CERN-HSF GSoC 2017

Big Data Tools For Physics Analysis

Exercise for Candidate Students

This document describes an exercise that will be used to evaluate those students interested in applying for the "Big Data Tools for Physics Analysis" project, included in the Google Summer of Code (GSoC) program and offered by the EP-SFT group and the IT department at CERN. The detailed description of the project can be found here.

The exercise is divided into two tasks. In order to complete the first one, the student will need to know and combine a set of technologies that are important for the project, namely Python, Spark, JavaScript and Jupyter notebooks. The second task, which is a bonus test, brings also ROOT into the equation.

Please follow the guidelines below to go through the exercise and work at a pace that suits you. We recommend that you start with the first task and, if you finish it and have some more time, you go for the second one. Do not worry if you cannot complete both and do not hesitate to ask us, the mentors, any question you might have.

TASK 1: Display live information about a Spark job in a notebook

The objective of this task is to write a Python function that spawns a Spark job and monitors its execution. In addition, the function should display some kind of information about the job in a Jupyter notebook via JavaScript graphics.

In preparation for this task you will need to:

- Install Spark on your machine and run some simple test locally.
- Install Jupyter on your machine and launch a local notebook server.
- Understand how to embed graphics in a Jupyter notebook, in particular JavaScript graphics.
- Understand how to query the Spark REST API and the kind of information it provides.

Your Python function needs to execute a Spark application and monitor it. It does not really matter what application you run, since the interesting part is the monitoring. As an example, you could write a Python function that calculates, using Spark, the sum of a list of numbers, which could have the following signature:

```
def calculate_sum(my_list):

   # Body of the function here

   return sum
```

The body of your function should launch the Spark job and, while it is running, request some information to the Spark REST API (for instance, completed and pending tasks). This information should be then converted into a JavaScript display that is automatically refreshed as the Spark job executes (for example, a progress bar).

You can assume that your function will be always invoked from a cell of a Python notebook. Therefore, the JavaScript display needs to be shown in the output of that cell, in the notebook itself.

The deliverables of this task are:

1) The Python code of the monitoring function and the JavaScript code that produces the graphics.
2) A Python Jupyter notebook that invokes your function and shows the JavaScript graphics you created.

In this test, we will evaluate your knowledge of each individual technology and your ability to combine them. On the other hand, creativity is always a plus! Feel free to work on a more sophisticated JavaScript display if you have an original idea!

OPTIONAL *- TASK 2: Execute a ROOT-Spark notebook*

Being able to finish task 1 of this exercise already demonstrates that you master most of the technologies to be used in the project. However, if you want to go the extra mile, we propose you a second task that adds the ROOT data analysis framework to the picture.

In preparation for this task you will need to:

- Install Spark on your machine (you should have done this already in task 1).
- Install ROOT on your machine.
- Download the DistROOT Python module from here.
- Download the DistROOT example notebook from here.

The objective of this task is simply to execute the aforementioned DistROOT notebook. By doing so, you will demonstrate that you have been able to combine ROOT and Spark via the DistROOT module, which is fundamental for the project.

The deliverable of this task is the DistROOT notebook after being executed and saved, that is, the notebook containing the code cells plus the generated output cells.

Once you complete any of the tasks of this exercise, please send us by e-mail the requested deliverables at:

etejedor@cern.ch, danilo.piparo@cern.ch, prasanth.kothuri@cern.ch, kacper.surdy@cern.ch