Spark Standalone模式应用程序开发

在本博客的《Spark快速入门指南(Quick Start Spark)》文章中简单地介绍了如何通过Spark shell来快速地运用API。本文将介绍如何快速地利用Spark提供的API开发Standalone模式的应用程序。Spark支持三种程序语言的开发：Scala (利用SBT进行编译), Java (利用Maven进行编译)以及Python。下面我将分别用Scala、Java和Python开发同样功能的程序：

一、Scala版本：

程序如下：

```
package scala
/**
 * User: 过往记忆
 * Date: 14-6-10
 * Time: 下午11:37
 * bolg:
 * 本文地址：/archives/1041
 * 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 * 过往记忆博客微信公共帐号：iteblog_hadoop
 */
import org.apache.spark.SparkContext
import org.apache.spark.SparkConf
object Test {
   def main(args: Array[String]) {
     val logFile = "file:///spark-bin-0.9.1/README.md"
     val conf = new SparkConf().setAppName("Spark Application in Scala")
     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: %s, Lines with b: %s".format(numAs, numBs))
   }
 }
}
```

为了编译这个文件，需要创建一个xxx.sbt文件，这个文件类似于pom.xml文件，这里我们创建一个scala.sbt文件，内容如下：

```
name := "Spark application in Scala"
```

```
version := "1.0"
scalaVersion := "2.10.4"
libraryDependencies += "org.apache.spark" %% "spark-core" % "1.0.0"
resolvers += "Akka Repository" at "http://repo.akka.io/releases/"
```

编译：

```
# sbt/sbt package
[info] Done packaging.
[success] Total time: 270 s, completed Jun 11, 2014 1:05:54 AM
```

二、Java版本

```
/**
 * User: 过往记忆
 * Date: 14-6-10
 * Time: 下午11:37
 * bolg:
 * 本文地址：/archives/1041
 * 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 * 过往记忆博客微信公共帐号：iteblog_hadoop
 */
/* SimpleApp.java */
import org.apache.spark.api.java.*;
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.function.Function;

public class SimpleApp {
  public static void main(String[] args) {
    String logFile = "file:///spark-bin-0.9.1/README.md";
    SparkConf conf =new SparkConf().setAppName("Spark Application in Java");
    JavaSparkContext sc = new JavaSparkContext(conf);
    JavaRDD<String> logData = sc.textFile(logFile).cache();

    long numAs = logData.filter(new Function<String, Boolean>() {
      public Boolean call(String s) { return s.contains("a"); }
    }).count();

    long numBs = logData.filter(new Function<String, Boolean>() {
      public Boolean call(String s) { return s.contains("b"); }
```

```
      }).count();

      System.out.println("Lines with a: " + numAs +",lines with b: " + numBs);
   }
}
```

本程序分别统计README.md文件中包含a和b的行数。本项目的pom.xml文件内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
          http://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>

   <groupId>spark</groupId>
   <artifactId>spark</artifactId>
   <version>1.0</version>

   <dependencies>
      <dependency>
         <groupId>org.apache.spark</groupId>
         <artifactId>spark-core_2.10</artifactId>
         <version>1.0.0</version>
      </dependency>
   </dependencies>
</project>
```

利用Maven来编译这个工程：

```
# mvn install
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 5.815s
[INFO] Finished at: Wed Jun 11 00:01:57 CST 2014
[INFO] Final Memory: 13M/32M
[INFO] ------------------------------------------------------------------------
```

三、Python版本

```
#
# User: 过往记忆
# Date: 14-6-10
# Time: 下午11:37
# bolg:
# 本文地址：/archives/1041
# 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
# 过往记忆博客微信公共帐号：iteblog_hadoop
#
from pyspark import SparkContext

logFile = "file:///spark-bin-0.9.1/README.md"
sc = SparkContext("local", "Spark Application in Python")
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)
```

四、测试运行

本程序的程序环境是Spark 1.0.0，单机模式，测试如下：
1、测试Scala版本的程序

```
# bin/spark-submit --class "scala.Test"  ₩
                --master local[4]    ₩
           target/scala-2.10/simple-project_2.10-1.0.jar

14/06/11 01:07:53 INFO spark.SparkContext: Job finished:
count at Test.scala:18, took 0.019705 s
Lines with a: 62, Lines with b: 35
```

2、测试Java版本的程序

```
# bin/spark-submit --class "SimpleApp"  ₩
                --master local[4]    ₩
           target/spark-1.0-SNAPSHOT.jar
```

```
14/06/11 00:49:14 INFO spark.SparkContext: Job finished:
count at SimpleApp.java:22, took 0.019374 s
Lines with a: 62, lines with b: 35
```

3、测试Python版本的程序

```
# bin/spark-submit --master local[4]    ₩
              simple.py

Lines with a: 62, lines with b: 35
```

₩本文地址：《Spark Standalone模式应用程序开发》
：/archives/1041，过往记忆，大量关于Hadoop、Spark等个人原创技术博客