Performance Evaluation

Test environment:

- Processor : Intel Core i5
- RAM : 6GB
- Server and clients running on the same system using different ports

Evaluation set 1:

Registering 10K files of size 1Kb from clients. Slowly the number of concurrent clients is increased ranging from 1 to 8

1. **Test 1:** Running Single client with 10K requests. The unit of time is ms

	Register	Search	Download
Client 1	1280	901	5637

Average response time per register request = 1280 / 10000 = **0.128 ms**
Average response time per search request = 901 / 10000 = **0.0901 ms**
Average response time per download request = 5637 / 10000= **0.5637 ms**

No of Register operations per second = (10000/1280)*1000 = **7810**
No of Search operations per second = (10000/901)*1000 = **11090**
No of Download operations per second = (10000/5637)*1000 = **1770**

2. **Test 2:** Running two clients simultaneously with 10K requests.

	Register	Search	Download
Client 1	1474	1023	7237
Client 2	1372	975	6835
Avg	**1423**	**999**	**7036**

Average response time per register request = 1423 / 10000 = **0.1423 ms**
Average response time per search request = 999 / 10000 = **0.0999 ms**
Average response time per download request = 7036 / 10000 = **0.7036 ms**

Average No of Register operations per second = (10000/1423)*1000 = **7020**
Average No of Search operations per second = (10000/999)*1000 = **10010**
Average No of Download operations per second = (10000/7036)*1000 = **1420**

3. **Test 4:** Running four clients simultaneously with 10K requests.

	Register	Search	Download
Client 1	2893	1184	12505
Client 2	2239	1264	12281
Client 3	2587	1040	12518
Client 4	2194	996	12300
Avg	**2478.25**	**1121**	**12401**

Average response time per register request = 2478.25 / 10000 = **0.247825 ms**
Average response time per search request = 1121 / 10000 = **0.1121 ms**
Average response time per download request = 12401 / 10000 = **1.2401 ms**

Average No of Register operations per second = (10000/2478.25)*1000 = **4030**
Average No of Search operations per second = (10000/1121)*1000 = **8920**
Average No of Download operations per second = (10000/12401)*1000 = **800**

4. **Test 5:** Running Six clients simultaneously with 10k requests.

	Register	Search	Download
Client 1	4542	1329	24024
Client 2	4710	1229	18281
Client 3	4823	1163	18636
Client 4	4150	1196	18352
Client 5	4994	1108	17892
Client 6	4489	1186	18235
Avg	**4618**	**1201.83**	**19236.66667**

Average response time per register request = 4618 / 10000 = **0.4618 ms**
Average response time per search request = 1201.83 / 10000 = **0.120183 ms**
Average response time per download request = 19236.7 / 10000 = **1.92367 ms**

Average No of Register operations per second = (10000/4618)*1000 = **2160**
Average No of Search operations per second = (10000/1201.83)*1000 = **8320**
Average No of Download operations per second = (10000/19236.7)*1000 = **520**

5. **Test 6:** Running Eight clients simultaneously with 100000 requests.

	Register	Search	Download
Client 1	6419	1492	35123
Client 2	6180	1402	29845
Client 3	6057	1485	28748
Client 4	6326	1303	27432
Client 5	6972	1283	29836
Client 6	5692	1263	26585
Client 7	5831	1222	29836
Client 8	4800	1103	28265
Avg	**6034.63**	**1319.13**	**29458.75**

Average response time per register request = 6034.63 / 10000 = **0.603463 ms**
Average response time per search request = 1319.13 / 10000 = **0.131913 ms**
Average response time per download request = 29458.75 / 10000 = **2.945875 ms**

Average No of Register operations per second = (10000/6034.63)*1000 = **1650**
Average No of Search operations per second = (10000/1319.13)*1000 = **7580**
Average No of Download operations per second = (10000/29458.75)*1000 = **330**

