Peer2Peer Replica Management System

Through JAVA RMI

Manohar Kotapati

CS Graduate Student

TEXAS TECH UNIVERSITY

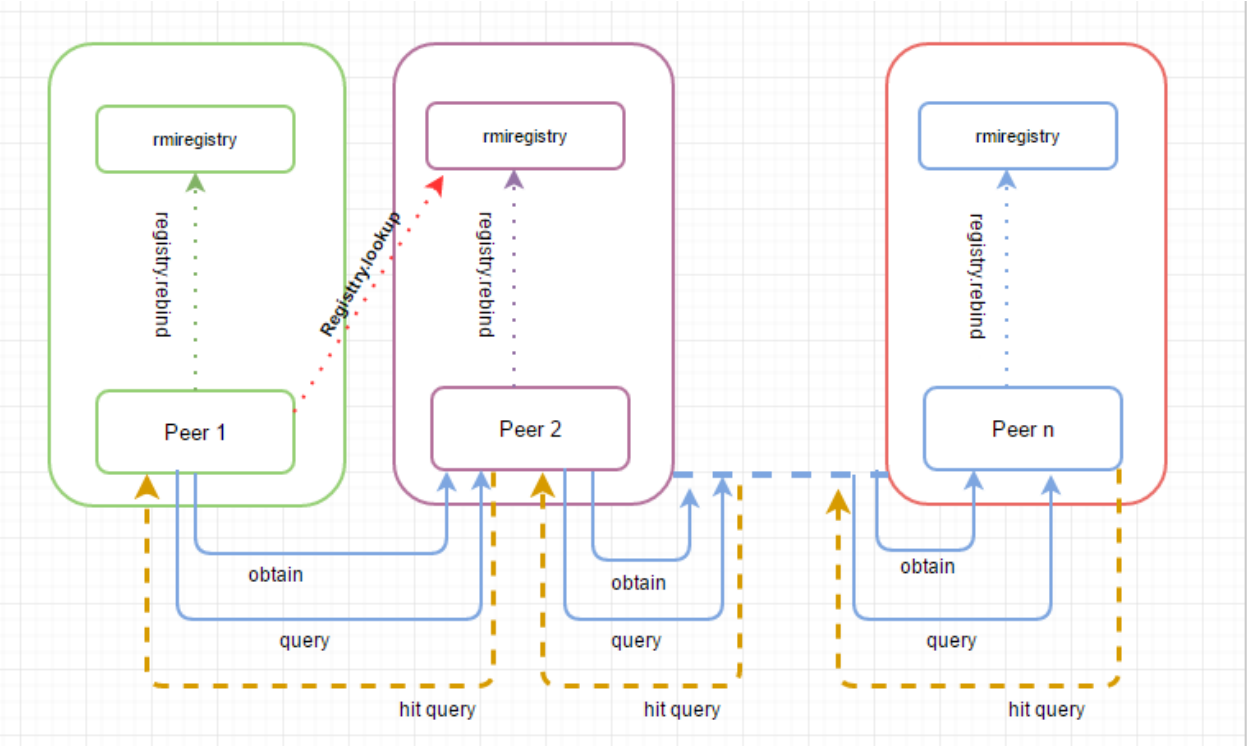
Submitted on 04h May, 2016.

Table of Contents

1. Project Statement

This project builds on two previous projects. The goal of this project is to add consistency mechanisms to a Gnutella-style P2P file sharing system for maintaining file consistency. While some files that are typically shared using today's P2P systems rarely change after creation, in the future, we can expect to use P2P systems to share other types of files that do change after creation. The objective of the project is to ensure that all copies of a file in the P2P system consistent with one another.

