Programming Assignment 1

NAPSTER PEER TO PEER SHARING SYSTEM

CS550 – Advanced Operating System

February 22, 2018

NIKITA V. JADHAV

A20401223

Introduction

- Peer-To-Peer (P2P) Technologies have been widely used
  - Content Sharing.
- Existing applications of P2P File sharing applications
  - Napster, Gnutella, Free net etc.
- The Design of these systems is the concept of files distributed throughout Nodes.
- The P2P system is different from the older Client/Server Models where the files would reside on one Central Server and all the transfers would happen only between the Central Server and the Clients.
- In P2P File Sharing Application, the File transfer can occur between the individual Nodes/Peers.

Design Overview

In this project there are two main entities. First is Indexing server and other is Peer. Peer can take both roles of server and client. Peer client registers its files by doing register request to the indexing server. Indexing server saves this data in a non-blocking thread safe List data structure. Peer client can make Lookup file requests to indexing server to download the files. To download the file Peer Client connects to Peer Server and makes the download request. Peer Server receives the download request from Client Peer and sends the file to that Client Peer.

Design

- Implemented using Java, Sockets and Threads.
- Built on: Linux Operating system
- The P2P file sharing system has two components:
  - **Central Index Server:**

    This server indexes the contents of all the peers that register with it. It also provides a search facility to the peers.
    The Central Index Server provides the following Interfaces to the Peers:
    a) **Registry (peer id, filename)** – invoked by a peer to register its files with the Centralized Index server. The CIS then builds an Index for the peers.
    b) Search (filename) – this procedure searches the index and then returns all the matching peers to the requestor.

  - **Peer:**

The peer acts as both a client and a server. As a client the user specifies the filename with the indexing server using “lookup”. The Indexing server then returns a list of all other peers that hold the file. The user can then pick one such peer and the client then connects to this peer and downloads the file. As a Server, the peer waits for requests from other peers and sends the requested file when receiving a request.
The Peer Server provides the following interface to the Peer client:

a) **Obtain/Download (Peer ID, filename)-** invoked by a peer to download file from another peer.

Implementation

The Implementation is done using Socket Libraries for communication between the Central Index Server and the Peers as well as between the Peer Servers. The CIS and the Peer-Servers are Multi-Threaded.
Following are the main operations:
a) Register
b) Search
c) Download

The Centralized Index Server [CIS]:

When the CIS is run, Initially the Index Server will be in the Idle State.The CIS Socket then keeps listening on a Port for incoming connection requests. Whenever a Peer makes a request on the CIS port, the CIS serves the request for the connected peer. The Registry and Search method is implemented on the Central Index Server.

a) Register Method:
RegisterRequestThread()-running and listening on a Port to serve the Register request.
RegisterWithIServer()-Socket connection is automatically established with the Central Index Server with the CIS_ip and the port number.
ArrayList – Maintained by the Index Server to hold all the values of the Peer ID’s and the Filenames with the corresponding Peers.

Registration Function:

- Index Server – Runs a RegisterRequestThread() and listens for active registration requests.
- Request for Register from the Peer
- Peer enters the Peer ID and the Filename it wants to register
- RegisterWithIServer()

- Request for Socket with (CIS_ip, Port No)
- Peer sends Peer ID and filename over the socket through the Output stream.
- Server stores the Peer ID and the Filename received over the Input Stream
- Index Server serves Register request for the Peer.

b) Search Method:

SearchRequestThread()-running and listening on a Port to serve the Search request.
SearchWithIServer()-Socket connection is automatically established with the Central Index Server with the CIS_ip and the port number.
ArrayList- to hold all the values of the Peer ID's and the Filenames with the corresponding Peers.

Search Function:

- Index Server – Runs a SearchRequestThread() and listens for active search requests.
- Request for search from the Peer
- Peer enters the Filename it wants to search
- Request for Socket with (CIS_ip, Port No)
- Peer sends the filename over the socket through the Output stream.
- Server receives the Filename over the Input Stream and then traverses the Index ArrayList.
- If a Match is found, it returns the Peer ID and the IP Address of the Peer which contains the searched File.
- If no match is found, it returns "File Not Found"
- Index Server serves Search request for the Peer.

c) The **Download Method**

AttendFileDownloadRequest ()- thread running and listening on a Port to serve the Download request.
DownloadFromPeerServer ()- a Socket connection is established with the Peer-Server which now acts as the Server to serve the Download request.
Filereader object- check for the file and read the contents
WriteObject- to store the contents from the output stream

Download Function:

- Peer-Server – Runs a AttendFileDownloadRequest () and listens for active download requests.
- Request for download from the Peer

- Peer enters the Peer ID, IP Address of the Peer holding the file and the Filename it wants to download.
- DownloadFromPeerServer ()
- Request for Socket with (Peer_Ip_Addr, Peer ID1)
- Peer sends the filename over the socket through the Output stream.
- Peer-Server serves Download request for the Peer.

3. Future Improvements

• Current system does not support file replication. If this project implements Replication, File Availability can be easily increased. This can be done by asking the Client Peer to act as a replication node or not. If Client Peer says 'Yes', files from other Peers can be stored on this node for the replication purpose.

• Current system does not encrypt the communication which happens between Client and Server. To provide security to the system encryption is must. We can use algorithmssuch as RSA to encrypt and decrypt the data by public and private keys. Implementation of public key cryptography will allow the confidentiality as well as authenticity in the system.

• For file integrity purposes, project can calculate the hash code for every file. Hash codes can be tested at client side to check the integrity of the file. Current IndexedFile class can be improved by adding FileHash field which will hold the information about the file hash.

• Project code can be improved for better efficiency in overall project. For eg. Automatic file indexing can be added and file lookup algorithm can be improved.