**Performance Evaluation**

Evaluation of the system is done to find its performance when current peers are sending requests concurrently to the distributed servers. The throughput of the system is also evaluated . Then a comparison is drawn between the centralized and distributed file sharing system.

Required files

Server Side

DistributedServer.java (to be run)

DistributedServerOperate.java

ServiceClient.java

Utility

Server1.properties

makefile(optional)

Client side

PeerClientPerfTest.java (to be run)

PeerServer.java

makefile(optional)

createFile.sh (used to create files of different sizes)

Test Bed Used:

OS : Linux

Amazon Cloud Instance:

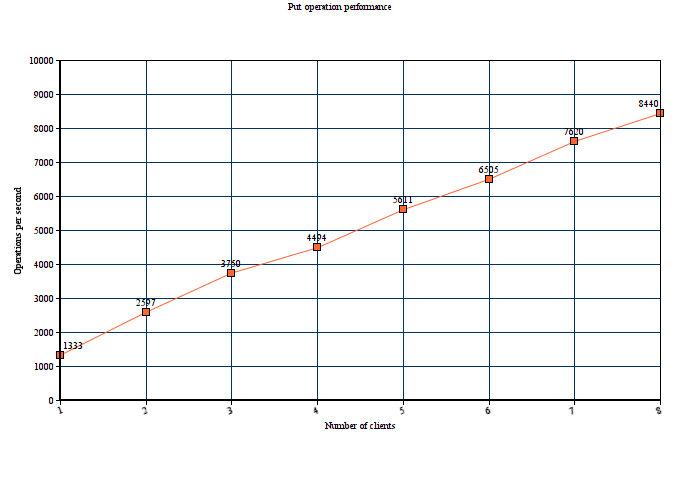
Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-f0091d91

Instance type**:** t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GB memory, EBS only)

**Experiment 1**

In this experiment 8 servers are up and running and implement the distributed hash table for storing the metadata. The number of put(register), get(download) and delete(deregister) operations per second are observed when incrementing the clients from 1,2 .. upto 8. Data resilience (both metadata and file)has been turned off while performing this experiment. The experiments are carried out on the amazon cloud using the free amazon web services resource. Using the AWS a more realistic data on the performance of the system can be evaluated

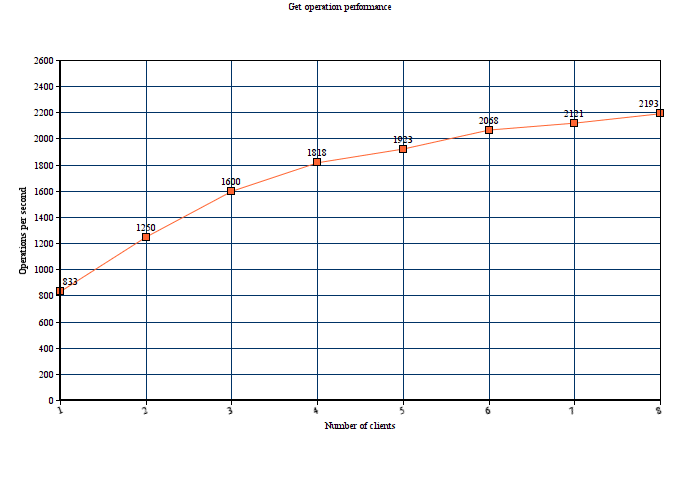
Put (register of file) operation performance



In put operation each client register the files it wants to share with other peers on the network. As the number of servers are 8 ,the load on the servers is well distributed (Djb2 algorithm used in hashcode function).The number of put operations per second increases as the number of clients are increased. This is because there are 8 servers and the resource utilization of the system increases as the number of clients are increased. The response time to the clients remain almost constant even after the number of clients are increased as the load of handling requests is shared among the distributed servers. Also as the client number increases the servers will be more loaded , increase in number of cllients will increase the performance of the servers until they are saturated with workload. Beyond that the operations per second may fall as the servers will take more time to respond to the requests.

