Big Data Systems

Semester Assignment

The objective of this assignment is for you to become more comfortable working with Apache Spark using Scala or Java API. It covers the fundamental functionality of Spark Core and Spark SQL library.

The deliverable of this assignment will be a PDF in the form of a small report containing **your code** for answering each one of the questions along with an **explanation** on why you use each Spark Transformation or Action on every step of each solution.

***Code without explanation will NOT be considered as valid.**

You can freely choose between Scala and Java as your programming language.

Extra reading material can be found at:
https://spark.apache.org/docs/1.6.0/
https://www.safaribooksonline.com/library/view/learning-spark/9781449359034/ch04.html

Datasets are residing in the specified HDFS paths at
http://83.212.102.157:8888/filebrowser

For executing your code you can use the Scala interactive console at:
83.212.102.157

User: b-analytics
Pass: B-@n@lytic$

Cluster Yarn URL: http://83.212.102.157:8088/cluster

Type (Console): spark-shell --deploy-mode client --master yarn

Question 1 (25 points)

a) Create a Spark program to read the airport data from "/user/b-analytics/assignment/airports.text", find all the airports which are located in Greece and output the airport's name, the city's name and the airports , IATA/FAA code **(15 points)**

Each row of the input file contains the following columns:

Airport ID, Name of airport, Main city served by airport, Country where airport is located, IATA/FAA code, ICAO Code, Latitude, Longitude, Altitude, Timezone, DST, Timezone in Olson format

Sample output:

```
("Alexion","Porto Heli","PKH")
("Andravida","Andravida","PYR")
("Agrinion","Agrinion","AGQ")
```

b) Create a Spark program to read the airport data from “/user/b analytics/assignment/airports.text”, find all the airports whose latitude are greater than 37 and lower than 39. Then output the airport's name and the airport's latitude. **(10 points)**

Sample output:

```
(("Sidi Ahmed Air Base","Bizerte",37.245447),
("Albacete","Albacete",38.948528),
("Alicante","Alicante",38.282169))
```

Question 2 (30 points)

"/user/b-analytics/assignment/nasa_19950701.tsv" file contains 10000 log lines from one of NASA's apache server for July 1st, 1995. "/user/b-analytics/assignment /nasa_19950801.tsv" file contains 10000 log lines for August 1st, 1995.

a) Create a Spark program to generate a new RDD which contains the hosts which are accessed on BOTH days. **(15 points)**

Sample output:

```
alyssa.prodigy.com,
www-d1.proxy.aol.com,
piweba4y.prodigy.com,
.....
```

Keep in mind, that the original log files contains the following header lines.

```
host    logname    time    method    url    response    bytes
```

Make sure the head lines are removed in the resulting RDD.

b) Create a Spark program to read the an article from "/user/b-analytics/assignment/word_count.text", output the number of occurrence of each word in descending order. **(15 points)**

```
Sample output:
 (the,71)
 (of,33)
 (in,21)
```

Question 3 (45 points)

a) Create a Spark program to read the house data from "/user/b-analytics/assignment//Real.csv", output the average price for houses with different number of bedrooms. **(25 points)**

```
Sample output:
  (3, 325000)
  (1, 266356)
  (2, 325000)
  ...
```

3, 1 and 2 mean the number of bedrooms. 325000 means the average price of houses with 3 bedrooms is 325000

b) From the same dataset now using spark SQL group by location, aggregate the average price per SQ Ft and max price, and sort by average price per SQ Ft. (Keep in mind, that the original log files contains header lines.) **(20 points)**

Sample output:

```
+-----------------+-----------------+---------+
|              loc|             avgp|      max|
+-----------------+-----------------+---------+
|       King City| 71.51333333333334| 167770.0|
|      Santa Ynez|391.33000000000004|1395000.0|
|        Lockwood|            283.33| 425000.0|
| Santa Margarita|             95.38|  59900.0|
```

The houses dataset contains a collection of real estate listings.

The dataset contains the following fields:
1. MLS: Multiple listing service number for the house (unique ID).
2. Location: city/town where the house is located.

3. Price: the most recent listing price of the house (in dollars).
4. Bedrooms: number of bedrooms.
5. Bathrooms: number of bathrooms.
6. Size: size of the house in square feet.
7. Price/SQ.ft: price of the house per square foot.
8. Status: type of sale. Three types are represented in the dataset: Short Sale, Foreclosure and Regular.

Each field is comma separated.