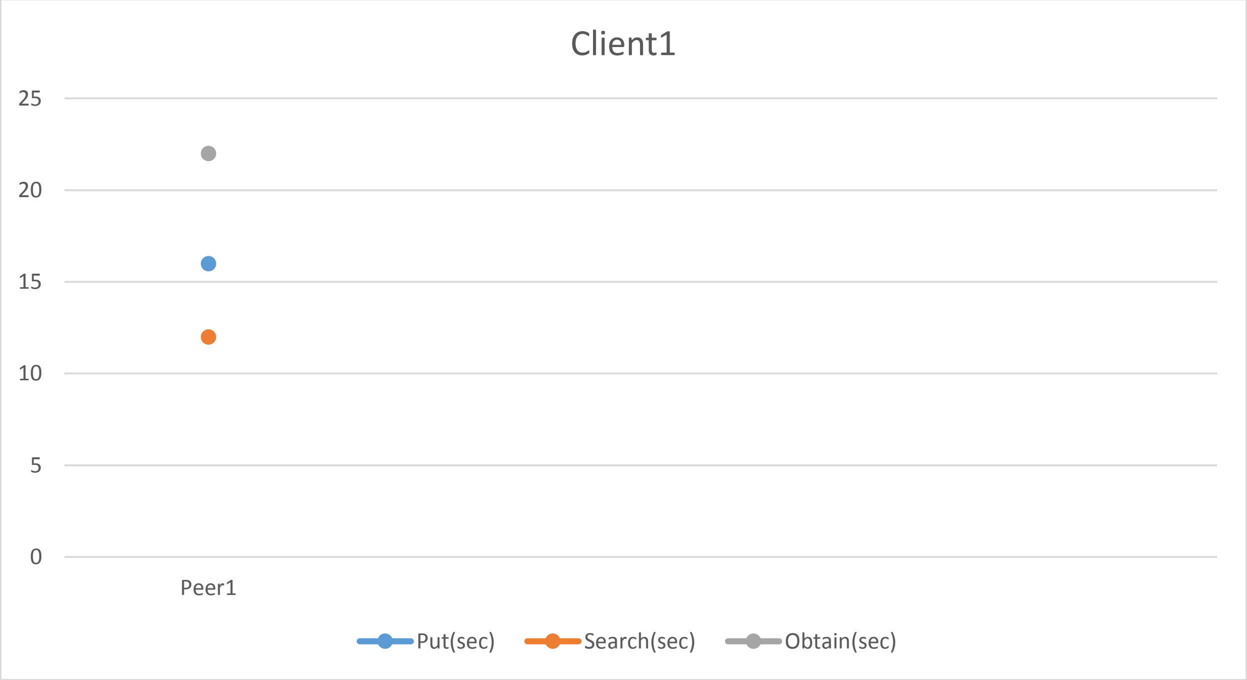
PERFORMANCE EVALUATION

- **First set of experiment:**
  8 Servers are up. First, 1 client performs 10k operations each of put, search and obtain. Then 2 concurrent clients perform 20k, 4 concurrent clients perform 40k and eventually 8 clients perform 80k.

