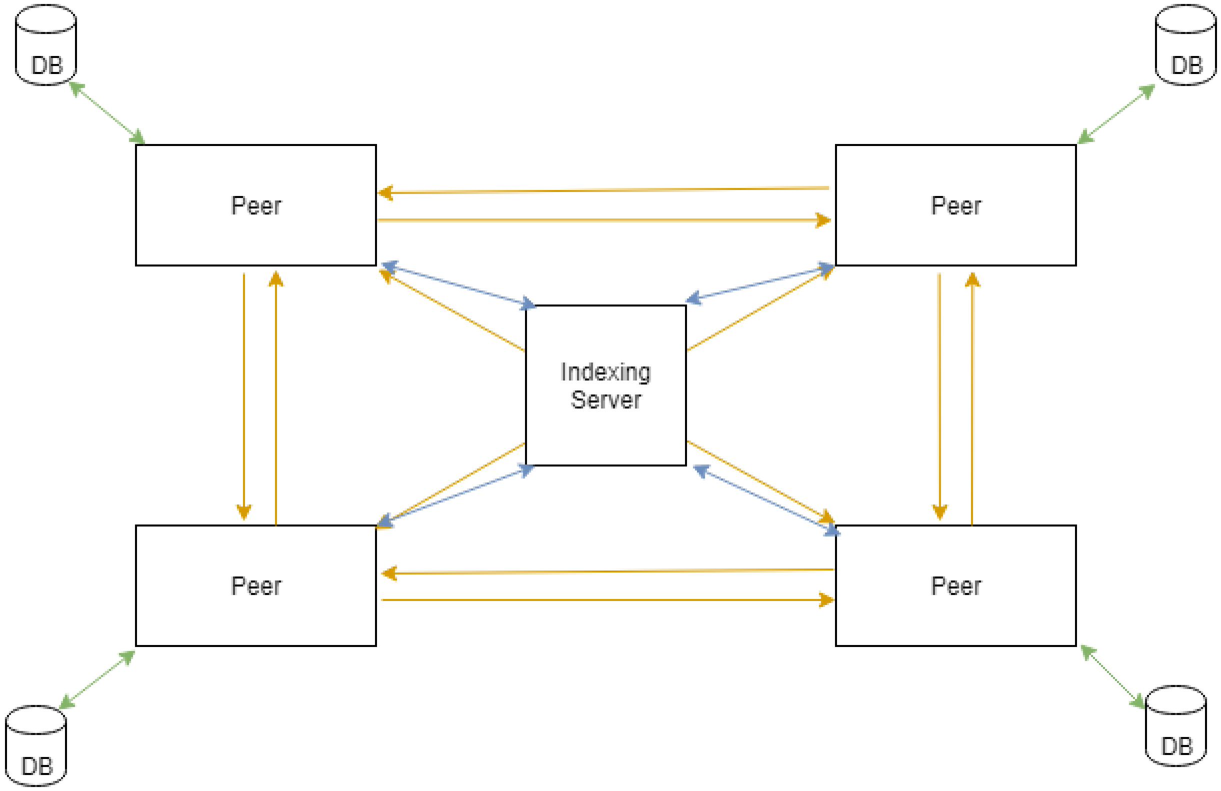
CS 550 Programming Assignment 1 (PA#1)

A Simple Peer to Peer File Sharing System

Abstract:

Peer-to-peer file sharing is the distribution and sharing of digital media using peer-to-peer (P2P) networking technology. P2P file sharing allows users to access media files such as books, music, movies, and games using a P2P software program that searches for other connected computers on the network to locate the desired content. Basically, the load caused of sharing data to every client is split among all the clients, thereby reducing network congestion. [1] Here, a basic P2P file sharing system using Java RMI (Remote Method Invocation) is presented. Java RMI allows a system running on java Virtual Machine to invoke objects running on another java virtual machine. It facilitates remote communication between two java programs. In our approach towards P2P, we have two main components (i) indexing server & (ii) multiple peers. In the remaining section, we describe how these two main components communicate with each other and evaluate performance measures such as average response time for a client making multiple sequential requests and concurrent file search by multiple clients.

System Design:



The above diagram illustrates the system design for a P2P file sharing protocol. It contains two main components:

i. Indexing Server: Indexing server maintains the lists of the file, whose clients are registered with it. It facilitates automatic update mechanism, when a file is added or deleted from a client directory, the same is updated on the server list.

ii. Client/Peer: Each client/peer has a shared folder. On bootup, client registers the files in the folder with the server. When a client wants a file, it issues a search to the server and gets back the lists of peers having that file. Client can then download the file from the list of clients informed by the indexing server.

Implementation:

For a P2P application we write two programs, a server program and a client program.

