Programming Assignment 3

Distributed Peer to Peer File Sharing System.

System Configuration

- I ran following experiments on a machine having 3 cores and 8GB ram.

Evaluation set 1:

Performance Evaluation: By increasing number of clients for operation Register, Search, Download (Obtain).

Number of Clients = 1

- Performance of a system is measured by generating **10,000 (10 Thousand) requests** of **file size 1KB** from a single client. And then varying the number of clients for concurrent requests.
- First a single client is run to make 10,000 requests for Register, Search, Download with a different Key-value pairs.
- Then number of clients are increased gradually and their average time for execution of 10,000 requests are noted.
- Following results are observed for a client performing various operations.

Operation	Time in seconds
Register	**1.834**
Search	**2.298**
Download	**62.185**

- **Observation**:
  - Download operation is taking more time than other two operations.
  - Search is taking slightly greater time than register operation.

Number of Clients = 2

- Now two client are run concurrently to make 10,000 requests for Register, Search, Download with a different Key.
- Following results are observed for 2 clients performing various operations.

Operation	client1 (time in mills)	client2 (time in mills)	average time in seconds
Register	3519	1490	**2.5045**
Search	2596	2914	**2.755**
Download	86583	87063	**86.823**

Number of Clients = 4

- Same experiment is repeated for four client and run to make 10,000 requests for Register, Search, Download with a different Key.
- Following results are observed for 4 clients performing various operations.

Operation	client1 (time in mills)	client2 (time in mills)	client3 (time in mills)	client4 (time in mills)	average time in seconds
Register	4722	4425	1593	1715	**3.11375**
Search	4995	3209	2074	1936	**3.0535**
Download	125104	123581	129988	131340	**127.50325**

Number of Clients = 8

- Eight client are run to make 10,000 requests for Register, Search, Download with a different Key.
- Following results are observed for 2 clients performing various operations.

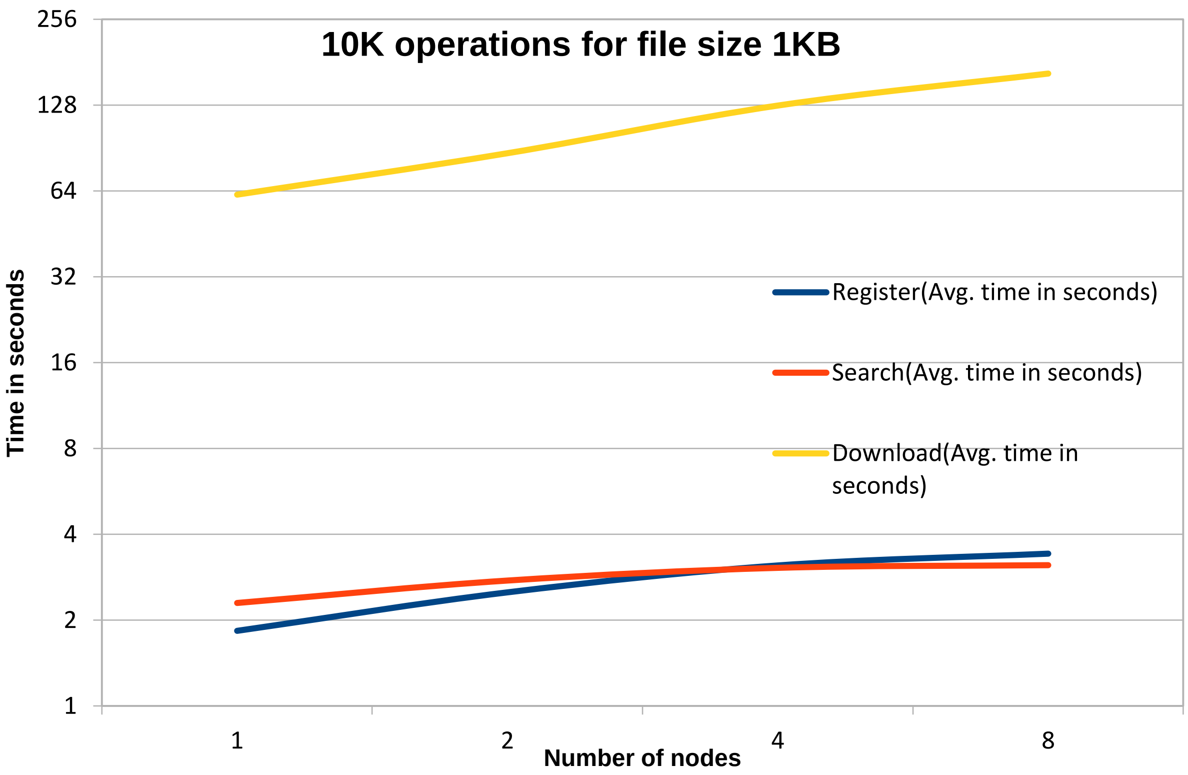
Operation	client1 (time in mills)	client2 (time in mills)	cient3 (time in mills)	client4 (time in mills)	client5 (time in mills)	client6 (time in mills)	client7 (time in mills)	client8 (time in mills)	average time in seconds
Register	5384	3948	3281	3391	1474	4950	2414	2561	**3425.38**
Search	4330	2955	3574	3053	2735	3014	2944	2358	**3120.38**
Download	162879	162193	165324	159306	159326	171399	171631	167699	**164970**

Below is the summarized table showing the average times in seconds for different operations when 1, 2, 4 and 8 clients are run concurrently for 10,000 operations.

Observations for different number of clients:

	Per operation 10K transaction of file size 1KB		
Number of concurrent Clients	Register (Average time in seconds)	Search (Average time in seconds)	Download (Average time in seconds)
1	1.834	2.298	62.185
2	2.504	2.755	86.823
4	3.11375	3.0535	127.50325
8	3.42538	3.12	164.97

Plotting an overall graph for above observations:



Final Conclusion:

- Time taken by Download operation is always greater than Register and Search operations.
- Download takes more time because in this operation a huge amount of data is transferred over a network and streams which increases its cost.
- As the number of client increases the time for performing each operation increases gradually. Time taken for registering and searching files is increased by very small amount as very small amount of data is transferred while performing this operations.
- For **downloading 10K files** of 10KB single system required **60 seconds**, however increasing to 8 clients total **time taken is just 160 seconds**. It's because of distributed multithreaded system.

