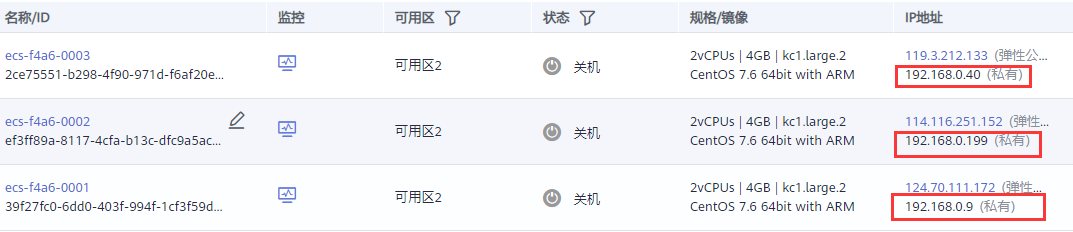
课程实验三：Spark Core Scala单词计数

实验时间：2021年04月17日

实验学生姓名：李志毅

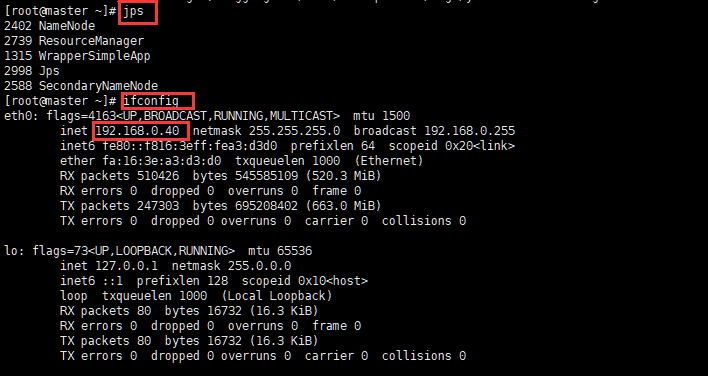
学生班号、学号：2018211314班 2018211582

**一、实验结果截图**

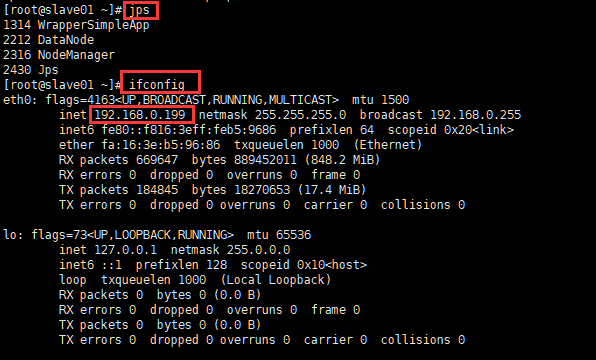
首先我本人华为云上三个服务器的内网Ipv4地址分别为：  


|  |  |  |
| --- | --- | --- |
| 节点 | 公网Ipv4地址 | 内网Ipv4地址 |
| master | 119.3.212.133 | 192.168.0.40 |
| slave01 | 114.116.251.152 | 192.168.0.199 |
| slave02 | 124.70.111.172 | 192.168.0.9 |

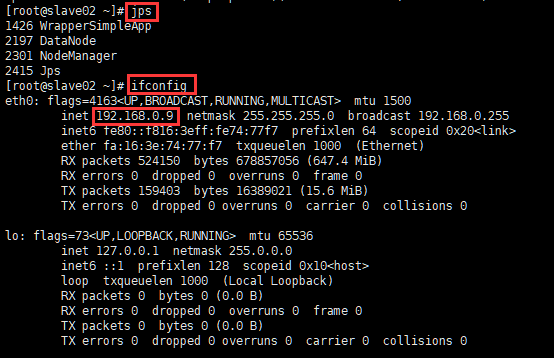
按照实验指导书部署Hadoop集群并启动后，在三个节点上执行Jps分别查看Java进程状态，可以得到如图29~31的截图：



