To run the code:

**Configuration Files:**

We are maintaining 2 configuration files:

1. Config.txt: In this, I have put all peers id, ip address, port. Registry port and the type of the peer (each entity separated by a ‘-‘) in the following format:
   1. PeerID-IP Address-Registry Port-Type
      1. Example: 1-127.0.01-1099-Seller

Registry by default runs on 1099 port, but sometimes, if incase this port is not free, then we can change it to another port like 2001 etc.

1. Neighbors.txt: In this, I have defined the topology (network) in the following format:
   1. PeerID-Comma Separated Immediate Neighbors
      1. Example: For 3 peers:
         1. 1-2,3
         2. 2-1,3
         3. 3-1,2

To run the code, follow the below mentioned steps: I am explaining the steps with 3 peers with id 1,2,3 (1 acting as seller and 2,3 are acting as buyers), and the registry is assumed to run on port 2001:

The number of peers, N: is equal to the number of entries in the config.txt file.

For Config 1: If all the peers are on present the same machine:

Steps: All the java code is present in the src folder: /lab-1-yadav/src

Corresponding Script is in: script.sh

Run it using ./script.sh after navigating to the src folder in the lab-1-yadav folder

1. Navigate to the src folder in the lab-1-yadav folder
2. Compile all the java files
   1. javac \*.java
3. export \_JAVA\_OPTIONS="-XX:ParallelGCThreads=2" // Limiting GC Threads to 2
4. It’s always better to free the port which you want to use in your code, So, suppose you want all your three peers run on 1901, 1902, 1903 ports respectively
   1. lsof –ti :2001 | xargs kill -9 // Free the port for registry (port : 2001)
   2. lsof –ti :1901 | xargs kill -9 // Free the port of other process as well in the similar manner
5. Start the registry
   1. If you want to start the registry on the default port (i.e., 1099), run,
      1. rmiregistry &
   2. Else, if you want to run your registry on a different port (let’s say, 2001), run
      1. rmiregistry 2001 &
   3. lsof -ti tcp:2001
6. Run all the peers : Main Function: Inputs: PeerID, Port Number on which the Peers runs(make sure that the port number is free), Port Number on which you want registry to run(again make sure that the port number is free), number of hops, type of peer (Seller/Buyer)
   1. java Main 1 1901 2001 4 seller config neighbors -Djava.rmi.server.codebase=file://nfs/elsrv4/users4/grad/ryadav/677/lab-1-yadav/src/ &
   2. java Main 2 1903 2001 4 seller config neighbors -Djava.rmi.server.codebase=file://nfs/elsrv4/users4/grad/ryadav/677/lab-1-yadav/src/ &
7. After all the peers are started, run the controller program which input as the config file.
   1. Java Controller config

Note: Number of entries in the config is equivalent to the number of the peers.

For Config 2: If some/all peers are present on different machines:

Change the config File with the ip address of the other machines, (as described above)

Now follow the following steps:

Corresponding Script is in: script.sh

* Navigate to the src folder in the lab-1-yadav folder on Machine 1
* Run the script ./script\_1.sh on one machine (suppose Machine Number 1)
* Navigate to the src folder in the lab-1-yadav folder on Machine 2
* Run it using ./script\_2.sh on the other machine (suppose Machine Number 2)

1. Navigate to the src folder in the lab-1-yadav folder
2. Compile all the java files
   1. javac \*.java
3. export \_JAVA\_OPTIONS="-XX:ParallelGCThreads=2" // Limiting GC Threads to 2
4. It’s always better to free the port which you want to use in your code, So, suppose you want all your three peers run on 1901, 1902, 1903 ports respectively
   1. lsof –ti :2001 | xargs kill -9 // Free the port for registry (port : 2001)
   2. lsof –ti :1901 | xargs kill -9 // Free the port of other process as well in the similar manner
5. Start the registry
   1. If you want to start the registry on the default port (i.e., 1099), run,
      1. rmiregistry &
   2. Else, if you want to run your registry on a different port (let’s say, 2001), run
      1. rmiregistry 2001 &
   3. lsof -ti tcp:2001
6. Follow the above 5 steps for both the machines (Machine Number 1 and 2)
7. Run all the peers: Main Function: Inputs: PeerID, Port Number on which the Peers runs(make sure that the port number is free), Port Number on which you want registry to run(again make sure that the port number is free), number of hops, type of peer (Seller/Buyer)
   1. Run the peers on Machine 1 (the peers which you want to run on Machine 1 – IP address set to the corresponding peer id in config)
      1. java Main 1 1901 2001 4 seller config neighbors -Djava.rmi.server.codebase=file://nfs/elsrv4/users4/grad/ryadav/677/lab-1-yadav/src/ &
   2. Run the peers on Machine 2 (the peers which you want to run on Machine 2 – IP address set to the corresponding peer id in config)
      1. java Main 2 1903 2001 4 seller config neighbors -Djava.rmi.server.codebase=file://nfs/elsrv4/users4/grad/ryadav/677/lab-1-yadav/src/ &
8. After all the peers are started on both the machines, run the controller program on any machine which input as the config file.
   1. Java Controller config
9. Also, the number of threads can be set to any number in the Peer.java file. Currently set to 2. The following line in Peer.java: Executors.newFixedThreadPool(2); (2: determines the number of threads )

**Output Files:**

The program will generate:

1. log files of all the peers (For instance Peer1.log, Peer2.log , where 1,2 are the peer ids)
2. Response Time of the remote calls (announce, buy, look\_up) for all the Peers.
3. All Peer log Files will be saved in PeerLogs/ Folder

Performance Metrics:

For Performance Metrics, average response time of all the remote calls for all the peers have been calculated and then the corresponding graphs have been plotted. All perfLogs will be saved in PerfMetricLogs/ folder.

Run plotGraphs.py python script for plotting the graphs.

Steps:

1. Navigate to the src folder in the lab-1-yadav folder
2. Run, python plotScripts.py

Test Results:

**TestCase 1:** **2 Peers (Both Running on Local Machine):**

PeerID 1 – Acting as Seller

PeerID 2 – Acting as Buyer

Registry running on 1099(default) Port Number

To Run this testCase will be generated:

1. Go to test/TestCase-1 directory
2. Run ./script.sh
3. Output files:
   1. Peer Logs -> /src/PeerLogs
   2. PerfMetric Logs -> /src/PerfMetricsLogs
   3. Graphs -> Perf-Metrics

**TestCase 2: 6 Peers (All Running on Local Machine)**

PeerID 1 – Acting as Seller

PeerID 2 – Acting as Seller

PeerID 2 – Acting as Seller

PeerID 2 – Acting as Buyer

PeerID 2 – Acting as Buyer

PeerID 2 – Acting as Buyer

Registry running on 1099(default) Port Number

To Run this testCase:

1. Go to test/TestCase-1 directory
2. Run ./script.sh
3. Output files will be generated:
   1. Peer Logs -> /src/PeerLogs
   2. PerfMetric Logs -> /src/PerfMetricsLogs
   3. Graphs -> Perf-Metrics