Laboratory 1 - Introduction to Spark

In this laboratory you will learn how to run a simple Spark program using the Big Data cluster of Politecnico.

Note that you have access to the Big Data cluster using the web interface at: `jupyter.polito.it` and using `ssh` to the machine `bigdatalab.polito.it` .

1. Run a simple Spark job

In this exercise you will run a simple Spark program in three different ways. You will first run the program in a Jupyter notebook. Then you will run the same code using the pyspark shell. Finally you will run it from the command line.

1.1 Jupyter notebook

Open a browser and connect to `jupyter.polito.it` . Log in and open a **Pyspark (local)** notebook. Then execute the following lines of code:

```
rdd = sc.textFile("/data/students/bigdata_internet/lab1/lab1_dataset.txt")
fields_rdd = rdd.map(lambda line: line.split(","))
value_rdd = fields_rdd.map(lambda l: int(l[1]))
value_sum = value_rdd.reduce(lambda v1, v2: v1+v2)
print("The sum is:", value_sum)
```

Question: Which value is printed by the print statement? Which is the purpose of each line of code?

Question: Where is the input file? On which file system?

Question: What happens if you open a **Pyspark (YARN)** notebook? Which is the difference in the Cluster manager interface at https://bigdatalab.polito.it? (https://bigdatalab.polito.it?)

Hint: look at the Jobs tab.

1.2 Execute in a pyspark shell

Always using the `jupyter.polito.it` interface, start a terminal and execute the `pyspark --master local --deploy-mode client` command to start the shell. Then, execute the same code inside the shell.

Question: What does `--master local` mean? And what about `--deploy-mode client` ? Where spark executors and driver executed?

1.3 Create a Spark script and run it from the command line

Now you have to create a Spark script and execute it the the `spark-submit` command line utility. To run a Spark script with `spark-submit`, your script must explicitely create the SparkContext object. To do this, prepend to the code the following lines:

```
from pyspark import SparkConf, SparkContext
conf = SparkConf().setAppName("My app")
sc = SparkContext(conf = conf)
```

Then, launch with:

```
spark-submit --master local --deploy-mode client <your_script_file>
```

Note that `<your_script_file>` must end by `.py` for Spark to interpret it as Python.

Question: In which file system are located your script and the `/data/students/bigdata_internet/lab1/lab1_dataset.txt` files? Are they on the same file system?

2. Play with HDFS

Now, try to manipulate files on HDFS For example, to list the files of a directory run:

```
hdfs dfs -ls /data/students/bigdata_internet/lab1
```

To print a help of the available options of the `hdfs` command, write:

```
hdfs dfs -usage
```

Now copy the HDFS file `/data/students/bigdata_internet/lab1/lab1_dataset.txt` in the local file system.

Note: You can obtain the same results with the web interface at: https://bigdatalab.polito.it/ (https://bigdatalab.polito.it/). Use it to check the files in your HDFS home directory.

Question: if you modify the local file, does the modifications automatically affect also the HDFS file?

Now create a file in the file system of the gateway and copy it to your home in HDFS.

Question: Which is the complete path of your file in HDFS? And on the gateway local file system?

3. Run a Job

Now, consider the same input file `/data/students/bigdata_internet/lab1/lab1_dataset.txt`. Write a Python script that reads the file, computes the sum of values for each person, and saves the output in a HDFS folder.

The output file should look like:

```
bob,34
alice,23
```

It is important that the output has the form `<name>,<sum>`.

Hint: use the `reduceByKey` transformation to compute the person-wise sum, and the `saveAsTextFile` action to save the RDD.

Note: `saveAsTextFile` cannot overwrite the content of an existing directory. You can use the command `hdfs dfs -rm -r <folder>` to delete a folder.

Question: Report the code you have written, and explain the goal of each instruction.

Question: How does the output folder on `HDFS` look like? Why do you find multiple `parts` file in the folder?

4. Bonus Task

Always considering `/data/students/bigdata_internet/lab1/lab1_dataset.txt`, write a script that reads the file, and concatenates all values for a name, separating them by `:`. Then, it saves the output in a HDFS file.

The output file should look like:

```
bob,5:3
alice,5:3:7
```

It is important that the output has the form `<name>,<v1>:<v2>...`, one line for each name.

Hint: You can accomplish this task in many ways. Consider the transformations `reduceByKey`, `groupByKey` and `aggregateByKey`.

Question: Report the code you have written, and explain the goal of each instruction.

How to write and submit the report

In your report, you must answer all questions and report the code/commands that you used to complete the exercise. You must follow the order in which question and exercise are posed.

Your report must be a PDF generated from a Jupyter notebook. Go on `File-> Export Notebook As ... -> Export Notebook As PDF`. It must be submitted through the Teaching Portal course page (didattica.polito.it).

Naming convention: The file must be named with the following schema:

```
s<id>_lab<n>.pdf
```

For example, if you student ID is 123456 and you are submitting lab 1, the file must be: `s123456_lab1.pdf` Reports with a wrong file name will **NOT** be considered.

Report lenght: the report length must not exceed 10 pages, otherwise it will **NOT** be considered valid.