1. Client 1 performing 10 k operations each of put, search and obtain.

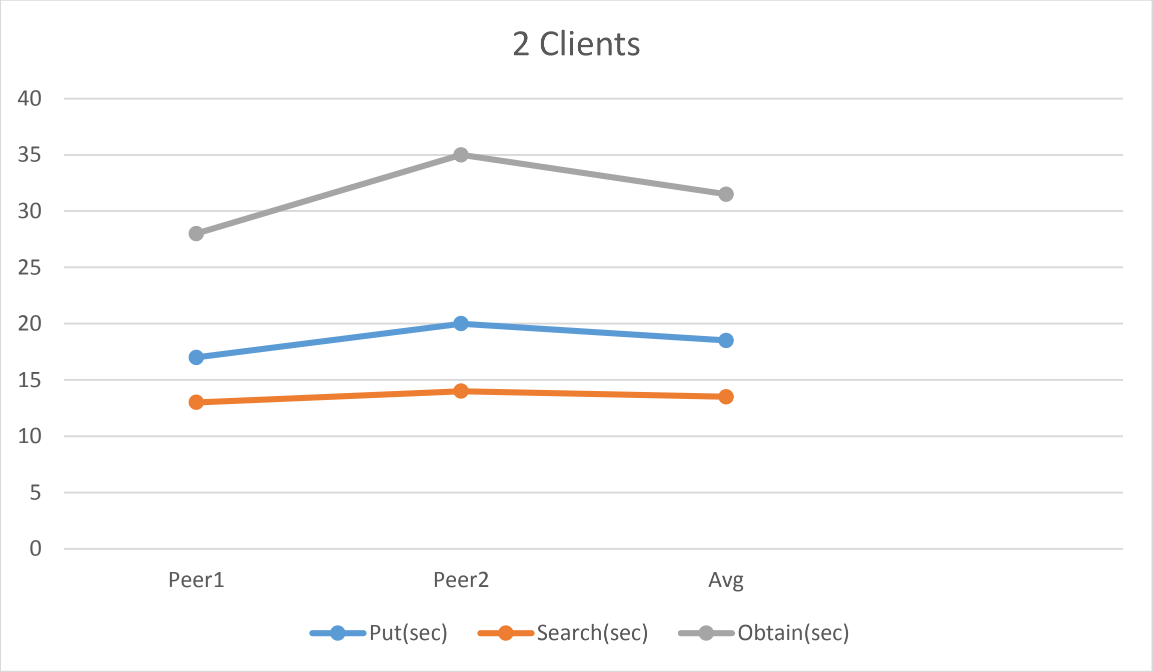
Client	Put(sec)	Search(sec)	Obtain(sec)
1	16	12	22



Here, 10k files are kept with Peer2. Peer1 registers, searches and obtains 10k files from peer2. Avg response time for peer 1 is shown above.

2. Client 1 and Client 2 each performing 10k operations each of put, search and obtain so in all 20k operations.

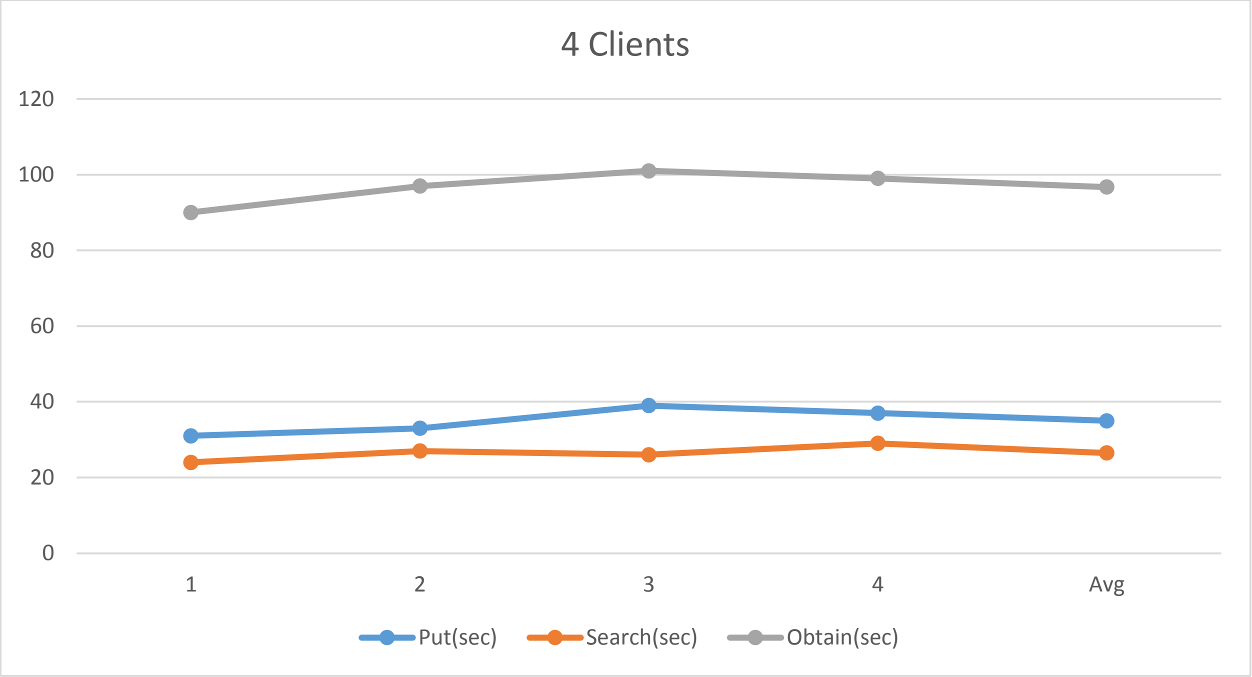
Client	Put(s)	Search(s)	Obtain(s)
1	17s	13s	28s
2	20s	14s	35s
Avg	18.5s	13.5s	31.5s



