Output files and Performance evaluation results:

1. **Deploying at least 3 peers and 1 indexing server in the same virtual machine:**

   - Indexing Server:

```
jorge@jorge-VirtualBox:~/Desktop/Prog 2/Code/server$ python3 server.py 127.0.0.
1 3000
Listening as  127.0.0.1  :  3000

Client ('127.0.0.1', 44926)  connected to server. Total connections:  1
Client ('127.0.0.1', 44928)  connected to server. Total connections:  2
Client ('127.0.0.1', 44930)  connected to server. Total connections:  3
```

   - Peers:

```
jorge@jorge-VirtualBox:~/Desktop/Prog 2/Code/client_1$ python3 client.py 127.0.
0.1 3000 5000 client_files

Connected to INDEXING SERVER:  127.0.0.1  :  3000
Listening as  127.0.0.1  :  5000
```

```
jorge@jorge-VirtualBox:~/Desktop/Prog 2/Code/client_2$ python3 client.py 127.0.
0.1 3000 7000 client_files

Connected to INDEXING SERVER:  127.0.0.1  :  3000
Listening as  127.0.0.1  :  7000
```

```
jorge@jorge-VirtualBox:~/Desktop/Prog 2/Code/client_3$ python3 client.py 127.0.
0.1 3000 9000 client_files

Connected to INDEXING SERVER:  127.0.0.1  :  3000
Listening as  127.0.0.1  :  9000
```

   a) Ensure you can transfer one file properly:

```
jorge@jorge-VirtualBox:~/Desktop/Prog 2/Code/server$ python3 server.py 127.0.0.
1 3000
Listening as  127.0.0.1  :  3000

Client ('127.0.0.1', 45156)  connected to server. Total connections:  1
Client:  ('127.0.0.1', 45156)  registered.
Client ('127.0.0.1', 45158)  connected to server. Total connections:  2
Client:  ('127.0.0.1', 45158)  registered.
Client ('127.0.0.1', 45162)  connected to server. Total connections:  3
Client:  ('127.0.0.1', 45162)  registered.
Client:  ('127.0.0.1', 45158)  added a file.
```

   b) Ensure multiple peer nodes could simultaneously upload and download files:

Log file of the server

```
INFO:root:Listening as 127.0.0.1 : 3000

INFO:root:Client('127.0.0.1', 57914) connected to server.  Total
connections: 1
INFO:root:Client('127.0.0.1', 57916) connected to server.  Total
connections: 2
INFO:root:Client('127.0.0.1', 57918) connected to server.  Total
connections: 3
INFO:root:Client('127.0.0.1', 57920) connected to server.  Total
connections: 4
INFO:root:Client: ('127.0.0.1', 57916) registered.
INFO:root:Client: ('127.0.0.1', 57918) registered.
INFO:root:Client: ('127.0.0.1', 57920) registered.
INFO:root:Client: ('127.0.0.1', 57914) registered.
INFO:root:Client ('127.0.0.1', 57920) requested to download file: test3.txt
INFO:root:Client: ('127.0.0.1', 57920) added a file.
INFO:root:Client ('127.0.0.1', 57918) requested to download file: test1.txt
INFO:root:Client ('127.0.0.1', 57916) requested to download file: test2.txt
INFO:root:Client: ('127.0.0.1', 57916) added a file.
INFO:root:Client ('127.0.0.1', 57914) requested to download file: test1.txt
INFO:root:Client ('127.0.0.1', 57920) requested to download file: test2.txt
INFO:root:Client: ('127.0.0.1', 57920) added a file.
INFO:root:Client: ('127.0.0.1', 57918) added a file.
INFO:root:Client: ('127.0.0.1', 57914) added a file.
```

