ILLINOIS DATA SCIENCE INITIATIVE

TECHNICAL REPORTS

Parsing Tweets using PySpark

Author:
Sameet Sapra & Joshua Chang

February 24, 2017

Parsing Tweets using PySpark

SAMEET SAPRA[1] AND JOSHUA CHANG[2]

[1] *National Center For Supercomputing Applications (NCSA)*
[2] *Laboratory for Computation, Data, and Machine Learning*
[3] *Illinois Data Science Initiative*

Compiled February 24, 2017

An introduction to using Apache Spark with Python by parsing tweets.

https://github.com/lcdm-uiuc

1. SPARK INTRODUCTION: WHY PYSPARK?

PySpark is an API that interfaces with RDD's in Python. It is built on top of Spark's Java API and exposes the Spark programming model to Python. PySpark makes use of a library called Py4J, which enables Python programs to dynamically access Java objects in a Java Virtual Machine. This allows data to be processed in Python and cached in the JVM.

2. PREREQUISITES TO PYSPARK

Assume that the environment is a cluster with Hadoop installed. Verify that Java and Scala are installed by checking the versions. If they are installed, skip to the Spark configuration steps, otherwise follow the installation instructions given below.

3. DOWNLOAD AND INSTALL

Install Java on the system with:

```
> sudo yum install java-1.7.0-openjdk-devel
```

Install Spark on the system with:

```
> wget http://downloads.lightbend.com/scala/2.11.8/scala-2.11.8.rpm
> sudo yum install scala-2.11.8.rpm
```

Let's configure Spark to only show errors on startup, rather than all info. This will make the Spark shell less cluttered when it is opened on startup.

```
> cd /usr/hdp/2.3.6.0-3796/spark/conf
> vi log4j.properties
```

Find a line that describe the rootCategory of log:

```
log4j.rootCategory=INFO, console
```

Replace INFO with ERROR:

```
log4j.rootCategory=ERROR, console
```

4. SETTING UP PYSPARK

Now let's ensure that python 2.7 is installed and configured.

```
> yum install -y centos-release-SCL
> yum install -y python27
> yum -y install python-pip
```

Finally, let's set up the environment variables for Spark and Python.

```
> vi ~/.bashrc
> export SPARK_HOME=/usr/hdp/2.3.6.0-3796/spark
> export PYTHONPATH=$SPARK_HOME/python
> export SPARK_HIVE=true
```

Then, reload the environment:

```
> source ~/.bashrc
```

5. WRITING OUR FIRST LINES OF PYSPARK

This example is taken from the Apache Spark website. It runs a parallelized operation to compute the value of pi.

```
def inside(p):
    x, y = random.random(), random.random()
    return x*x + y*y < 1

count = sc.parallelize(xrange(0, NUM_SAMPLES)) \
            .filter(inside).count()
print "Pi␣is␣roughly␣%f" % (4.0 * count / NUM_SAMPLES)
```

Once you've written your PySpark code, you can compile and deploy it with:

```
> spark-submit --master yarn-cluster MY_PYTHON_FILE.py
```

The *'–num-executors 10'* flag (arbitrary number) may be added to specify how many executors, the objects responsible for executing tasks, are to be used. Using as many executors as data nodes is recommended to decrease minimize runtime.