

Lab 6

In this lab, we continue our work on the Amazon dataset using Apache Spark. Your task is similar to that of Lab 3: given the original Amazon food dataset (that you can find in the HDFS file system at `/data/students/bigdata-01QYD/Lab3/Reviews.csv`), find all the pairs of items frequently reviewed together.

In the following Ex. 1 you find the steps that you are required to perform on the dataset.

Ex. 1

Write a single Spark application that:

- Transposes the original Amazon food dataset, obtaining a PairRDD of the type:
`<user_id> → <list of the product_ids reviewed by user_id>`
- Counts the frequencies of all the pairs of products reviewed together;
- Writes on the output folder all the pairs of products that appear more than once and their frequencies. The pairs of products must be *sorted by decreasing frequency*.

The input Amazon food dataset (available in the HDFS shared folder of the BigData@Polito cluster: `/data/students/bigdata-01QYD/Lab3/Reviews.csv`) lists all the reviews per-row (one review per line), and is comma-separated. In each line, two of the columns represent the user id and product id. The schema of `Reviews.csv` is the following:

```
Id,ProductId,UserId,ProfileName,HelpfulnessNumerator,HelpfulnessDenominator,Score,Time,Summary,Text
```

Inspect the output of your application to search for interesting facts, and analyze the job execution as usual (performances, number of executors, etc...).

Pay attention that the line starting with "Id," is the header of the file and must not be considered.

On the web site you can download the file `ReviewsSample.csv`. It contains a sample of `Reviews.csv`. You can use it to perform some initial tests locally.

Bonus task

Extend the implemented application in order to write on the standard output the top 10, most frequent, pairs and their frequencies.

Note that Spark 1.6, or above, provides the following actions that can be applied on an RDD of type `JavaRDD<T>`:

- 1) `List<T> top(int n, java.util.Comparator<T> comp)`
- 2) `List<T> takeOrdered (int n, java.util.Comparator<T> comp)`

top returns the **n largest elements** of the RDD based on the specified Comparator

takeOrdered returns the **n smallest elements** of the RDD based on the specified Comparator

Note that the standard output of the driver is stored in the log files of your application. Use the following steps to access the log files.

How to access logs by using the job history web page

The logs of your applications on the cluster is available at <https://ma1-bigdata.polito.it:18489/>
If there is any error and you need to analyze the content of the error log files follow these steps:

1. Open a new terminal on your local machine (do **not** use the previous terminal connected to bigdatalab.polito.it).
2. On the local terminal command line, type the command **kinit sXXXXXX**
(then, you will be asked for your BigDataLab password)
This command is used to obtain a Kerberos ticket (without it you cannot access the log web page)
3. Now you should have access to the spark job history and logs at <https://ma1-bigdata.polito.it:18489/>
4. Select your job and click on its App ID
You can use the search box to select only your jobs
5. Click on the executor tab and analyze the stderr associated with your application.
Pay attention that there is one log file for the driver and one log file for each executor.
If your application raised an exception, analyze all the stderr files to identify the reason of the error (depending on the error type, the log containing the needed information can be the one associated with the driver or that of an executor)

How to access logs without the web page and the Kerberos ticket

If you are connecting from outside Polito, and hence 1) you cannot request a Kerberos ticket and 2) you cannot access the history web page, you can proceed as follows to retrieve the log files from the command line:

1. Open a shell on the gateway by using ssh
`ssh smatricola@bigdatalab.polito.it`
2. Execute the following command in the remote shell:
`yarn logs -applicationId application_1521819176307_2195`
The last parameter is the application/job ID. You can retrieve the job ID of your application on the HUE interface: <https://bigdatalab.polito.it:8080/hue/jobbrowser#!jobs>