HOMEWORK 3

Outline of the homework

YOU MUST

1. Write a Spark program implementing a 2-round coreset-based MapReduce algorithm for k-center with z outliers:
   - **Round 1:** compute a weighted coreset by extracting $k+z+1$ points from each of L partitions using Farthest-First Traversal. Each point is assigned, as weight, the number of points closest to it, from the same partition.
   - **Round 2:** compute the final solution by running the sequential algorithm developed for HW2 (`SeqWeightedOutliers`), on the union of the weighted coresets, whose size is $(k+z+1)*L$.
2. Run the program on the CloudVeneto cluster, testing accuracy and scalability.

GOAL: you should (hopefully) observe a substantial gain in efficiency w.r.t. running the sequential algorithm on the entire dataset, at the cost of a moderate loss of accuracy.

Specific tasks

1. Download the template (Java or Python)
2. Complete the template adding the missing code:
    - Insert `SeqWeightedOutliers` from HW2;
    - Complete Round 2 of the MapReduce algorithm to extract the final solution by running `SeqWeightedOutliers` on the weighted coreset using $\alpha = 2$
    - Measure and print separately the times required by Round 1 and Round 2;
    - Add the code to compute the value of the objective function.
3. Test and debug your code on your local PC (or virtual machine)
4. Run your program on the cluster testing various datasets and configurations of the parameters, and report the results on a table provided as a word file.

Time measurements in Spark

You can use standard methods **but:**

BE AWARE OF THE LAZY EVALUATION

- If the region of code to be measured includes a transformation of an RDD, make sure that an action (e.g., the invocation of methods such as `count` or `collect`) is executed on the resulting RDD before stopping the time measurement. The time for the action will be included in the measurement, but it will hopefully be negligible.
- If you do not want to include previous transformations to an RDD in your measurement, make sure that after these transformations you invoked an action and cached the resulting RDD.

Refer to the section of Introduction to Programming in Spark, dedicated to time measurements.

Use of the cluster on CloudVeneto

(Refer to User guide for the cluster on CloudVeneto in Moodle Exam)

Main points:

- Cluster of 10 machines, each equipped with 8 cores and 16 GB of RAM.
- One of the machines (147.162.226.106) acts as frontend.
- Each group has an account on the frontend: groupXXX where XXX is the 3-digit group number. The initial password, *to be changed upon the first access*, is:
- The frontend can be accessed through remote login using the ssh protocol but only from machines on the unipd network. When you work from home, you must first connect to a machine on the unipd network (e.g., `login.dei.unipd.it` for students @DEI).

How to run a program on the cluster

(Refer to User guide for the cluster on CloudVeneto in Moodle Exam)

- (For Java users only) Create a jar file
- Transfer your program to the home of your group's account on the frontend using the scp protocol.
- Execution of your **GxxxHW3.java** program:

  ```
  spark-submit –num-executors X –class GxxxHW3 BDC-all.jar argument-list
  ```

  where X is the number of workers that you want, and argument-list is the list of CLI arguments (e.g., filename k z L)
- Execution of your **GxxxHW3.py** program:

  ```
  spark-submit –num-executors X GxxxHW3.py argument-list
  ```

  where X and argument-list are as above.

RULES

To ensure a fair use of the cluster to all (many) groups,

STRICTLY FOLLOW THESE RULES

- Groups with even (resp., odd) group number must use the clusters in even (resp., odd) days.
- Do not run several instances of your program at once.
- Do not use more than 16 executors.
- Try your program on a smaller dataset first. If your program is stuck for more than 1 hour, its execution will be automatically stopped by the system.

Downtime

Two days ago CloudVeneto's system administrators have informed us that the infrastructure will be unavailable, for unplanned maintenance

from May 20 at 9.30am until the morning of May 24

We will send you a reminder on May 19 and will inform you as soon as maintenance is over.