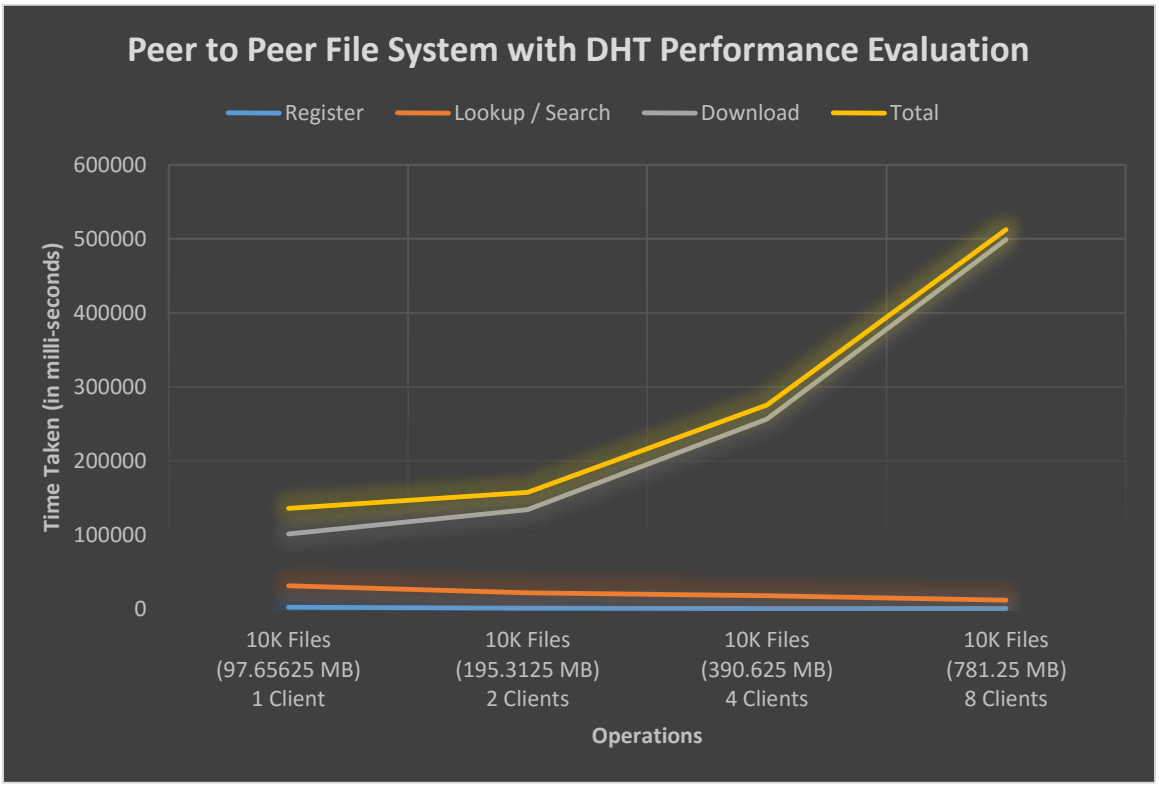
1. Performance Evaluation

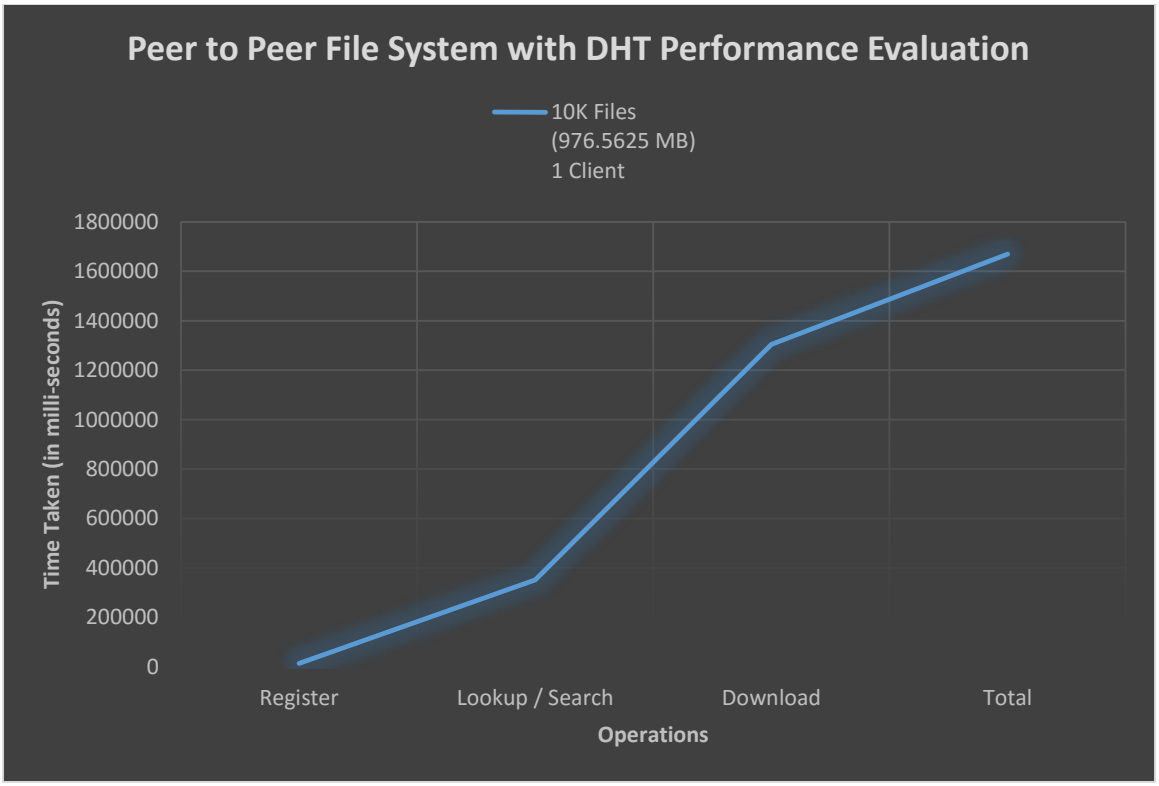
a. 10K Files each of size 10K

Sr. No.	Operation	10K Files (97.65625 MB) 1 Client	10K Files (195.3125 MB) 2 Clients	10K Files (390.625 MB) 4 Clients	10K Files (781.25 MB) 8 Clients
1.	Register	2616 mSec	1500 mSec	1037 mSec	929 mSec
2.	Lookup / Search	31719 mSec	21927 mSec	18063 mSec	12143 mSec