- From the summarised table we can compute the **Average response time** per operation. Table gives value for 10,000 operations. So computing for a single operation we get following values.
  - **Average response time per Register operation**: 0.3 milliseconds
  - **Average response time per Search operation**: 0.2 milliseconds
  - **Average response time per Obtain/ Download operation**: 11.03 milliseconds

- As the number of concurrent nodes increases the number of concurrent operations also increases and hence time increases in linear fashion. Thus we can conclude that the performance of a system is good but when 8 nodes concurrently perform operations average time increases.

Evaluation set 2:

Performance Evaluation: By increasing File Size for measuring throughput in bytes per seconds of a distributed file sharing system.

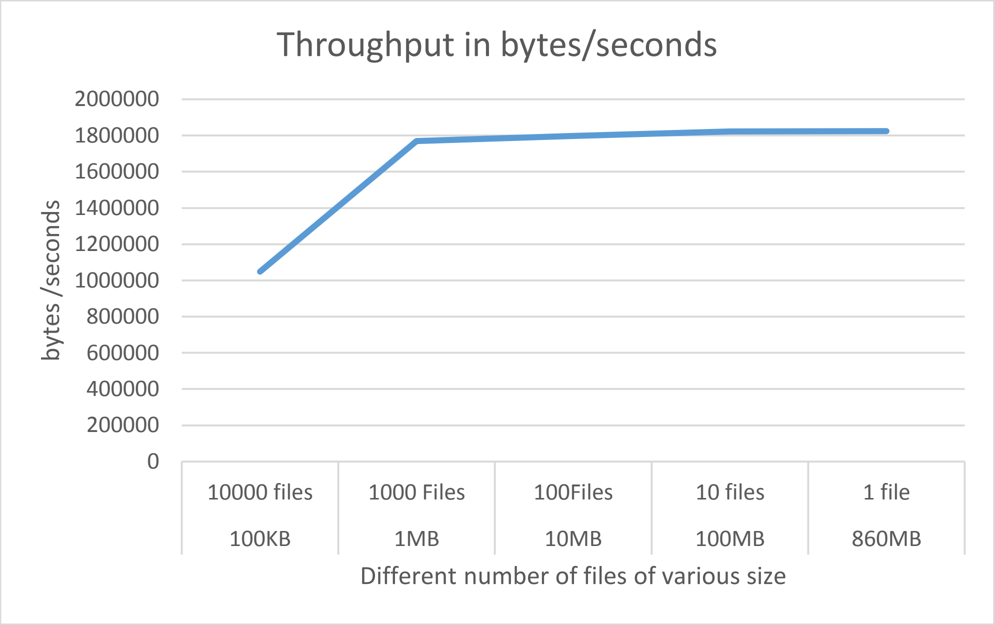
Here I have kept fixed number of clients = 8

1. **Here system are performing operations on files such that system is running for at least a minute or more.**

- Performed download operation by increasing the file size and decreasing number of files.
- In each operation same amount of bytes is transferred in a download operation among all clients.
- For all 8 clients time is measured and aggregate time is computed.
- Aggregate throughput is calculated by dividing total number of bytes transferred to total time taken.

File Size	Number Of Files	Per Client Data in Bytes	client1 (time in mills)	client2 (time in mills)	cient3 (time in mills)	client4 (time in mills)	client5 (time in mills)	client6 (time in mills)	client7 (time in mills)	client8 (time in mills)
100KB	10,000 Files	**1024000000**	100167	104872	1308131	1314491	1243011	1284946	1248809	1206679
1MB	1000 Files	**1024000000**	473934	580566	538400	503536	620735	631683	636178	644649
10 MB	100 Files	**1024000000**	379144	525252	528530	587682	647691	634829	611967	640121
100MB	10 files	**1024000000**	625107	611878	613832	601052	570505	502724	389124	579491
860MB	1 file	**9027,27,572**	577839	450677	508601	471686	541761	501439	489827	415794

File Size	Number Of Files	Average time in seconds	Throughput in bytes/seconds
100KB	10,000	**976.388**	**1048763.1**
1MB	1000	**578.710125**	**1769452**
10MB	100	**569.402**	**1798378**
100MB	10	**561.714125**	**1822991**
860MB	1	**494.703**	**1824787**



X-axis: Different number of files of different size.

Y-axis: Throughput in bytes per seconds.

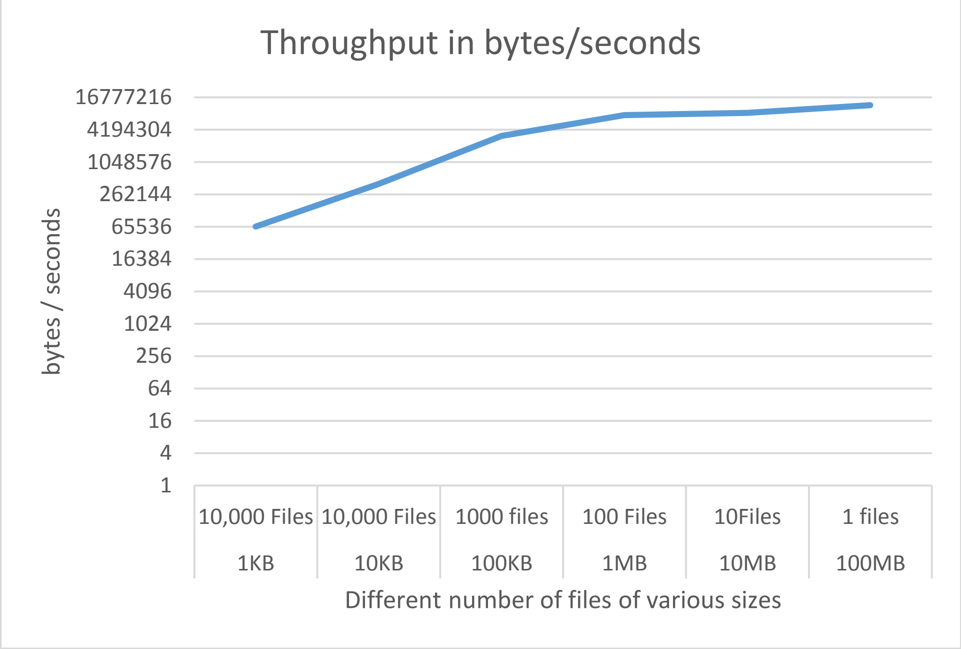
- **Conclusion:** From the above graph we can observe that the system work good for small number of files of large size **than** large number of files of small sizes.

