Deliverables

For Peer to Peer File sharing system, we have integrated 12 java files into a jar file "**GnutellaP2P_win.jar**" (for windows) and "**GnutellaP2P_ios.jar**" (for Mac/Linux machine), external jar named "javax.ws.rs-api-2.0.jar" for implementing MultiValuedHashMap Data structure with 5 config files. We have also created 3 Windows Executable batch files and 3 bash scripts.

1. **SetupConfig.java**

   This java file contains the main method for reading the hardcoded property file and use it initialize super peers, leaf nodes and their topology.

2. **SuperPeer.java**

   This java file contains the main method for setting up the Super Server via property file and registering its service in the rmi naming registry and binds its address.

3. **SuperPeerInterface.java**

   This java file extends Remote class and declare Super Peer's Remote methods (RegistryFiles, Search, Query, queryHit, broadCastSP, broadCastSS, poll, getVersionNum, and notifyPoll)

4. **SuperPeerImpl.java**

   This java file implements the "SuperPeerInterface" interface and define the remote methods. And also called various Leaf node's remove methods for consistency mechanism.

   (RegistryFiles, Search, Query, queryHit, broadCastSP, broadCastSS, poll, getVersionNum, and notifyPoll)

5. **LeafNode.java**

   This java file contains the main method for setting up the leaf node via property file and registering its service in the rmi naming registry and binds its address.

6. **LeafNodeInterface.java**

   This java file extends Remote class and declare Leaf node's Remote method (fileDownload, queryHit, poll, invalidate, outOfDate, getStatus)

7. **LeafNodeImpl.java**

   7.1. This java file implements the "LeafNodeInterface" interface and define the remote method in it (fileDownload, queryHit, poll, invalidate, outOfDate, getStatus)

   7.2. The Leaf node get the service from RMI and obtains target address (Super Peer's address).

   7.3. Call various Leaf node's and super peer's remove methods for consistency mechanism (PUSH, PULL 1, and PULL2).

8. **MultiClient.java**

   This java file contains the main method for setting up Peer and registering each of its service in the rmi naming registry and binds its address.

9. **AvgRespFileSearch.java**

   9.1. This java file implements the "LeafNodeInterface" interface and define the remote method in it (fileDownload, queryHit, poll, invalidate, outOfDate, getStatus) and Call various Leaf node's and super peer's remove methods for consistency mechanism (PUSH, PULL 1, and PULL2).

   9.2. Along with that it also calculate the total invalid percentage for all queryHits it received.

10. **GetPeerDetails.java**

    This java file is used to store all peer details from the config file passed to SetupConfig.java

11. **GetSuperPeerDetails.java**

    This java file is used to store all Super peer details from the config file passed to SetupConfig.java

12. **GetTopologyDetails.java**

    This java file is used to store Topology details of Super Peers from the config file passed to SetupConfig.java

Windows Batch files:

1. **run_peer.bat**
   a. Setting the java path

      set path=%PATH%;C:\Program Files\Java\jdk1.6.0_14\bin

   b. Executing the Peer class in the jar file by passing supporting jars as classpath

   java -cp GnutellaP2P.jar;javax.ws.rs-api-2.0.jar com.gfiletransfer.LeafNode

2. **run_server.bat**
   a. Setting the java path

   set path=%PATH%;C:\Program Files\Java\jdk1.6.0_14\bin

   b. Executing the Peer class in the jar file by passing supporting jars as classpath

   java -cp GnutellaP2P.jar;javax.ws.rs-api-2.0.jar com.gfiletransfer.SuperPeer

3. **run_test.bat**
   a. Setting the java path

      set path=%PATH%;C:\Program Files\Java\jdk1.6.0_14\bin

   b. Executing the Peer class in the jar file by passing supporting jars as classpath

      java -cp GnutellaP2P.jar;javax.ws.rs-api-2.0.jar com.gfiletransfer.MultiClient

Linux bash files:

4. **peer_ script.sh**
   a. Setting the java path and classpath

JAVA_HOME=/usr/bin/java/

CLASSPATH=/Users/vjstark/Documents/Consistency/GnutellaP2P_ios.jar:/Users/vjstark/Documents/Consistency/javax.ws.rs-api-2.0.jar:.

b. Executing the Peer class in the jar file by passing supporting jars as classpath

$JAVA_HOME -cp $CLASSPATH com.gfiletransfer.LeafNode

5. **script.sh**

a. Setting the java path and classpath

JAVA_HOME=/usr/bin/java/

CLASSPATH=/Users/vjstark/Documents/Consistency/GnutellaP2P_ios.jar:/Users/vjstark/Documents/Consistency/javax.ws.rs-api-2.0.jar:.

b. Executing the Peer class in the jar file by passing supporting jars as classpath

$JAVA_HOME -cp $CLASSPATH com.gfiletransfer.SuperPeer

6. **test_ script.sh**

a. Setting the java path and classpath

JAVA_HOME=/usr/bin/java/

CLASSPATH=/Users/vjstark/Documents/Consistency/GnutellaP2P_ios.jar:/Users/vjstark/Documents/Consistency/javax.ws.rs-api-2.0.jar:.

b. Executing the Peer class in the jar file by passing supporting jars as classpath

$JAVA_HOME -cp $CLASSPATH com.gfiletransfer.MultiClient

Config files:

TopologyDetails_config.txt, Superpeer_config.txt, peer_config.txt, Consistency_config.txt and Topology.txt