Log file of one client:

```
INFO:root:
Connected to INDEXING SERVER : 127.0.0.1 : 3000
INFO:root:Listening as 127.0.0.1 : 4456

INFO:root:Registered for Peer-to-Peer downloads
INFO:root:
 Download file: test3.txt from peer node: ["('127.0.0.1', 57916)", '9456']
INFO:root:
Connected to peer : 127.0.0.1 : 9456
INFO:root:
File test3.txt (2099 bytes ) successfully downloaded. Time taken:
0.0006139278411865234
INFO:root:Client('127.0.0.1', 35374) connected.  Total connections: 2
INFO:root:File specs received
INFO:root:
File test1.txt (31985 bytes) sent to client: ('127.0.0.1', 35374). Time
taken: 0.0003631114959716797
INFO:root:Client('127.0.0.1', 35378) connected.  Total connections: 3
INFO:root:File specs received
INFO:root:
File test1.txt (31985 bytes) sent to client: ('127.0.0.1', 35378). Time
taken: 0.0002815723419189453
```

```
INFO:root:
 Download file: test2.txt from peer node: ["('127.0.0.1', 57916)", '9456']
INFO:root:
Connected to peer : 127.0.0.1 : 9456
INFO:root:
File test2.txt (8324 bytes ) successfully downloaded. Time taken:
0.00016880035400390625
INFO:root:
File test1.txt successfully sent to client: ('127.0.0.1', 35374)
INFO:root:
File test1.txt successfully sent to client: ('127.0.0.1', 35378)
```

As it can be seen in the previous figures, multiple nodes can download and send files between them without any problems. However, to implement the bash script to run the evaluation, the command sleep is used to ensure that all the clients have been connected and their servers launched before registering into the Index server and start the downloads.

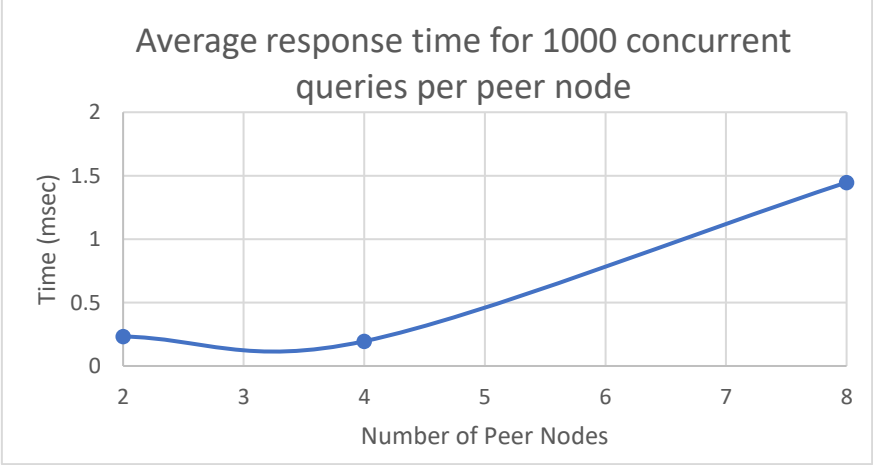
2. **Measuring the average response time when multiple peer nodes are concurrently querying file from the indexing server node.**

To perform this evaluation, some arrangements have been made to the code. All the logs have been disabled except for the one that prints the time it takes to the peer nodes to receive, from the indexing server, the list of nodes that host the file it wants to download. This is needed so ease the averaging of the 1000 requests per node.

Then, a bash script is created to automate the evaluation by creating different screens and run all the clients simultaneously:

q2.sh <num_peers> <num_requests>

Finally, the results that have been obtained are the following:



Is important to note that these results have been obtained while running the server and the peer nodes in a virtual machine with 4 CPUs and 2 GB of RAM. Hence, as it can be seen in the Figure, the average response time with 2 and 4 peer nodes is low and pretty much identical, however, when the number of peers increases up to 8, the average response time of the indexing server also increases tenfold. This happens as there are more simultaneous threads than available physical CPUs and to mimic concurrency, the threads have to be constantly switching with each other, decreasing the performance.

The logs are just the times taken to receive the queries from the 1000 requests from all the clients.

