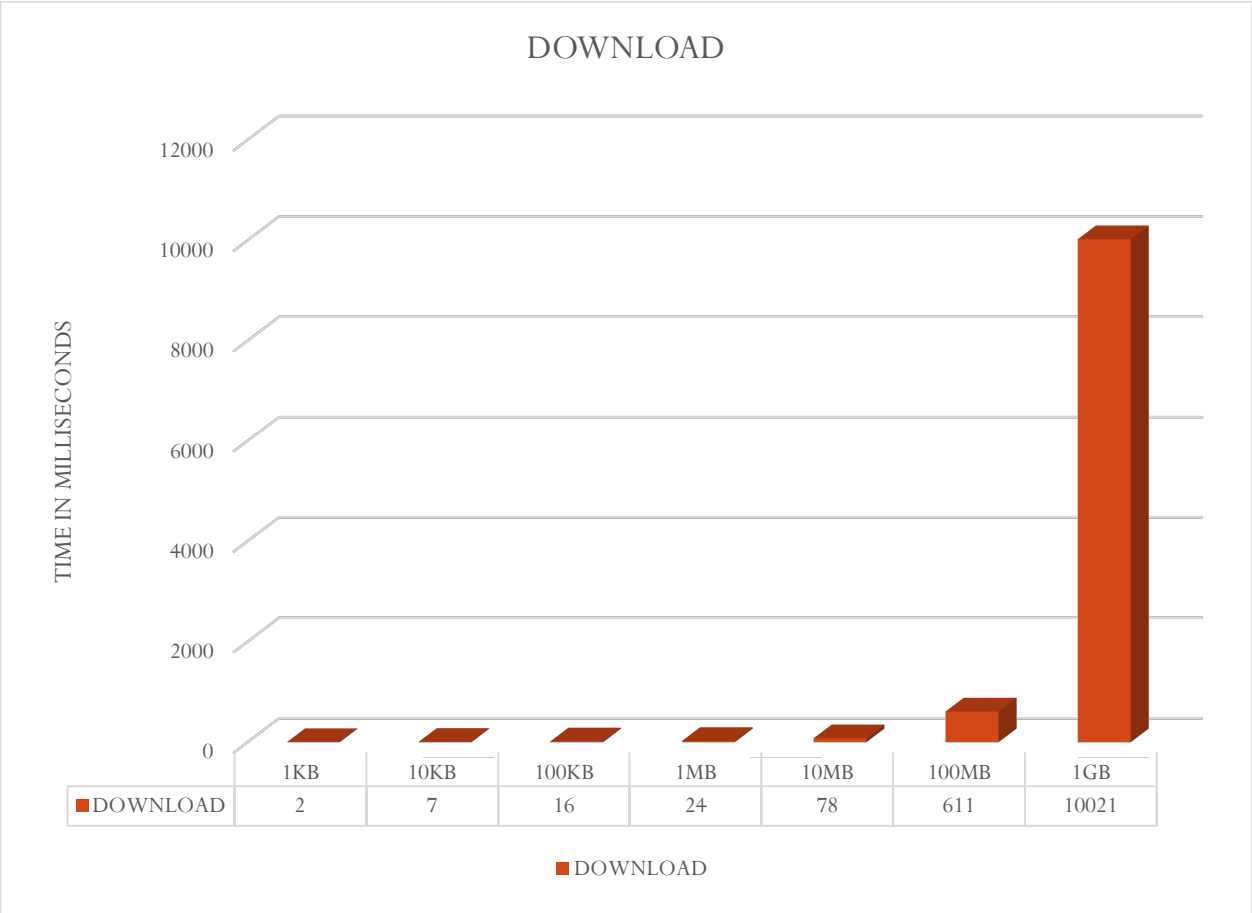
A. ON LOCAL MACHINE:

The following evaluation was first done on my Local machine's (Apple Macbook Pro) Terminal console:-

- 8 servers was made to run on 8 different host machines and all of them were connected in a Distributed P2P environment.
- Then 7 Clients were brought in, containing Files of varying sizes:
  - 1GB – Client 1
  - 100 MB – Client 2
  - 10 MB – Client 3
  - 1 MB – Client 4
  - 100 KB – Client 5
  - 10 KB – Client 6
  - 1 KB – Client 7

