In this document, we talk about how to run our programs.

Socket-Based

To run this project, JDK 1.8 was required. Thus you may need to modify your ".bashrc.custom" file in the server to include the JDK 1.8.

To modify this file, simply login any server, and edit ".bashrc.custom". Add the following lines into the file.

export JAVA_HOME=/usr/local/jdk1.8.0_101
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

You can modify the path of JAVA_Home if you have your own JDK 1.8. Save it, and reconnect to the server to active this profile.

Then you can go to the program folder by using the following command.

cd /afs/cs.pitt.edu/usr0/colinzhang/public/Prj2HaoranZhang/socket_based

In this folder, there is a src fold contains the source code, a class fold contains compiled code. In addition, there are some .sh file to let you compile and run our program.

You can re-compile the program if you want by using the following command.

sh compile.sh

Now you can start the Server by using the following command.

sh startServer.sh [number of searching query master]

You can define the number of searching query master by yourself, the default number is 10 if you don't input the parameter.

Then you can start the Helpers by using the following command.

sh startHelper.sh [number of thread]

You can define the number of thread of a helper by yourself, the default number is 1 if you don't input the parameter.

The you can start the Client by using the following command.

sh startClient.sh

Now you can see the simple ui in shell format. Follow the instruction and input the commands and see the results.

Hadoop-Based

Since the version of Hadoop API of the cluster is too old to support new JDK version, such as JDK 1.8. Thus you have to delete or comment the lines we just added into the ".bashrc.custom" file (we mentioned it in the beginning of the socket-based part above). Re-connect to the server to active the default JDK which supported by the Hadoop API in our cluster.

Then you can go to the program folder by using the following command.

cd /afs/cs.pitt.edu/usr0/colinzhang/public/Prj2HaoranZhang/hadoop_based

In this folder, there is a src fold contains the source code, a class fold contains compiled code. In addition, there are some .sh file to let you compile and run our program. Also, there is file named "indexing.jar", this is the pre-compiled jar file for the project.

You can re-compile the program if you want by using the following command:

sh compile.sh

Before the you start Indexing and Searching, you may have to modify the sh files to make sure the path of output is suitable for yourself, so that you don't have to type in too much times of your own path.

In indexing.sh, you need to define a temp folder, the output folder, and an indexing folder. In searching.sh, you need to define the input folder same to the indexing folder in the indexing.sh. In addition, you also need to define the output folder for searching.sh.

For the indexing.sh:

For the searching.sh:

After you define all your path, you can create index for a folder by using the following command.

sh indexing.sh [prefix of folder] [path of input]

The prefix of folder is necessary, it is better to use the folder name as the prefix, so that we can assign an id for the result. Also, you can define the path of input by yourself.

e.g.

```
sh indexing.sh smallInput /user/colinzhang/prj2/smallinput
```

Then you can search keywords by using the following command.

sh searching.sh "[keywords]"

You can define the keywords, but be aware of the double quote in the command. They are necessary because the system may not recognize your multiple keywords, if you use space to split all keywords.

e.g.

For single word:

```
sh searching.sh "you"
```

For multiple words:

```
sh searching.sh "hello world"
```