2. Design

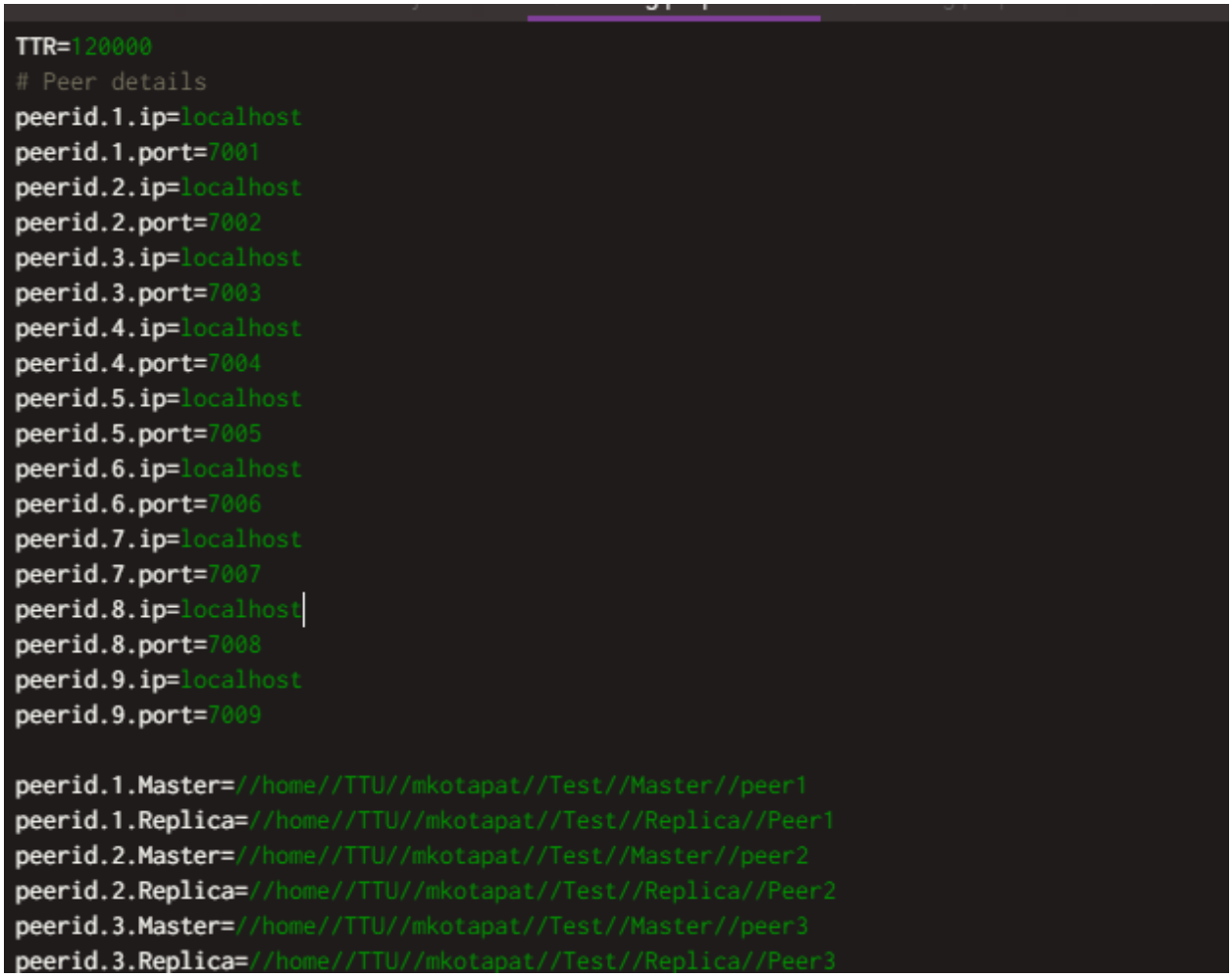


Gnutella style Peer to Peer file management design with RMI

3. Implementation

3.1 Config

As we are not implementing the dynamic initialization of the network as in Gnutella. To keep things simple, assume that the structure of the P2P network is static. Declare the neighbor peer details in the config.properties file. Maintain config file for topology, master , replica directories and TTR



```
peerid.4.Master=//home//TTU//mkotapat//Test//Master//peer4
peerid.4.Replica=//home//TTU//mkotapat//Test//Replica//Peer4
peerid.5.Master=//home//TTU//mkotapat//Test//Master//peer5
peerid.5.Replica=//home//TTU//mkotapat//Test//Replica//Peer5
peerid.6.Master=//home//TTU//mkotapat//Test//Master//peer6
peerid.6.Replica=//home//TTU//mkotapat//Test//Replica//Peer6
peerid.7.Master=//home//TTU//mkotapat//Test//Master//peer7
peerid.7.Replica=//home//TTU//mkotapat//Test//Replica//Peer7
peerid.8.Master=//home//TTU//mkotapat//Test//Master//peer8
peerid.8.Replica=//home//TTU//mkotapat//Test//Replica//Peer8
peerid.9.Master=//home//TTU//mkotapat//Test//Master//peer9
peerid.9.Replica=//home//TTU//mkotapat//Test//Replica//Peer9
# Neighbor details
#3X3 Mesh
peerid.1.neighbors=peerid.2,peerid.4
peerid.2.neighbors=peerid.1,peerid.3,peerid.5
peerid.3.neighbors=peerid.2,peerid.6
peerid.4.neighbors=peerid.1,peerid.5,peerid.7
peerid.5.neighbors=peerid.2,peerid.4,peerid.6,peerid.8
peerid.6.neighbors=peerid.3,peerid.5,peerid.9
peerid.7.neighbors=peerid.4,peerid.8
peerid.8.neighbors=peerid.5,peerid.7,peerid.9
peerid.9.neighbors=peerid.6,peerid.8
```

3.2 p2pReplica Package

This package consists of the necessary classes for the peer to participate in the network.

PeerInterface.java – Interface with three remote methods

- `updateVersion(String fileName,int version,long modifiedTime)throws RemoteException;`
- `checkFileStatus(String fileName,int version,long lastModifiedTime)throws RemoteException;`
- `invalidation(int fromPeerId,String msgId,String fileName,int version, int originPeerId,long modifiedTime)throws RemoteException;`

Peer.java

- It creates the RMI registry for PeerInterfaceRemote on a specific port that provided by the user. So that peer can act as a server on rmi://<ip>:<port>/peerServer
