Big Data and Artificial Intelligence

Lab Assignment (TP) 2: Big Log Data Analytics

Vassilis Christophides

In this lab assignment you will :

- Compare the functionality of RDDs and Dataframes
- Understand data typing (i.e., schema) for analyzing structured files
- Learn how to use SQL queries on Dataframes
- Familiarize with performance issues when processing large volumes of data

Exercise 0: Download and Prepare your Log File

The Wikimedia Foundation supports hundreds of thousands of people around the world in creating the largest free knowledge projects in history. The work of volunteers helps millions of people around the globe discover information, contribute knowledge, and share it with others no matter their bandwidth. In this assignment you are going to explore the page views of Wikimedia projects. Download or copy in Google drive the zip file `pagecounts-20160101000000_parsed.out.zip` with the page view statistics generated between 0-1am on Jan 1, 2016 from the following URL:
`https://drive.google.com/file/d/1qr-SBzlojgxzXu2fJx0xaWH9P6vrDpnS/view`

Each line, delimited by a white space, contains the statistics for one Wikimedia page according to the following schema:

Field	**Meaning**
`Project`	The project identifier for each page
`Page title`	A string containing the title of the page
`Page hits`	Number of requests on the specific hour
`Page size`	Size of the page

Then, load (locally in your PC or in Google colab) and read in your Spark Instance the contents of the file as an RDD and a DataFrame and compare the respective data types.

Exercise 1: Explore Web Logs with Spark RDDs

In this exercise you need to convert the Spark type of the Data Frame you created for the file `pagecounts` from `RDD[String]` into `RDD[Log]` according to the following instructions:

1. Create a schema Log of the `pagecounts RDD` (using `namedtuple`) using the above four field names and types

2. Write a function that takes a string, split it by white space and converts it into an object of type Log
3. Convert an `RDD[String]` to an `RDD[Log]` using the `map()` function
4. Use the operator `.attname` on your rdd to access the value of the attribute `attname`

For each of the questions below, implement a Scala or Python function that takes as input an RDD[Log] and prints the requested values. As your logRDD will be used multiple times, in each of the questions bellow, it is better to *persist*/*cache* the *logRDD* in memory. Note that we finally *unpersist* the *logRDD* from the memory, when all executions are completed. You should include in your report both the code you wrote to implement the queries as well as their respective results.

Question 1

Retrieve the first 20 records and beautify the results.

Hint: As the `take()` operation returns the first k (=20) records of an RDD and prints an array of its element separated by a comma, you can make the output more readable by traversing the array to print each record on its own line. To beautify the prints, you can create a function `print_record()`, which takes as input a Log objects and prints a new line with all of its fields to be separated by the tab character, as it follows:

```
"ProjectCode:"+…+"\t PageTitle:"+…+"\t PageHits:"+…+"\t PageSize:"+…+"
```

Question 2

Find the total number of records in the dataset.

Question 3

Compute the min, max, and average page size.

Hint: Use `map()` function in conjunction with `max()`, `min()` and `mean()` provided by the RDD API.

Question 4

Find the record(s) with the largest page size. If multiple records have the same size, list all of them.

Question 5

Find the most popular record(s). If multiple records have the same popularity, list all of them in decreasing page size.

Question 6

Use the results of Question 3, and create a new RDD with the records that have greater page size than the average.

Question 7

Report the 10 most popular pageviews of all projects, sorted by the total number of hits. Then report the 5 most popular projects based on the pageviews of their pages.

Hint: To sort the contents of an RDD you will need to execute `sortByKey()`. You will also need to group pages per project and sum their hits using `reduceByKey()`.

Question 8

Find the unique words occurring in the page titles.

Hint: Note that in page titles, words are delimited by "_" instead of a white space. You can use any number of normalization steps (e.g. lowercasing, removal of non-alphanumeric characters) as we shown in lab assignment 1.

Question 9

Find the most frequently occurring page title words in this dataset.

Exercise 2 – Query Web Logs with Spark SQL

First convert the `pagecounts` from `RDD[String]` into `DataFrame` using the `toDF()` function with appropriate arguments similarly to the examples found in the following URL: `https://sparkbyexamples.com/pyspark/convert-pyspark-rdd-to-dataframe/`

Hint: You may need to transform `RDD[String]` into a `DataFrame` of type `Log` using `StructType`. The resulting `DataFrame` (DF) should look similar to the following Figure as returned by `show(n)`:

```
+-------+--------------------+----+------+
|project|               title|hits|  size|
+-------+--------------------+----+------+
|     aa|             271_a.C|   1|  4675|
|     aa|    Category:User_th|   1|  4770|
|     aa|  Chiron_Elias_Krase|   1|  4694|
|     aa|    Dassault_rafaele|   2|  9372|
|     aa|              E.Desv|   1|  4662|
|     aa|File:Wiktionary-l...|   1| 10752|
|     aa|Indonesian_Wikipedia|   1|  4679|
|     aa|           Main_Page|   5|266946|
|     aa|Requests_for_new_...|   1|  4733|
|     aa|Special:Contribut...|   1|  5812|
|     aa|Special:Contribut...|   1|  5805|
|     aa|Special:Contribut...|   1|  5808|
|     aa|Special:Contribut...|   1|  5812|
|     aa|Special:ListFiles...|   1|  5035|
|     aa|Special:ListFiles...|   1|  5036|
|     aa|Special:ListFiles...|   1|  5032|
|     aa|Special:Log/Md._F...|   1|  5529|
|     aa|Special:Log/MikeL...|   1|  5368|
|     aa|Special:MyLanguag...|   1|  4701|
|     aa|Special:RecentCha...|   1|  6152|
+-------+--------------------+----+------+
```

Next, you should use the DF to answer again to the questions 3, 5, 6, 8 and 9 of Exercise 1, but this time by *running SQL queries programmatically* (see the tutorial available at `https://spark.apache.org/docs/3.0.1/sql-programming-guide.html`). You should also include in your report both the code you wrote to answer the queries as well as their results in tabular format that is each to read. You should also include the runtime of the

code you run along with your comments regrading the performance of the functionality you implemented in RDD (Exercise 1) and DataFrame (Exercise 2).

Hint: From the DF API, you have to use the following functions: `sql`, `show()`, `createTempView()`