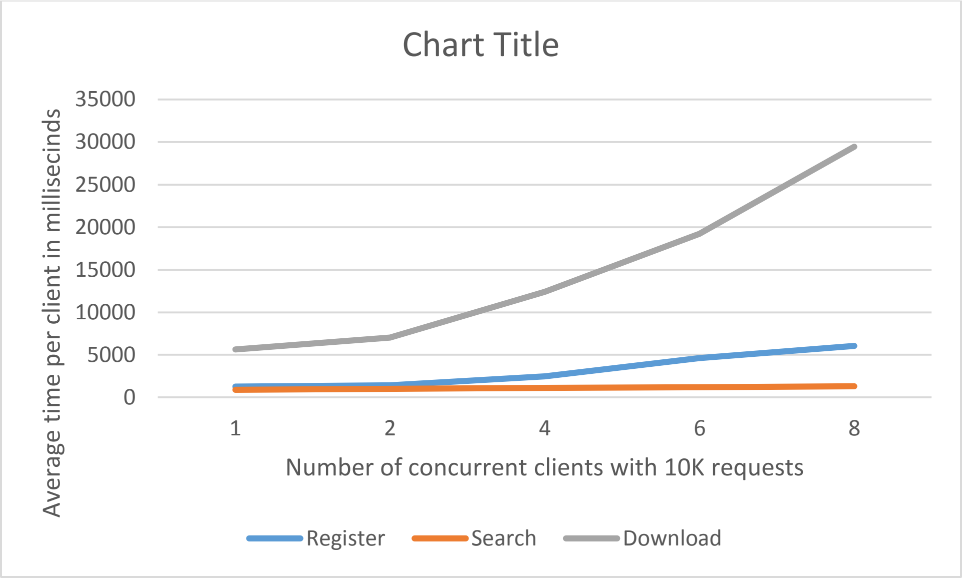
Analysis:

Below is the analysis of the system with increasing number of simultaneous clients

The table shows the average time for register, search and download requests (10K requests) per client. The average time per client for all the operations is increasing with increasing number of concurrent clients.

No of Clients	Register	Search	Download
1	1280	901	5637
2	1423	999	7036
4	2478.25	1121	12401
6	4618	1201.83	19236.66
8	6034.63	1319.13	29458

Below is graphical representation of the testing



Observation and Conclusion:

- Among all three operations download takes more time compared to register and search. This is because of the data transferred over the network. Large **amount of data is transferred** in case of download request Search takes less time since less amount of data is transferred in search request.
- As the number of **concurrent clients increases the average time per client also increases**. This is because severs are now getting multiple concurrent requests. If it's a register request then the sever writes data to a map. In case of concurrent requests the threads lock the data for writing. Hence increases waiting time.
- The experiment conducted above is **on a single system** so in case of multiple download request, the **disk IO** plays important role. All the clients are using same hardware resources for writing and reading from the disk.
- In case of 8 clients, Disk IOs create a huge latency slowing down the entire system
- **Throughput of the system decreases with increased number of concurrent clients and at the same time the response time increases.**

Evaluation Set 2:

Performance of the system will be measured in terms of the throughput (bytes per seconds)

Test performed:

- The number of clients and server is fixed to 8.
- Files with different sizes are transferred to see the change in throughput.

Below table shows the measurements in different cases

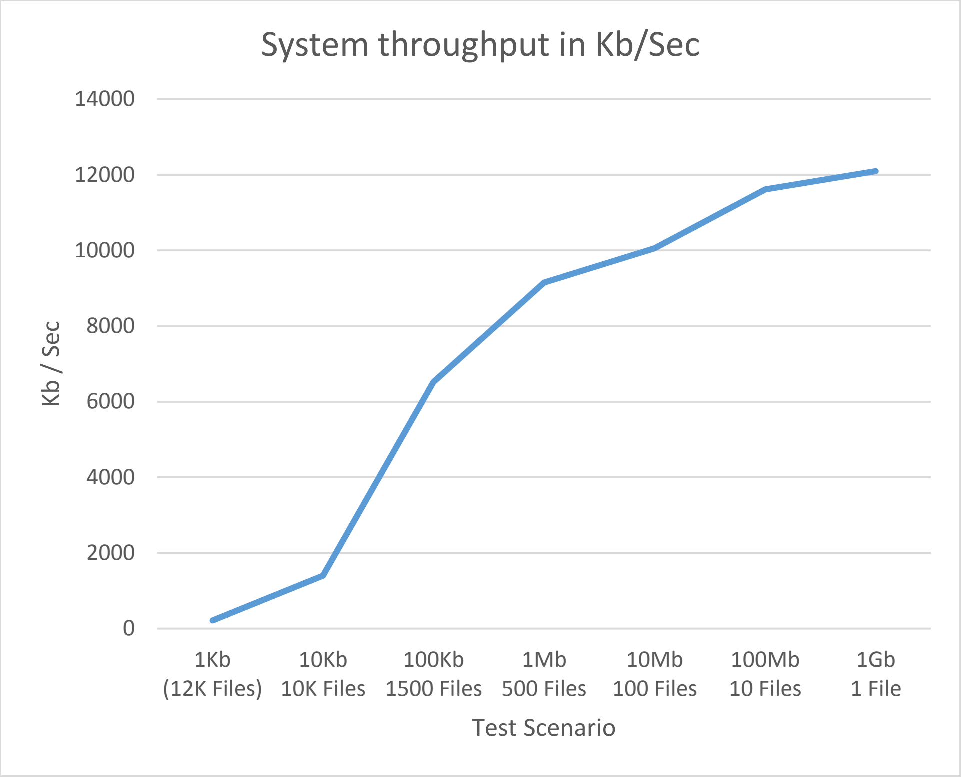
File size	No of files	Total bytes transfered	Client1	Client2	Client3	Client4	Client5	Client6	Client7	Client8
1Kb	12000	12000000	60234	59372	58934	59421	54934	56093	55233	52378
10Kb	10000	100000000	94224	73765	69866	70895	105434	54324	58654	43655
100Kb	1500	150000000	35627	37234	35637	15928	16827	13968	14983	13826
1Mb	500	500000000	62836	49454	54234	47344	55857	58345	48435	60586
10Mb	100	1000000000	92384	120656	114865	114857	113968	107435	60533	70543
100Mb	10	1000000000	90475	92385	84576	87645	89584	92355	76485	75342
1Gb	1	1000000000	47245	91534	83245	93245	75342	91345	93857	85432

