Spark搭建手册

# 一、准备工作

## 1、版本选择及服务器

### 1.1、版本选择

Hadoop：2.6.5

JDK：1.8

scala：2.11.8

### 1.2、服务器地址

node 192.168.2.100

### 1.3、准备工作

搭建Hadoop的伪分布式环境

# 二、安装Scala

## 1、在本地window环境中使用winscp工具将scala-2.10.6.tar.gz 上传虚拟机的/usr/local/QST文件夹下

## 2、使用tar命令将scala-2.11.8.tar.gz 压缩包解压

tar -xzf scala-2.11.8.tar.gz

## 重命名scala文件夹

mv scala-2.11.8 scala

## 4、配置scala的环境变量

执行vim /etc/profile命令将scala的环境变量添加至文件末尾

export SCALA\_HOME=/usr/local/QST/scala

export PATH=$PATH:$SCALA\_HOME/bin

# 安装Spark

## 1、在本地window环境中使用winscp工具将spark-2.3.3-bin-hadoop2.6.tgz文件上传至虚拟机的/usr/local/QST文件夹下

## 2、使用tar命令将spark-2.3.3-bin-hadoop2.6.tgz压缩包解压

tar -xzf spark-2.3.3-bin-hadoop2.6.tgz

## 3、重命名spark文件夹

mv spark-2.3.3-bin-hadoop2.6 spark

## 4、配置spark环境变量

执行vim /etc/profile命令将spark的环境变量添加至文件末尾

export SPARK\_HOME=/usr/local/QST/spark

export PATH=$PATH:$SPARK\_HOME/bin:$SPARK\_HOME/sbin

## 5、配置spark的spark-env.sh文件

cd $SPARK\_HOME/conf

cp spark-env.sh.template spark-env.sh

vim spark-env.sh

在spark-env.sh文件末尾追加如下配置

export JAVA\_HOME=/usr/local/QST/jdk #此处修改为对应的jdk目录

export HADOOP\_HOME=/usr/local/QST/hadoop #此处修改为对应的hadoop目录

export HADOOP\_CONF\_DIR=/usr/local/QST/hadoop/etc/hadoop #此处修改为对应的hadoop配置文件目录

export SPARK\_MASTER\_IP=localhost

## 6、验证

### 6.1加载spark环境变量

source /etc/profile

### 6.2执行spark的wordcount例子

1、生成wordcount源文件wc.txt，文件内容如下

I love hadoop

I love hdfs

I love mapreduce

I love hive

I love hbase

2、将wc.txt文件上传至HDFS根目录

hdfs dfs -put wc.txt /

1. 进入spark命令行（本地模式）

执行spark-shell命令，在spark-shell命令行中执行如下代码并观察结果

val file = sc.textFile("/wc.txt")

file.collect

val word = file.flatMap(\_.split(" "))

word.collect

val wordOne = word.map((\_,1))

wordOne.collect

val result = wordOne.reduceByKey(\_+\_)

result.collect

1. 进入spark命令行（YARN）

执行spark-shell --master yarn

sc.textFile("/wc.txt").flatMap(line => line.split(" ")).map(word => (word,1)).reduceByKey((a,b) => a+b).saveAsTextFile("/result")

