This instruction shows how to install spark with Scala and sbt on Ubuntu system. Spark and Scala require at least Java 8. Since we have installed Java 8 when installing Hadoop, this step can be skipped.

1. Install git with following commands:

$ sudo apt-get update

$sudo apt-get install libjansi-java

$sudo apt-get upgrade

$sudo apt-get install git

2. Install Scala and sbt in your system. These can be under any path you want.

$sudo wget www.scala-lang.org/files/archive/scala-2.12.8.deb

$sudo dpkg -i scala-2.12.8.deb

$sudo apt-get install curl

$sudo wget https://dl.bintray.com/sbt/debian/sbt-1.2.8.deb

$sudo dpkg -i sbt-1.2.8.deb

3.Download and install Spark

$sudo wget <https://archive.apache.org/dist/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz>

Install Spark to /usr/local

$ sudo tar -zxvf ./spark-2.4.0-bin-hadoop2.7.tgz -C /usr/local

$ cd /usr/local/

$ sudo mv ./spark-2.4.0-bin-hadoop2.7 ./spark

Set Spark Path:

$vi ~/.bashrc

Add the following command to the file:

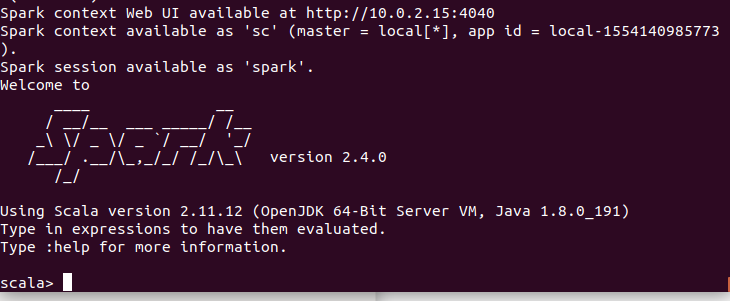
export PATH=/usr/local/spark/bin:$PATH

Save and close the file. Activate the setting immediately with:

$source ~/.bashrc

Run the command to check Spark whether installed successfully:

$spark-shell



When you see these in your terminal, it means you have installed spark successfully.

4.Run example

Reference website: <http://spark.apache.org/docs/latest/quick-start.html>

We will use self-Contained applications:

4.1

We will create a very simple Spark application in Scala-so simple. Create a file and name it as SimpleApp.scala. Copy the following codes to the file SimpleApp.scala.

/\* SimpleApp.scala \*/

import org.apache.spark.SparkContext

import org.apache.spark.SparkContext.\_

import org.apache.spark.SparkConf

object SimpleApp {

def main(args: Array[String]) {

val logFile = "**YOUR\_SPARK\_HOME/**README.md"

val conf = new SparkConf().setAppName("Simple Application")

val sc = new SparkContext(conf)

val logData = sc.textFile(logFile, 2).cache()

val numAs = logData.filter(line => line.contains("a")).count()

val numBs = logData.filter(line => line.contains("b")).count()

println(“Lines with a: $numAs, Lines with b: $NumBs“)

sc.stop()

}

}

This program just counts the number of lines containing ‘a’and the number containing ‘b’ in the Spark README. Note that you’ll need to replace YOUR\_SPARK\_HOME with the location where Spark is installed. In this instruction, logFile =“/usr/local/spark/README.md”.

4.2

We pass the SparkContext constructor a SparkConf object which contains information about our application. Our application depends on the Spark API, so we’ll also include an sbt configuration file simple.sbt (create a file and name it as simple.sbt. Copy the following into simple.sbt), which explains that Spark is a dependency. This file also adds a repository that Spark depends on. Please make sure that you set scalaVersion correctly.

name := "Simple Project"

version := "1.0"

scalaVersion := "2.11.12"

libraryDependencies += "org.apache.spark" %% "spark-core" % "2.4.0"

4.3

For sbt to work correctly, we’ll need to layout SimpleApp.scala and simple.sbt according to the typical directory structure. Once that is in place, we can create a JAR package containing the application’s code, then use the spark-submit script to run our program.

# set up a new directory (say, spark\_test) and the directory layout of spark\_test should look like this

$ find .

.

./simple.sbt

./src

./src/main

./src/main/scala

./src/main/scala/SimpleApp.scala

#The following steps are carried out in spark\_test folder

# Package a jar containing your application and add sbt path to your bashrc

$ sbt package

...

[info] Packaging {..}/{..}/target/scala-2.11/simple-project\_2.11- 1.0.jar

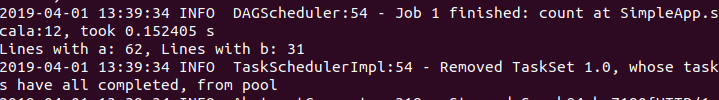
# Use spark-submit to run your application

$ /usr/local/spark/bin/spark-submit \

--class "SimpleApp" \

--master local[4] \

target/scala-2.11/simple-project\_2.11-1.0.jar



...

Lines with a: 62, Lines with b: 31