Since the put operation is lightweight ,it does not does require much computations the reponse time is low.

Get (download of file) operation performance



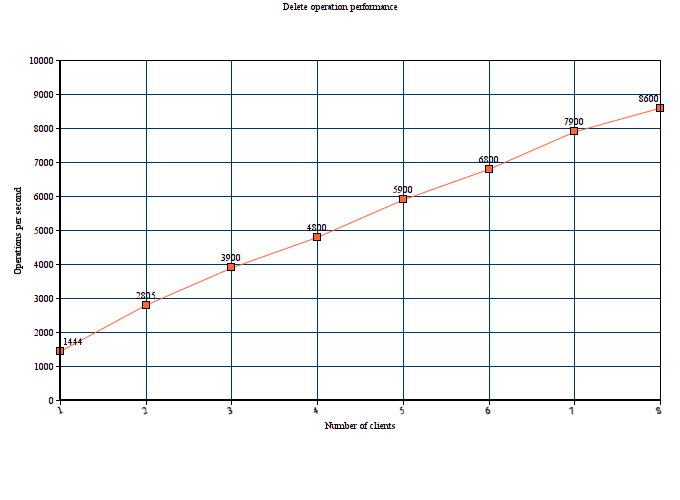
In the get operation the peers requests one of the distributed server for a file that it wants to download. The server responds with the metadata i.e the location (ip,port number , file directory) of the peer which hosts the file. The peer then communicates directly with the peer which has the file and downloads it. The server has no role in the download of file. Hence the get operation response time is the sum of response time of the server to send the metadata to the peer and the time taken by the peer to download the file.

While conducting this experiment the filesize is kept constant at 1 KB. From the graph it can be observed that as the number of clients increase the operations per second also increase this is because there are 8 servers and the load is well distributed , resource utilization increases as the number of client increases.

Compared to put operation the number of operations per second is lower . This is because the peers take time to download files . The operations per second may further decrease as the size of the file is increased.

Note that if the file to be download the response time is lower.

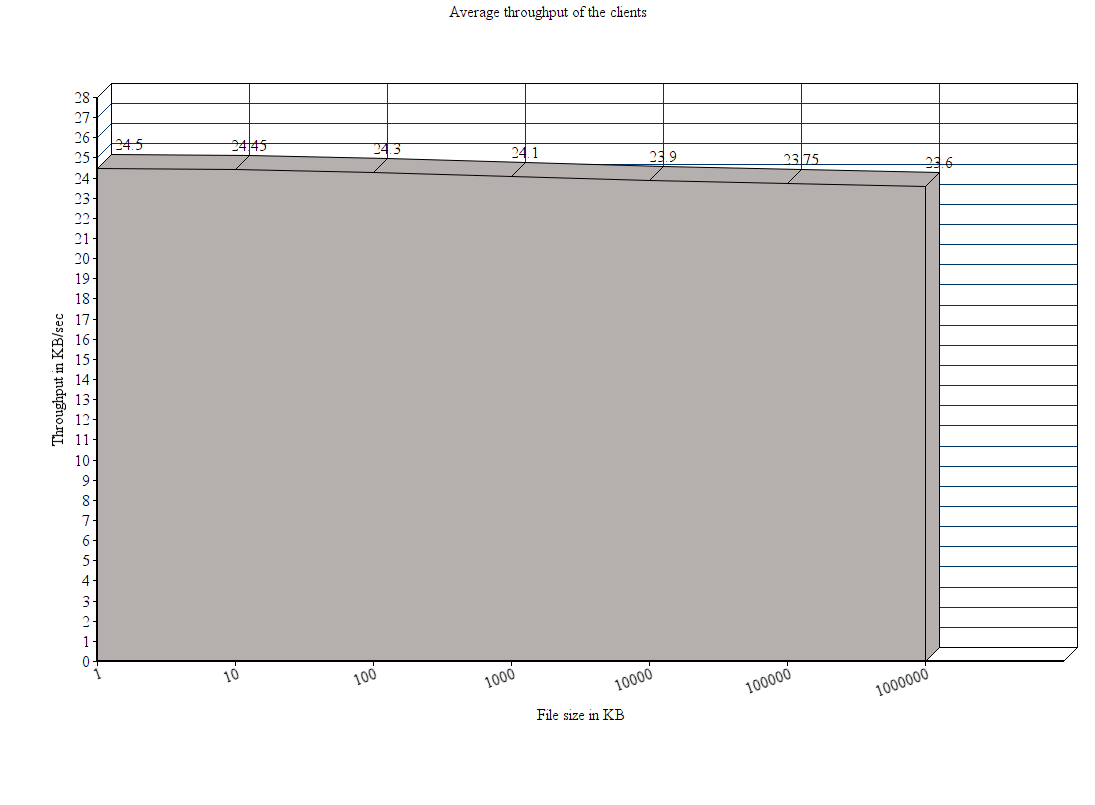
Delete (Deregister ) operation performance