Here, 10k files are kept with Peer1 and 10k with Peer2. Peer1 and Peer2 now concurrently run 10k operations for put, search and obtain. Peer1 registers its files and searches and downloads files from Peer2. Peer2 does vice-versa. As it can be seen from table and graph above when 2 clients are running concurrently, avg response time for search increases negligibly while for registering and obtaining increases slightly. This is obvious from the fact that with increase in number of requests the time taken to serve request increases. To verify this, on single peer 20k operations are performed. It was seen that results didn't vary much.

3. Clients 1,2,3,4 running concurrently each performing 10k operations each of put, search and obtain so in all 40k operations.

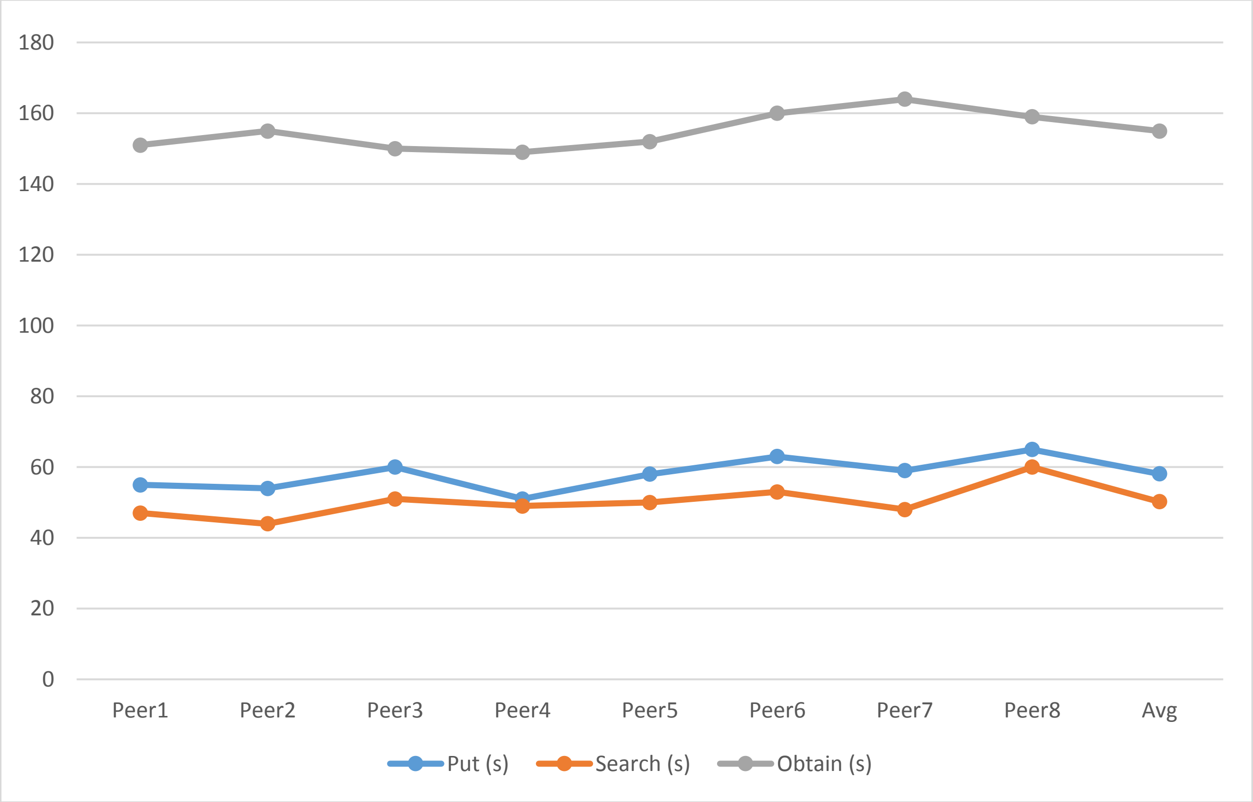
Client	Put	Search	Obtain
1	31	24	90
2	33	27	97
3	39	26	101
4	37	29	99
Avg	35	26.5	96.75