1. 可能出现的问题

1）spark-shell —master yarn启动失败

给节点分配的内存小了yarn kill了spark application

修改yarn-site.xml，重启yarn

<property>

<name>yarn.nodemanager.pmem-check-enabled</name>

<value>false</value>

</property>

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

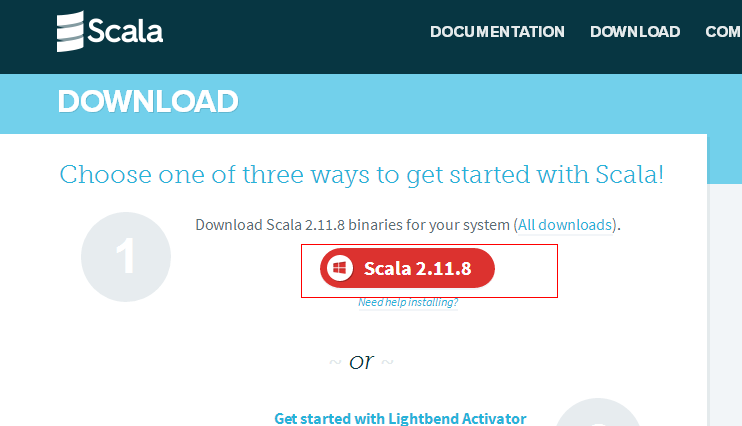
<value>false</value>

</property>

# 四、安装开发环境

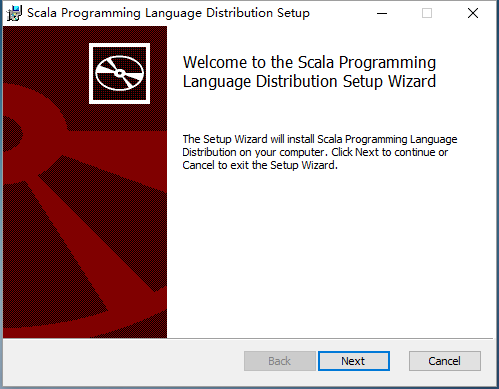
## 1、下载Scala

在官方网站https://www.scala-lang.org/下载对应版本的scala

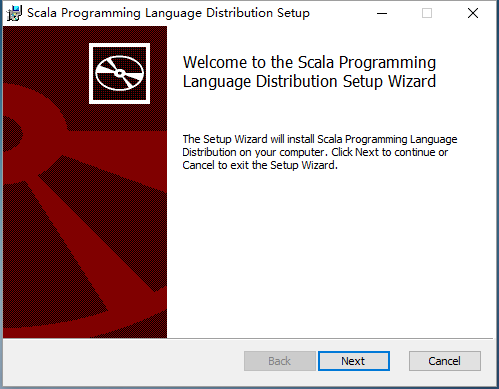


## 2、安装Scala

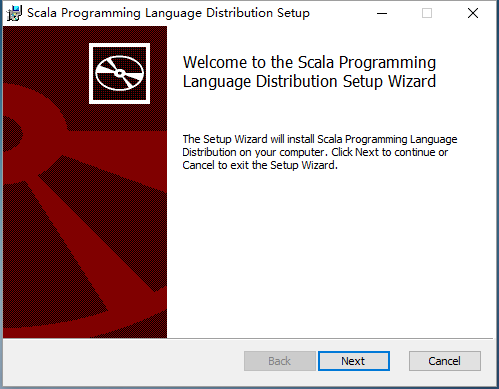
双击scala-2.11.8.msi安装，启动安装程序



单击Next下一步，然后选择安装位置，后续全部选择默认即可进入安装



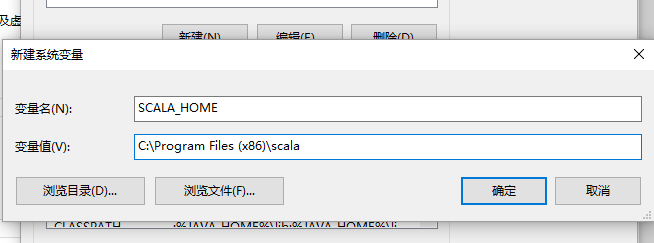
安装完毕后，点击Finish退出



## 配置环境变量

安装Scala之后，需要配置一下环境变量，从而直接从命令行访问，配置过程和JDK蕾丝

进入Windows环境变量，此处不在详细描述，然后在系统变量中新增SCALA\_HOME变量，值是安装位置，类似”C:\Program Files(x86)\scala”



然后点击确定，编辑Path变量，在后面添加如下配置：

;%SCALA\_HOME%\bin;%SCALA\_HOME%\jre\bin

## 4、下载ScalaIDE

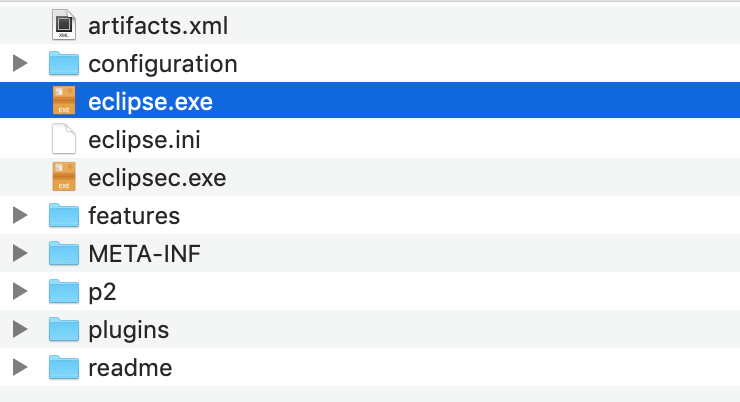
在官网<http://scala-ide.org/download/sdk.html>下载对应系统的ScalaIDE



