Egyptian Big Data Geeks 3rd Event

Installing Spark and Scala

Mostafa Alaa Mohamed
mustafaalaa.mohamed@gmail.com

December 18, 2016

Abstract

This Document introduce How to install spark into local Machine. This document doesn't have any copy-writes and I collected and copied from Internet content to be easy for who need to install spark into one document. This document generated by LaTeX and code will be found into the following link

1 Installing Apache Spark and Scala

1.1 MacOS

- Install Apache Spark using Homebrew.
  - Install Homebrew if you donâĂŹt have it already by entering this from a terminal prompt: /usr/bin/ruby -e "$(curl fsSL https://raw.githubusercontent.com/Homebre
  - Enter brew install apache-spark
  - Create a log4j.properties file via cd /usr/local/Cellar/apache-spark/2.0.0/libexec/conf cp log4j.properties.template log4j.properties (substituted 2.0.0 for the version actually installed)
  - Edit the log4j.properties file and change the log level from INFO to ERROR on log4j.rootCategory.
- Install the Scala IDE from http://scala-ide.org/download/sdk.html
- Test it out!
  - ./bin/spark-shell
  - val textFile = sc.textFile("README.md") // textFile: org.apache.spark.rdd.RDD[String] = README.md MapPartitionsRDD[1] at textFile at <console>:25
  - textFile.count() // Number of items in this RDD
  - res0: Long = 126
  - You should show a count of the number of lines in that file.
  - Hit control-D to exit the spark shell, and close the console window

1.2 Linux

Note That: The below code only tested under Fedora

- Installing Scala
  - curl -O http://downloads.typesafe.com/scala/2.11.7/scala-2.11.7.rpm
  - rpm -ivh scala-2.11.7.rpm
  - curl -O http://ftp.unicamp.br/pub/apache/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz

  - tar -xvzf apache-maven-3.3.9-bin.tar.gz
  - cd apache-maven-3.3.9
  - rm /usr/bin/mvn 2> /dev/null
  - alternatives –install /usr/bin/mvn mvn on $HOME/apache-maven-3.3.9/bin/mvn 200000
  - sudo alternatives –config mvn
- Downloading and compiling Spark.
  - curl -O http://mirror.nbtelecom.com.br/apache/spark/spark-2.0.1/spark-2.0.1.tgz
  - cd spark-2.0.1/
  - Note That: Starting Spark installation. Be very, very, very patient.
  - build /mvn -DskipTests clean package
- Run ./spark-shell
- Test it out!
  - ./bin/spark-shell
  - val textFile = sc.textFile("README.md") // textFile: org.apache.spark.rdd.RDD[String] = README.md MapPartitionsRDD[1] at textFile at <console>:25
  - textFile.count() // Number of items in this RDD
  - res0: Long = 126
  - You should show a count of the number of lines in that file.
  - Hit control-D to exit the spark shell, and close the console window

1.3 Windows

- Install a JDK (Java Development Kit) from http://www.oracle.com/technetwork/java/javase/downloads/ . Keep track of where you installed the JDK; youâĂŹll need that later.
- Download a pre-built version of Apache Spark from https://spark.apache.org/downloads.html
- If necessary, download and install WinRAR so you can extract the .tgz file you downloaded. http://www.rarlab.com/download.htm
- Extract the Spark archive, and copy its contents into C:\spark after creating that directory. You should end up with directories like c:\spark bin, c: \\spark\conf, etc.
- Download winutils.exe from https://github.com/steveloughran/winutils/blob/master/hadoop-2.7.1/bin/winutils.exe and move it into a C:\winutils\bin folder that youâĂŹve created. (note, this is a 64-bit application. If you are on a 32-bit version of Windows, you will need to search for a 32-bit build of winutils.exe for Hadoop.)

- Open the the c:\spark\conf folder, ahttps://github.com/steveloughran/winutils/blob/master/hadoop-2.7.1/bin/winutils.exend make sure âĂIJFile Name ExtensionsâĂİ is checked in the âĂIJviewâĂİ tab of Windows Explorer. Rename the log4j.properties.template file to log4j.properties. Edit this file (using Wordpad or something similar) and change the error level from INFO to ERROR for log4j.rootCategory
- Right-click your Windows menu, select Control Panel, System and Security, and then System. Click on âĂIJAdvanced System SettingsâĂİ and then the âĂIJEnvironment VariablesâĂİ button. \
- Add the following new USER variables:
  - SPARK_HOME c:\spark
  - JAVA_HOME (the path you installed the JDK to in step 1, for example C:\Program Files\Java\jdk1.8.0_101)
  - HADOOP HOME c:\winutils
- Add the following paths to your PATH user variable:
  - %SPARK_HOME%\bin.
  - %JAVA_HOME%\bin.
- Close the environment variable screen and the control panels.
- Install the Scala IDE from http://scala-ide.org/download/sdk.html
- Test it out!
  - ./bin/spark-shell
  - val textFile = sc.textFile("README.md") // textFile: org.apache.spark.rdd.RDD[String] = README.md MapPartitionsRDD[1] at textFile at <console>:25
  - textFile.count() // Number of items in this RDD
  - res0: Long = 126
  - You should show a count of the number of lines in that file.
  - Hit control-D to exit the spark shell, and close the console window

2 Spark on Scala IDE

- Download the Scala IDE from http://scala-ide.org/download/sdk.html
- Open eclipse
- choose the working directory Ex: /run/media/moustafaalaa/Main_Hard/Work/SparkWS/
- File -> new -> scala project.
- projectname : scalademo -> next -> finish.

- Here you have two choices

1. First One import spark jars into the project direct as below

   - Right click on the project "scalademo" -> buildpath -> configure build path -> libraries -> Add External Jars =\> choose your spark home directory -> jars -> select All jars -> Ok -> Ok
   - Right click on the project -> new -> scala object -> choose a name such as "scalatest" then ok.
   - copy the below code into the object

```
import scala.math.random
import org.apache.spark._

object scalatest {
  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setAppName("Spark Pi")
      .setMaster("local")

    val spark = new SparkContext(conf)
    val slices = if (args.length > 0) args(0).toInt
        else 2
    val n = math.min(100000L * slices,
        Int.MaxValue).toInt // avoid overflow
    val count = spark.parallelize(1 until n,
        slices).map { i =>
      val x = random * 2 - 1
      val y = random * 2 - 1
      if (x * x + y * y < 1) 1 else 0
    }.reduce(_ + _)
    println("Pi is roughly " + 4.0 * count / n)
    spark.stop()

  }
}
```

2. Second one "recommended" using maven Will update it soon.