Spark: Setting up a Single Node Cluster

Qi Song 08/25/2015

This instruction is derived from Spark Standalone Mode. If you meet any errors during installation or execution, please try to google them and solve them!

Prerequisites:

Supported Platforms

Windows is also a supported platform but the followings steps are for Linux (Ubuntu 14.04 – 64bit) only.

Software install:

For Java and SSH, please see document "Hadoop howto". Other related software:

```
$ sudo apt-get install maven
$ sudo apt-get install scala
```

Hadoop version

Spark can build upon Yarn(Hadoop 2.0), please see document "Hadoop howto" and install single node hadoop first.

Yarn on a single node:

Also in hadoop-2.7.1 folder

1. Configure parameters as follows: *etc/hadoop/mapred-site.xml:*

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

etc/hadoop/yarn-site.xml:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

2. Start ResourceManager daemon and NodeManager daemon:

```
$ sbin/start-yarn.sh
```

Apache Spark on a single node

1. Download spark from Download Spark, here we use version 1.4.1.
2. Compile

```
$ export MAVEN_OPTS="-Xmx1300M -XX:MaxPermSize=512M -XX:ReservedCodeCacheSize=512m"
$ mvn -Pyarn -Phadoop-2.7 -Dhadoop.version=2.7.1 -DskipTests clean package
```

You should see something like this by the end of the compilation process:

```
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] Spark Project Core ............................... SUCCESS [17.713s]
[INFO] Spark Project Bagel .............................. SUCCESS [15.760s]
[INFO] Spark Project Streaming .......................... SUCCESS [57.492s]
[INFO] Spark Project ML Library ......................... SUCCESS [35.794s]
[INFO] Spark Project Examples ........................... SUCCESS [2:23.954s]
[INFO] Spark Project Tools .............................. SUCCESS [9.144s]
[INFO] Spark Project REPL ............................... SUCCESS [28.911s]
[INFO] Spark Project YARN Support ....................... SUCCESS [46.324s]
[INFO] Spark Project Assembly ........................... SUCCESS [33.443s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6:28.869s
[INFO] Finished at: Fri Oct 11 21:57:32 CEST 2013
[INFO] Final Memory: 31M/289M
[INFO] ------------------------------------------------------------------------
```

3. Start Spark

```
$ cp conf/slaves-template conf/slaves
$ ./sbin/start-master.sh
$ ./sbin/start-slaves.sh
```

We can use *conf/spark-env.sh* to set up some configurations.

4. Submit applications

See Spark Quick Start, here we use python as an example. Create a simple Spark application, SimpleApp.py:

```
"""SimpleApp.py"""
from pyspark import SparkContext

logFile = "/home/qsong/spark-1.4.1/README.md"  # Should be some file on your system
sc = SparkContext("local", "Simple App")
logData = sc.textFile(logFile).cache()

numAs = logData.filter(lambda s: 'a' in s).count()
numBs = logData.filter(lambda s: 'b' in s).count()

print "Lines with a: %i, lines with b: %i" % (numAs, numBs)
```

Run this application using the bin/spark-submit script:

```
# Use spark-submit to run your application
$ /home/qsong/spark-1.4.1/bin/spark-submit \
  --master local[4] \
  SimpleApp.py
```

We can see some result like this:

```
Lines with a: 60, lines with b: 29
```