## 5、安装

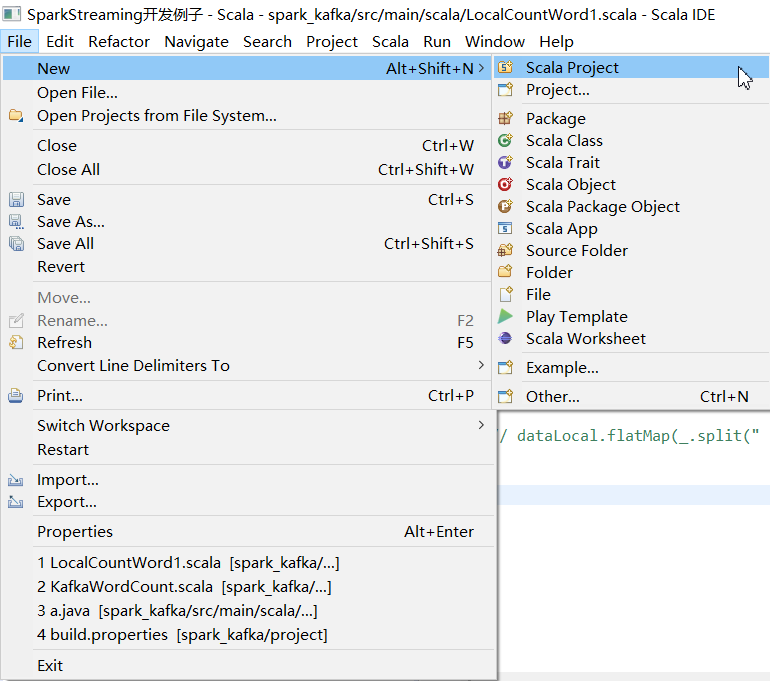
解压scala-SDK-4.7.0-vfinal-2.12-win32.win32.x86\_64.zip后生成一个eclipse文件夹