2. Decreasing Number of Files and increasing File Size for any time period:

- Here I have generated 10,000 files of 1KB size and downloaded on all 8 clients. Then number of files are reduced and File size is increased by multiple of 10 and same as that of above experiment is repeated.

File Size	Number Of Files	Per Client Data in Bytes	client1 (time in mills)	client2 (time in mills)	cient3 (time in mills)	client4 (time in mills)	client5 (time in mills)	client6 (time in mills)	client7 (time in mills)	client8 (time in mills)	Average time in seconds
1KB	10000 Files	**10240000**	164267	166332	164361	161297	121116	158360	158266	159204	**156.650375**
10KB	10000 Files	**102400000**	237114	237286	238582	238801	275294	237346	275793	276499	**252.089375**
100KB	1000 files	**102400000**	13370	17613	15170	15547	141045	17630	18226	18679	**32.16**
1MB	100 Files	**102400000**	8824	11813	13095	14221	13548	14158	14408	14266	**13.041625**
10MB	10Files	**102400000**	7046	13879	10900	12691	12441	13143	12769	12136	**11.875625**
100MB	1 files	**102400000**	11088	7799	9173	7064	10141	7917	8379	7111	**8.584**

File Size	Number Of Files	Throughput in bytes/seconds
1KB	10000 Files	**65368.5**
10KB	10000 Files	**406205.14**
100KB	10000 files	**3184079.6**
1MB	1000 Files	**7851782.3**
10MB	100Files	**8622704.1**
100MB	10 files	**11929171**



X-axis: Different number of files of different size.

Y-axis: Throughput in bytes per seconds.

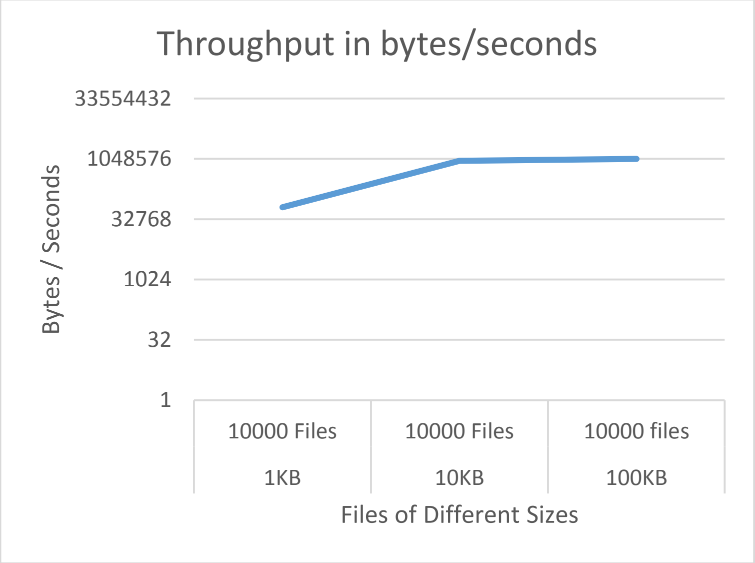
- **Conclusion:** From the above graph we can observe that the system work good for small number of files of large size **than** large number of files of small sizes.
- **Throughput** of a system **increases first** and **then becomes stable** after particular time.
- Throughput becomes stable as file size increases above 1MB size.

3. For Same number of files of varying size:

- In this experiment I have **fixed the number of files to constant (10,000)** and only **file size is increased** to verify my previous results.

File Size	Number Of Files	Per Client Data in Bytes	client1 (time in mills)	client2 (time in mills)	cient3 (time in mills)	client4 (time in mills)	client5 (time in mills)	client6 (time in mills)	client7 (time in mills)	client8 (time in mills)	Average time in Seconds
1KB	10000 Files	**10240000**	164267	166332	164361	161297	121116	158360	158266	159204	**156.650375**
10KB	10000 Files	**102400000**	117783	118771	114093	111502	79315	116686	111388	102069	**108.950875**
100KB	10000 files	**1.024E+09**	100167	104872	1308131	1314491	1243011	1284946	1248809	1206679	**976.38825**

File Size	Number Of Files	Throughput in bytes/seconds
1KB	10000 Files	**65368.49976**
10KB	10000 Files	**939873.1309**
100KB	10000 files	**1048763.133**



X-axis: 10,000 files of different size.

Y-axis: Throughput in bytes per seconds.

Conclusion: We can see that **as the file size increases throughput increases.**

4. **I have also considered one more set of experiment in which system registers 8 various different files of different file size.**

   Total size of data on each client = 1.32 GB

   = 1,42,19,56,723 KB

operations	client1	client2	cient3	client4	client5	client6	client7	client8	average time in seconds
Register	5393	4500	3772	3694	1656	4578	3397	3538	**3.816**
Search	68	19	18	20	28	23	22	19	**0.027125**
Download	752559	853936	750813	750191	842814	748620	749259	749570	**774.72025**

So throughput of system = Total size of data transfer by a client/ Average time taken

