Cloud Computing and Big Data Ecosystems Design

Hbase Application

- **Description:** The goal of this assignment is to implement a Java application that stores trending topics from Twitter into HBase and provides users with a set of queries for data analysis. The trending topics to load in HBase are stored into text files with this format:
    - 1 file per language.
    - Each line of the file in CSV format "timestamp_ms, lang, tophashtag1, frequencyhashtag1, tophashtag2, frequencyhashtag2, tophashtag3, frequencyhashtag3".

  Students must define the schema of a table named **twitterStats**, load the data into it, and program these three queries that access the data stored in HBase:
    1. **Given a language (lang), find the Top-N most used words for the given language in a time interval defined with a start and end timestamp. Start and end timestamp are in milliseconds.**
    2. **Find the list of Top-N most used words for each language in a time interval defined with the provided start and end timestamp. Start and end timestamp are in milliseconds.**
    3. **Find the Top-N most used words and the frequency of each word regardless the language in a time interval defined with the provided start and end timestamp. Start and end timestamp are in milliseconds.**
- **Deadline:** 29th January 2017
- **TO BE RELEASED: All the required files must be uploaded to Moodle by the deadline.** The file name must be ID.rar (ID is the id of the students) and has the structure of a maven project (without the target directory):
    - ID.rar
        - hbaseApp/
            - pom.xml/

- src/

The pom.xml of the hbaseApp must be configured with the appassembler maven plugin and with the java compiler set to 1.7. The name of the appassembler script to launch the hbaseApp must be **hbaseApp.sh**.

- **Groups:** The same groups as for the project 1 assignment.
- **Requirements:**
  - The application must be developed using the versions of the software (Oracle Java 7) installed in the computers available for the students (CESVIMA) and deployed using Ubuntu 14.04.
  - Hadoop version: hadoop-2.5.0-cdh5.3.5
  - HBase version: hbase-0.98.6-cdh5.3.5
  - HBase cluster must have 3 Region Servers (each one running in a different machine).
- **How to use the application:**
    - Script name: **hbaseApp.sh**
    - Script parameters:
      - *Mode*, integer whose value can be:
        - 1: run first query
        - 2: run second query
        - 3: run third query
        - 4: create the table **twitterStats** and load data files
      - zkHost: string with the format IP:PORT
      - startTS: timestamp in milliseconds to be used as start timestamp.
      - endTS: timestamp in milliseconds to be used as end timestamp.
      - N: size of the ranking for the top-N.
      - Languages: a cvs list of languages.
      - dataFolder: path to the folder containing the files with the trending topics (the path is related to the filesystem of the node that will be used to run the HBase app). File names have the format "*lang*.out", for example en.out, it.out, es.out…
      - outputFolder: path to the folder where the files with the query results are stored.
  - **According to the *mode* parameter, the script will be used with the following parameters:**
    - Load: ./hbaseApp.sh mode ZKHOST:ZKPORT dataFolder
      - Ex:./hbaseApp.sh 4 cesvima123:2181 /local/data
    - Query1: ./hbaseApp.sh mode ZKHOST: ZKPORT startTS endTS N language outputFolder
      - Ex:./hbaseApp.sh 1 cesvima123:2181 1450714465000 1450724465000 7 en /local/output/
    - Query2: ./hbaseApp.sh mode ZKHOST: ZKPORT startTS endTS N language outputFolder
      - Ex:./hbaseApp.sh 2 cesvima123:2181 1450714465000 1450724465000 5 en,it,es /local/output/
    - Query3: ./hbaseApp.sh mode ZKHOST: ZKPORT startTS endTS N outputFolder

- Ex:./hbaseApp.sh 3 cesvima123:2181 1450714465000 1450724465000 10 /local/output/

- Output with results:
  - One output file for each query to be stored in the folder specified with the outputFolder input parameter.
  - Filenames must be: ID_query1.out, ID_query2.out, ID_query3.out
  - File format: language, position, word, startTS, endTS
    where *startTS* and *endTS* are the ones used as input parameters
  - In case of words with the same frequency the ranking is done according with the alphabetic order.
  - Multiple executions of the same query must use the same file to store the results without overwriting previous results.

- The script **hbaseApp.sh** must be tested on the Ubuntu nodes available for students and must be created using the appassembler maven plugin.