进入eclipse文件夹双击eclipse.exe打开ScalaIDE

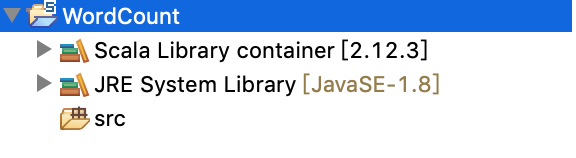


# 使用ScalaIDE开发环境编写WordCount实例

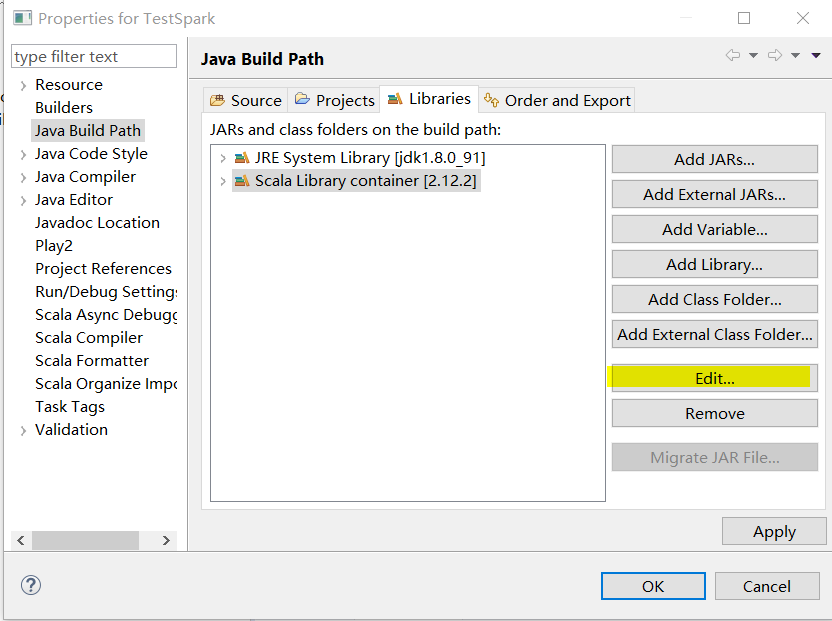
## 1、新建一个scala project

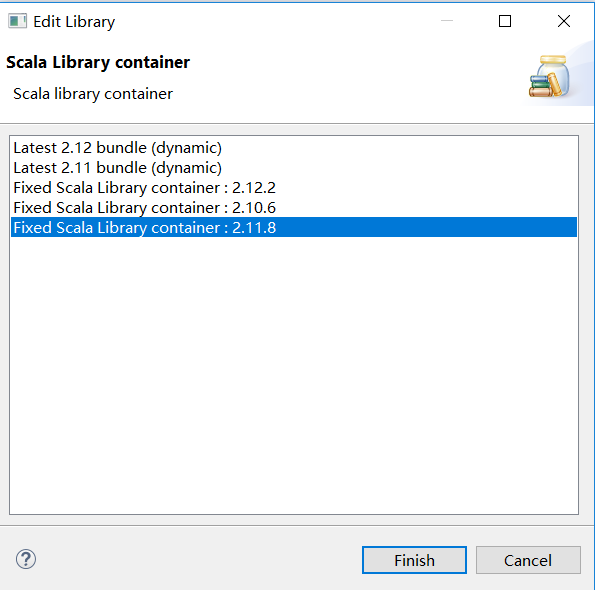


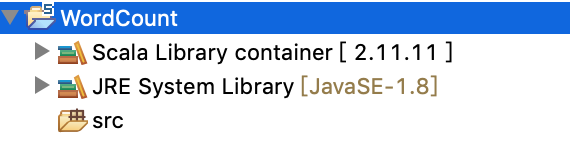
## 2、新建一个WordCount工程



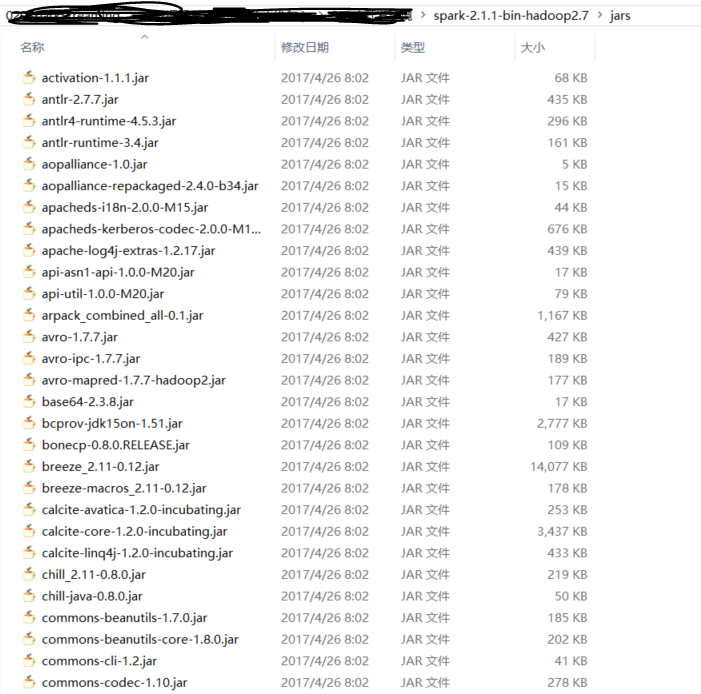
在这里scala版本默认使用的是2.12.3，我们要吧scala的版本设置为2.11.8



