✔ Terminé

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

Start the pyspark shell

(if you are working in Python, see the first Spark TD for instructions in Scala)

After logging in the LSD machine, to open the pyspark shell, run (`kinit` will ask for your password):

```
kinit
use_spark3
pyspark --master yarn
```

To help you in this lab session, don’t hesitate to investigate the PySpark SQL documentation and the documentation for the functions

The Intrepid data set

In the HDFS cluster, you will find the `/user/fzanonboito/CISD/ANL-Intrepid.csv` file, obtained from the ANL Intrepid Log. It contains information about 68,936 jobs submitted to the Intrepid supercomputer (Argonne National Laboratory, USA) over 240 days of 2009. At the time, the machine of 40,960 quad-core nodes was among the world's 10 fastest supercomputers. About each job, the dataset contains (among other things) its submission time (when a user has asked the system to run it), number of requested nodes, amount of requested time, and its actual execution time.

Start by reading the csv file into a dataframe and looking at its structure with show() and printSchema(). Then, use dataframe operations to answer the following questions. **Notice that the columns will be of type string, that means you may have to do some conversions. You may also use the inferSchema option to have Spark infer the schema of the table (but it may need to read the data more than once for that, so be careful when using large data sets).**

1. Before starting, verify if any columns contain lines with unknown values (in this dataset, those are represented by -1). If there are, filter them out. Hint: `df.columns` gives the list of the names of the columns of dataframe `df`.

2. Make a dataframe that only contains jobs of user 7. How many there are? What is the average number of processors this node requested ? How does this average compare to the global average (for all users) ?

3. Prepare a dataframe containing, to each job id, the number of core-hours it used (knowing the provided run time is in seconds and that nb_proc is the number of used processing cores).

4. Show statistics of core-hours usage (min, max, average).

5. The Cobalt scheduler, used in Intrepid, sometimes allocated to jobs more processors than requested by the user. Did it ever give less resources than requested?

6. Obtain, to each queue, its number of jobs, total resource usage and maximum wait time. Order this dataframe by queue number in descending order (from the highest to the shortest).

7. Write the dataframe of the previous question to a new csv file.

8. Investigate the Pearson correlation between these columns (which you can obtain with the `corr()` function, passing two column names as arguments). But do not be tempted to take too many conclusions from this!

Modifié le: Friday 2 December 2022, 10:41