= 1,42,19,56,723 / 774.72025

= 1835445 bytes/ seconds

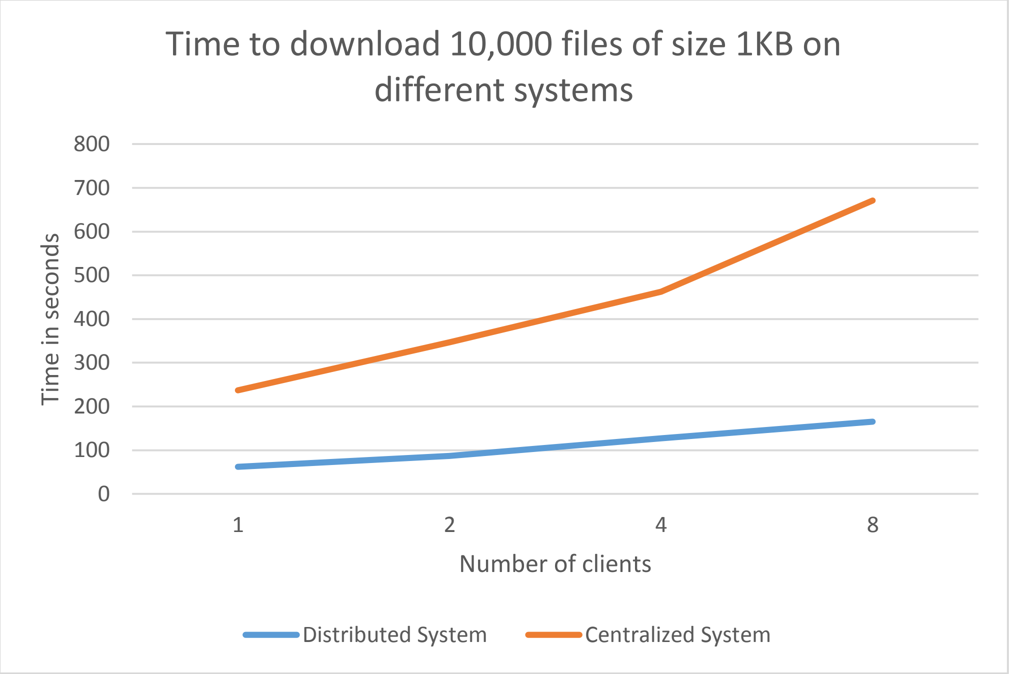
= 1.75 MB/second

Evaluation set 3:

Comparing Performance of Centralized System vs Decentralized system.

- In this set of Experiment I have performed same kind of evaluation on Centralized system to find time for downloading of 10K files of size 1KB.
- Again same experiment is performed by increasing number of clients to two, four and eight performing each operation concurrently.
- Total number of files = 10,000
- File size = 1KB
- Following are the observations recorded after running

Number of Peers running concurrently	Time Required for Download (Seconds)	
	Distributed System	**Centralized System**
1	62.185	236.65
2	86.823	346.45
4	127.50325	462.509
8	164.97	671.34



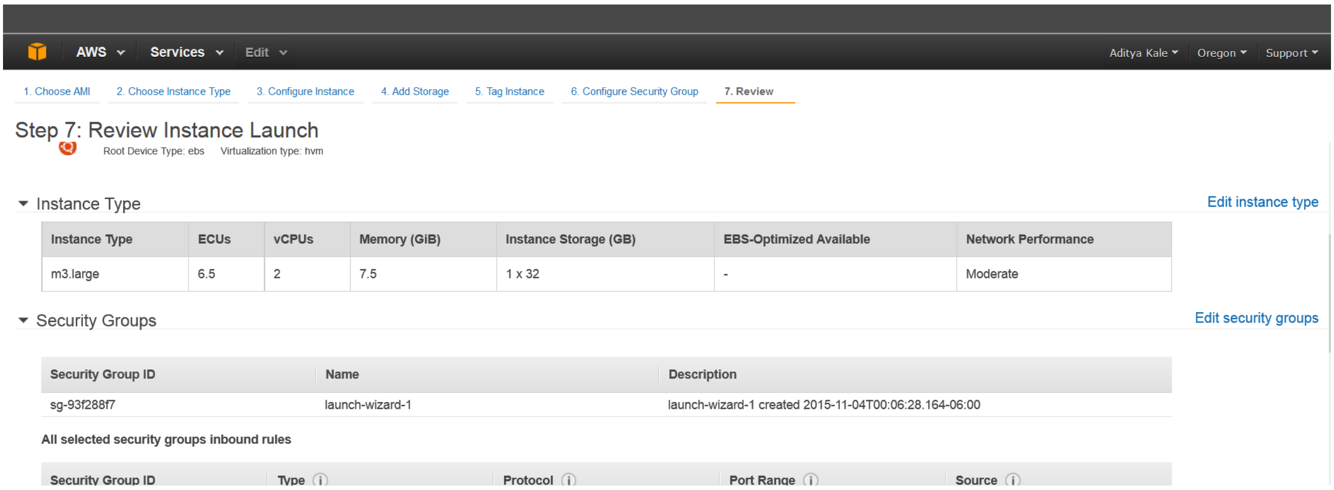
Conclusion:

- **As the number of clients increases time required to download files also increases for both the system**. Because as concurrent request increases amount of bytes transfer on stream also increases which increase the cost of operation.
- **Time taken** for downloading 10,000 files through **distributed system is less than** the time taken by a **centralized system**.
- Distributed system take less time because request for searching a file machine address goes to a particular servers based on a file hash code. Which reduces the load of all other servers of a distributed system.
- However in centralized system all search request goes to a Centralized Indexing server increasing its load. As the concurrent request increases Centralized Indexing server takes more time to respond to a request thus increasing total time of further operation.

Evaluation on Amazon Web Service

Instance: EC2 M3.large

Configuration:



Step 7: Review Instance Launch

Root Device Type: ebs   Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
m3.large	6.5	2	7.5	1 x 32	-	Moderate

Security Groups

Security Group ID	Name	Description
sg-93f288f7	launch-wizard-1	launch-wizard-1 created 2015-11-04T00:06:28.164-06:00

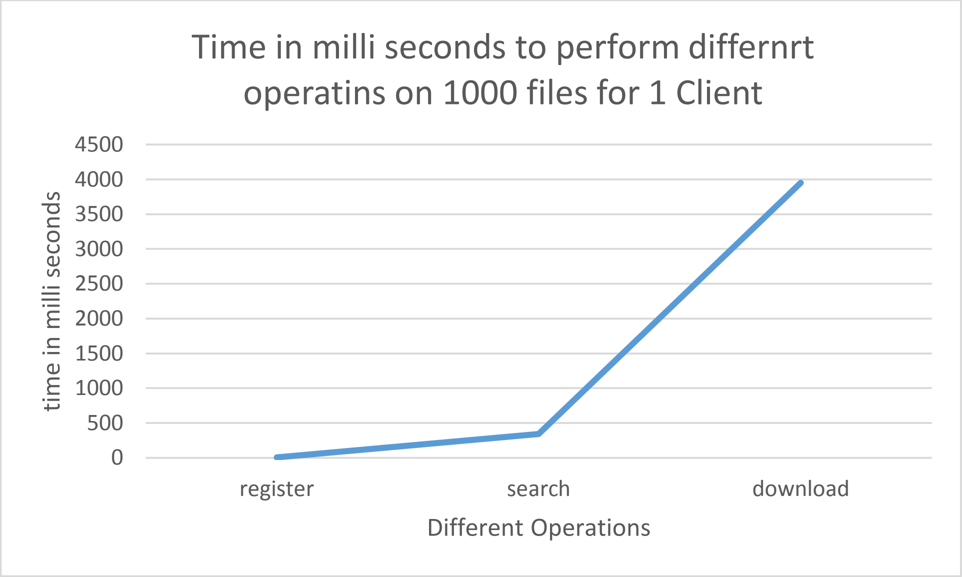
All selected security groups inbound rules

Security Group ID	Type	Protocol	Port Range	Source

- Performance of a Distributed file sharing system is measured by generating **1000** sequential **request** of **file size 100KB** from a single client. And then varying the number of clients for concurrent requests.
- First a single client is run to make 1000 requests for Register, Search, Download with a different Key-value pairs.
- Then the number of clients are increased gradually and their average time for execution are noted.
- Following results are observed for multiple clients performing various operations

operation	client1(Time in mills)
Register	8
Search	341
Download	3953

- ➢ A line graph is plotted using the above observations.