Here, 10k files are kept with Peer1, 10k with Peer2, 10k with Peer3 and 10k with Peer4. Peer1, Peer2, Peer3 and Peer4 now concurrently run 10k operations for put, search and obtain. Peer1 registers its files and searches and downloads files from Peer2, Peer3 and Peer4. Same for Peer2, Peer3 and Peer4. As it can be seen from tables and graphs above when 4 clients are running concurrently, avg response time for registering, searching and obtaining increases compared to that of 2 clients. This is also obvious from the fact that with increase in number of requests the time taken to serve request increases. To verify this, 40k operations were done by single peer and it was observed that avg response time remained almost same.

4. **Clients 1,2,3,4,5,6,7,8 running concurrently each performing 10k operations each of put, search and obtain so in all 80k operations.**

Clients	Put (s)	Search (s)	Obtain (s)
1	55	47	151
2	54	44	155
3	60	51	150
4	51	49	149
5	58	50	152
6	63	53	160
7	59	48	164
8	65	60	159
Avg	58.125	50.25	155



Here, 10k files are kept with each of Peer1, Peer2, Peer3, Peer4, Peer5, Peer6, Peer7 and Peer8. Now all 8 peers concurrently run 10k operations for put, search and obtain. Peer1 registers its files and searches and downloads files from remaining 7 peers. Same for rest of peers. As it can be seen from tables and graphs above when 8 clients are running concurrently, avg response time for registering, searching and obtaining increases compared to that of 4 clients. This is very much obvious from the fact that the number of requests has increased significantly and so the time taken to serve request increases. To verify this, 80k operations were done by single peer and it was observed that avg response time didn't vary much.

Throughput when 8 clients and 8 servers are running in terms of operations per second:

= 10k/58.12 put operations/sec

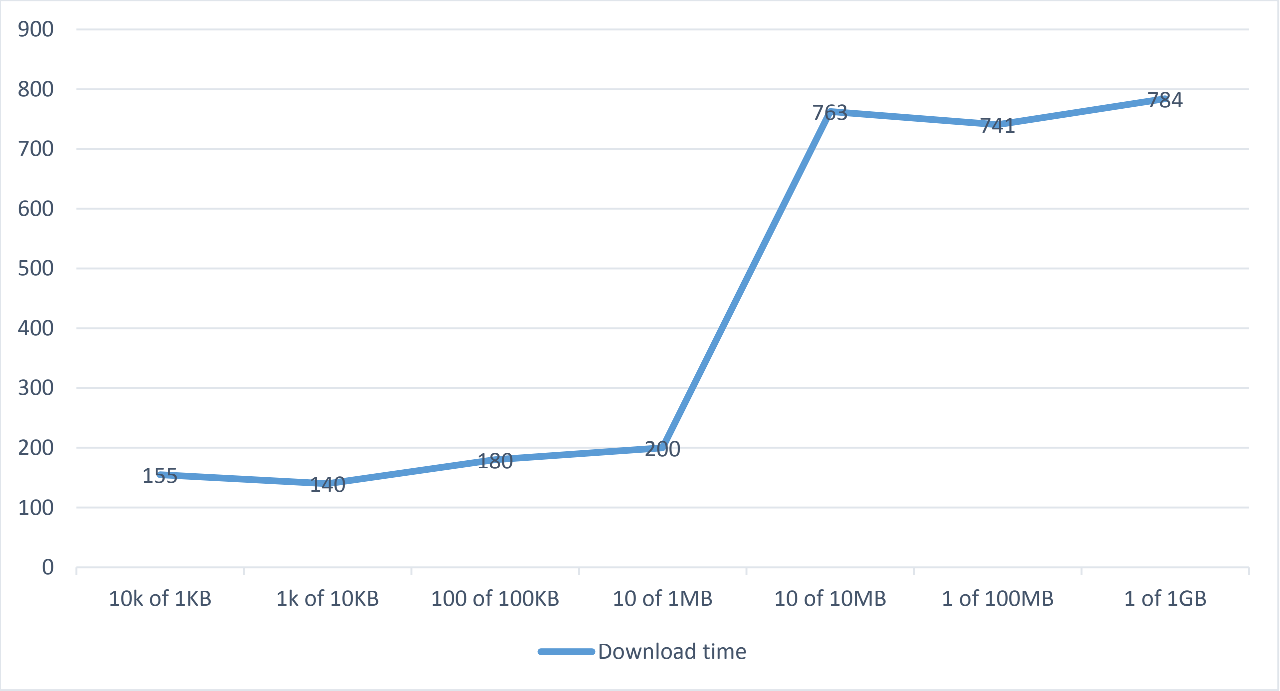
= 10k/50.25 search operations/sec

= 10k/155 obtain operations/sec

- ## Second set of experiments:

To measure throughput with different file sizes, experiments were carried out with 8 peers having 10k files of 1 KB, 1k files of 10KB, 100 files of 100KB, 10 files of 1MB, 10 files of 10MB, 1 file of 100MB and 1 file of 1GB in 7 different rounds of experiment.

10k of 1KB	1k of 10KB	100of100KB	10 of 1MB	10 of 10MB	1 of 100MB	1 of 1 GB
155 sec	140 sec	180 sec	200 s	763 sec	741 sec	784 sec



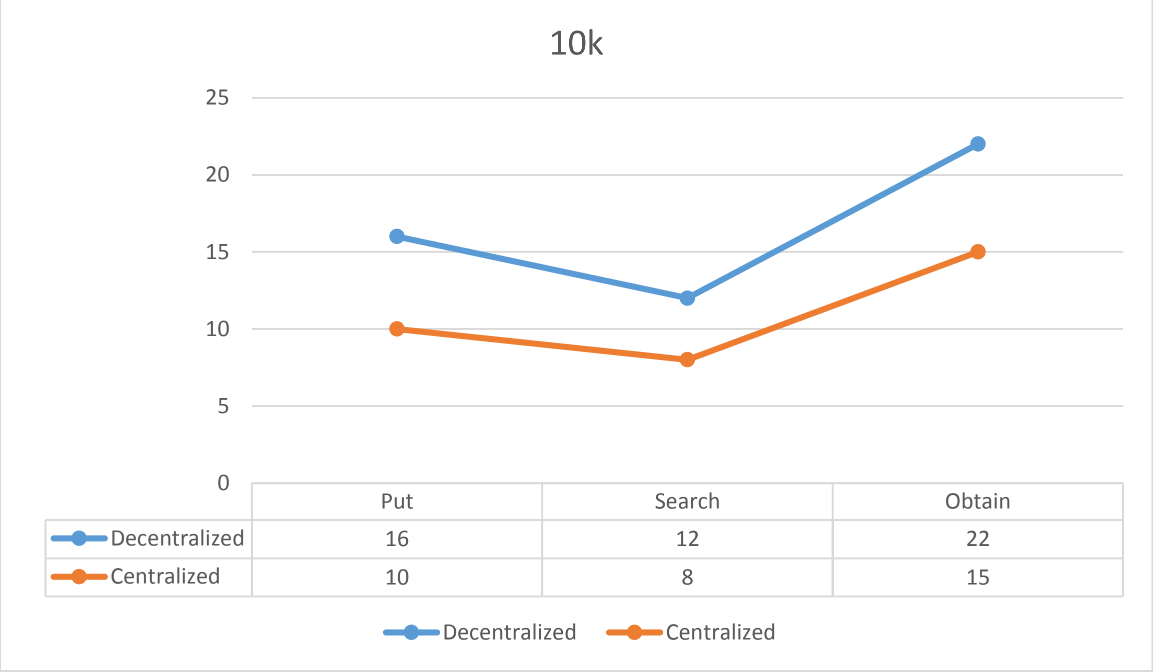
Average throughput = sum(bytes)/ sum(sec) = 114748 bytes per second.

