华东师范大学数据科学与工程学院实验报告

课程名称： 分布式模型与编程	**年级：** 2016 级	**上机实践成绩：**
指导教师： 徐辰	**姓名：** 张宏伟	
上机实践名称： Spark 编程	**学号：** 10165101180	**上机实践日期：**
上机实践编号： 07	**组号：**	**上机实践时间：** **20181026**

一、实验目的

配置 Spark 编程环境，熟悉 Spark 在 scala 和 java 下的编程接口。

二、实验任务

用 scala 和 java 分别编写能够实现统计 spark/README 文件中包含 a 和 b 行数的程序，并在本地和集群环境下运行。

三、使用环境

Sbt1.2.6 + apache-maven-3.3.9 + spark2.3.2 + spark-core 2.11

四、实验过程

1.本地环境

a.安装 sbt，显示 sbt 版本号以验证是否安装成功。

```
hadoop@PC-honwee:/usr/local/sbt-1.2.6/sbt$ sbt sbtVersion
[warn] No sbt.version set in project/build.properties, base directory: /usr/loca
l/sbt-1.2.6/sbt
[info] Set current project to sbt (in build file:/usr/local/sbt-1.2.6/sbt/)
[info] 1.2.6
```

b.创建符合符合结构的 sparkapp 文件目录，需要注意的是 simple.sbt 中的配置内容要和 spark-shell 版本对应，启动 spark-shell 时会显示版本内容。

```
hadoop@PC-honwee:~$ cd sparkapp
hadoop@PC-honwee:~/sparkapp$ find .
.
./simple.sbt
./src
./src/main
./src/main/scala
./src/main/scala/SimpleApp.scala
```

c.打包程序

```
[info] Packaging /home/hadoop/sparkapp/target/scala-2.11/simple-project_2.11-1.0
.jar ...
[info] Done packaging.
[success] Total time: 409 s, completed Oct 26, 2018, 1:24:50 PM
```

d.运行

```
hadoop@PC-honwee:~/sparkapp/target/scala-2.11$ spark-submit --class "SimpleApp"
/home/hadoop/sparkapp/target/scala-2.11/simple-project_2.11-1.0.jar 2>&1 | grep
"Lines with a:"
Lines with a: 61, Lines with b: 30
hadoop@PC-honwee:~/sparkapp/target/scala-2.11$
```

e.java 同理，直接贴运行结果

```
hadoop@PC-honwee:~/sparkapp2$ spark-submit --class "SimpleApp" target/simple-pro
ject-1.0.jar 2>&1 |grep "Lines with a"
Lines with a: 61, lines with b: 30
```

2.集群环境

Scala 代码修改

```
/*** SimpleApp.java ***/
import org.apache.spark.api.java.*;
import org.apache.spark.api.java.function.Function;

public class SimpleApp {
    public static void main(String[] args) {
        String logFile = "hdfs://10.11.6.91:9000/README.md"; // Should be some file on your system
        JavaSparkContext sc = new JavaSparkContext("spark://10.11.6.91:7077", "Simple App",
            "file:///usr/local/spark/", new String[]{"target/simple-project-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);
    }
}
```

Java 代码修改

```
/* 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 = "hdfs://10.11.6.91:9000/README.md" // Should be some file on your system
        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: %s, Lines with b: %s".format(numAs, numBs))
    }
}
```

7.集群下运行结果

Scala 运行结果

```
hadoop24@ubuntu16g-1:~/cluster_scala$ spark-submit --class "SimpleApp" /home/hadoop24/cluster_scala/t
arget/scala-2.11/simple-project_2.11-1.0.jar 2>&1 | grep "Lines with a:"
Lines with a: 61, Lines with b: 30
hadoop24@ubuntu16g-1:~/cluster_scala$
```

Java 运行结果

```
hadoop24@ubuntu16g-1:~/cluster_java$ spark-submit --class "SimpleApp" /home/hadoop24/cluster_java/target/simple-project-1.
0.jar 2>&1 | grep "Lines with a:"
Lines with a: 61, lines with b: 30
hadoop24@ubuntu16g-1:~/cluster_java$
```

五、 总结

1. 配 maven 时因为不小心开了网络代理，导致无法成功解析域名打包程序，具体表现为可以用 shell ping 网址但无法用浏览器上网。
2. 编写配置文件时需要注意 spark-shell 的版本。