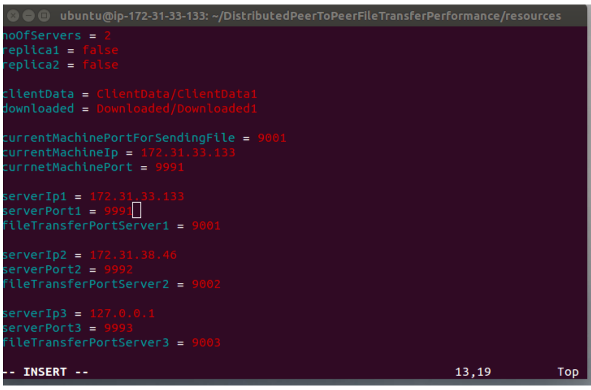


- ➢ **Observation**:
  - ○ Download operation is taking more time than other two operations.
  - ○ Search is taking slightly greater time than register operation.

Number of Clients = 2

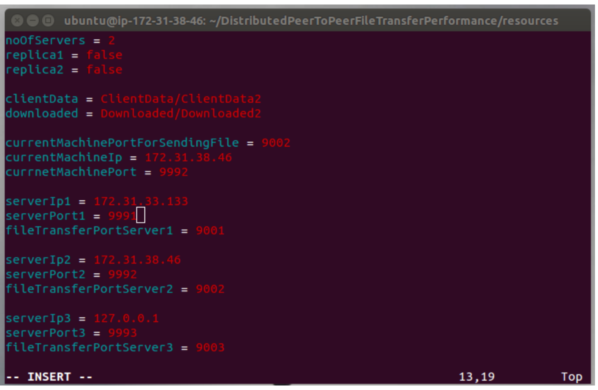
- ➢ Two client are run to make 1000 requests for Register, Search, Download with a different Key.
- ➢ Following results are observed for 2 clients performing various operations.

Configuring clients on AWS instances:



ubuntu@ip-172-31-33-133: ~/DistributedPeerToPeerFileTransferPerformance/resources

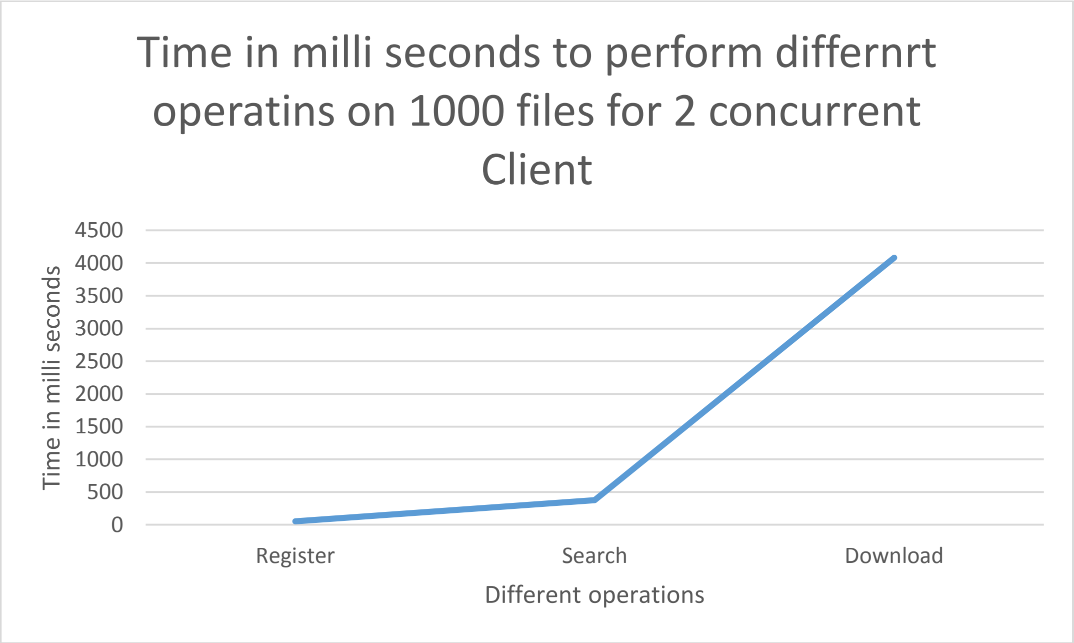
```
noOfServers = 2
replica1 = false
replica2 = false

clientData = ClientData/ClientData1
downloaded = Downloaded/Downloaded1

currentMachinePortForSendingFile = 9001
currentMachineIp = 172.31.33.133
currnetMachinePort = 9991

serverIp1 = 172.31.33.133
serverPort1 = 9991
fileTransferPortServer1 = 9001

serverIp2 = 172.31.38.46
serverPort2 = 9992
fileTransferPortServer2 = 9002

serverIp3 = 127.0.0.1
serverPort3 = 9993
fileTransferPortServer3 = 9003

-- INSERT --                                    13,19         Top
```



ubuntu@ip-172-31-38-46: ~/DistributedPeerToPeerFileTransferPerformance/resources

```
noOfServers = 2
replica1 = false
replica2 = false

clientData = ClientData/ClientData2
downloaded = Downloaded/Downloaded2

currentMachinePortForSendingFile = 9002
currentMachineIp = 172.31.38.46
currnetMachinePort = 9992

serverIp1 = 172.31.33.133
serverPort1 = 9991
fileTransferPortServer1 = 9001

serverIp2 = 172.31.38.46
serverPort2 = 9992
fileTransferPortServer2 = 9002

serverIp3 = 127.0.0.1
serverPort3 = 9993
fileTransferPortServer3 = 9003

-- INSERT --                                    13,19         Top
```

operations	client1(Time in mills)	client2(Time in mills)	Average time in milliseconds
register	44	61	**52.5**
search	379	375	**377**
download	4291	3875	**4083**

