

# Assignment 2: Chord File Distribution

Course ISE 431: Distributed Information Systems

Due December 19, 2009

In this assignment you will extend the web server program that you wrote for assignment 1 with support for peer-to-peer file storage. The substrate that you will use to support the peer-to-peer storage is Chord. We have discussed the mechanics of Chord in class already.

The overall behavior of the system is as follows:

- The network has a few file/web servers which may answer queries via HTTP
- When a client sends a URL request to one server, it must determine whether the file corresponding to the URL is stored locally or remotely.
- If the file is stored locally, the server responds with a normal response and sends back the file requested.
- If the file is stored on a remote machine, the server must first contact the remote server and retrieve a copy of the requested file. The server can then send it back to the requestor.
- If the file is not found locally or anywhere in the Chord system, an error message should be sent back to the requestor (404).

## 1 What to do

You may use any Chord library set which you find on line to keep track of the ids, finger tables, and lookup for your peer-to-peer network. Your job is to use the library as a platform for implementing your distributed file storage application.

Design and develop a Java application which performs the following jobs:

1. It listens on a given port (say 5000) for HTTP requests. The server should be multithreaded so that it can respond to multiple HTTP requests at a time.
2. It listens for internal file transfer requests (from other servers in the Chord network) on a different port (say 6000). The server should be multithreaded as well so that it can respond to several file transfer requests at a time.

The two jobs should be carried out by a multithreaded server application which listens on both of the ports listed and launches threads to handle requests as needed. The server application should store all of its local files in a single directory (it may be the run directory for the Java code).

When a server receives a file from another computer to send back, it should keep a copy of the file long enough to return it to the requestor, not permanently store it on the server.

### 1.1 Addition and Exit of Nodes

Your application must take into consideration what to do when nodes join and leave the Chord network. You may assume that nodes won't exit unexpectedly (crash), however, you must include logic to handle the transfer of files to a new node and the from a node which wishes to exit. The algorithms for how to handle node entries and exits are shown in the Chord paper and may be read online.

## 1.2 A note on pre-existing code

The main part of this exercise is the use of Chord as a substrate for a peer-to-peer networks, handling storage, search, and retrieval as well as the addition and exit of nodes. You may use a preexisting Chord library to help you (you don't need to implement it from scratch). Similarly you may use existing libraries or code for your application so long as they do not directly relate to the main part of the exercise (ex. You may use GUI libraries, code for networked file transfer, socket communication code, HTTP request processing code).

## 1.3 User interface

The client interface for the application can just be an internet browser (Firefox, Internet Explorer, etc) which can be pointed to different web servers and port numbers.

The server application (the application which you must write) must have a user interface which enables the following actions:

- Enter an existing Chord network by providing it with the IP address of an existing node in the Chord network.
- Start a new Chord network with it as the only node.
- Show a list of the files it currently is holding and their Chord IDs
- Show the server's Chord ID
- Produce a trace of what the server did for each request that it receives (local checks, Chord lookups, file transfers) and each file that it sends to another Chord peer. The trace should be stored in a log file.
- Exit the Chord network that the server is currently participating in.
- Add a new file to the Chord network that the server is currently participating in.

## 2 Scenarios and Testing

Test your code in the following scenarios:

1. A Chord network with 5 peers storing 100 randomly generated HTML files. You may find random file generators easily online (ex. <http://tinyurl.com/yh9zhhd>). Send at least 10 requests to each node in the network.
2. A Chord network with 4 peers storing 100 randomly generated HTML files. After a 10 requests have been sent to each node in the network, one new Chord node is added. All of the nodes are sent 10 new requests each. Then, one of the original four peers (not the one which newly joined) leaves the Chord network. All of the nodes are sent 10 new requests each.
3. A Chord network with 5 peers storing 50 randomly generated HTML files. One node produces 50 new randomly generated HTML files and distributes them appropriately around the network.

## 3 What to turn in by December 19

You may work in groups of two students for this project. I will allow one group of three students since there are an odd number of students in the class.

For this project you must turn in the following items:

1. A listing of all members of the group, the tasks they performed for the project, and the approximate number of hours that each spent on project.
2. A compressed file with the full source code for the entire program.
3. An instruction document indicating how to configure and use the application. The instructions should be sufficiently detailed for me to reproduce the above scenarios.
4. A listing of all external code sources used and their locations (in books or online).

You may turn in all of the above by sending it as a single compressed file to `ise431@gmail`. You may alternatively turn it in to me in person.

## 4 Questions and Clarifications

I have detailed some of the internal workings of the server applications, but there are some details still left unspecified. They are for you to decide. In case of questions, please send me email to `mjmaj@kinneret` and I will attempt to respond promptly. Questions which are of importance to all students will be answered on the Telem course announcements page and/or on the course web page at `www2.kinneret.ac.il/mjmaj/ise431-5770.html`