Analysis of the experiment:

Below table gives the average time per client and throughput in each case

Avg time per Client	Avg in sec	Throught (bytes per seconds)	Kb/Sec
57074.875	57.074875	210250.132	210.25
71352.125	71.352125	1401499.955	1401.50
23003.75	23.00375	6520675.977	6520.67
54636.375	54.636375	9151412.406	9151.41
99405.125	99.405125	10059843.49	10059.84
86105.875	86.105875	11613609.41	11613.61
82655.625	82.655625	12098390.16	12098.39

Graphical representation of the throughput change in all cases



Observation and Conclusion:

- As the number of files transferred increases the **throughput of the system decreases**.
- Let's take an example to understand this. In case of 1 file to be transferred the code for processing a download request executes for only one time on a server. In case of multiple download request the code for processing the download request runs for same times as the number of requests. This is one of the reasons of low throughput of the system
- If the downloading code takes 1 ms to to execute then for 1 file request it will contribute only 1 ms in the total time and in case of 10K request it will contribute 10000 ms in the total time.

Evaluation set 3:

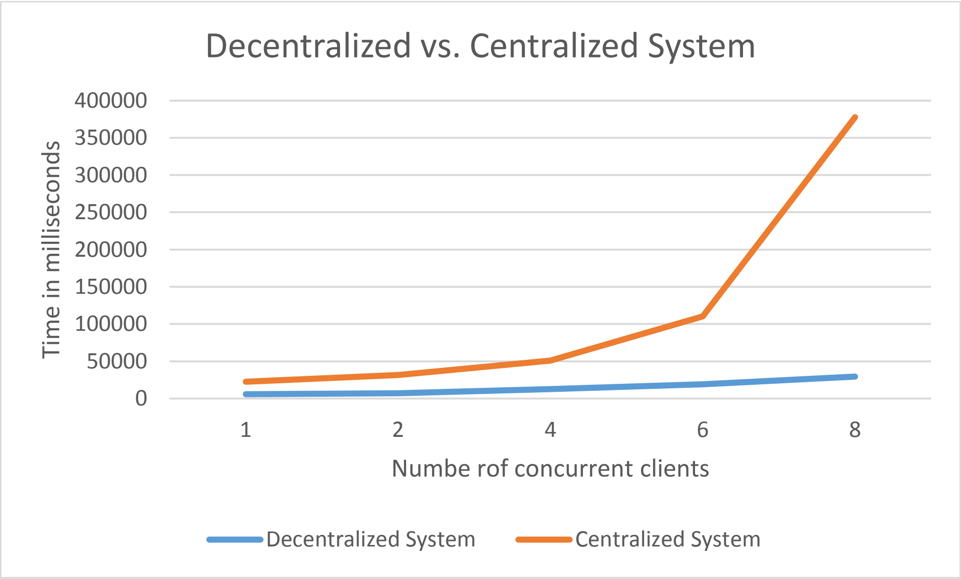
Comparing performance of centralized system with the distributed system

The comparison of the systems is done by transferring 10K 1bk files with increasing number of concurrent clients.

The table below gives the average time per client in each case for transferring 10k files.

No of concurrent clients	Decentralized System	Centralized System
1	5637	22560
2	7036	31604.7
4	12401	50697
6	19236.66	110285
8	29458	377545

The graphical representation of the comparison



Observation and Conclusion:

- From the above graph it is clear that the decentralized system has a better performance than the centralized system.
- The centralized system cannot handle load. The response time grows exponentially with increasing number of concurrent requests.

- In decentralized system the load gets balanced among multiple servers. In our assignment the hash function evenly distributes the requests among the servers.
- So the number of requests per server are less compared to the centralized system as all the requests go to only one sever.