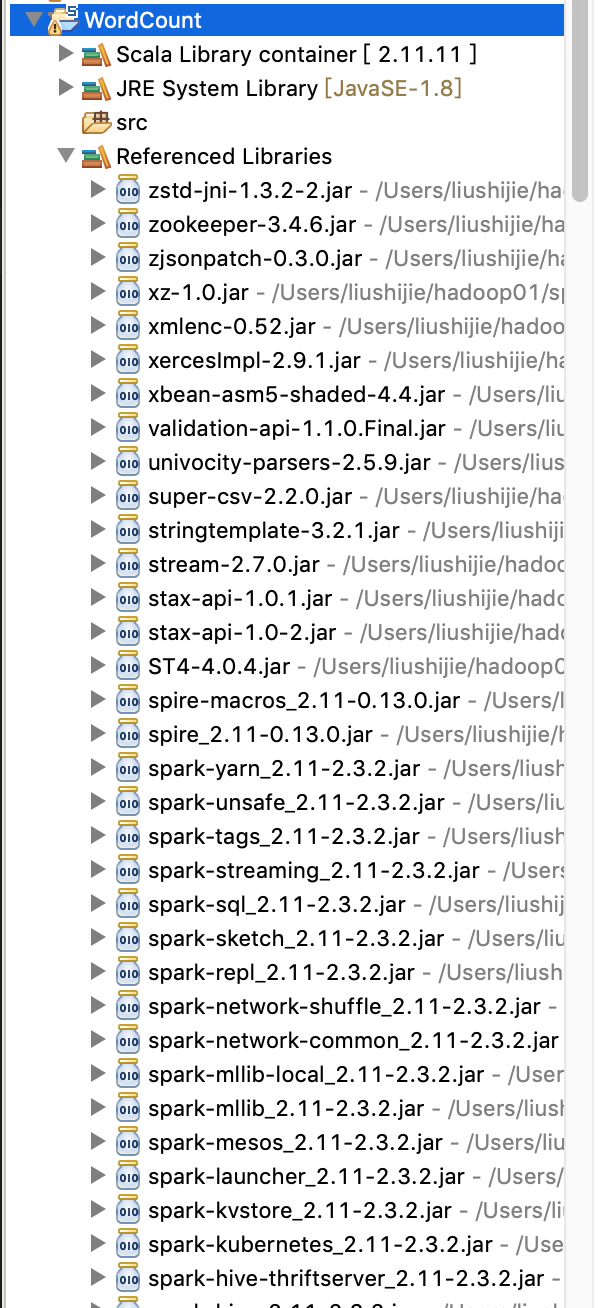




## 3、导入spark的所有jar包

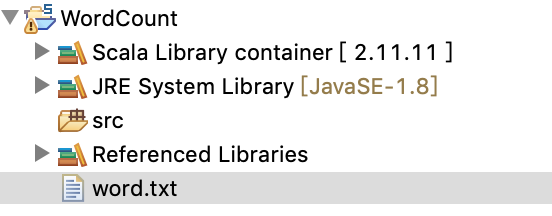


如图：



# WordCount示例

## 在WordCount工程下新建一个word.txt文件



word.txt内容如下：

I love hadoop

I love hdfs

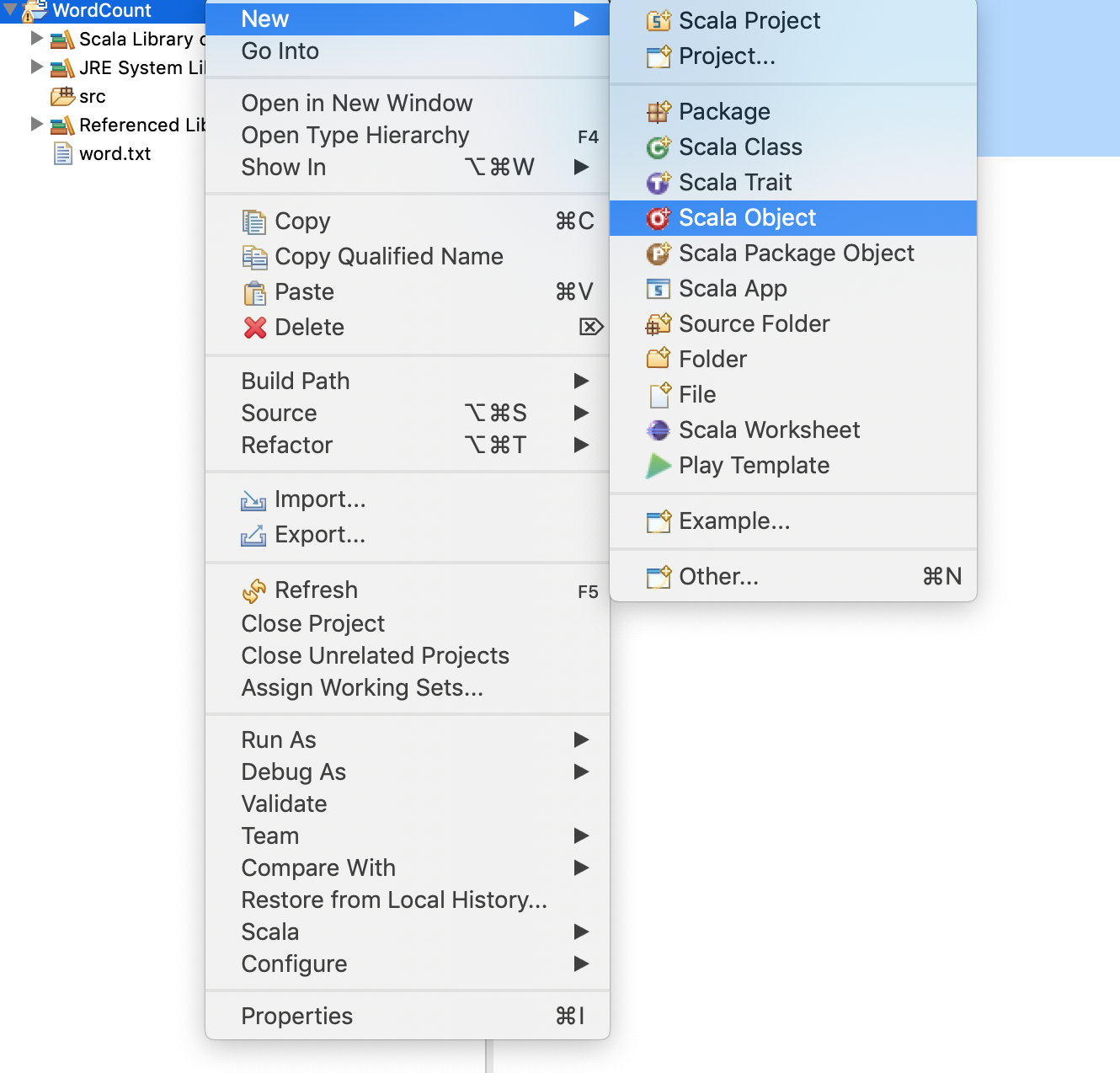
I love hive

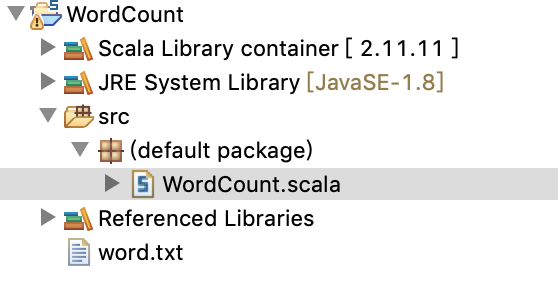
I love hbase

I love spark

## 本地模式新建一个WordCount.scala

右键New------>选择Scala Object





本地模式WordCount.scala代码如下：

import org.apache.spark.SparkConf

import org.apache.spark.SparkContext

import org.apache.spark.SparkContext.\_

object WordCount {

def main(args: Array[String]) {

val conf = new SparkConf()

conf.setMaster("local");

conf.setAppName("Word Count");

val sc = new SparkContext(conf);

val textFile = sc.textFile("word.txt");

