UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Setting-up of a Spark cluster

Petar Jovanovic and Sergi Nadal

November 18th, 2016

In this document, we are going to set up a Spark cluster. Specifically, we are going to install a Spark Standalone cluster in its version 2.0.1, this entails that no external scheduler system will be used, e.g., YARN or Mesos.

First of all you will need to obtain Spark.

```
wget -P tarballs/ http://d3kbcqa49mib13.cloudfront.net/spark-2.0.1-
    bin-without-hadoop.tgz
mv tarballs/spark-2.0.1-bin-without-hadoop.tgz ~/spark-2.0.1.tgz
tar -xf spark-2.0.1.tgz
mv ~/spark-2.0.1-bin-without-hadoop ~/spark-2.0.1
rm ~/spark-2.0.1.tgz
```

1 Installing Java 8

When developing in Spark we will use some specific functionalities provided by Java 8. To this end, as previously we installed Java 7, we need to use a parallel installation of Java 8. You can do it by following these steps:

```
wget http://bit.ly/2fGOOl7 -O ~/tarballs/java8.tar.gz
cp ~/tarballs/java8.tar.gz ~/.
tar -xzvf java8.tar.gz
rm java8.tar.gz
```

2 Configure Spark

As we did in the previous session, this setup must be done only in the master node and then it will be replicated to other nodes. The configuration of Spark is simpler than that of Hadoop and HBase. Here you only need to, first create the *slaves* file.

```
touch ~/spark-2.0.1/conf/slaves
```

And fill in the following content:

```
slave1
slave2
```

Similarly, create a file *spark-env.sh*.

```
touch ~/spark-2.0.1/conf/spark-env.sh
```

With the following content:

```
export JAVA_HOME="/home/bdma**/jre1.8.0_111"
export SPARK_DIST_CLASSPATH=$(/home/bdma**/hadoop-2.5.1/bin/hadoop
    --config /home/bdma**/hadoop-2.5.1/etc/hadoop/ classpath)
```

The first *export* locates the Java folder (in case you are using Spark without running HDFS or HBase), while the second makes Spark inherit the classpath from that defined in Hadoop's configuration.

That's all the required configuration, as we are mostly using default values. However, if some specific parameters were required for your system, you'd need to specify them in the file *\/spark-2.0.1/conf/spark-env.sh*. You can check all possible parametrizations in the file *spark-env.sh.template*.

3 Replicate the configuration

Finally, as for the case of Hadoop and HBase, we need to replicate Spark to the slave nodes. Thus, run the following commands:

```
scp -r ~/jre1.8.0_111/ ~/spark-2.0.1/ bdma**@slave1:.
scp -r ~/jre1.8.0_111/ ~/spark-2.0.1/ bdma**@slave2:.
```

4 Starting up the cluster

Now the cluster should be perfectly configured and, hence, it should be ready to run. In order to start the cluster, in the master node, run:

```
~/spark-2.0.1/sbin/start-all.sh
```

After running those command, the Spark web UI should be available at *http://MASTER:8080/*(check your email for port redirection).

5 Stopping the cluster

To stop the Spark cluster, you can run:

```
~/spark-2.0.1/sbin/stop-all.sh
```