Team:
Preeti Yadav (201751039)
Vaidehi Vaishnav (201751059)
Prakhar Gupta (201751036)
Keshav Purohit (201751021)

Book Sharing

9th November 2020

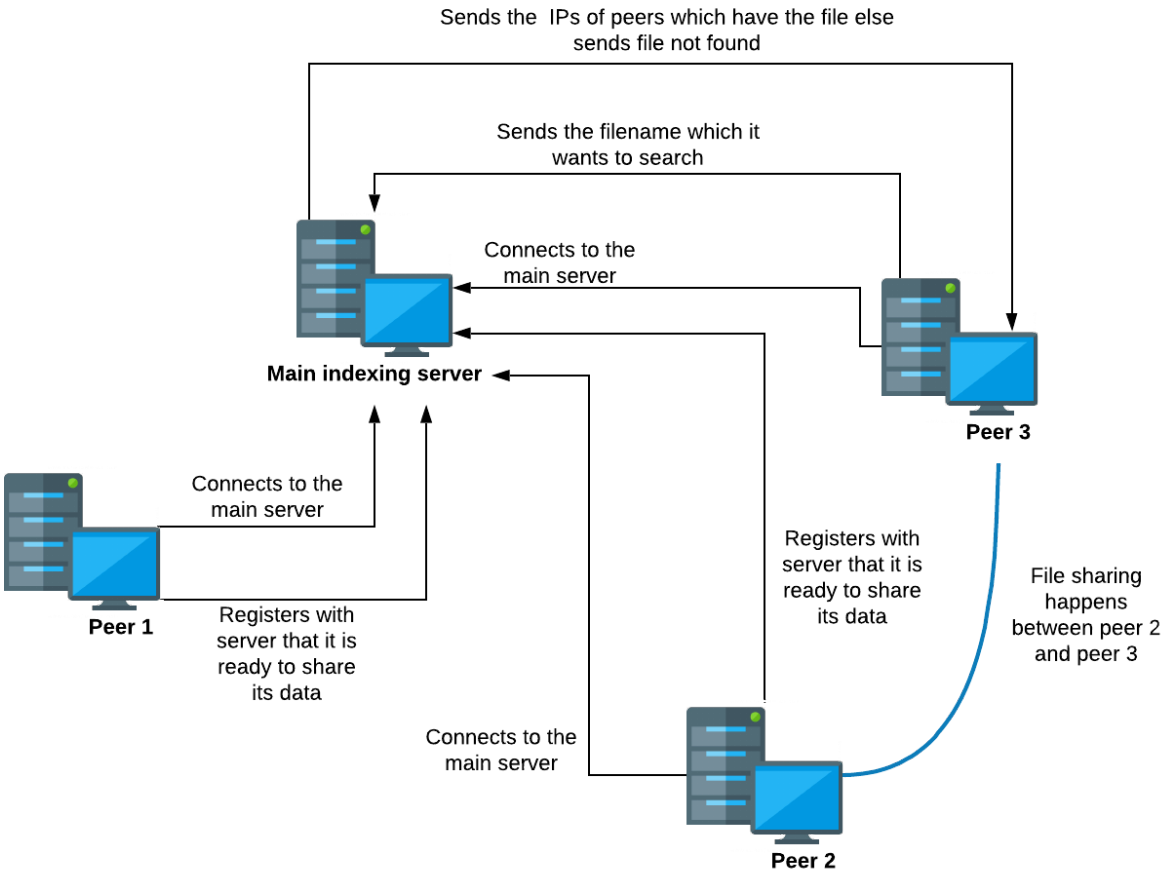
PROJECT ANALYSIS

- Peer to Peer is a file sharing technology, allowing users to access mainly the multimedia files like videos, music, e-books, games etc.
- The individual users in this network are referred to as peers.
- The peers request for the files from other peers by establishing TCP or UDP connections.
- When one peer makes a request, it is possible that multiple peers have the copy of that requested object.
- The problem is how to get the IP addresses of all those peers such that the pdf transfer takes place between two peers only.
- Centralized servers can become overloaded when there are too many clients requesting for the data which will slow down the process.
- The same files may be used by several clients, it is a great opportunity to coordinate the file transfer among clients in the peer-to-peer (P2P) manner to reduce the data transfer time.
- So, using this we create a book sharing system which is free and distributed for everyone on the network who is ready to share his books and enjoy all the books across the network.

APPROACH

1. There is one **indexing server** which stores the peer id attached with each registered peer.
2. **Peer**
    a. **Register** and listen to clients who want to download pdf - Peer waits for requests from other peers and sends the requested file when receiving a request.
    b. **Search** for a filename and ask to download it - User can request a file name to the indexing server. The indexing server returns a peer list which have the file. Peer can then choose one peer from the list and ask for the file.
3. Peers here act both as server and client.

