**Manual**

This document contains the procedure to run the decentralized file sharing system.Run the server first and then the client.

**Server side**

The following are the files required to run the program on Server Side

DistributedServer.java (to be run)

DistributedServerOperate.java

ServiceClient.java

Utility

Server1.properties

makefile(optional)

First execute Server program, find the steps to execute below

1. Execute the ‘make’ command in the source code folder

$ make

1. Step1 generates the .class files in case the compilation of source code is successful
2. Make the necessary changes to the property file such as IP, port,secondary server IP
3. Execute the following command to begin the Server program

$ java DistributedServer

1. On execution of step ‘4’ there is a pop up window to choose the property file for Server program from your system location.(Server1.properties)
2. Bring up all 8 servers using step 1 to 4, enter the number 100 in each of the server window at the prompt, this connects each of the server to all the other servers and the server is ready to accept requests from client.

**Client Side**

The following are the files required for running the client side program

PeerClient.java (to be run)

PeerServer.java

PeerClientPerfTest.java (to br run for performance evaluation)

makefile(optional)

createFile.sh (please change permissions of this file using the command:

chmod 777 createFile.sh )

First execute Peer program, find the steps to execute below

1. Execute the ‘make’ command in the source code folder

$ make

1. Step1 generates the .class files in case the compilation of source code is successful
2. Make the necessary changes to the property file such as IP, port,secondary server IP
3. Execute the following command to begin the Server program

$ java PeerClient

1. On execution of step ‘4’ there is a pop up window to choose the property file for Server program from your system location.(Peer1.properties)

**NOTE :** Each peer/server has its own config file

**Property files**

Server1.properties

Changes required for the properties file

IP , port number ,localhost, selfID, ServerPortNumber, NumberOfSystem

Update all the IP’s and port numbers of all the other servers as well

ServicePortNumber (port number where peers send requests)

Peer1.properties

Changes required in file

serverIP, serverPort,sharedFileLocation,

Note : in case the number of peers in the systems has to be reduced

NumberOfSystem value to be changed in property file and

number

fServer value in hashCode method in ServiceClient.java

Make file commands

Command to compile

Command : make

Output: .class files are generated

To remove the old .class file (in case the java files have to be recompiled)

Command: make clean

To recompile only a specific java file

Command make < filename.class>

Note :

If your running the server or clients on vm make sure the IP is not using NAT

Key points to be noted when running on amazon cloud

While creating the instance make the security setting to allow data over different protocols like icmp(ping), tcp(used for socket connect).

If java not installed on the instance

Using the following command to install it

yum install java-devel