Big Data Technologies and Physics Analysis with Apache Spark

Exercise 0: Get to know HDFS and the Hadoop Service

The client machines of the CERN Hadoop and Spark Service are accessible with the alias "it-hadoop-client". Login with your CERN account to the client machines with the following command:

```
$ssh <student_username>@it-hadoop-client
```

There are 3 different clusters that you can access within the Hadoop Service.

These are:

- analytix
- hadalytic
- lxhadoop

The default cluster is analytix. You can change clusters with the following command:

```
. hadoop-setconf.sh <cluster_name>
```

Do not change the cluster yet. Inspect the HDFS structure of the analytix cluster with the following command:

```
$hdfs dfs –ls /
```

Then, check your own home folder in HDFS with the following command:

```
$hdfs dfs –ls /user/<student_username>
```

This folder should be empty. Let's put the first file in our home folder.

```
$touch testfile.txt
$hdfs dfs –put testfile.txt /user/<student_username>
```

Then, retry:

```
$hdfs dfs –ls /user/<student_username>
```

You should now see testfile.txt inside your home folder. You can delete the file with:

```
$hdfs dfs –rm /user/<student_username>/testfile.txt
```

As we can see, we interact with HDFS in the same way that we interact with our local filesystem. You can see more HDFS commands in the official HDFS documentation:

https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-common/FileSystemShell.html

Exercise 1: Running a Physics Analysis workload directly on the Hadoop and Spark Service of CERN

Important Note: For Step 1, you will need SBT and JDK installed.

If you do not already have JDK installed, you can find the instructions here:

(Linux) https://docs.oracle.com/javase/8/docs/technotes/guides/install/linux_jdk.html#BJFGGEFG

(macOS) https://docs.oracle.com/javase/8/docs/technotes/guides/install/mac_jdk.html#CHDBADCG

The download page can be found in the following link:

https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

To install SBT, please follow the instructions:

(Linux) https://www.scala-sbt.org/1.0/docs/Installing-sbt-on-Linux.html

(macOS) https://www.scala-sbt.org/1.0/docs/Installing-sbt-on-Mac.html

It is strongly advised to follow this step in order to compile the project on your own. However, if you want to skip it, you can find the output JAR file in the following link:

http://cern.ch/go/bQq8

Step 0: Inspecting the repository

Navigate to

https://github.com/olivito/spark-root-applications

You can find multiple examples of physics analysis in this repository.

For this exercise, we will run the "DimuonReductionAOD" example.

Navigate to:

https://github.com/olivito/spark-root-applications/blob/master/src/main/scala/org/dianahep/sparkrootapplications/examples/DimuonReductionAOD.scala

Inspect the code.

The code is doing one specific operation on the input files. Can you detect which one it is?

Answer:

Step 1: Compiling the Project

On your laptops, create a folder "PhysicsOnAnalytix" (e.g. in your Desktop).

```
$mkdir PhysicsOnAnalytix
$cd PhysicsOnAnalytix
```
We then need to clone the “spark-root-applications” repository with the following command:

```
$git clone https://github.com/olivito/spark-root-applications.git
```
Then we need to navigate inside the folder:

```
$cd spark-root-applications
```
and then compile it with the following command:

```
$sbt package
```
This can take several seconds or minutes.

The result of the above command will be a jar file, which will be located inside the folder “spark-root-applications/target/scala-2.11/” with the name “spark-root-applications_2.11-0.0.11.jar”.

We must navigate in that folder and then move this jar file to our home AFS directory in order to run it automatically on the analytix cluster.

```
$cd /target/scala-2.11/ $scp spark-root-applications_2.11-0.0.2.jar
<student_username>@lxplus.cern.ch:
```
In case there are any issues and the compilation fails, the jar is available here:

http://cern.ch/go/bQq8

Please download it and place it in your AFS home directory with the above command.

Step 2: Running the Job on Analytix through the it-hadoop-client

Then, we need to connect via SSH to the it-hadoop-client machines and submit the JAR file for execution on the analytix cluster.

Connect, once again, to the it-hadoop-client machines with:

```
$ssh <student_username>@it-hadoop-client
```

Then, we will use the “spark-submit” command in order to submit our JAR file for execution to the analytix cluster.

The command is the following:

```
spark-submit \
--master yarn \
--class org.dianahep.sparkrootapplications.examples.DimuonReductionAOD \
--packages org.diana-hep:spark-root_2.11:0.1.15,org.diana-hep:histogrammar-
sparksql_2.11:1.0.3 \
spark-root-applications_2.11-0.0.11.jar
root://eospublic.cern.ch/eos/opendata/cms/Run2012B/SingleMu/AOD/22Jan2013-
v1/20000/FEFB25CE-6171-E211-BDDB-001EC9D27648.root
hdfs:/user/<student_username>/myoutputfolder/ ZZTo4mu
```

Please note that the last 3 arguments are “required” by the JAR file as the syntax is the following:

```
spark-submit [options]
spark-root-applications_2.11-0.0.8.jar \
<input_directory> \
<output_directory> \
<output_label>
```

You can browse:

http://namenode-analytix.cern.ch:8088

in order to track the progress of your application.

You can also check the output of this workload on your output path:

```
/user/<student_username>/myoutputfolder/
```

Which HDFS command will you use to check this output?

Answer:

Please note that the above "spark-submit" command will only execute the jar on the file

"FEFB25CE-6171-E211-BDDB-001EC9D27648.root"

which is located inside the directory

"eos/opendata/cms/Run2012B/SingleMu/AOD/22Jan2013-v1/20000//eos/opendata/cms/Run2010B/MuOnia/AOD/Apr21ReReco-v1/0000/"

within the EOS Storage Service.

We can also give any other root file located inside the same or a different directory, on HDFS or on EOS.

We can even give the whole directory as an argument, which will run the jar on all the files of this directory.

In that case, the <input> argument would be the following:

```
root://eospublic.cern.ch/eos/opendata/cms/Run2012B/SingleMu/AOD/22Jan2013-v1/
```

***[Important note**: The above folder is 22 TB. This job has been executed on 1 PB of data and finished smoothly in 3.8 hours. However, please do not execute the job over the full folder right now for obvious reasons! ☺]*

Exercise 2: Running a Machine Learning Workload with Apache Spark via the SWAN Service.

For this exercise, we will use the SWAN Service.

Please navigate to the SWAN Gallery:

https://swan.web.cern.ch/content/basic-examples

Then click on “Apache Spark”

and then select “Machine Learning with Apache Spark” and proceed with the configuration listed below.

You can inspect the notebook in the following link:

https://nbviewer.jupyter.org/github/prasanthkothuri/swan-spark-notebooks/blob/master/ML_Spark_MLlib.ipynb

Notes:

This notebook demonstrates ML using Apache Spark MLlib and has been developed in the context on Hadoop and Spark service at CERN.

To run this notebook we need the following **configuration**:

- *Software stack*: Developing bleeding edge (94) (it has spark 2.3.1)
- *Platform*: centos7-gcct
- *Number of cores*: 4
- *Memory*: 10G
- *Spark cluster*: Analytix

The source code can be also found in github under the following repository:

https://github.com/cerndb/SparkML

Please do not hesitate to request assistance if you face any issues. ☺