DESIGN

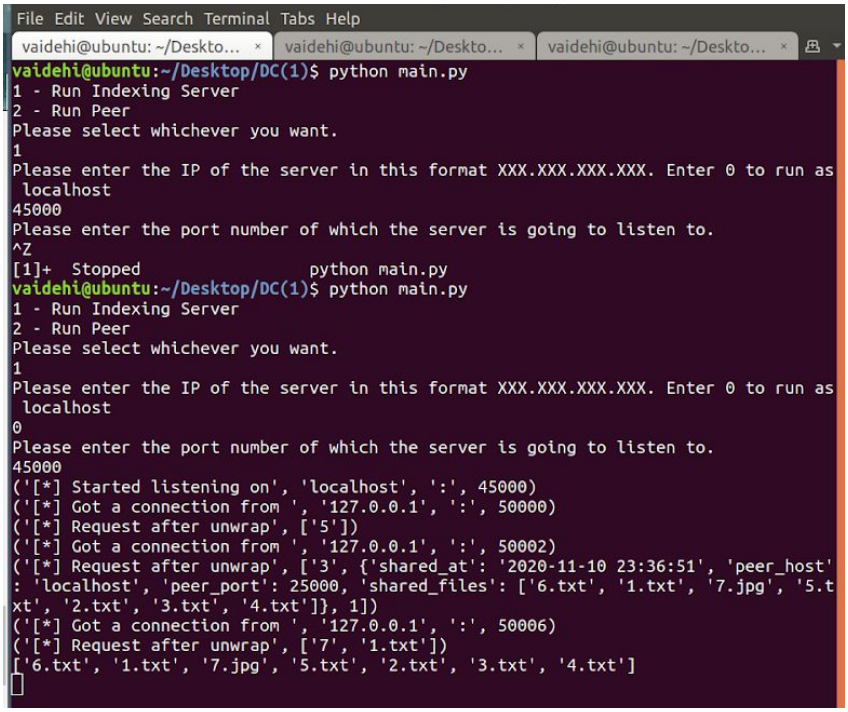


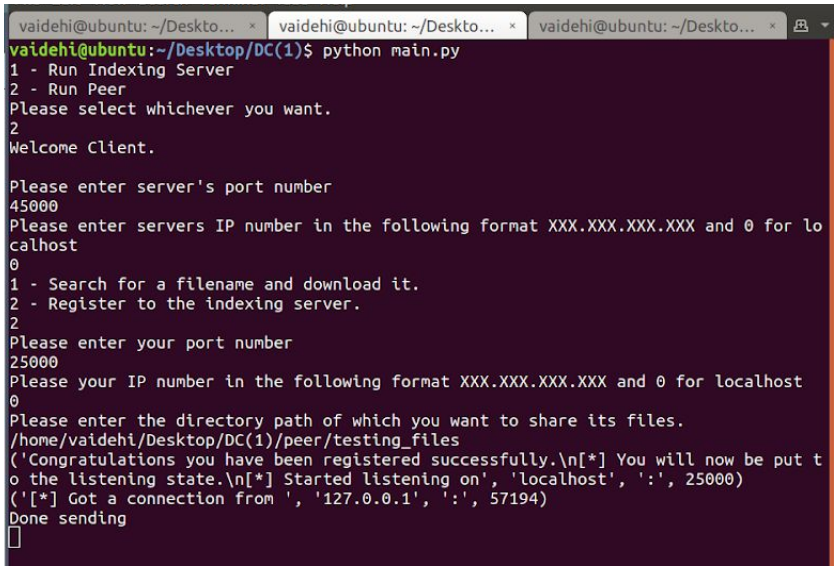
STEPS TO EXECUTE

1. Run main.py :
2. Run the indexing server first. So choose "*1*"
3. Now choose "*0*" to run as *localhost* (You can enter your machine IP), and for the listening port we choose *45000*.
4. Now the server is waiting(listening) for incoming peers' requests.
5. Run *main.py* one more time to run peer. This time choose "*2*"
6. Enter the server's port number which it is currently listening to, which is *45000*(check indexing server configurations) in our case.

7. Choose "*0*" since we are running on a local machine.
8. Choose "*2*" in order to register files to the indexing server(choose "*1*" if you have registered files already).
9. Choose a random port (*25000*) as a peer to listen to for incoming requests to download specific files, and "*0*" for localhost.

OUTPUT





```
vaidehi@ubuntu:~/Desktop/DC(1)$ python main.py
1 - Run Indexing Server
2 - Run Peer
Please select whichever you want.
2
Welcome Client.

Please enter server's port number
45000
Please enter servers IP number in the following format XXX.XXX.XXX.XXX and 0 for lo
calhost
0
1 - Search for a filename and download it.
2 - Register to the indexing server.
1
Please enter filename you want to search for.
1.txt
File 1.txt was found in the following one or more peers. Peer/s details are:

Peer ID: 1

Peer port: 25000

Peer host: localhost

File shared at: 2020-11-10 23:36:51

-----------------------------------
Do you want to download it (Y/N):
y
Successfully got the file
connection closed
vaidehi@ubuntu:~/Desktop/DC(1)$
```

TEST CASE OUTPUT.

```
vaidehi@ubuntu:~/Desktop/DC(1)$ python main.py
1 - Run Indexing Server
2 - Run Peer
Please select whichever you want.
2
Welcome Client.

Please enter server's port number
45000
Please enter servers IP number in the following format XXX.XXX.XXX.XXX and 0 for lo
calhost
0
1 - Search for a filename and download it.
2 - Register to the indexing server.
1
Please enter filename you want to search for.
2.txt
File 2.txt was found in the following one or more peers. Peer/s details are:

Peer ID: 1

Peer port: 25000

Peer host: localhost

File shared at: 2020-11-10 23:36:51

-----------------------------------
Do you want to download it (Y/N):
y
Successfully got the file
connection closed
vaidehi@ubuntu:~/Desktop/DC(1)$
```