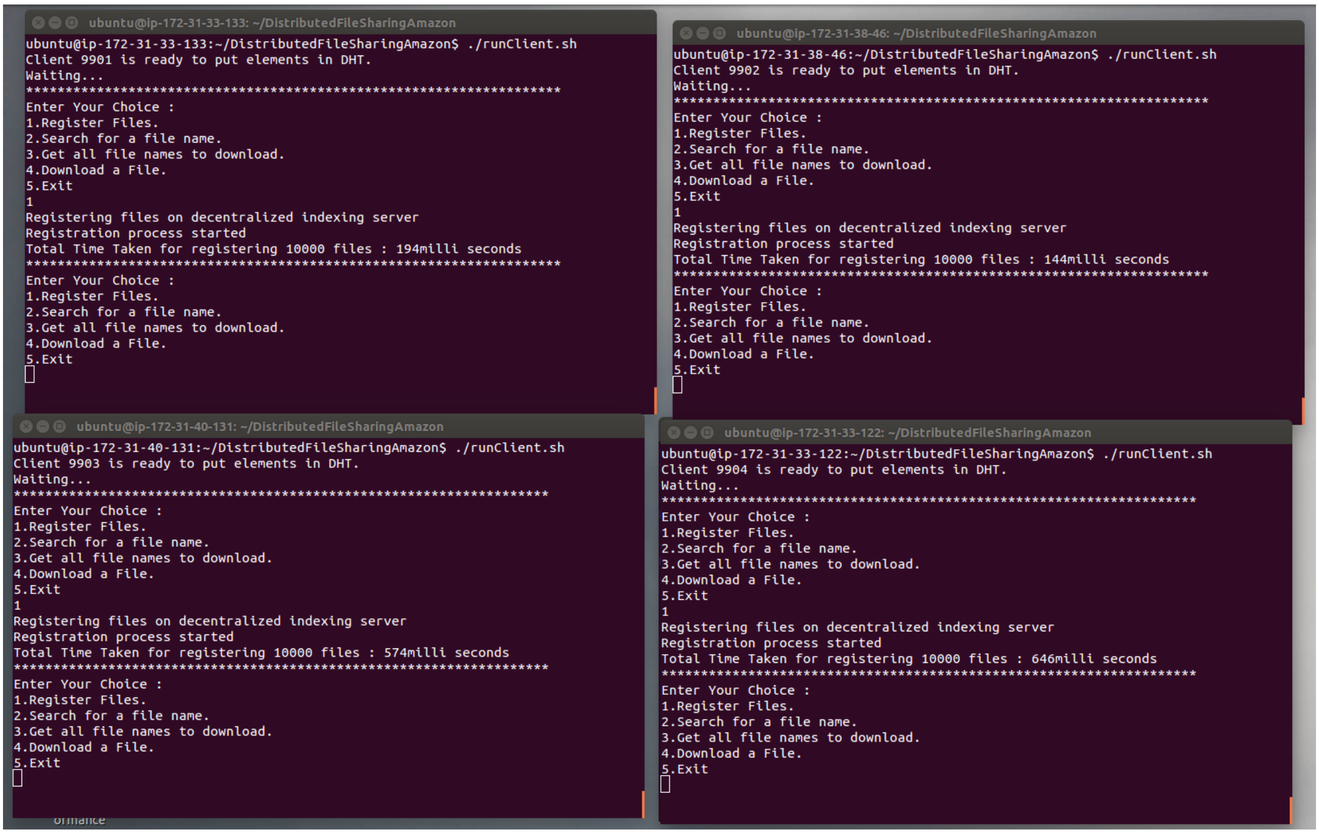


- **Observation**:
  - Download operation is taking more time than other two operations.
  - Search is taking slightly greater time than register operation.

Number of Clients = 4

- Four client are run to make 1000 requests for Register, Search, Download with a different Key.
- Following results are observed for 4 clients performing various operations.

Register on 4 clients.



ubuntu@ip-172-31-33-133: ~/DistributedFileSharingAmazon
ubuntu@ip-172-31-38-46: ~/DistributedFileSharingAmazon
ubuntu@ip-172-31-40-131: ~/DistributedFileSharingAmazon
ubuntu@ip-172-31-33-122: ~/DistributedFileSharingAmazon
ormance

