Spark Tutorial (Local Machine)

Duke CompSci 516

1 Overview

In this tutorial, you will learn how to install Spark and run a simple Spark application on your local machine. As this tutorial simply concatenates parts of documents provided in Spark's main website (http://spark.apache.org/), please refer to it for more information.

Note that this tutorial is based on Ubuntu, a Linux distribution. Other Linux distros and Mac OS users can follow this tutorial as the procedures are very similar. However, if you are a Windows user, I highly recommend creating a Ubuntu virtual machine using VirtualBox or VMWare.

2 Installation

2.1 Ubuntu

1. Install Java (jdk-8u144).
   - If you already have Java installed on your machine, you can skip this.
   - Download the latest version of Java Development Kit.
   - Follow the command lines below to untar the file.

     ```
     $ mkdir -p ~/application/java
     $ tar -xvzf jdk-8u144-linux-x64.tar.gz -C ~/application/java
     ```

   - Add `JAVA_HOME` environment variable and add java binaries to `PATH`.

     ```
     $ echo 'export JAVA_HOME=$HOME/application/java/jdk1.8.0_144' >> ~/.bashrc
     $ echo 'export PATH=$JAVA_HOME/bin:$PATH' >> ~/.bashrc
     $ source ~/.bashrc
     ```

   - Test whether the Java installation is successful.

     ```
     $ java -version
     ```

2. Install SBT (v-1.0.2).
   - Download the latest version of SBT (http://www.scala-sbt.org/).
   - Follow the command lines below to untar the file.

```
$ mkdir -p ~/application/sbt
$ tar -xvzf sbt-1.0.2.tgz -C ~/application/sbt
$ mv ~/application/sbt/sbt ~/application/sbt/sbt-1.0.2
```

- Add `SBT_HOME` environment variable and add sbt binaries to `PATH`.

```
$ echo 'export SBT_HOME=$HOME/application/sbt/sbt-1.0.2' >> ~/.bashrc
$ echo 'export PATH=$SBT_HOME/bin:$PATH' >> ~/.bashrc
$ source ~/.bashrc
```

- Test whether the sbt installation is successful.

```
$ sbt sbtVersion
```

3. Install Spark (v-1.6.0). Do NOT use Spark 2.* because the script `spark-ec2` is not supported yet.

   - Download spark-1.6.0-bin-hadoop2.6.tgz, which is a prebuilt Spark for Hadoop 2.6 or later (http://spark.apache.org/).
   - Follow the command lines below to untar the file.

```
$ mkdir -p ~/application/spark
$ tar -xvzf spark-1.6.0-bin-hadoop2.6.tgz -C ~/application/spark/
```

   - Add `SPARK_HOME` environment variable and add Spark binaries to `PATH`. (Backslash is the linebreak in the terminal. Just press enter after the backslash.). Note we also need the binaries in the `ec2` folder of the Spark installation.

```
$ echo 'export SPARK_HOME=$HOME/application/spark/spark-1.6.0-bin-hadoop2.6' \
>> ~/.bashrc
$ echo 'export PATH=$SPARK_HOME/bin:$SPARK_HOME/ec2:$PATH' >> ~/.bashrc
$ source ~/.bashrc
```

   - Try running Spark interactive shell.

```
$ spark-shell
```

2.2 Mac OS

Mac OS X users can follow the same procedure above. If you do not have the `~/.bashrc` file, then you will need to create one.

Or you can install all three of them using homebrew, which takes care of the `PATH` for you.

3 Spark Application

3.1 Build

Download the example code of wordcount.zip and unzip it. This is an app to count the number of occurances for each word in a text file. You will find the following directory and file layout:

```
./build.sbt
./project/build.properties
./src/main/scala/WordCount.scala
./input.txt
```

The `WordCount.scala` is the main programming logics. It invokes the APIs of Apache Spark, so you may want to search online for its API usage. It is also recommended to use an IDE like IntelliJ or Eclipse, which enables you to view the source and doc easily.

We will build this project with sbt. This example uses the recommended directory layout for an sbt project. There are two configuration files: `build.properties`, which is currently only used to specify the version of sbt to use; and `build.sbt`, which configures the application with necessary like name, version number, scala version, and dependencies. You can notice that the dependency of `spark-core` is included in `build.sbt`.

Now go the the base directory of this sbt project, where you can see `./build.sbt`, and run:

```
$ sbt package
...
[info] Done packaging.
[success] Total time: ...
```

After packaging is done successfully, you can find the packaged jar file at
`target/scala-2.10/wordcountapp_2.10-1.0.jar`

3.2 Run

Finally, we can run the application using `spark-submit` script inside `spark-1.6.0-bin-hadoop2.6/bin` directory. We can directly call it since it is already added to `PATH`.

```
$ spark-submit \
--class WordCount \
target/scala-2.10/wordcountapp_2.10-1.0.jar
```

The scala program is hardcoded to read in a file named `"input.txt"`. So it will be fine if the command is invoked at the base directory of the project, where you can find `./input.txt`.