

Spark 0.9.1 Standalone模式简单例子测试

在本博客的[《Spark 0.9.1 Standalone模式分布式部署》](#)详细的介绍了如何部署Spark Standalone的分布式，在那篇文章中并没有介绍如何来如何来测试，今天我就来介绍如何用Java来编写简单的程序，并在Standalone模式下运行。

程序的名称为SimpleApp.java，通过调用Spark提供的API进行的，在程序编写前现在pom引入相应的jar依赖：

```
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-core_2.10</artifactId>
  <version>0.9.1</version>
</dependency>
```

如果你的测试文件是在HDFS上，还需要引入HDFS的相关依赖。具体的程序如下：

```
package com.wyp;

import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.api.java.function.Function;

/**
 * User: wyp
 * Date: 14-4-24
 * Time: 上午11:42
 * Blog: /
 */
public class SimpleApp {
    public static void main(String[] args) {
        String logFile = "file:///home/wyp/README.md";
        JavaSparkContext sc = new JavaSparkContext("local", "Simple App",
            "/export1/spark/spark-bin-0.9.1",
            new String[]{"target/spark-1.0.jar"});
        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);
}
}
```

利用Maven进行打包，并运行：

```
[wyp@master ~/spark]$ sudo mvn package
[wyp@master ~/spark]$ sudo mvn exec:java -Dexec.mainClass="com.wyp.SimpleApp"
```

运行的结果如下：

Lines with a: 62, lines with b: 35

需要注意，（1）、在程序SimpleApp.java文件中运到的file:///home/wyp/README.md需要确保每个Worker节点都存放一份，或者你把这个文件上传到HDFS上面，确保所有的Worker节点都能正常访问；

（2）、/export1/spark/spark-bin-0.9.1这个是你Spark的安装目录

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#)（）