3.	Download	101637 mSec	134451 mSec	256701 mSec	499236 mSec
4.	Total	135972 mSec	157878 mSec	275801 mSec	512308 mSec



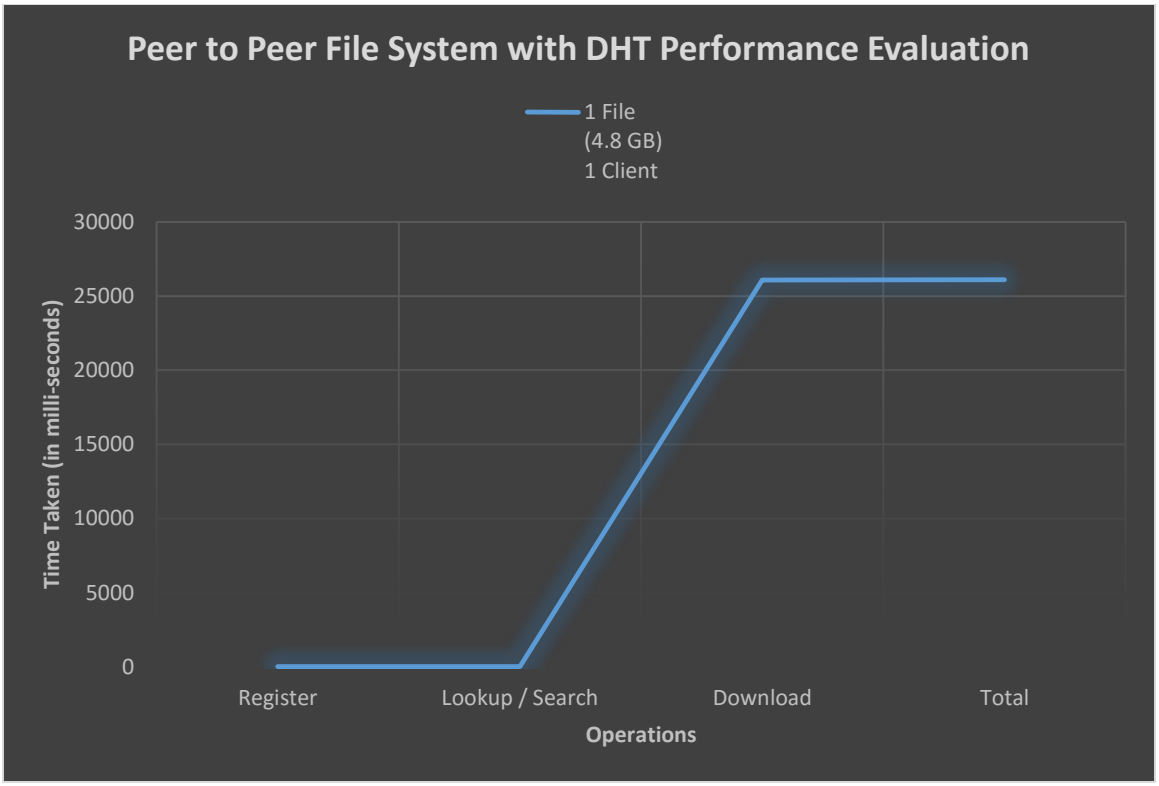
b. 100K Files each of size 10K

Sr. No.	Operation	100K Files (976.5625 MB) 1 Client
1.	Register	13611 mSec
2.	Lookup / Search	351682 mSec
3.	Download	1304557 mSec
4.	Total	1669850 mSec