**图29 master节点执行jps效果**



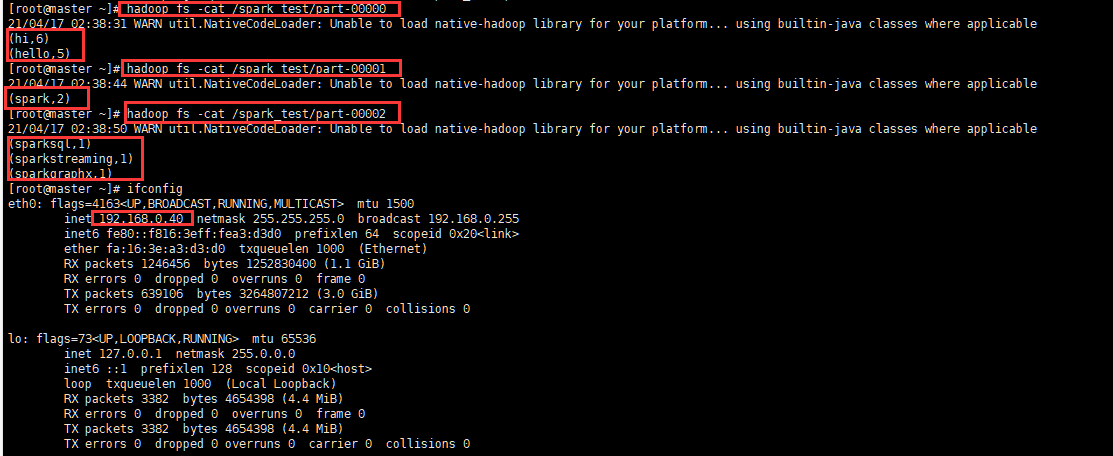
**图30 slave01节点执行jps效果**

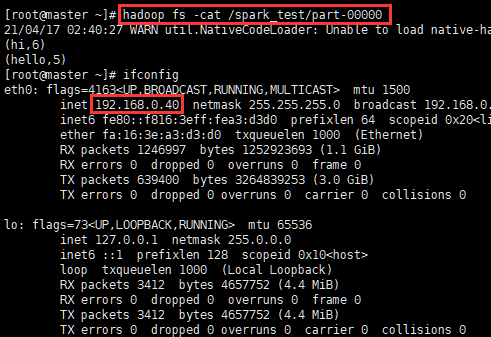


**图31 slave02节点执行jps效果**

可以看到，Hadoop集群部署完成并启动成功

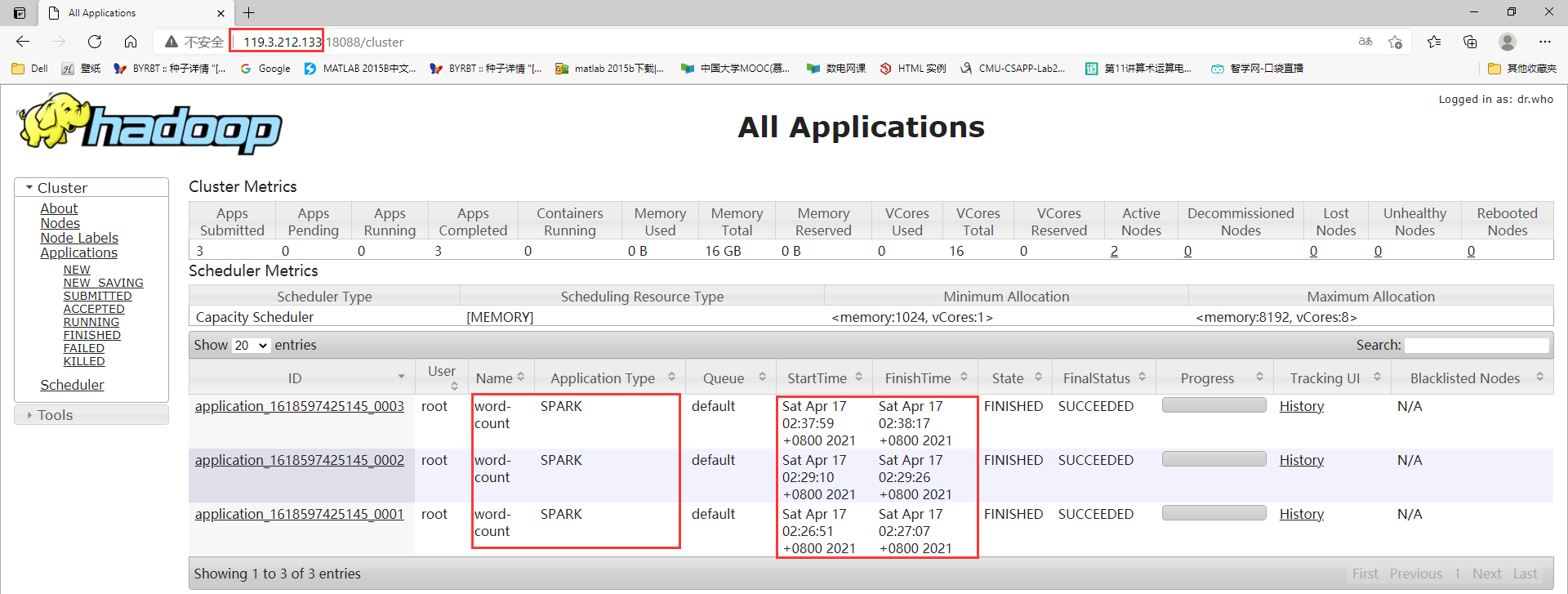
本地编写scala程序并导出jar包，在主节点使用spark-submit命令运行后可以得到如下的截图73：





**图73 spark统计词频结果**

选做，开放华为云所有安全组，访问<http://119.3.212.133:18088>可以看到任务完成情况：



