Spark – Hands-On

Objetivo: O objetivo deste hands-on é instalar e testar o Spark.

1. Edite o arquivo Vagrantfile com a descrição da máquina virtual e inclua uma nova operação para repasse de portas dentro do bloco principal (abaixo dos repasses de porta do HDFS e YARN)

```
  # Spark Application Web UI
  config.vm.network "forwarded_port", guest: 4040, host: 4040, host_ip: "127.0.0.1",
auto_correct: true
```

2. Reinicie a máquina virtual

```
$ vagrant reload
```

3. Entre na máquina virtual e instale o Spark versão 2.3.1.

```
$ vagrant ssh
$ cd /vagrant
$    wget    -c    http://ftp.unicamp.br/pub/apache/spark/spark-2.3.1/spark-2.3.1-bin-
hadoop2.7.tgz
$ tar -zxvf spark-2.3.1-bin-hadoop2.7.tgz
$ export SPARK_HOME=/vagrant/spark-2.3.1-bin-hadoop2.7
$ export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
$ export JAVA_HOME= /usr/lib/jvm/java-8-openjdk-amd64
```

4. Abra o PySpark

```
$ $SPARK_HOME/bin/pyspark
```

5. Acesse na máquina local a página http://localhost:4040 para ver a Spark Application UI.

6. Execute o programa abaixo (WordCount) dentro do pyspark (OBS. Confirme que o arquivo shakespeare.txt usado no exercício de MapReduce está armazenado na pasta /user/vagrant/shakespeare do HDFS)

```
text_file = sc.textFile("hdfs:///user/vagrant/shakespeare/shakespeare.txt")
counts = text_file.flatMap(lambda line: line.split(" ")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs:///user/vagrant/shakespeare_result")
counts.collect()
```

7. Verifique na Spark Application UI (http://localhost:4040) os detalhes sobre a execução da aplicação

8. Saia do PySpark

```
exit()
```

9. Execute o exemplo Pi através do comando spark-submit

```
$SPARK_HOME/bin/spark-submit --class org.apache.spark.examples.SparkPi --master local \
$SPARK_HOME /examples/jars/spark-examples_2.11-2.3.1.jar 1
```