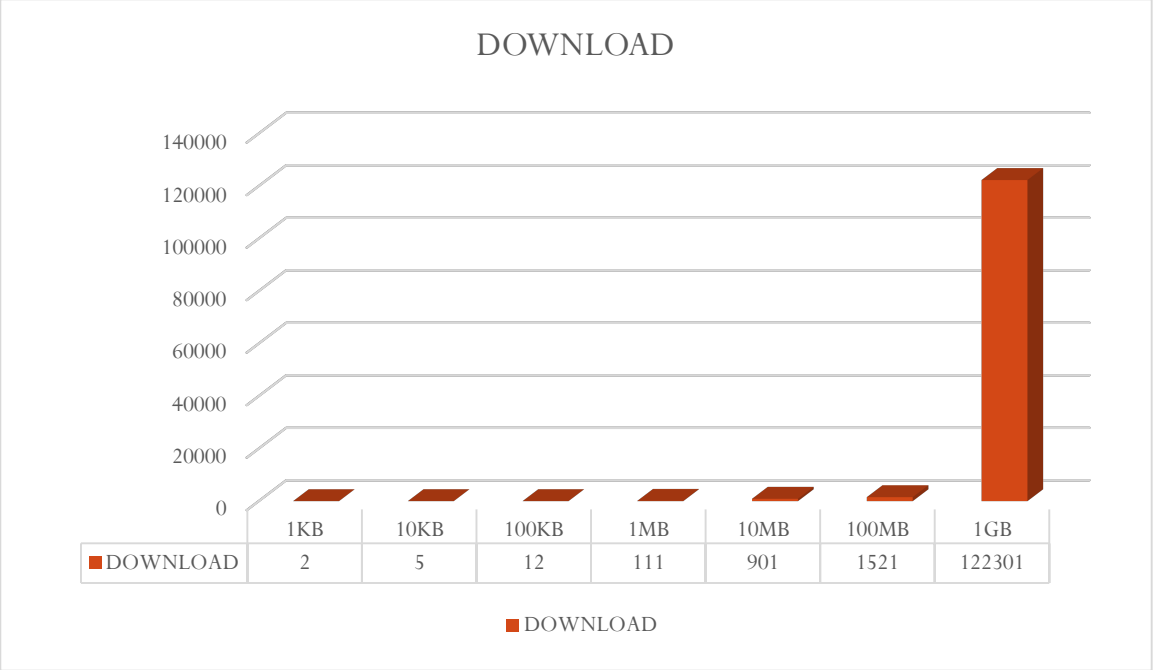
- Then all the clients above performed **Download Operation** on each of their file. The tests were conducted by using timers in the peer application. A timer was started in the code before the execution of each operation and stopped after the operation. The difference was calculated in seconds.

The Performance evaluation for Download files of various sizes is shown by a graph below:



B. IN AMAZON AWS

Similarly files of varying sizes were downloaded through different Client instances on Amazon AWS cloud and the performance evaluation is demonstrated by a chart below:

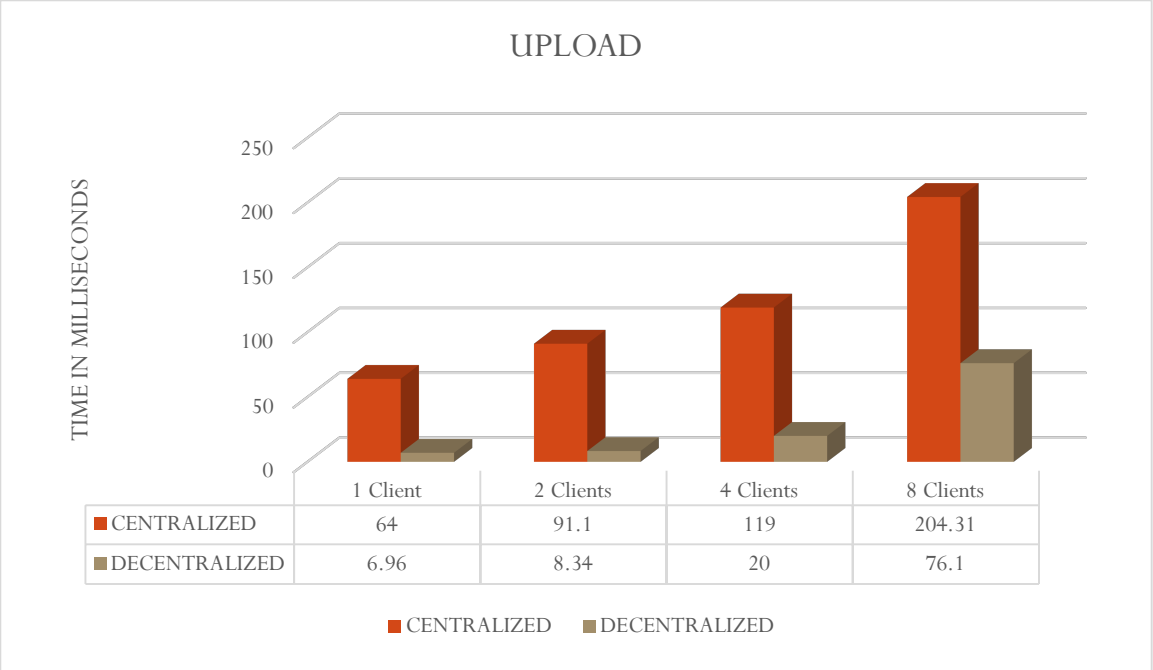


COMPARISON OF PROJECT ASSIGNMENT 1 (CENTRALIZED P2P) AND PROJECT ASSIGNMENT 3(DDECENTRALIZED P2P)

- For the above comparison, I ran both the application simultaneously on local machine with 10,000 files of 1KB each.
- The performance evaluation was done on three operations: UPDATE, SEARCH and DOWNLOAD and the average time was calculated and put into Graph
- The Centralized P2P(PA1) consisted of 1 Central Index Server and 8 Clients.

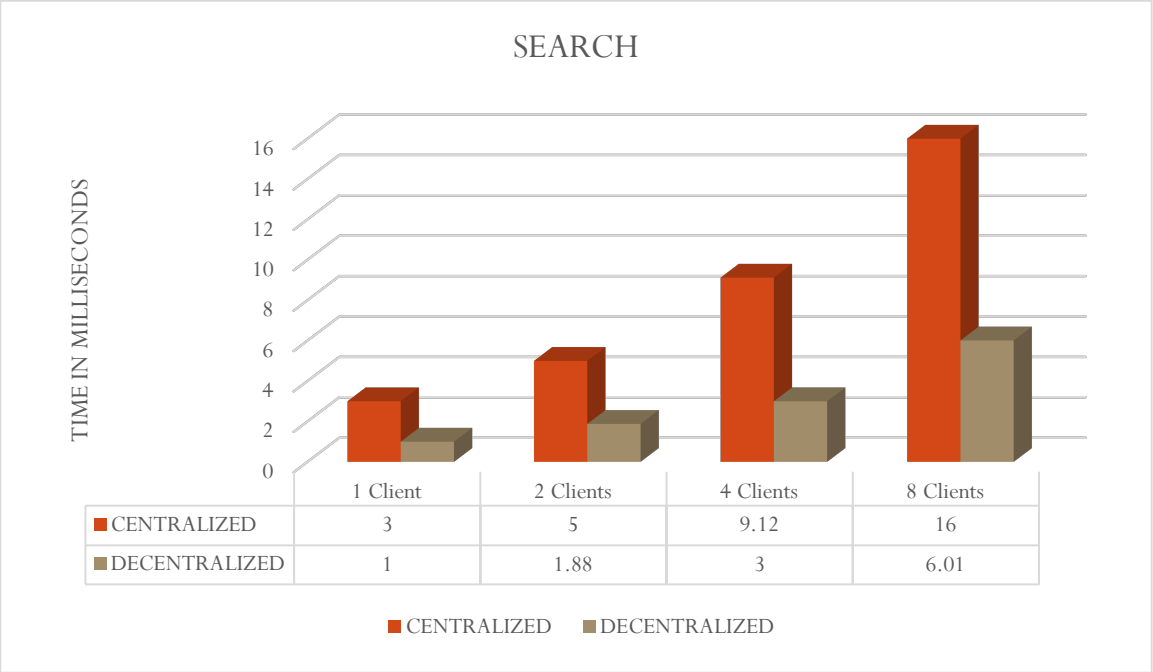
- The Decentralized P2P(PA2) consisted of 8 Decentralized Servers and 8 Clients.
- The applications were run and the average values were put into graphs as below:

UPDATE OPERATION (AVERAGE VALUES):



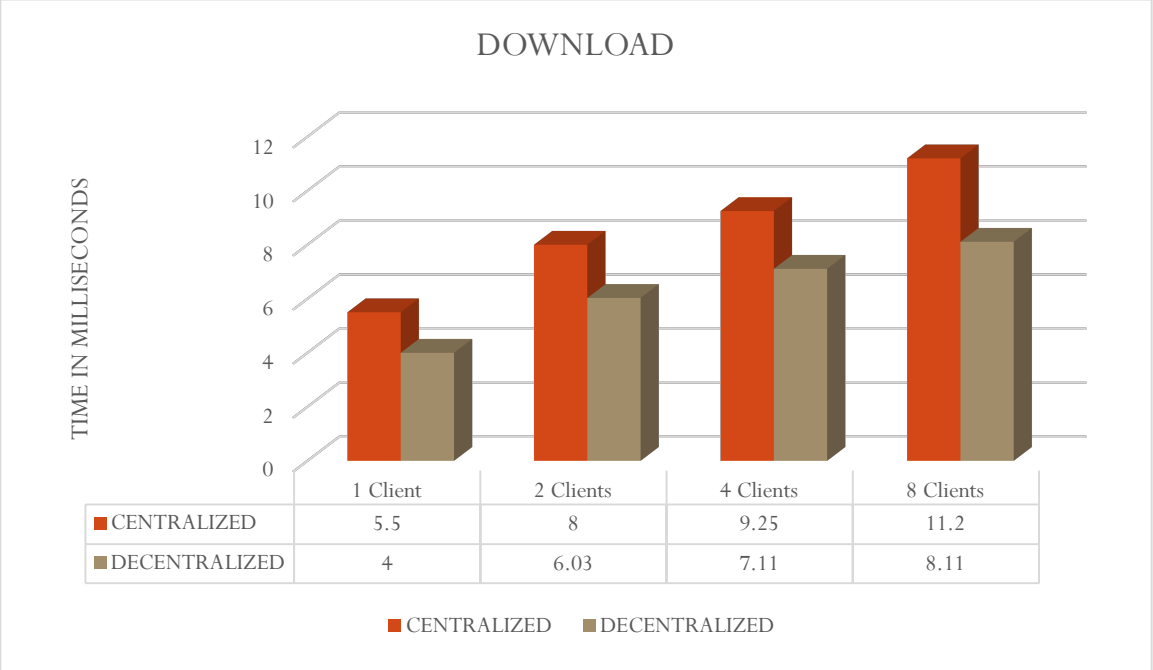
So with the above graph it becomes very clear the with a Distributed architecture in Green, it takes a lot less time to UPDATE 10,000 files on the servers than the Centralized one.

SEARCH OPERATION (AVERAGE VALUES):



So with the above graph, we can see the difference in SEARCH operation between the two P2P systems. And it can be clearly seen that the one marked in Green (Decentralized P2P) has a very good response time than the other one marked in Red (Centralized P2P).

DOWNLOAD OPERATION (AVERAGE VALUES):



From the above graph we can see that this time there is a very slight difference the DOWNLOAD operations of the two systems. That is because the downloading function in both the applications is among the Peers. The Servers in both the systems indexes and provides the requesting Peer with the details of the Peer having the requested file. And hence there is very slight difference.