**图选做 查看任务完成情况**

**二、简要描述实验做了哪些工作？**

本次实验分为云服务器端和本地端，

云服务器端首先在华为云上购买三台云主机，对三台服务器分别配置了java环境，配置了三节点互信，部署了Hadoop集群并启动成功。之后通过On Yarn模式搭建了spark环境。

本地端在本地windows安装jdk1.8和scala 2.13.5后，创建一个maven项目，并修改pom.xml文件，编写scala程序并导出完整jar包。

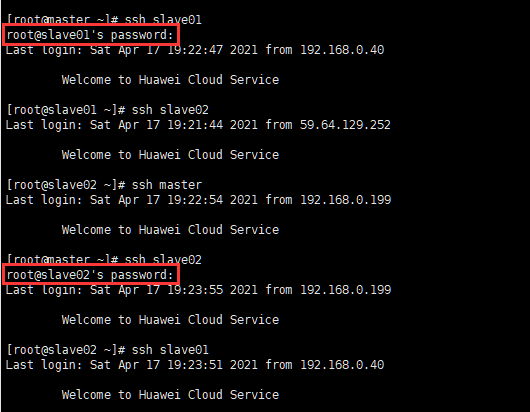
之后在云服务器上使用搭建好的环境执行命令spark-submit，该spark程序完成了单词计数任务，以空格分词计算每个单词的出现次数。