Important components of a P2P system are explained further:

1. Interface

We define two interfaces one is implemented by the server and the other is implemented by the client.

i. P2PInterface: This is implemented by the indexing server, it contains two functions.
Registry: This method registers all clients and their files.
Search: When clients want to download file from each other, they use search function from the indexing server to locate which client/peer has the file. Function returns client name, port number and directory name.

ii. ClientInterface: This is implemented by all the clients which participate in P2P file sharing system. This interface defines method:
Retrieve: This function returns the file requested by another client in byte format.

2. Server

Main function of a server is to provide service to the client. A server creates an object which implements the P2PInterface and export it. Java RMI registry is a namespace which has all the objects registered by a unique name called bind name. Java RMI registry has details about exported objects.

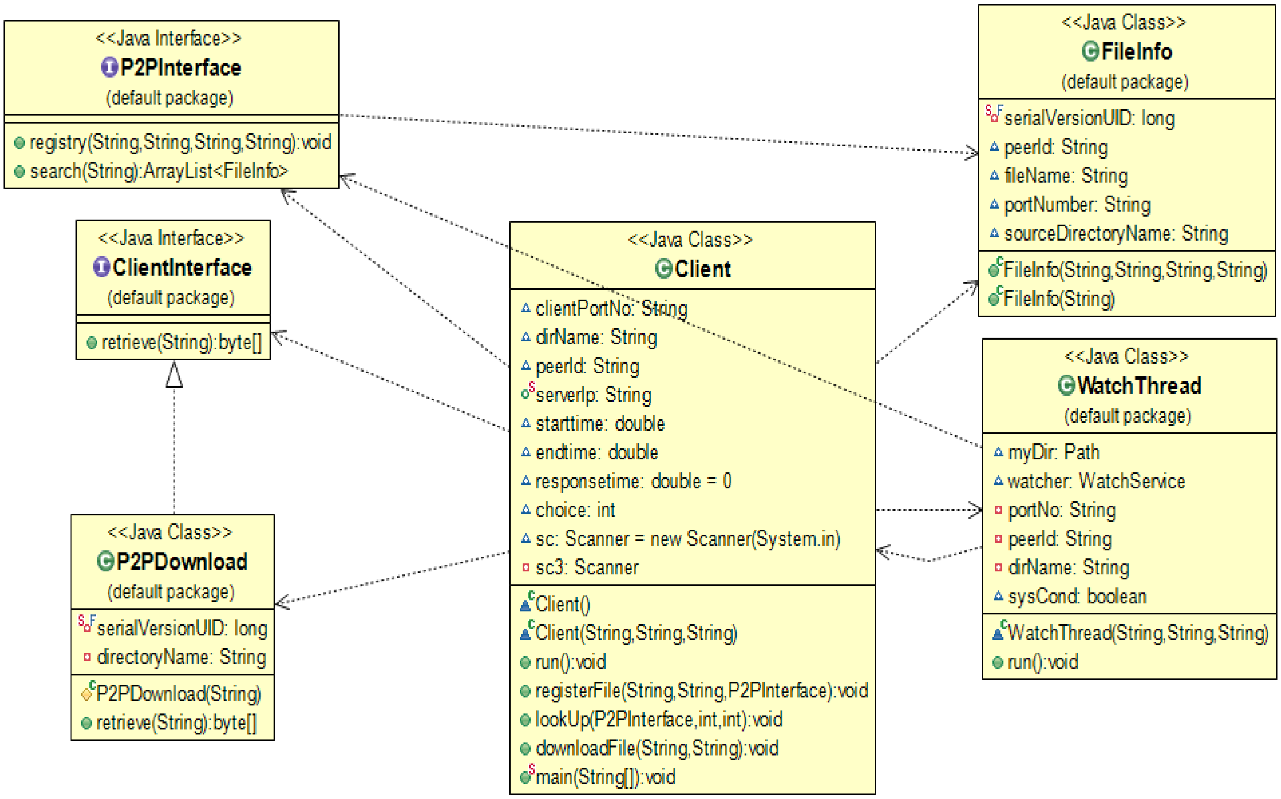
When client needs to access methods implemented by the server, it runs a lookup with RMI registry, which returns a reference to the object. Client uses this reference to access methods implemented by the server.

