

# Performance Evaluation

## **Experiment 1:**

For this experiment I took 10000 files of 10KB to perform register, search and obtain operations on 1, 2, 4 & 8 clients simultaneously. As per the design of my assignment 1 and 3 I came across below comparisons:

### ***Register:***

Register operation is giving a throughput of around 1475 operations per sec. It takes around 9-15 seconds individually to complete registration of 10,000 files. Whereas collectively it comes around 50-55 seconds. Whereas in assignment 1, it takes around 400-500 milliseconds individually and roughly the same when run collectively.

This difference can be easily highlighted due to distributed hashing across all the servers, padding and replicating of key/value pair across replication servers. Where in assignment 1, there was only one centralized server with no padding and key/value pair replication.



### ***Search:***

Search operation is giving a throughput of around more than 200K operations per sec.

Since there is nothing changed in the code for Search operation other than the key padding, the search performance remains almost the same across single & multiple clients in both assignments 1 and assignments 3.



**Obtain:**

Obtain operation is giving a throughput of around 451 operations per sec. It takes around 12-15 seconds individually and 170-180 seconds collectively to download 10,000 files of 10KB. As compared to obtain method in assignment 1, where in it takes around 180 seconds individually to download 10,000 files of 10KB.

This huge difference was because of design implementation of assignment 1 wherein I was creating new thread for each file download operation so when I tried to download 10,000 files, it created 10,000 threads which used up all my CPU cycles and frozen my system. Whereas in assignment 3, the socket communication between each server and each client is already established, open and saved in HashMap. So each file download requests is served through this communication link without creating any extra thread.



### **Experiment 2:**

As per the requirement, I calculated the throughput number of bytes per second for various size and quantity of files. While evaluating, I have noticed that for small size but larger amount of files it takes longer amount of time giving us a low throughput whereas larger but small amount of files takes a bit longer time but provides far better throughput. This can be easily seen in the below graph. To cross check my evaluation I compared downloading 1000 & 10000 files of size 100KB, the time difference was almost 20 times and the throughput was double for 1000 files.

