Deadline:

June 6, 2019

Submission Instructions:

Submit report as described below to ahmeda@cs.umu.se
Your email should explain what role did each team member play in the project

Assignment Instructions:

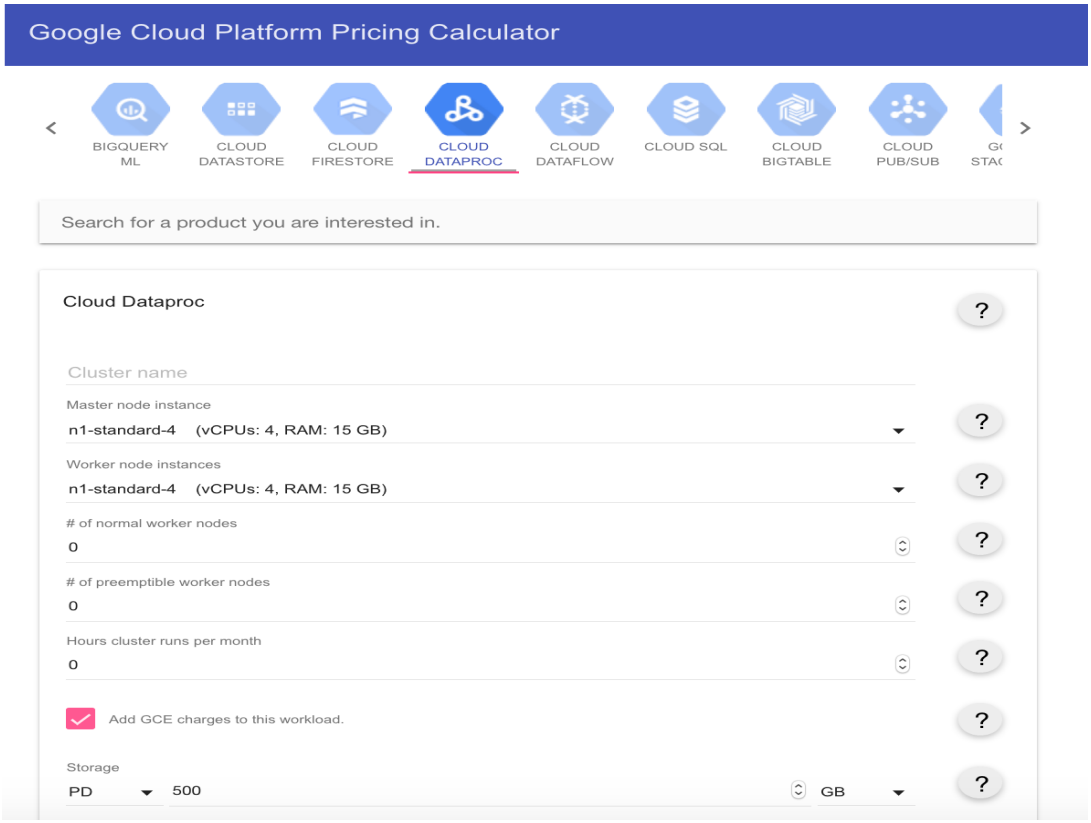
The assignment aims to introduce you to using the Cloud to run apache Spark. We will be using the Google Cloud (GCE) platform. You have $300 worth of free usage when you register, which should be sufficient to try many of the Google Cloud offerings and finish your assignments. Please start by creating an account here. You need a credit card to do so, but it is completely free for a year. The system is safe and reliable. **Make sure that you do not cross the $300 limit in your project.**

Now once you have an account, please familiarize yourselves with the system. You will be using the Cloud Native Apache Spark offered by GCE. Read the documentation here [1]. Also, make sure you go through the Apache Spark tutorial as you need to [2].

The project is to be done in teams of at most 3 people. If you want to work on your own, fine, but I would highly suggest that you team up with someone that would complement your expertise. You will form your own teams. Talk to your peers. Ideally, people in the same team would be able to meet physically, but not necessarily.

You need to carefully design your experiments to not consume all your free tier before you are done. Try testing on a local machine first to know if your code works.

