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：**吴双	
上机实践名称： Spark 编程	**学号：** 10164102141	**上机实践日期：**
上机实践编号：#7	**组号：** **23**	**上机实践时间：**

一、实验目的

使用 Scala 和 Java 进行基于 Spark RDD 的程序开发与本地、集群运行

二、实验任务

使用 Scala 和 Java 进行基于 Spark RDD 的程序开发（实现统计 spark/README 文件里的 “a” 和 “b” 行数的程序）与本地、集群运行。

三、使用环境

Ubuntu LTS 18.04
Hadoop 2.7.3
Sbt 1.2.6
Maven 3.3.9
Spark 2.3.2
Spark-core 2.11

四、实验过程

1. 本地环境搭建，使用 Scala 语言编写 Spark 程序，并使用 sbt 编译打包：

$ sudo apt-get install sbt
$ sbt sbtVersion

```
hadoop@Master ~/sparkapp/target/scala-2.11 sbt sbtVersion 1 ↵ 687 03:39:13
[warn] No sbt.version set in project/build.properties, base directory: /home/hadoop/sparkapp/target/sc
ala-2.11
[info] Set current project to scala-2-11 (in build file:/home/hadoop/sparkapp/target/scala-2.11/)
>[info] 1.2.6
```

2. 创建相应结构的 sparkapp 文件夹并使用 sbt 打包程序：

$ cd sparkapp
$ find .

```
hadoop@Master ~ cd sparkapp 689 03:42:29
hadoop@Master ~/sparkapp find . 690 03:42:37
.
./target
./target/streams
./target/streams/compile
./target/streams/compile/internalDependencyClasspath
./target/streams/compile/internalDependencyClasspath/$global
./target/streams/compile/internalDependencyClasspath/$global/streams
./target/streams/compile/internalDependencyClasspath/$global/streams/export
./target/streams/compile/compile
./target/streams/compile/compile/$global
>./target/streams/compile/compile/$global/streams
```

3. 将生成的 jar 包提交给 Spark 中运行：

$ spark-submit –class “SimpleApp” ~/sparkapp/target/scala-2.11/simple-project_2.11-1.0.jar 2>&1 | grep “Lines with a:”

```
hadoop@Master ~/sparkapp spark-submit --class "SimpleApp" ~/sparkapp/target/scala-2.11/simple-pro
ject_2.11-1.0.jar 2>&1 | grep "Lines with a:"
>Lines with a: 61, Lines with b: 30
```

4. 本地环境搭建，安装 maven 使用 Java 语言编写 Spark 程序，并使用 maven 编译打包，类似之前的步骤：

$ mvn -v

```
hadoop@Master ~/sparkapp mvn -v                                    1 ↵ 697 03:51:31
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)
Maven home: /usr/local/maven
Java version: 1.8.0_181, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java8-2018-10-13/jre
Default locale: en_US, platform encoding: UTF-8
>OS name: "linux", version: "4.15.0-42-generic", arch: "amd64", family: "unix"
```

$ spark-submit –class “SimpleApp” ~/sparkapp2/target/simple-project-1.0.jar 2>&1 | grep “Lines with a:”

```
hadoop@Master ~/sparkapp spark-submit --class "SimpleApp" ~/sparkapp2/target/simple-project-1.0.j
ar 2>&1 | grep "Lines with a:"
>Lines with a: 61, lines with b: 30
```

5. 集群运行准备：

a) Scala 代码修改

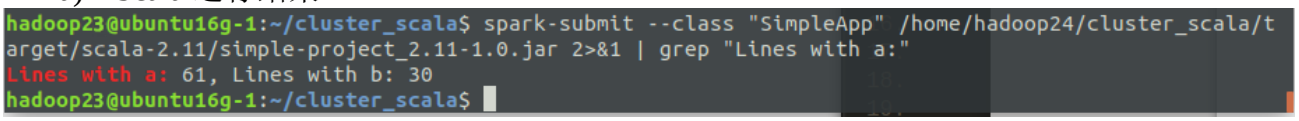
```scala
/*** SimpleApp.java ***/
import org.apache.spark.api.java.*;
import org.apache.spark.api.java.function.Function;

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);
    }
}
```

b) Java 代码修改

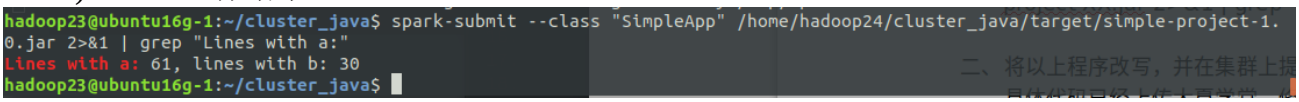
```scala
/* SimpleApp.scala */
import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf

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))
    }
}
```

6. 集群下运行结

a) Scala 运行结果



b) Java 运行结果



五、总结

集群上出现很严重的问题，总是显示“native-hadoop unfound”，后来发现是集群使用错误了。