✔ Terminé

In this TP, you must write your code in Scala or Python. The use of the interactive shell is strongly recommended. But remember to copy your code into a file so you can save it.

Study the RDD programming guide

The guide is available here. Read the "Resilient Distributed Datasets (RDDs)" section. Notice you can click to choose the language (Scala, Python, or Java) in which to show the examples.

Using Spark

Start by reading and understanding the provided examples in `spark_examples.tgz` (available on Moodle): `miserables.py` for Python, `miserables_scala/src/main/scala/miserables.scala` for Scala.

Python

To start the interactive shell, use the following command:

```
pyspark --master yarn
```

If you don't want to use the shell, and want to run a program from a .py file instead, use the following command. You can test it with `miserables.py`

```
spark-submit --master yarn [YOUR FILE]
```

Scala

To start the interactive shell, use the following command:

```
spark-shell --master yarn
```

If you don't want to use the shell, and want to run a program from a Scala source file instead, you'll need to create a project and compile it with maven.

```
cd miserables_scala
mvn clean
mvn package
spark-submit --master yarn [.jar file created under target/]
```

Controlling the number of executors

Notice that, when using Yarn (as we are using in LSD), we can control the number of executors used for the Spark application by passing the `--num-executors` option to `spark-submit`.

Climatological Database for the World's Oceans

This exercise was designed in collaboration with Thomas Ropars, from the Université Grenoble Alpes

The data set we will study comes from the project Climatological Database for the World's Oceans. It has been created starting from the logbook of ships that were navigating the oceans during the 18th and 19th centuries. These logbooks were maintained by the crew of the ships and contain a lot of information regarding the weather conditions during that period.

Each observation reported in the data set includes many entries including the date of the observation, the location, the air temperature, the wind speed, etc. A detailed description of all included fields is available here. The data set is available in the Hadoop cluster of LSD as `/user/fzanonboito/CISD/cliwoc15.csv`.

Notice that in the data set, an entry with no value is represented by the string "NA" (stands for "Not Available")

1. Read the file and make an RDD containing one element per line of the file (except the first line, which must be ignored). Each element must be a list of strings (separate each line using split(',')).

2. Count the total number of observations included in the data set (each line corresponds to one observation).

3. Count the number of years over which observations have been made (Column `Year` should be used, it is column 40).
4. Display the oldest and the most recent years of observation. You may use the `min()` and `max()` actions.
5. Display the years with the minimum and the maximum number of observations (and the corresponding number of observations).
6. Count the distinct departure places (column `VoyageFrom`, 14) using two methods (i.e., using the function `distinct()` or `reduceByKey()`) and comparing the execution time. Check the links for help on how to measure elapsed time in Python and in Scala.
7. Display the 10 most popular departure places.
8. Display the 10 roads (defined by a pair `VoyageFrom` and `VoyageTo`, columns 14 and 15 respectively) the most often taken. Here you can start by implementing a version where a pair `VoyageFrom`-`VoyageTo` A-B and a pair B-A correspond to different roads. Then, implement a second version where A-B and B-A are considered as the same road.
9. Compute the hottest month (defined by column `Month`, 41) on average over the years considering all temperatures (column `ProbTair`, 117) reported in the data set.
10. Re-execute questions 1 to 4 above, measuring the execution time of each command. Explain the differences between them.

Modifié le: Friday 18 November 2022, 12:07