University of Zurich

Distributed Systems - Assignment 1

Deadline: 23:59, 22 October 2013

October 2, 2013

In this course project, we will develop a search engine. First, we will work with a dataset of text documents and build an inverted index for the dataset. The engine will allow the user to enter a query as a sequence of words. By using the inverted index, the engine will retrieve a list of documents matching the user's query. We will develop this engine progressively through the assignments. We begin with Assignment 1, which is described below.

In this assignment, we will build the index as a in-memory data structure. We will work with a single document "Alice in Wonderland." Usually an inverted index maps a term to the documents that contain the term. In this assignment however, we will map a term to the line numbers that contain the term in the file.

Your job is to implement:
- a client and a server that communicate with each other using Sockets
- an inverted index for the terms in the file "Alice in Wonderland."

1 What we have provided:

- Engine.scala: This is the entry point to the Search Engine. The user passes a port number as command line argument. It then starts both the web server and the text processing server.

- a web server: WebServer.scala. The web server is started from Engine.scala and it listens for requests on port 8080. After receiving requests, it replies the String returned by RetrievalSystem.getResult(). Requests are queries provided by the user and the response is the result provided by the inverted index for the query.

- a web client: client.html. This is provided for you to test communication with the web server. You can load this file on a web browser and see how the communication takes place by clicking on the "Load Server Response" button. You can enter query in the text box and receive the server's response in the area below the button.

- An example test: In Engine.scala the test() method sends the query "pigs" to the web server. The getResult() method in the web server sends a reply. The response is checked and correctness is verified.

2 What you need to implement:

1. Fill the body of TextProcessingServer.scala. Specifically, you need to do the following:

- implement the Inverted Index logic. For this, you need to read the text file for "Alice in Wonderland" and maintain an in-memory data structure to act as the index

- please convert all text to lowercase and remove all commas (,) from the file before indexing the terms. Other text transformations are not necessary

- implement a server that listens on a port provided by the user.

- the server listens for query from a client process and replies with the result for the query.

- the result is a list of line numbers in which the terms in the query are present. You can see an example in the code we have provided: for the query "pigs", the server should reply "1463 2319" which are the line numbers containing the term pigs. For queries with more than one term, please take an intersection of the line numbers containing the terms.

2. Implement the method getResult in WebServer.scala.

- This method receives the query passed by the user. You need to implement a client that connects to the TextProcessingServer on the user-specified port. Then you should pass the query and handle the response.

- Please make sure to handle queries with multiple terms. In the example we have provided, the query contains only one term, i.e., "pigs," but the query passed from the web client may be of the form "pigs%20fly"- you should transform it into "pigs fly".

3 How to get started:

Extract the archive file we have provided. In the Scala IDE on your machines, go to File -> Import -> Existing Projects into Workspace -> and provide the location of the extracted directory titled "Eclipse." The text file for "Alice in Wonderland" is also included in the archive.

4 Submission Instruction:

Create a jar file with your matriculation number (e.g.: 1-234-5.jar) and upload it to the course system. To create the jar file, right click on the project in your IDE, click on Export -> Java -> JAR File -> (enter the name for the jar file, check "Export all output folders for checked projects" and "Export Java source files and resources") -> Finish.

5 Evaluation:

We will evaluate your submission by running a command like this on the command prompt: java -cp 1-234-5.jar:lib/scala-library.jar assignment1.Test 8090 query_file.txt

We have not included the Test class and query_file.txt in the code provided to you. It will include the following test cases and some additional ones (not provided to you). You will receive full credits if your submission passes all the tests.

The test cases provided to you are:

query	**response**
"lewis carroll"	"33 1 3375 11"
"alice in wonderland"	""
"web site search"	"3728"
"goldfish"	"3094 3099"
"queen king"	"2008 2180 2349 2831 1964"

Other test cases not provided to you will be similar to the ones above.

6 Deadlines:

Please try to start working on the assignments on time so that you can submit early. However, in cases you need some extra time, everybody is provided a pool of 4 days. If you need extra days for solving the assignments, you can use them in any combination you want. For example, you can use 2 slack days for 1st assignment and 2 for 2nd. Then, you won't have any extra late days for the 3rd assignment.