val wordCounts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey((a, b) => a + b);

wordCounts.saveAsTextFile("result");

println("Word Count Done");

sc.stop();

}

}

右键Run As-------->Scala Application

## 集群模式

修改代码如下：

import org.apache.spark.SparkConf

import org.apache.spark.SparkContext

import org.apache.spark.SparkContext.\_

object WordCount {

def main(args: Array[String]) {

val conf = new SparkConf()

conf.setAppName("Word Count");

val sc = new SparkContext(conf);

val textFile = sc.textFile("/word.txt");

val wordCounts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey((a, b) => a + b);

wordCounts.saveAsTextFile("/result");

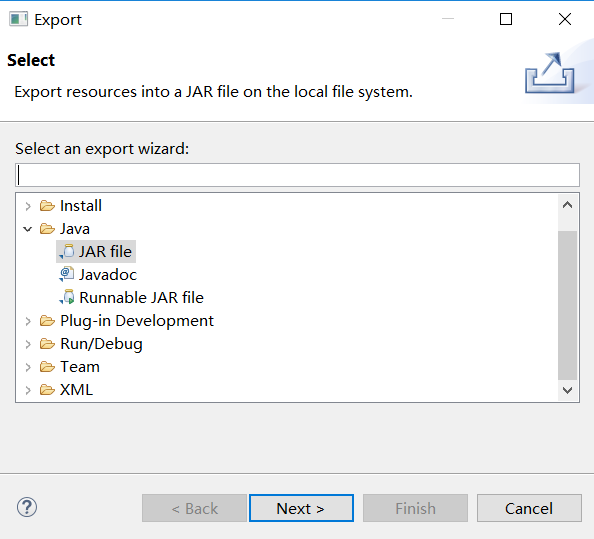
println("Word Count Done");

