Big Data Processing

Apache spark lab

Author: François Amat

Contact: amat.francois@gmail.com

Introduction:

This lab is an Introduction to the scala programming langage and spark ecosystem.

The data used are tweets from this link.

The tool used is the Community Edition of Databricks which include a small cluster, enough for this lab.

The scala notebook with all source code can be found here

Instructions:

Write a notebook on the following tasks, writing the code in Scala:

1. Find hashtags on tweets
2. Count hashtags on tweets
3. Select the 10 most frequent hashtags
4. Select the 10 users with more tweets
5. Detect trending topics

First we have to import the database :

```
import sys.process._
import org.apache.spark.sql._
import org.apache.spark.sql.functions._
```

```
import org.apache.spark.sql.types._
"wget -P /tmp https://www.datacrucis.com/static/www/datasets/stratahadoop-BCN-2014.
val localpath="file:/tmp/stratahadoop-BCN-2014.json"
dbutils.fs.mkdirs("dbfs:/datasets/")
dbutils.fs.cp(localpath, "dbfs:/datasets/")
display(dbutils.fs.ls("dbfs:/datasets/stratahadoop-BCN-2014.json"))
val df = sqlContext.read.json("dbfs:/datasets/stratahadoop-BCN-2014.json")
val rdd = df.select("text").rdd.map(row => row.getString(0))
```

1. The hashtags are found by this command :
   2.

```
val hashtags = df.select("entities.hashtags.text").as[String].collect()
```

2. To count hashtags on tweets we use map reduce.

```
val hashtags = df.select("entities.hashtags.text").as[String]
val rdd2 = hashtags.rdd.map(word => word.
                            slice(1, word.length - 1 ).
                            toLowerCase().replaceAll("\\s", ""))
val wc2 = rdd2.
  flatMap(_.split(",")).
  map(word => (word,1)).
  reduceByKey((a,b) => a+b)

wc2.sortBy(-_._2).take(100).foreach(println)
```

3. To print the first 10 most frequent hashtags we sort before printing: `wc2.sortBy(-_._2).take(10).foreach(println)`

4. We also use map reduce to get the 10 most active users.

```
val users = df.select("user.id").as[String]
val rdd3 = users.rdd.map(row => row)
val wc3 = rdd3.map(word => (word,1)).reduceByKey((a,b) => a+b)
wc3.sortBy(-_._2).take(10).foreach(println)
```

5. We use the hashtags and the time to get the trending topics. It can be noted that for detecting trending topic a stream could be better in real life situation.

```
val all = df.select("created_at","entities.hashtags.text").as[(String,String)]
val rdd4 = all.rdd.map(word => (word._1.slice(4,11) ,
            word._2.
            slice(1, word._2.length - 1 ).
            toLowerCase().replaceAll("\\s", "")))
val wc4 = rdd4.
      map(word => ((word._1,word._2),1)).
      reduceByKey((a,b) => a+b)
wc4.sortBy(r => (-r._2,r._1)).take(10).foreach(println)
```

Conclusion

This lab was useful in order into getting started with apache spark and scala.
After few hours playing with the spark dataframe it became really useful.
I think the next step will be to implement a spark streaming program using the tweeter api to detect trending topics.