

Performance Evaluation

Part 1:

- To evaluate the performance of the system, especially, REGISTER, SEARCH and OBTAIN operations, I have created a Test program (file TestClient.java)
- The Test programs asks you the type of operation you want to evaluate – REGISTER/SEARCH/OBTAIN and then it executes (sends) 10k requests for the selected operation.
- This program was run on various combinations of the input parameters. So the test environment was as follows:

Total no. of peers connected: 1

Configuration of each peer: 2 core processor, 512 MB RAM virtual machine (Ubuntu Linux OS)

Total no. of requests: 10,000 (10k requests each)

Results are as follows:

Sr. No.	REGISTER Operation	SEARCH Operation	OBTAIN Operation
1	53 milliseconds	16 milliseconds	66 seconds

Average response time for REGISTER request: **0.0053 milliseconds**

Average response time for SEARCH request: **0.0016 milliseconds**

Average response time for OBTAIN request including download: **6.6 milliseconds**

Average File Transfer Speed: **10.24 MBps**

Time is less because all the files will hash to the same client and will be registered (entry stored in hash table) to self because there is only one client in the network.

Total no. of peers connected: 2

Configuration of each peer: 2 core processor, 512 MB RAM virtual machine (Ubuntu Linux OS)

Total no. of requests: 20,000 (10k requests each)

Results are as follows:

Sr. No.	REGISTER Operation	SEARCH Operation	OBTAIN Operation
1	213.69 seconds	215.1 seconds	82 seconds
2	213.69 seconds	214.7 seconds	80 seconds
AVG.	213.69 seconds	214.9 seconds	81 seconds

Average response time for REGISTER request: **21.36 milliseconds**

Average response time for SEARCH request: **21.49 milliseconds**

Average response time for OBTAIN request including download: **8.1 milliseconds**

Average File Transfer Speed: **8.33 MBps**

Total no. of peers connected: 4

Configuration of each peer: 2 core processor, 512 MB RAM virtual machine (Ubuntu Linux OS)

Total no. of requests: 40,000 (10k requests each)

Results are as follows:

Sr. No.	REGISTER Operation	SEARCH Operation	OBTAIN Operation
1	327.55 seconds	315.82 seconds	107 seconds
2	328 seconds	316.37 seconds	116 seconds
3	328 seconds	315.02 seconds	105 seconds
4	325.1 seconds	310.86 seconds	98 seconds
AVG.	327.16 seconds	314.51 seconds	106.5 seconds

Average response time for REGISTER request: **32.71 milliseconds**

Average response time for SEARCH request: **21.49 milliseconds**

Average response time for OBTAIN request including download: **10.65 milliseconds**

Average File Transfer Speed: **6.31 MBps**

Total no. of peers connected: 8

Configuration of each peer: 2 core processor, 512 MB RAM virtual machine (Ubuntu Linux OS)

Total no. of requests: 80,000 (10k requests each)

Results are as follows:

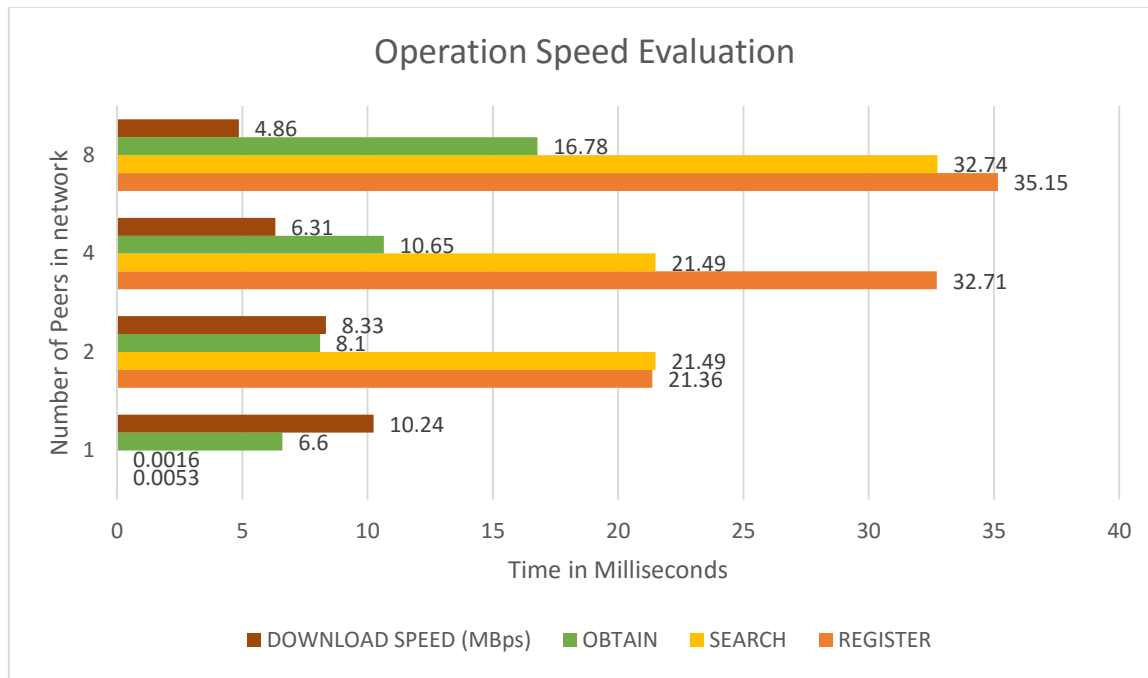
Sr. No.	REGISTER Operation	SEARCH Operation	OBTAIN Operation
1	352.12 seconds	328.87 seconds	168 seconds
2	352.30 seconds	328.77 seconds	179 seconds
3	352.29 seconds	328.02 seconds	170 seconds
4	348.90 seconds	326.12 seconds	174 seconds
5	350.23 seconds	327.10 seconds	163 seconds
6	351.83 seconds	325.45 seconds	160 seconds
7	354.52 seconds	329.14 seconds	158 seconds
8	350.13 seconds	326.40 seconds	171 seconds
AVG.	351.54 seconds	327.48 seconds	167.87 seconds

Average response time for REGISTER request: **35.15 milliseconds**

Average response time for SEARCH request: **32.74 milliseconds**

Average response time for OBTAIN request including download: **16.78 milliseconds**

Average File Transfer Speed: **4.86 MBps**



Observation:

- In the above evaluations, we can observe that the average time for REGISTER, SEARCH and OBTAIN requests increase as the number of peers increase in the network. While the average download (file transfer) speed decreases as the number of peers in the network increase because the peers constantly demand files and hence the communication link between the peers get more and more busy.
- These tests were performed on a peer having low configuration. The average response time for the REGISTER, SEARCH and OBTAIN requests will decrease and the average download speed will increase as the configuration of peer becomes strong (i.e. faster processor, more number of cores and more RAM).
- The response time for serving REGISTER and SARCH request if there is only one peer in the network is very small (i.e. 0.0016 milliseconds) because the file names are stored and retrieved from itself (same peer).
- The response time stated above includes connection to the peer time, server processing time and sending the results and connection closing time.

Part 2:

- This test was done to check the throughput which we achieve through different file sizes. 8 peers were connected in the network and file of size 1KB, 50KB, 1MB, 50MB, 400MB, and 1GB were registered by each peer.
- The file was downloaded (obtained) by all the eight peers concurrently and average download speed was calculated. This program was run on various combinations of the input parameters. So the test environment was as follows:

Total no. of peers connected: 8

Configuration of each peer: 2 core processor, 512 MB RAM virtual machine (Ubuntu Linux OS)

Total no. of file downloading requests: 8 (1 request each)

File size tested: 1KB, 50KB, 1MB, 50MB, 400MB, and 1GB

Peer Number	File Size (Speed in Bytes per second)							
	1 KB	50 KB	500 KB	1 MB	50 MB	400 MB	1 GB	AVERAGE
1	26497	461261	2426540	4017532	6247473	8078896	2532134	4827190
2	12047	360563	1418282	4017532	6171724	5831577	5083265	3270713
3	4357	656410	2089795	6432981	6439302	7196321	5881541	4100101
4	17694	1600000	4571428	5461333	7180060	6881323	5110953	4403256
5	4266	320000	1651612	6168094	4360708	6531555	5487952	3503455
6	36571	269473	1689768	5518821	4087378	7525845	4915478	3434762
7	6168	742028	1199063	5115004	4045432	8202894	5548954	3551363
8	6781	292571	1187935	3666349	531835	7633779	5369874	2669875
AVERAGE	14297	587788	2029303	5049706	4882989	8485274	4991269	3541518

Average file transfer speed for the entire test is **35, 41, 518 bytes per second = 3.377 MBps**

Observation:

The average file transfer speed is 3.377 MBps. While developing the application and debugging, I got somewhat same transfer speed. As we can see from the last row in the above table, the average speed is somewhat same for files ranging from 500 KB – 1 GB. There is a minute difference because of we can't say how much memory or processing power is available at this time. Moreover, three virtual machines out of 8 were clone of first 3 machines. So, the performance of these cloned virtual machines were dependent on their parent virtual machines.

