GlooGloo

SD Report First Part

Introduction:

So, in this project our goal is to create a search engine with similar functionalities to a very well-known search engine we all use on daily basis google. To achieve this goal, we will present a series of codes and scripts that make the clients able to interact with the functionalities the main scripts are as follows:

* Gateway.java
* Downloader.java
* IndexStorageBarrel.java
* Client.java

Structure:

**Gateway**: works as the brain of the operation interacting with the **Queue**, **Downloaders** and **Barrels**. In all scripts most of the operations are connected to the gateway and in a sense the gateway is our tool to interact with the program.

**Barrels**: They work as the main storage compartment of the system. Each barrel is Implemented as a separate database, but it is important to note that the database server for all barrels is the same. It is important to note that the Barrels do not communicate with each other, the synchronization process will be explained in detail later**.**

**Downloaders:** They work as the main workers of the system, they will visit the input URLs stored in the queue in order, and they proceed to index the page of the mentioned URL then they will send the data to the barrels to save them.

For a client to be able to run the program the following requirements must be met:

* Having installed the following: Maven, Postgres, Gradle, javac22
* Paying attention to check if all imported packages are working and are recognized

Functionalities:

* In the first functionality the client will input the desired URL after having established a valid connection to Gateway which will later be sent to a queue where it is saved and processed, later it will be sent to the downloaders which will index URL pages and save the data in the barrels, the choosing of which barrel to store the data in is done randomly. This functionality relates to the first functionality implemented to input and index a URL.
* The second functionality is for a client to search some terms(single word or a group of them) the way this functionality works is as follows the client will need to firstly connect to the gateway and input the desired terms the terms later will be compared to the existing indexation data in the barrels and the output of this search will include the series of links and URL which are the most related to the searched words.
* The third functionality is implemented to work exactly like the second with the difference that the input in this case will be URL, and the output will be the series of URL that are most connected and related to the inserted URL by the client.
* Finally, we have a fourth functionality aimed at administrative use which provides a series of statistics measures related to the words and URLs.

All the communication processes mentioned were done using RMI to invoke the necessary methods and to establish the necessary connections.

To execute the code, the client must run the Main but it is important that previously and in order the following files have to be online and running Queue,Gateway,Barrels,Main by a host or another machine.

After the successful execution of Main, the client will see a menu with the following options:

1. Insert URL
2. Search
3. Consult URL connections
4. Administrative page
5. Exit

Testing:

We provide the following table as confirmation of the tests we have done to make sure of the correct functioning of our code.

|  |  |  |
| --- | --- | --- |
| Description | Pass/Fail | Results and outputs |
| Client inserts entry that does not exist in the menu. |  |  |
| Client connects without having the Gateway connected. |  |  |
| Downloader connects without having Gateway connected. |  |  |
| Index Storage Barrel connecting without Gateway connected. |  |  |
| Index Storage Barrel breaks down, recover status? |  |  |
| Load balancing on storage barrel searches Is the information identical across all storage barrels? |  |  |
| Are indexing requests only answered by one downloader? |  |  |
| Does the gateway recover from any internal failures? |  |  |
| Do downloaders run in parallel? |  |  |

Work distribution:

The strategy in this matter was that 2 of the elements work and focus on the implementation and the other element focusing on the documentation sides of things regarding the report, how to set up md and other necessary documentation. It is important to note though that all members were present during the process of the development to be in synchronization with the updates and states.

Documentation

In addition to this report a mark down file (README) and a Javadoc file will be available to the clients to better understand and work with the programs.