- It reads the Master and Replica directories for the specified peer id
- Gives the menu with the options of PUSH, PULL, Modify and PUSH, Just Modify
- On Push it will broadcast the Master copy to all the peers
- On Pull it check the local replica folder and if TTR expires it requests the New TTR or file download if version changed
- Modify and PUSH modifies the specific file in Master folder and broadcast the INVALIDATION.
- Just Modify option is to modify the version of the file so that it will be useful for testing the PULL

HitQuery.java

This class is used to handle the details of the result of every query. It contains properties of the peer details of the broadcasting

FileDetails.java

This class is to handle the file details like origin peer id, last modified, filename, version

NeighborConnectionThread.java

This class handles the thread to broadcast the messages to the neighbor peers

3.2 Performance Evaluation.

Assume mesh network of 3X3 peers. Find the response time of Peer1 for broadcasting PUSH requests in following scenarios

- When it is the only peer performing the PUSH in the network.
- When there is one more peer performing the PUSH in the network
- When there are two more peers performing the PUSH in the network

Modify the class ***PerformanceLauncher.java*** file as per each scenario given above.

PerformanceEvalution.java file handles PUSH option and find the average response time.

4. Instructions to run the software:

Please follow the below instructions to run the software.

1. **Config:** Modify the config.properties file as per the required network topology
2. **Run the peer**: Run the class Peer
   E.g. Java p2pReplica/Peer
   We can run more than one peers with different ports in the same machine.

Note: If you are testing the whole software in one machine, then maintain separate command interfaces for server and client.

5. Improvements

Currently peers are configured in config file. It can be enhanced by configuring the peers in the network dynamically.
Algorithm used to search files in the arrays for version comparisons can also be improved.