Comparison with Centralized File Transfer System (Assignment 1):

- Similar tests as above were performed on Centralized File Transfer System as done in Assignment 1.

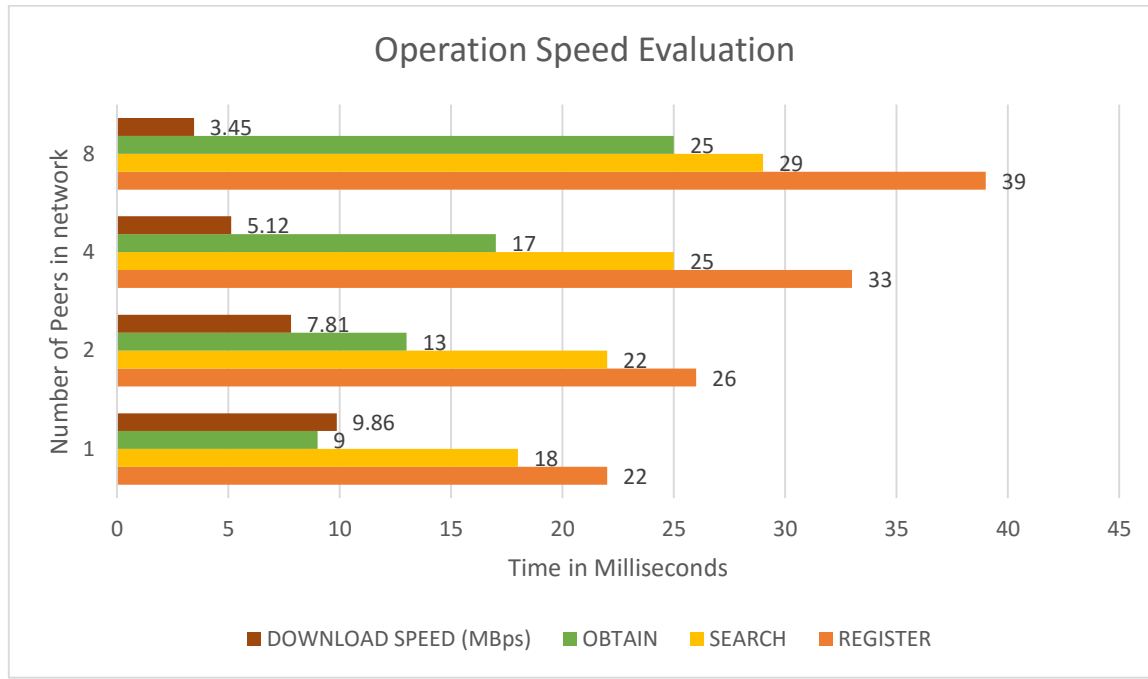
Configuration of each peer: 2 core processor, 512 MB RAM virtual machine (Ubuntu Linux OS)

Total no. of requests: 10,000 (10k requests each)

Results are as follows:

Peer Count	REGISTER Operation (Avg. Response Time)	SEARCH Operation (Avg. Response Time)	OBTAIN Operation (Avg. Response Time)
1	22 milliseconds	18 milliseconds	9 milliseconds
2	26 milliseconds	22 milliseconds	13 milliseconds
4	33 milliseconds	25 milliseconds	17 milliseconds
8	39 milliseconds	29 milliseconds	25 milliseconds

Average File Transfer Speed: **3.87 MBps**



Observation:

- In the above evaluations, we can observe that the average time for REGISTER, SEARCH and OBTAIN requests increase as the number of peers increase in the network because the communication link between the peers and the indexing server gets more and more busy. While the average download (file transfer) speed decreases as the number of peers in the network increase because the peers constantly demand files and hence the communication link between the peers gets more and more busy.

Conclusion:

- As we can see that the average response time for serving the requests in case of centralized system is more than compared to the average response time in de-centralized system because in case of centralized system all the requests except the download request is served by only one machine i.e. the indexing server. Whereas in the case if de-centralized system, the requests are divided among the different peers using the hashing mechanism.
- The difference between the response times in both the systems may seem to be less here in a real-time scenario where millions of files can be shared using the system, the difference will be more, and thus we'll see the benefit of distributed system vs. the centralized system.