3. Client

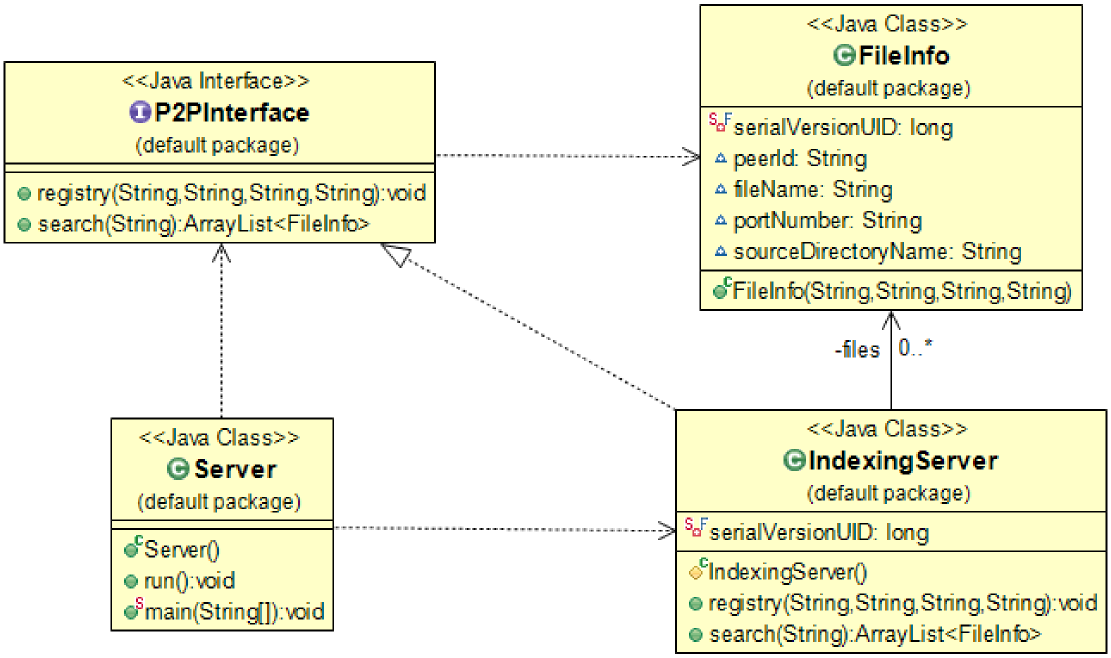
Each client must register itself with an indexing server and bind its name to the RMI registry. On file request by the client, the server returns the list of all the peers having the requested file. The Client can use its preferred choice of peer from the returned list to download the requested file.

UML Class Diagram

Client:



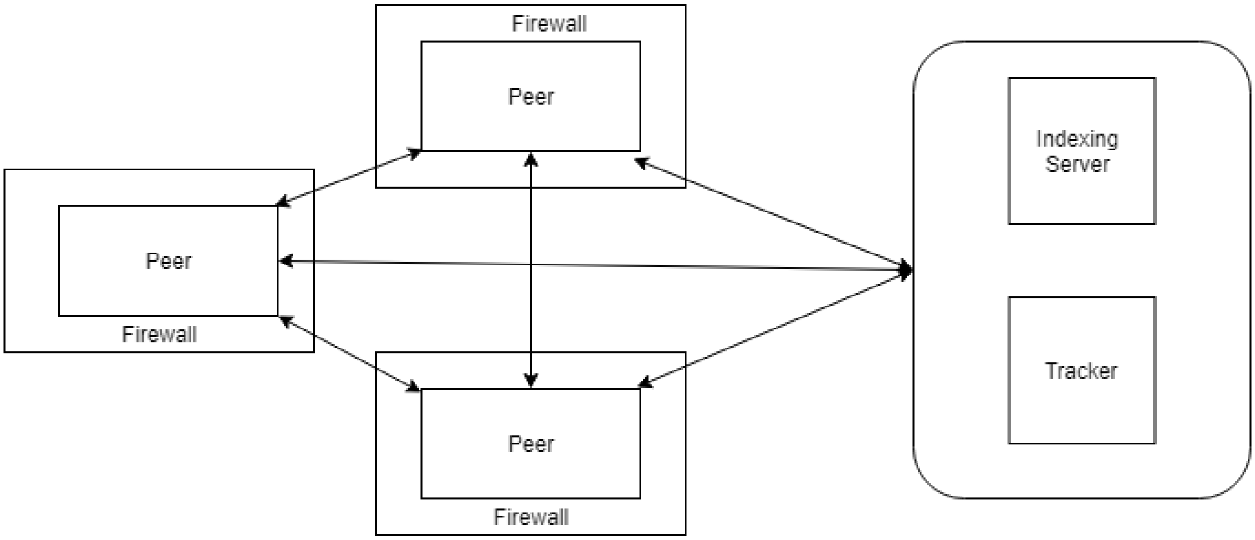
Server:



Design Trade Offs: The following design tradeoffs are made acknowledging the scale of the system.

1. For this implementation of P2P we have not used sophisticated searching algorithm to search a file but rather a exact string match is used.
2. In our implementation, even if there are multiple clients with the same file, a client downloads the file only from a single client.
3. We have used only a single indexing server, we can replicate more if there are more number of clients.
4. There are no security measures taken in our application.
5. There is no tracking whether a client is active or not.

Further Proposed Improvements to P2P:



1. File download methods can be improved, where a client can download parts of a file from multiple clients. This improves availability of the file (in case one or more clients go offline).
2. Sophisticated search algorithm can be implemented for efficient search.
3. Multiple indexing server can be deployed to facilitate area-based query processing.
4. Tracker can be implemented which keeps track of active clients.

References:

1. https://www.tutorialspoint.com/java_rmi/java_rmi_introduction.htm
2. https://en.wikipedia.org/wiki/Peer-to-peer_file_sharing