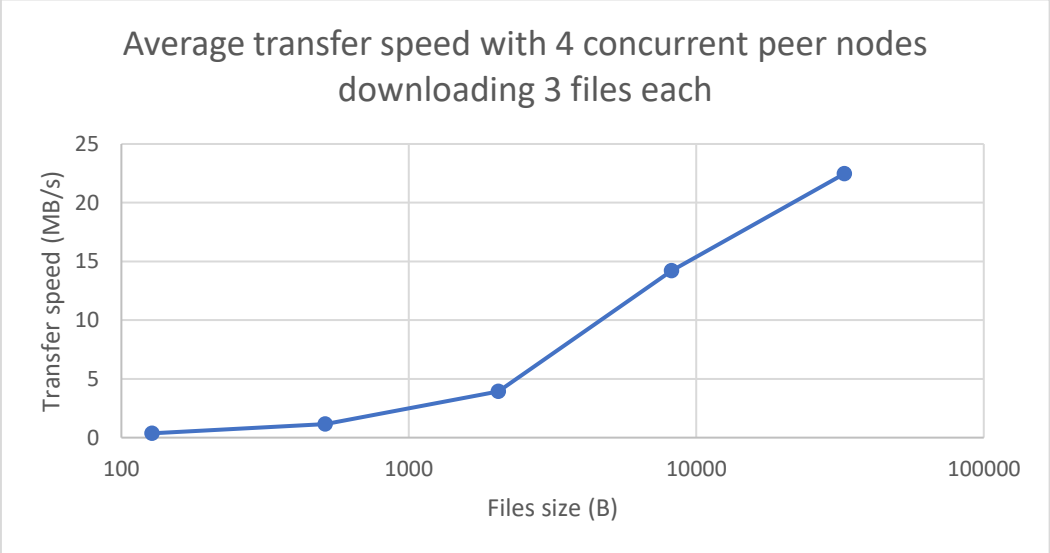
3. **Measuring how the transfer speed changes when varying the download file size.**

For this last evaluation, 5 files have been created with different sizes (128, 512, 2k, 8k, 32k Bytes) and 4 peer nodes concurrently request to download a fixed number of files with the same size.

As with the second evaluation, a bash script is created to automate the evaluation by creating different screens and run all the clients simultaneously:

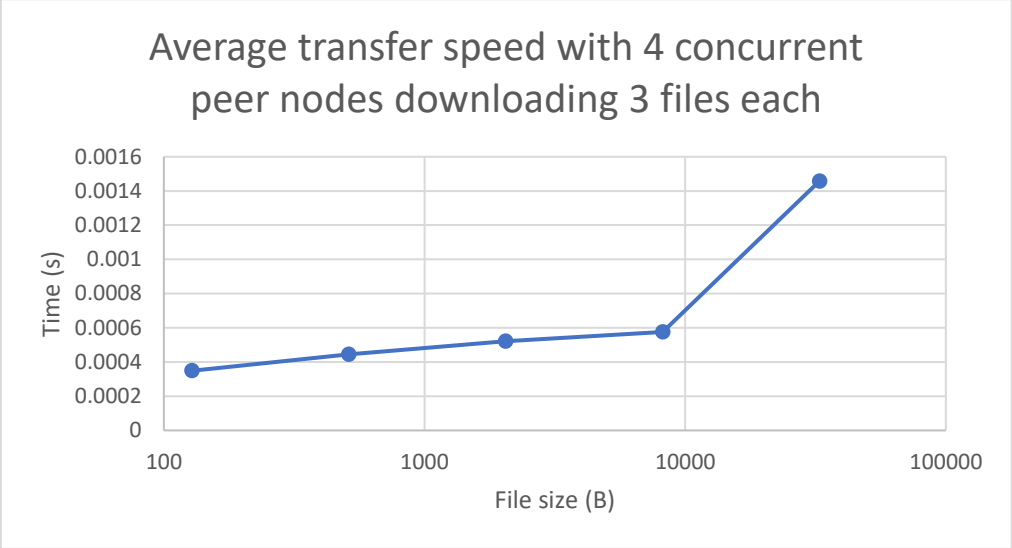
Q3.sh <file_name> <num_times>

Finally, the results that have been obtained for num_times = 3 are the following:



These results indicate that the transfer speed increases with bigger files. Although it may seem weird that the transfer speed increases with the file size, it is just a misconception.

Actually, the time measured in the graph is the one needed to receive all the data in the clients. However, this is not the transfer time but the time it took the client to read the data from the “buffer”.
Therefore, as all the sizes are relatively small, the time taken to read the data is approximately the same for all the files, see next Figure, and, when these times get scaled by their file size, it looks as if the speed in increasing.



Similarly to evaluation 2, the logs are just the times taken to receive the files from all the clients with the following order: time for 128 B, for 512, for 2k, for 8k and for 32k.