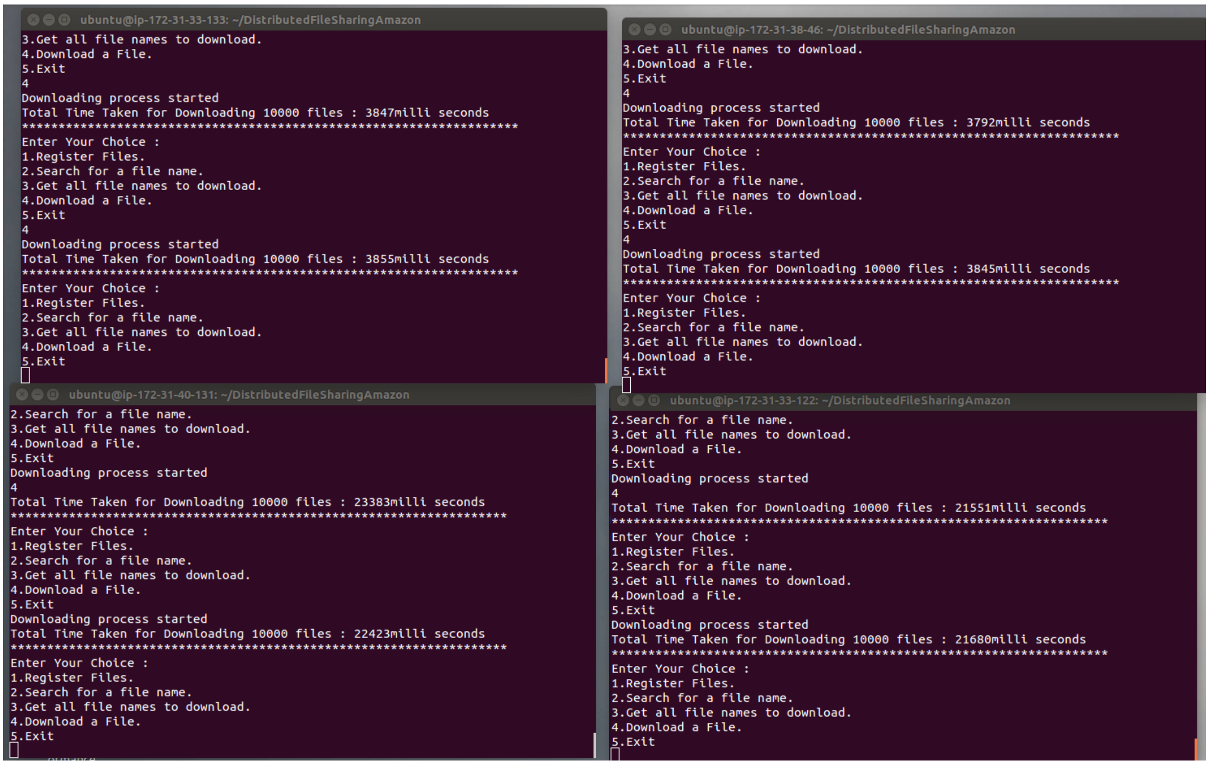
```
ubuntu@ip-172-31-33-133:~/DistributedFileSharingAmazon$ ./runClient.sh
Client 9901 is ready to put elements in DHT.
Waiting...
*********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
1
Registering files on decentralized indexing server
Registration process started
Total Time Taken for registering 10000 files : 194milli seconds
*********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
```

```
ubuntu@ip-172-31-38-46:~/DistributedFileSharingAmazon$ ./runClient.sh
Client 9902 is ready to put elements in DHT.
Waiting...
*********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
1
Registering files on decentralized indexing server
Registration process started
Total Time Taken for registering 10000 files : 144milli seconds
*********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
```

```
ubuntu@ip-172-31-40-131:~/DistributedFileSharingAmazon$ ./runClient.sh
Client 9903 is ready to put elements in DHT.
Waiting...
*********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
1
Registering files on decentralized indexing server
Registration process started
Total Time Taken for registering 10000 files : 574milli seconds
*********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
```

```
ubuntu@ip-172-31-33-122:~/DistributedFileSharingAmazon$ ./runClient.sh
Client 9904 is ready to put elements in DHT.
Waiting...
*********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
1
Registering files on decentralized indexing server
Registration process started
Total Time Taken for registering 10000 files : 646milli seconds
*********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
```

Download on 4 clients:


ubuntu@ip-172-31-33-133: ~/DistributedFileSharingAmazon
ubuntu@ip-172-31-38-46: ~/DistributedFileSharingAmazon
ubuntu@ip-172-31-40-131: ~/DistributedFileSharingAmazon
ubuntu@ip-172-31-33-122: ~/DistributedFileSharingAmazon

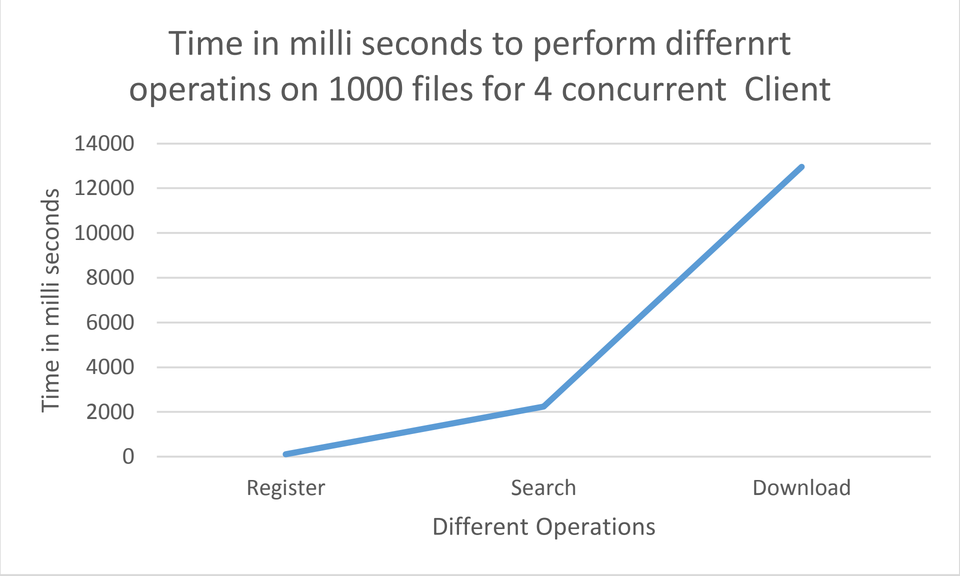
```
3.Get all file names to download.
4.Download a File.
5.Exit
4
Downloading process started
Total Time Taken for Downloading 10000 files : 3847milli seconds
**********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
4
Downloading process started
Total Time Taken for Downloading 10000 files : 3855milli seconds
**********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
```

```
3.Get all file names to download.
4.Download a File.
5.Exit
4
Downloading process started
Total Time Taken for Downloading 10000 files : 3792milli seconds
**********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
4
Downloading process started
Total Time Taken for Downloading 10000 files : 3845milli seconds
**********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
```