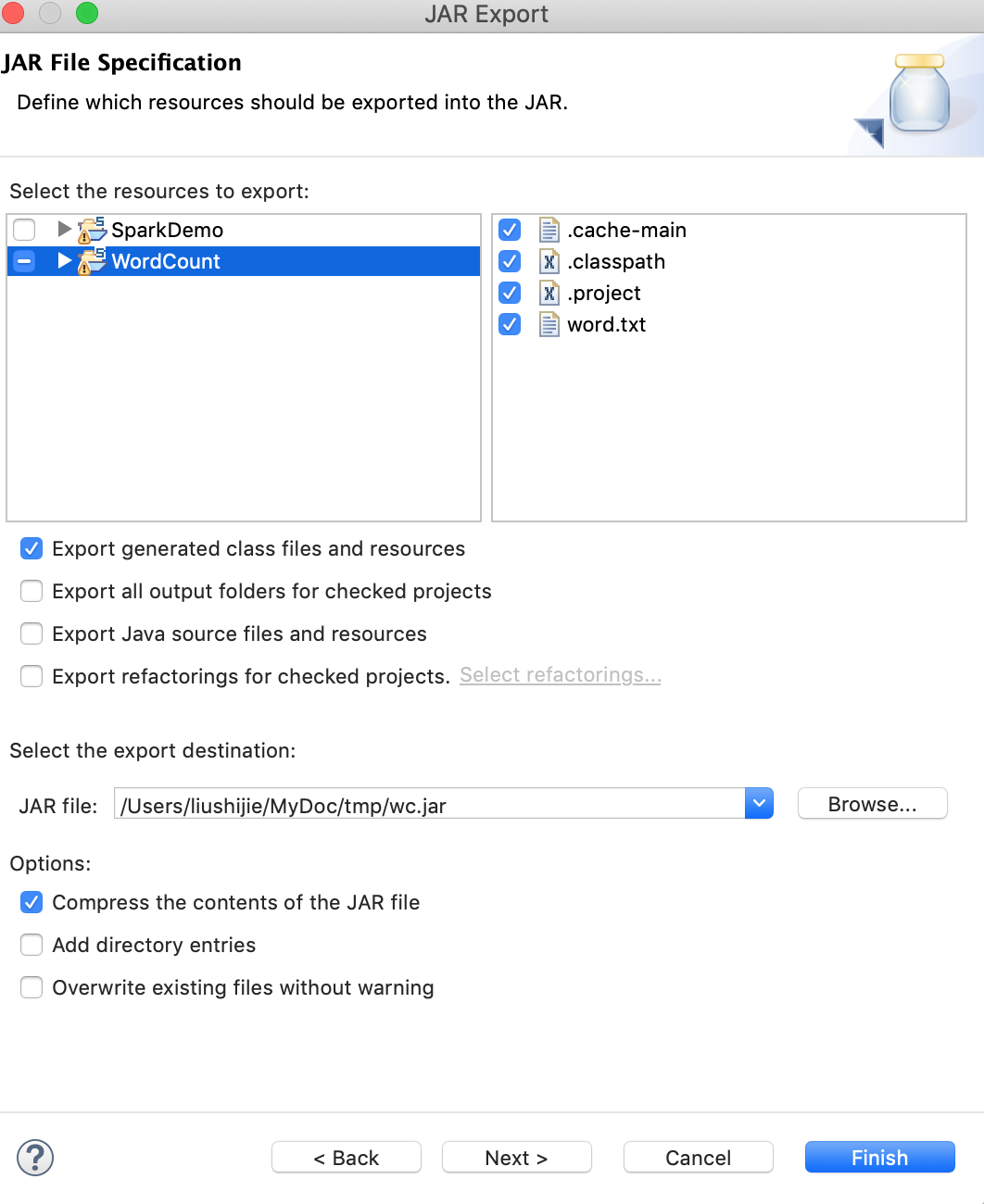
sc.stop();

}

}

打包：右键、export、Java、jar File





把wc.jar包上传到spark服务器/usr/local/QST下

确定HDFS的/word.txt文件存在

提交spark任务

spark-submit --class "WordCount" --master yarn /usr/local/QST/wc.jar