Lab2: Spark RDD

Step1:

1. **Enter the master container**:

   `docker exec -it hadoop-master bash`

2. **Navigate to Spark's configuration directory**:

   `cd /usr/local/spark/conf`

3. **Create or edit the `slaves` file**:

   `vim slaves`

4. Add the following:

   `hadoop-slave1`
   `hadoop-slave2`

   → don't miss too save the vim before quitting!, write `:wq`

5. after that go to : `cd /usr/local/spark/sbin/`

6. and run: `./start-all.sh`

Step2: Enable Python Support for Spark

1. Navigate to the config directory: `cd /usr/local/spark/conf`
2. Copy the template file: `cp spark-env.sh.template spark-env.sh`
3. Edit `spark-env.sh` : `vim spark-env.sh`
4. inter `i` to have access to write , go to the end of file and add this line:
   `PYSPARK_PYTHON=/usr/bin/python3`

Step3: (without hdfs)

1. We will do the execution locally:

```
from pyspark import SparkConf, SparkContext
sc = SparkContext("spark://hadoop-master:7077", "count_lines")
rdd = sc.textFile("file:///root/arbres.csv")
```

```
nbr = rdd.count()
print("nbr of lines: ",nbr)
sc.stop()
```

Note!

in this line `sc = SparkContext("spark://hadoop-master:7077", "count_lines")` if your bash is written in this way for example: root@b524f35852c2 , try to change the code to `sc = SparkContext("spark://b524f35852c2:7077", "count_lines")` .

2. Copy arbre.csv to both master and slaves:

```
docker cp arbres.csv hadoop-master:/root
docker cp arbres.csv hadoop-slave1:/root
docker cp arbres.csv hadoop-slave2:/root
```

3. Copy the code to the container:

```
docker cp count_lines.py hadoop-master:/root
docker cp numberlines.py hadoop-master:/root
docker cp averageheight.py hadoop-master:/root
docker cp tallesttree.py hadoop-master:/root
docker cp nbtrgenus.py hadoop-master:/root
```

4. Then we run this command to get the result:

```
spark-submit --master spark://b524f35852c2:7077 count_lines.py > ou
tput_q1.txt
                                ^
                    #change this depends on your bash name
```

5. To see the result write:

```
cat output_q1.txt
```

Step 4: (with hdfs)

- We do the same steps as precedent :

1. The code:

```
from pyspark import SparkConf, SparkContext
sc = SparkContext("spark://b524f35852c2:7077", "count_lines")
#rdd = sc.textFile("file:///root/arbres.csv")
rdd = sc.textFile("hdfs://b524f35852c2:9000/input/arbres.csv")
nbr = rdd.count()
print("nbr of lines: ",nbr)
sc.stop()
```

2. Copy arbre.csv to hdfs:

```
docker cp arbres.csv hadoop-master:/root
hadoop fs -put arbres.csv input
```

3. and 4. and 5. are as precedent.

NOTES:

- You may have problems when you try to put arbre.csv to hdfs, so test with this commend `jps` if you did not see `NameNode` in the list, restart Hadoop:

```
start-dfs.sh
```

`docker exec -it sp-master bash`