Information Retrieval and Web Search - Project Phase 2

Lukas Pfahler Tejas Umakanth

March 10, 2014

1 Collaboration Details

Lukas implemented the web interface, the lucene search engine backend and the hadoop backend including the ranking algorithm used for hadoop search. Tejas implemented the mapreduce jobs that created the index json files.

2 Architecture

2.1 Indexing

2.2 Webinterface and Search Engine

For the web interface we decided to go with node.js again. We start a http server on port 80 and wait for requests. Once a user enters a query, our webserver opens a TCP connection to our searchengine backbone, specifying whether to use Lucene or Hadoop and relaying the query. The backbone executes the query and returns the results as JSON. The resultsare then output to the user by the webserver. Additional information about the documents is loaded in an asynchronus fashion using AJAX and JSON.

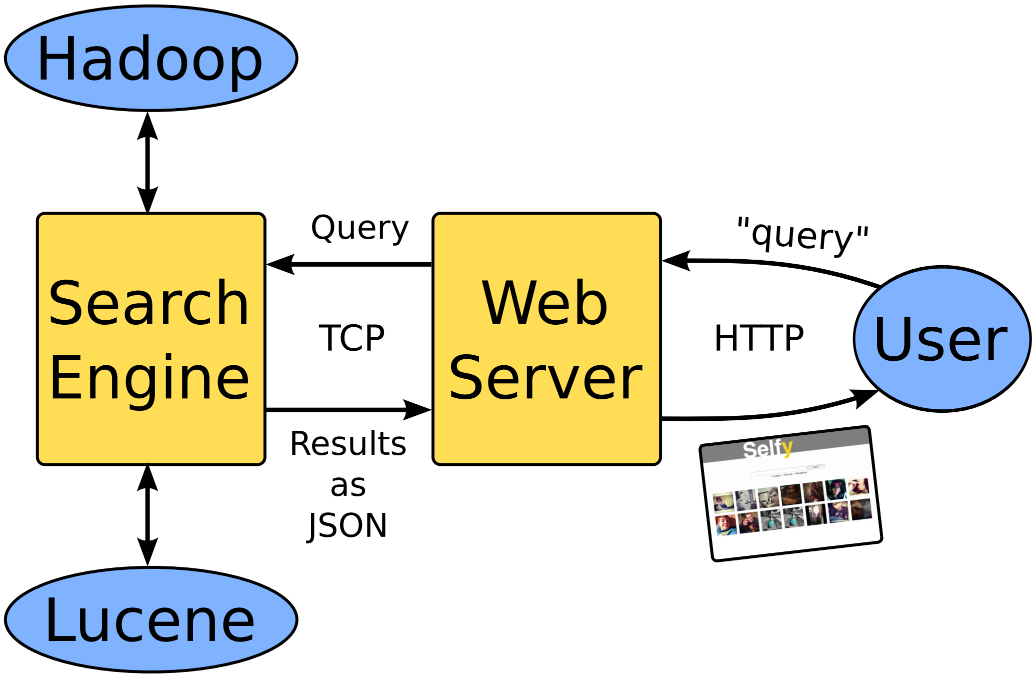

Figure 1: An overview of our search engine architecture.

2.3 Searching

3 Usage

3.1 Crawler

To run our crawler, you first have to install node.js, which you can download at

`http://nodejs.org/download/`

We are also using the request node-module, which you can download and install by running the following command:

```
npm install request
```

Now you can run our crawler by calling

```
node instagramStream.js
```

which will start a http-server on port 1337 of your local machine. Please note that port 1337 of your machine has to be open to the public. To actually start crawling, you have to subscribe to instagram media updates. This is done by calling the following command:

```
curl -F 'client_id=620c46f7e7da46149b0ad5d3cd8268ba' \
     -F 'client_secret=c57beab5c2864e608b0dc4e46e5f8b50' \
     -F 'object=tag' -F 'aspect=media' -F 'object_id=selfie' \
     -F 'callback_url={http://YOUR_IP:1337}' \
     https://api.instagram.com/v1/subscriptions/
```

This will subscribe to all updates of the (very popular) hashtag #selfie; the crawler will start storing media items in a newly created folder selfies.

You can unsubscribe from updates by running

```
curl -X DELETE 'https://api.instagram.com/v1/subscriptions?client_secret=\
     c57beab5c2864e608b0dc4e46e5f8b50&object=all&\
     client_id=620c46f7e7da46149b0ad5d3cd8268ba'
```

You should always unsubscribe before shutting down the crawler (`CTRL-C`), if you fail to do so you might not be able to subscribe to media for a while. We suspect that the servers at Instagram need some time to realize that our crawler is not running anymore and continue to try to send the new subscriptions to our already shutdown crawler.

3.2 Indexer

To deploy our indexer, you have to execute the jar file by typing

```
java -jar indexer.jar {datadir}
```

An 'index' folder will be created in the file containing the all the files of the project.

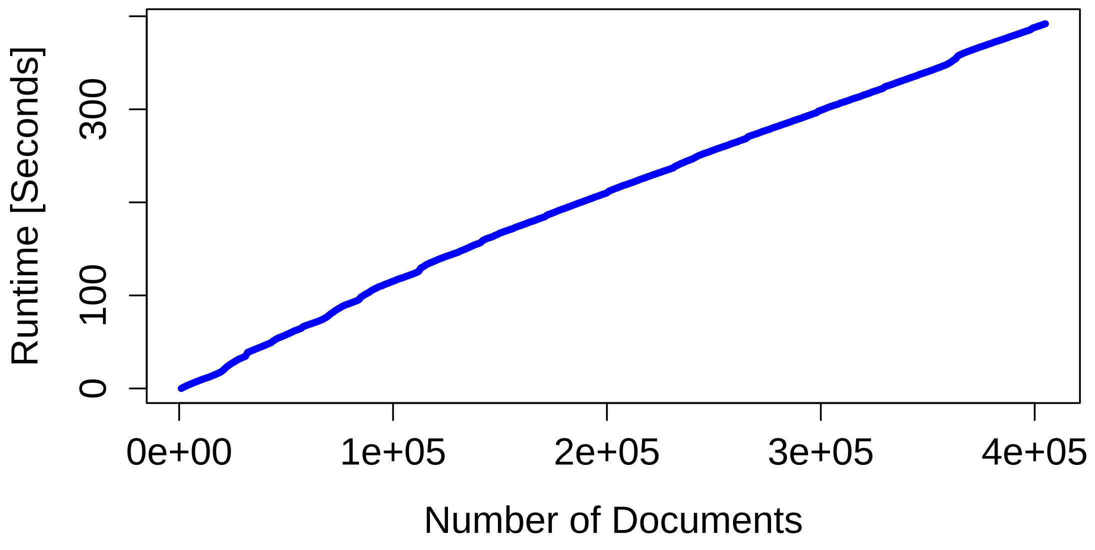

Figure 2: Indexing Performance

4 Performance

The time needed to for indexing is dependent on the system on which the indexing is done. This performance analysis was carried out on a laptop with a 2.4 GHz CPU. For performing this test, we use a document containing 3GB worth of JSON files that are returned by our crawler. Then we try to index the JSON files in the document and calculate the time taken to index. As we can see in figure 1, the indexing time increases almost linearly as the number of JSON files that are indexed increases.

5 Screenshots