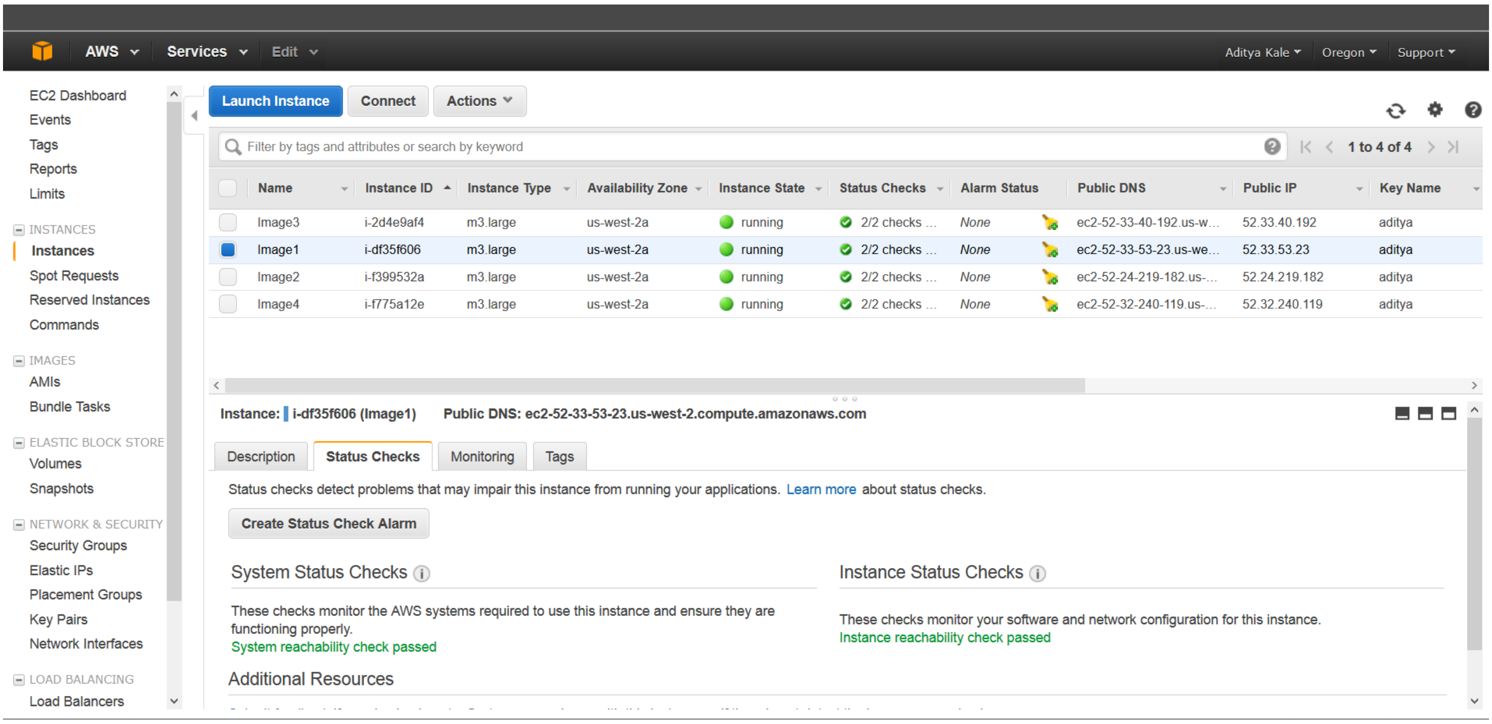
```
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
Downloading process started
4
Total Time Taken for Downloading 10000 files : 23383milli seconds
**********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
Downloading process started
Total Time Taken for Downloading 10000 files : 22423milli seconds
**********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
```

```
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
Downloading process started
4
Total Time Taken for Downloading 10000 files : 21551milli seconds
**********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
Downloading process started
Total Time Taken for Downloading 10000 files : 21680milli seconds
**********************************************************************
Enter Your Choice :
1.Register Files.
2.Search for a file name.
3.Get all file names to download.
4.Download a File.
5.Exit
```

operations	client1(Time in mills)	client2(Time in mills)	client3(Time in mills)	client4(Time in mills)	Average time in milliseconds
register	43	44	181	180	**112**
search	338	341	4085	4207	**2242.75**
download	3855	3845	22423	21680	**12950.75**



- **Observation**:
    - Download operation is taking more time than other two operations.
    - Register and search time is not increasing that much.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name
Image3	i-2d4e9af4	m3.large	us-west-2a	running	2/2 checks ...	None	ec2-52-33-40-192.us-w...	52.33.40.192	aditya
Image1	i-df35f606	m3.large	us-west-2a	running	2/2 checks ...	None	ec2-52-33-53-23.us-we...	52.33.53.23	aditya
Image2	i-f399532a	m3.large	us-west-2a	running	2/2 checks ...	None	ec2-52-24-219-182.us-...	52.24.219.182	aditya
Image4	i-f775a12e	m3.large	us-west-2a	running	2/2 checks ...	None	ec2-52-32-240-119.us-...	52.32.240.119	aditya

Instance: i-df35f606 (Image1)   Public DNS: ec2-52-33-53-23.us-west-2.compute.amazonaws.com

Status checks detect problems that may impair this instance from running your applications. Learn more about status checks.

System Status Checks

These checks monitor the AWS systems required to use this instance and ensure they are functioning properly.
System reachability check passed

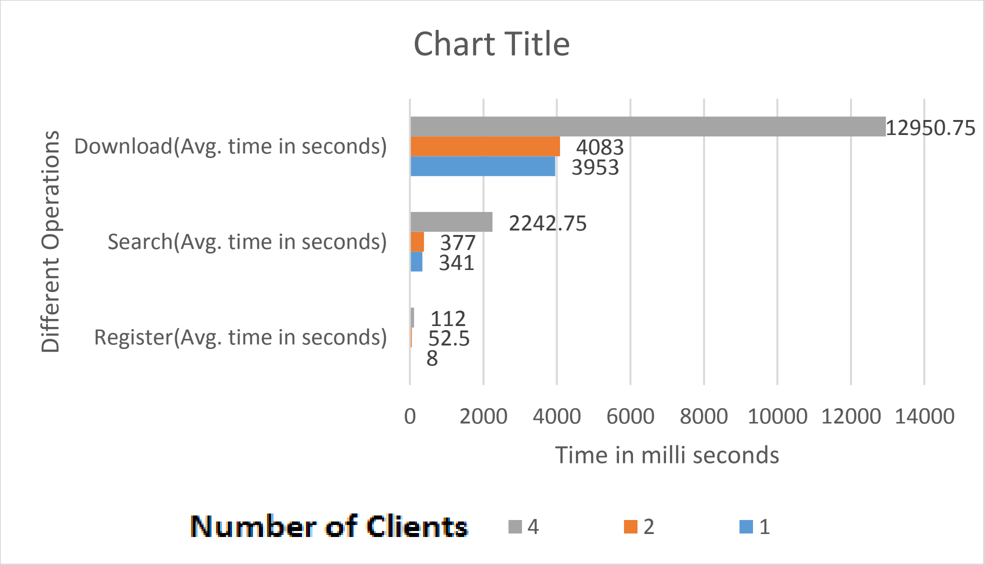
Instance Status Checks

These checks monitor your software and network configuration for this instance.
Instance reachability check passed

Additional Resources

Final Result:

Observations for different number of clients:



Final Conclusion:

- As the number of client increases the time for performing each operation increases gradually. Time taken for registering and searching files is increased by very small amount.
- Time required to perform all operation is reduced by huge amount.