IDC – Cloud Computing
Exercise 3 – 20%
Due Date 20/8/2013

Spring Semester 2013
Instructor: Mr. Dan Amiga

Objective

- Write a Map-Reduce program on hadoop.
- This is not a joint work, this homework is **not** to be done in couples. You should do it on your own.

Submission

- You are required to submit a zip file that contains (1) the solution itself. (2) A CSV file with the inverted index output for 5 websites in the internet.
- You will place a single zip file in S3 in the following structure: **ALL LOWER CASE LETTERS**
  https://s3.amazonaws.com/lastnamefirstname/homework/lastnamefirstnameex3.zip
- Please use the following google form to submit the homework:
- https://docs.google.com/forms/d/1x6u-yCqAkQdyo7kwzOQFWgXI2eO5_1cxRmec-WX8tpM/viewform
-

Brain dump instructions

- Start simple on your laptop and the move to Hadoop (locally or on AWS).
- Use Google for search…
- Tip – two easy ways to get started on Hadoop quickly!
  - Amazon EMR (go through the document and actually do it; it's a great document).
    http://docs.amazonwebservices.com/gettingstarted/latest/emr/
  - Cloudera downloadable virtual machine and tutorial
    - https://ccp.cloudera.com/display/SUPPORT/CDH+Downloads#CDHDownloads-CDH4PackagesandDownloads
    - www.cloudera.com/content/cloudera-content/cloudera-docs/HadoopTutorial/CDH4/Hadoop-Tutorial.html

Part 1 – Building a simple inverted index + very simple ranking system

Read carefully!

1. An inverted index is a very simple concept that is used by most modern search engines. Essentially it's a hash table where the key is a word, and the value is a list of all the documents the word resides in. for example:

Key	Value
Dan	http://www.ynet.co.il, http://www.cnn.com
Israel	http://www.idc.ac.il, http://www.ynet.co.il, http://www.cnn.com, http://.....
IDC	http://www.idc.ac.il, http://....

   You can see how easy it is given a word to return all the documents it appears in. Google for more info.
2. You are to write a map reduce program that runs on Hadoop or Amazon EMR (or both) that builds an inverted index.
   a. The only constrain is that the output should be a file (or list of files) in the form of CSV, where the first column is the key and the rest are the values (as the table above).
   b. There is a small caveat though: sometimes the key itself (e.g. the current word you are working on in the current document) is actually a link to a different page, that is, the key is an http:// URI. In this case instead of outputting the key and the document it resides on, just output the URL as Key and the number of occurrences that the current link was found (similar to wordcount).
   c. A partial example of what you will get:
      Dan, http://www.ynet.co.il, http://www.cnn.com
      Israel, http://www.idc.ac.il, http://www.ynet.co.il, http://www.cnn.com, http://....
      IDC, http://www.idc.ac.il, http://....
      http://www.ynet.co.il, 20
      http://www.idc.ac.il, 4
      http://www.cnn.com , 980
   d. The bigger the occurrence number is, it means more pages are linking to it, which probably means this is a very popular site. Now, if you had too (you don't) build a search application, whenever the user types in a keyword you can look that keyword in the inverted index, retrieve all the URLs and then sort them by their ranking. (note this is very simplicitc for the sake of the exercise and could be improved in a lot of ways).

3. Decide on an efficient data model and write a program that inserts the inverted index CSVs into Amazon Dynamo DB. Note that you have at least two entities – Rank and InvertedIndex.