Estimate your usage:
To calculate the pricing of your run use the provided Google Calculator:
https://cloud.google.com/products/calculator/
You want to choose the DataProc product and to include the GCE charges in your cost estimations



Preemptable machines:
Read on Google's Preemptible instance here [3]. They are extremely cheap compared to normal instances and are well suited for the project. It is your decision what you use for the project. This is more or less just a suggestion.

Assignment:

Please choose only one of the following two projects to work on based on the competence in your team. The second project will involve slightly more SW development if you do not choose a ready made implementation.

Project 1
As the foremost research team in data science, a guy called Ahmed came to your team and asked you to evaluate for him if he should use Spark on the Google Cloud for improving the performance of his matrix computations instead of just running his computations on his local machine. He said you can use any programming language of your choice to evaluate for him a

Spark library called LINALG [4], [5]. As data scientists, you asked Ahmed for test data. He pointed you to the following dataset of Matrices that you can use for your testing [6] where you will find multiple matrices in different formats from many applications, including, control and optimization, networking, and privacy data.

To evaluate the library, you need to choose two of the available methods in the library, and one of the matrices available in the datasets. You will then evaluate the speedup when using Spark with a low number of nodes, versus a large number of nodes for each of the methods. Choose the matrix wisely. A very small matrix will show you nothing. A very large one will take forever to operate on (and consume all $300).

You will write a report where you will explain your choice of the methods, and the dataset. You will describe your setup, and your conclusion. As scientists, you should be able to describe why you have reached that conclusion, e.g., by showing performance speedup graphs or slowdowns. Please comment on the behaviour you see, if it is linear in the amount of resources, sublinear, or something else, or if the CPU was the bottleneck, or the Memory, etc?

Please follow the data science process, and comment on how you have followed it.

Project 2

As the foremost research team in data science, a guy called Ahmed came to your team and asked you to evaluate for him if he should use Spark on the Google Cloud for improving the performance of his link prediction algorithms instead of just running computations on his local machine. Link predictions is a fundamental problem in complex networks science, used to predict malware spread in networks, disease spread, social network connections, among many other things [7].
Ahmed said you can use any programming language of your choice to evaluate link prediction on top of Spark. You can look at projects such as Sparkling-Graph[8]. As data scientists, you asked Ahmed for test data. He pointed you to the following dataset of networks that you can use for your testing [9] where you will find multiple datasets from many large scale networks.

To evaluate link prediction, you need to implement the algorithm yourself or use one of the existing source codes on github, and apply it to one of the network datasets in the repository. Your choice!
Please be aware that there is a lot of rubbish on github, and sometimes it is much more efficient to do your own thing, or search the internet for people who are using what you plan to use. You will then evaluate the speedup when using Spark with a low number of nodes, versus a large number of nodes.
You will write a report where you will explain your choice of the methods, and the dataset. You will describe your setup, and your conclusion. As scientists, you should be able to describe why you have reached that conclusion, e.g., by showing performance speedup graphs or

slowdowns. Please comment on the behaviour you see, if it is linear in the amount of resources, sublinear, if the CPU was the bottleneck, or the Memory, etc?

Please follow the data science process, and comment on how you have followed it.

For both project:
In your report, include a link to your code on github.
Include a section where you comment on what you have learned, what did you think about the assignment, if you feel it is too easy, too hard, or something else. Please comment on what you liked an what you did not like.

[1] https://cloud.google.com/dataproc/
[2] https://data-flair.training/blogs/spark-tutorial/
[3] https://cloud.google.com/compute/docs/instances/preemptible
[4]https://spark.apache.org/docs/1.5.1/api/java/org/apache/spark/mllib/linalg/package-frame.html
[5] https://shivaram.org/publications/matrix-spark-kdd.pdf
[6] https://sparse.tamu.edu/
[7]http://www.leonidzhukov.net/hse/2016/networks/papers/NowellKleinberg07linkprediction.pdf
[8] https://sparkling-graph.github.io/
[9] https://snap.stanford.edu/data/index.html