This was verified by considering the hardware of the system on which this decentralized p2p file sharing system was executed, this throughput value approximately matches that of the system.

Comparison of Centralized and Decentralized p2p file sharing system:

- We know that centralized approach is good when the number of nodes in the distributed system is less. Decentralized approach is preferred is the number of nodes in the distributed system is significantly large.
- From the experiments performed it was seen that using centralized approach average response time for registering a file was 3 microseconds and for lookup was 1 microseconds while using decentralized approach it is 5 milliseconds and 1 millisecond. This is because in decentralized approach calculating hash function causes overhead.
- However, for downloading, the average response time remains almost same as after all in both centralized and decentralized system the file is downloaded from other peers.

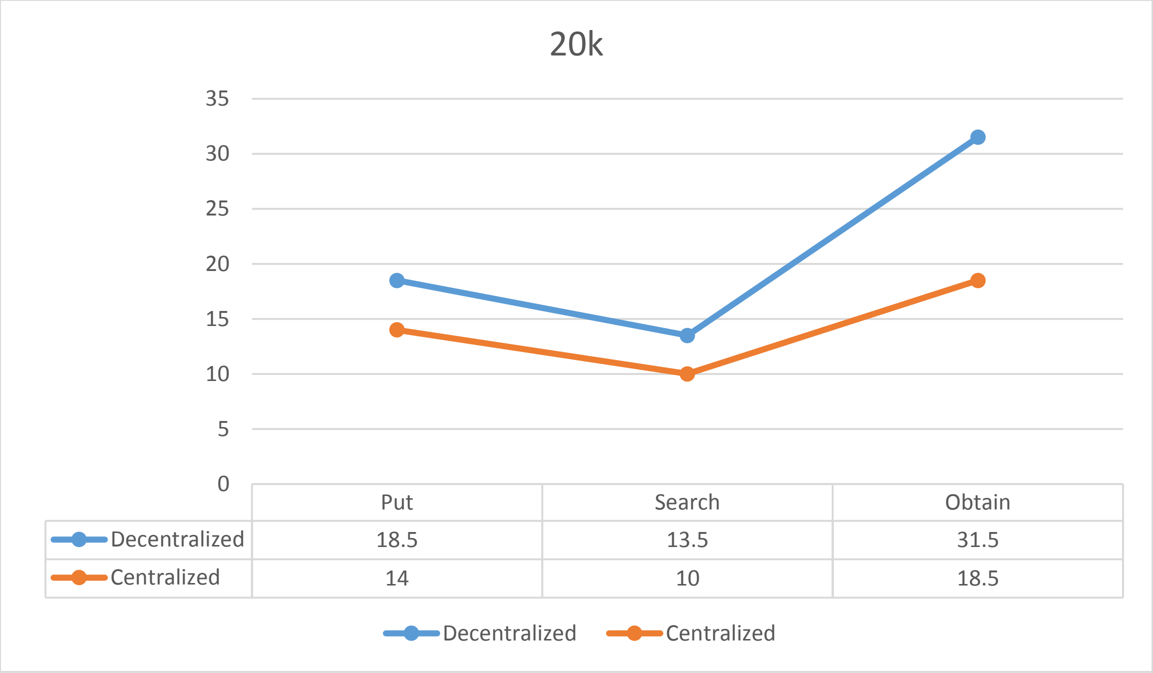
- ## Comparison for 10k operations:



	Put	Search	Obtain
Decentralized	16	12	22
Centralized	10	8	15

➢ Here, Y-axis represent the average time in seconds.
➢ The graph shows comparison for 10k operations of centralized and decentralized p2p file sharing system by taking average response time for 10k put, search and obtain operations.
➢ Files of 1kb size were used.

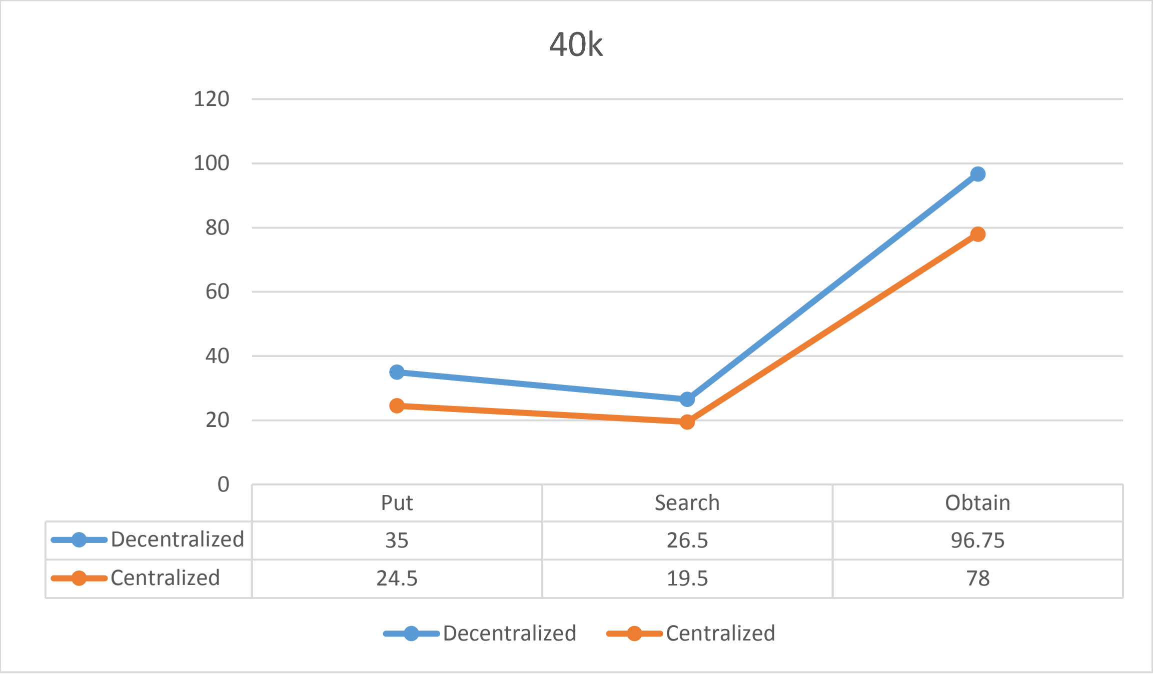
- **Comparison for 20k operations:**



	Put	Search	Obtain
Decentralized	18.5	13.5	31.5
Centralized	14	10	18.5

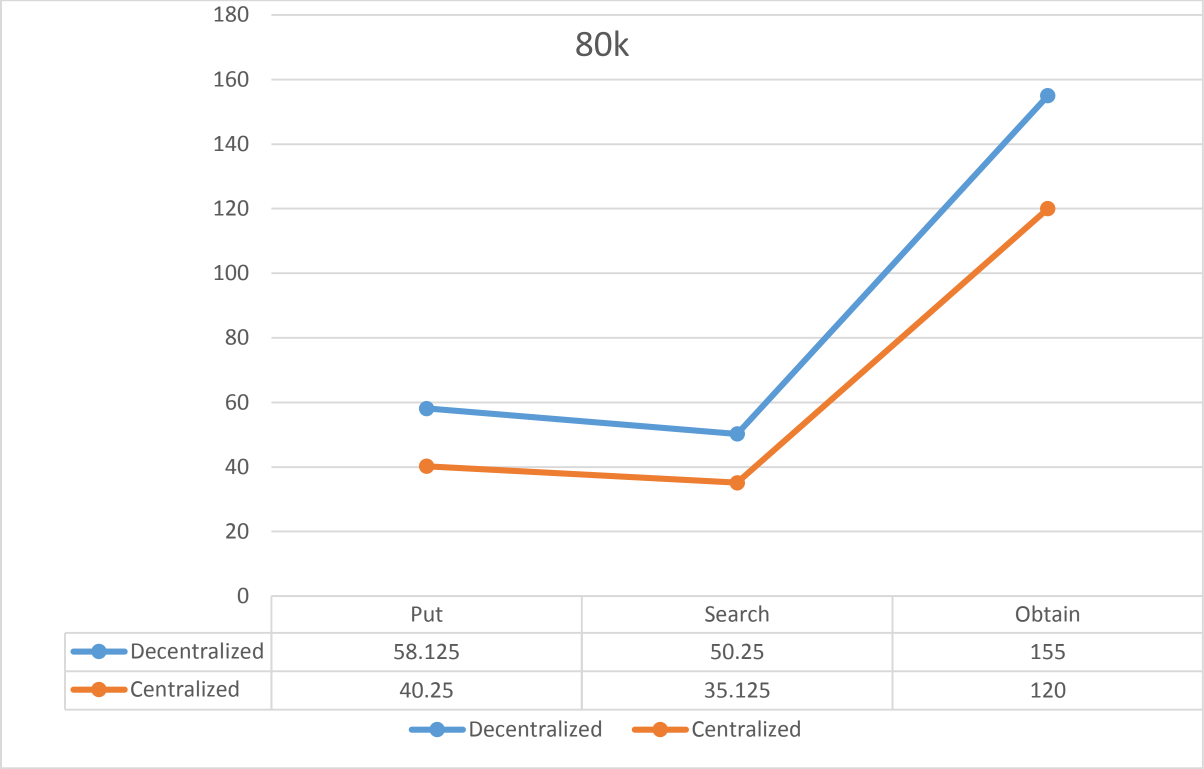
- ➢ Here, Y-axis represent the average time in seconds.
- ➢ The graph shows comparison for 20k operations of centralized and decentralized p2p file sharing system by taking average response time for 20k put, search and obtain operations.
- ➢ Files of 1kb size were used.

- **Comparison for 40k operations:**



- ➢ Here, Y-axis represent the average time in seconds.
- ➢ The graph shows comparison for 40k operations of centralized and decentralized p2p file sharing system by taking average response time for 40k put, search and obtain operations.
- ➢ Files of 1kb size were used.

- **<u>Comparison for 80k operations:</u>**



	Put	Search	Obtain
Decentralized	58.125	50.25	155
Centralized	40.25	35.125	120

- ➢ Here, Y-axis represent the average time in seconds.
- ➢ The graph shows comparison for 80k operations of centralized and decentralized p2p file sharing system by taking average response time for 80k put, search and obtain operations.
- ➢ Files of 1kb size were used.

<u>NOTE:</u> As expected the average response time taken by decentralized system is greater than centralized system as the maximum number of nodes here taken is 8 which is not that big number.