M-Koda Peer to Peer System

Arun Natarajan Manikandan Sivanesan Kousik Krishnakumar Dinesh Radhakrishnan
anatara@ncsu.edu msivane@ncsu.edu kkrishn2@ncsu.edu dradhak@ncsu.edu

---

MySql Setup:

This provides the steps to install and configure MySQL database at the server side. Please note that these are the instructions to be followed for Ubuntu.

Steps to be followed:

1. Open the terminal in Ubuntu using Applications → Accessories → Terminal
2. MySql can be installed by entering the following command

   $ sudo apt-get install mysql-server mysql-client

3. Once mysql is installed root password can be set in the installation process itself or it can be set using the below command for the first time.

   $ mysqladmin -u root password mysqlpwd

4. Login into mysql using the above set password

   $ mysql -u root -p

   This command asks for the password and you can login into mysql by entering the above set root password.

5. Once logged in you can create databases for the centralized bootstrap server (bootstrap) and distributed server (bootdist) ans start using the databases by following the commands shown below.

   mysql > create database bootstrap;

   mysql > create database bootdist;

   mysql > use bootstrap;

   mysql > use bootdist;

6. Quit the database using the command

   mysql > quit

7. Tables(IPTABLE and FILETABLE) used in centralized bootstrap server can be imported from the file bootstrap.sql using the given below command in the terminal. Password should be set as **mysqlpwd.**

   $ mysql -u root -pmysqlpwd bootstrap < bootstrap.sql

   Note: No space in between -p and PASSWORD

8. In the same way IPTABLE used in distributed server can be imported from the file bootdist.sql using the given below command in the terminal.

   $ mysql -u root -pmysqlpwd bootdist < bootdist.sql

Centralized System :

Mkoda project contains the Centralized peer to peer system. There are 5 packages in this project

1. Client Package
- client_gui.java
  Client application is started by running this program.
- LogMessages.java

2. general Package contains the general functions used in this project.
- Genfunc.java contains all the conversion methods used.
- Myinet4addr.java retrieves the IP address used by the system other than the loopback .

3. packetformat Package
- pack.java defines format of the packet used. It defines header format and the fields used in the header.

4. server package
   server.java contains the server code and the server can be started by running this program.

5. trigest package
   trigest.java contains the code to get the trigest for a file or for a string.

6. lib folder contains all the library functions required for this project.
7. Resolver.Properties file is required in order to configure the server IP address and server port
8. triplet_frequency_2 contains all the trigrams in english alphabets and the number of occurrences. This file is required in order to run the trigest program.
9. Use JRE System Library (Java 6-sun-1.6.0.16)
10. Inorder to track the files published by each client, we create a new file locally when the user try to publish for the first time named SHA_path_[username]. It contains SHA of each file and the absolute path of the file.

Steps to run the Centralized code in Eclipse:

1. Create an empty project in Eclipse
2. Import Mkoda centralized code to this project. Refresh the project.
3. Configure the Build path by adding the external jar's provided in the lib folder.
4. Setup the mysql as mentioned in the mysql readme file.
5. Set the server address and server port number in Resolver.properties file.
6. Server program can by initiated by running server.java program.
7. Client program can be initiated by running the client_gui.java program. We can run multiple clients on the same machine by running multiple client_gui.java programs.

Distributed System:

Mkoda project contains the Distributed peer to peer system. There are 5 packages in this project

1. Client Package

- client_gui.java
  Client application is started by running this program.
- LogMessages.java
- filetable_rowarray.java declares the class for a specific row in the file table
- super_peer.java contains the methods to manage the file and hash space by the super peers.

2. general Package contains the general functions used in this project.

- Genfunc.java contains all the conversion methods used.
- Myinet4addr.java retrieves the IP address used by the system other than the loop back .

3. packetformat Package

- pack.java defines format of the packet used. It defines header format and the fields used in the header.

4. server package
   server.java contains the server code and the server can be initiated by running this program.

5. trigest package
   trigest.java contains the code to get the trigest for a file or for a string.

6. lib folder contains all the library functions required for this project.
7. Resolver.Properties file is required in order to configure the server IP address and server port
8. triplet_frequency_2 contains all the trigrams in english alphabets and the number of occurrences. This file is required in order to run the trigest program.
9. Use JRE System Library (Java 6-sun-1.6.0.16)
10. In order to track the files published by each client, we create a new file locally when the user try to publish for the first time named SHA_path_[username]. It contains SHA of each file and the absolute path of the file.

Steps to run the Distributed code in Eclipse:

1. Create an empty project in Eclipse
2. Import Mkoda distributed code to this project. Refresh the project.
3. Configure the Build path by adding the external jar's provided in the lib folder.
4. Setup the mysql as mentioned in the mysql readme file.
5. Parameters for the distributed system can be set in Resolver.properties file. Set the server address and server port number in Resolver.properties file. Client Port Number can be made set as random when used in single machine for testing purposes. Other parameters such as load factor I.e maximum files managed by peer, minimum search results, search_hash_depth _split(depth of the search) can be configured.
6. Server program can by initiated by running server.java program. Server will be the first super peer in distributed system. There should be at least one super peer in the system and if the server goes down, the system goes down along with it.
7. Client program can be initiated by running the client_gui.java program. We can run multiple clients on the same machine by running multiple client_gui.java programs.