In the delete operation the metadata of a particular file is removed from the distributed hash map. From the above graph it can be observed that as the number of servers increases the number of operations per second increases. The distributed servers share the load of the incoming requests and resource utilization increases. The number of delete operations per second is slightly higher than the get operation. The increase in number of operations will be observed until the servers are loaded , then the response time will take longer.

**Experiment 2**

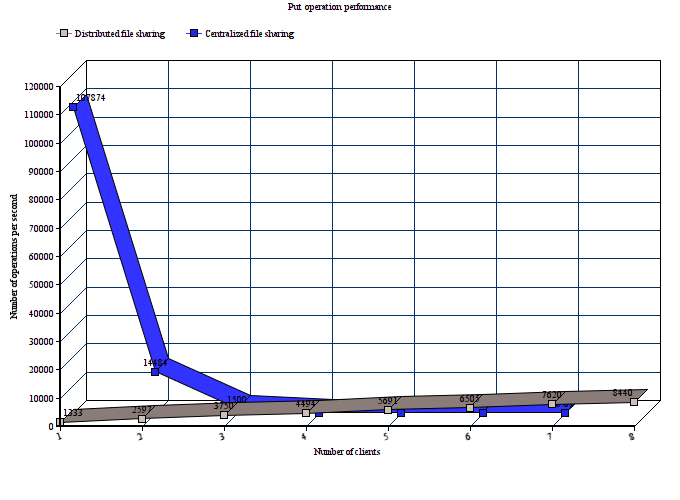
In this experiment the number of clients and servers are kept constant (8). The main aim of this experiment is to observe the throughput of clients for different file sizes.



As seen from the graph the average throughput of the clients is almost constant for the different file sizes. The throughput mainly depends on the network speed. For file download the peers communicate with each other , the throughput depends on the rate at which the two peers interacting (one sending the file the other downloading it) upload or download the file. As each peer has its own allotted bandwidth and the throughput depends on the network speed.

Comparison of centralized file sharing system

We now can compare the performance of a centralized file sharing system with the decentralized file sharing system.



From the above graph it can be deduced that for a centralized file sharing system the performance decreases as the number of clients are increased and in case of decentralized file sharing system the performance increases with the increase in number of clients , although this increase becomes stagnant for a large number of clients.

The decrease in performance in centralized file sharing system as clients increase is because the indexing server that stores the metadata has to respond to more number of requests . The response time becomes slower as the server is overloaded with requests, the decrease in operations per second in the graph is due to this reason. This kind of file sharing system is optimal in networks where the number of clients are low.

In the centralized file sharing system the performance increases with the increase in the clients. This is because the distributed servers’ resource utilization is maximized as number of clients increased. Therefore the number of operations per second increases. But this increase will become stagnant when the servers are loaded by the requests from more number of clients than it can handle optimally.

Also we can observe that when the number of clients are lower then the centralized file sharing system gives a much better performance than the distributed file sharing system.

Appendix

AWS

To run the experiments on the amazon cloud , instances of linux machines can be created and configured.

Connected to a t2 instance