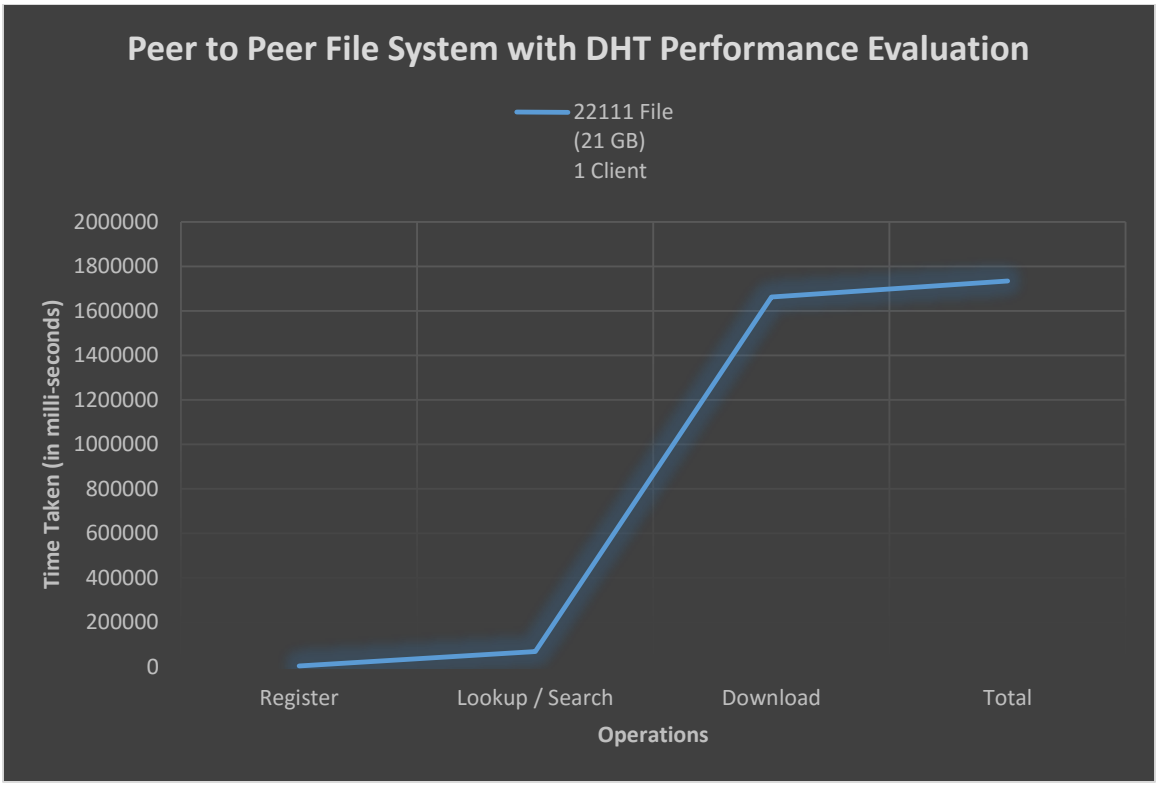
c. 1 file of 4.8 GB

Sr. No.	Operation	1 File (4.8 GB) 1 Client
1.	Register	9 mSec
2.	Lookup / Search	19 mSec
3.	Download	26082 mSec
4.	Total	26110 mSec



d. 22111 file of 21 GB

Sr. No.	Operation	22111 File (21 GB) 1 Client
1.	Register	4335 mSec
2.	Lookup / Search	68591 mSec
3.	Download	1661789 mSec
4.	Total	1734715 mSec



e. Comparative Analysis & Summary

- Distributed Indexing Server / architecture (Assignment 3) definitely helps to support larger databases as compared to the Centralized Indexing Server / architecture (Assignment 1).
- It also helps from the Data Resilience perspective as Indexing Server is still assumed to be a server class node instead of some Desktop / Laptop / PC nodes.
- Assuming the Download Size to be constant, distributed / decentralized architecture will yield in performance improvement.
- Distributed / decentralized architecture will also help to scale the System by adding new nodes as Indexing Server. This will help in in better load balancing.