**三.实验过程中遇到的问题和解决办法？**

在本地实验中，我遇到了如下几个问题，并最终解决了这些问题

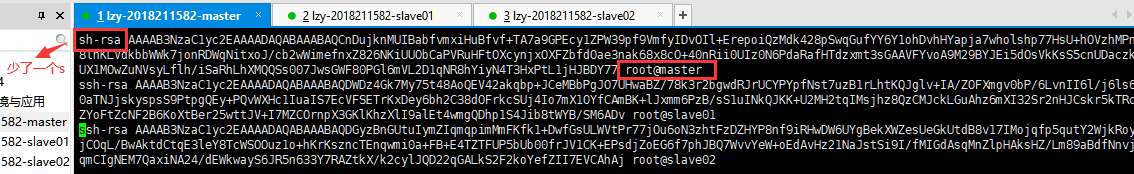
**1.节点互信配置问题**

**问题描述：**配置节点互信时，通过ssh检验始终不成功，如下图所示，在master节点使用ssh slave01命令登录slave01节点时，并未直接登录成功，而是需要输入了密码，同理在登录slave02节点时也出现了这个问题



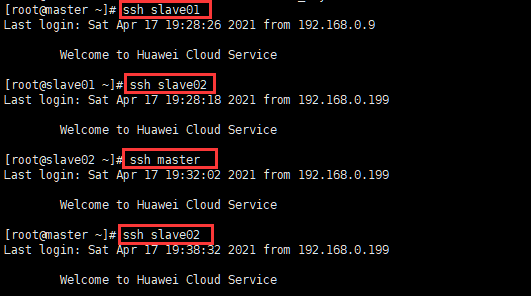
**图-1 互信检测问题**

**错误分析：**我测试了所有节点之间互相执行ssh后的结果，发现只有在master节点上执行ssh slave01和ssh slave02命令登录需要输入密码，因此将问题定位到文件/root/.ssh/authorized\_keys上，同时由于是只有master节点登录其他节点需要输入密码，因此是master节点的秘钥出错，因此我使用vim命令查看了该文件，细致检查后发现在复制master节点上的id\_rsa.pub文件内容时少复制了一个字符‘s’，如下图：



**图-2 缺少字符’s’**

将s添加上并发给另外两个节点后，我解决了这个问题也成功启动了Hadoop集群

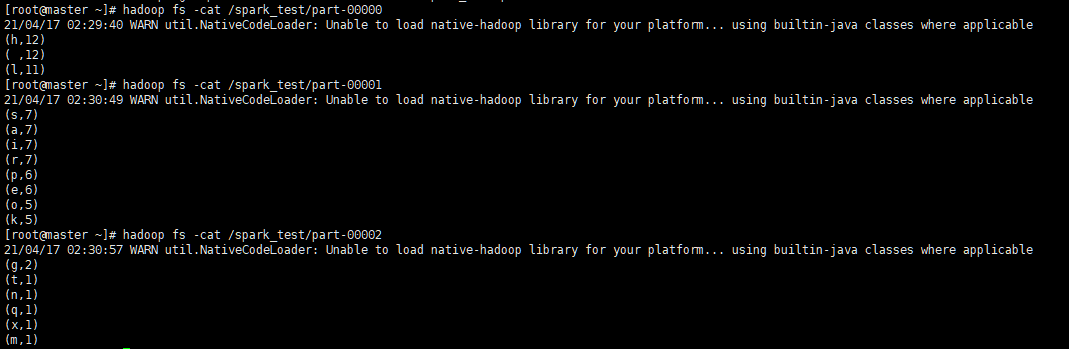


**图-3 互信配置成功**

**问题思考：**这个问题暴露出我在做实验时对于每一步操作的检查不够细致，问题产生的原因为在Linux命令行中复制整个文件后并没有仔细检查粘贴时粘贴的内容是否完整，因此犯了这样的小错误，这也警示自己做实验时一定要细致检查每一步操作，因为这些步骤都是环环相扣的

**2.scala程序编写问题**

**问题描述：**本地编写完scala程序并打包成jar包在服务器上运行时，发现运行结果不正确，运行结果与实验指导相差很大，程序并没有完成计算单词出现次数功能，或者说程序分词是按照字母进行了分词，并没有按照单词进行计算，运行截图如下：



**图-4 程序执行结果不正确**

**错误分析：**出现这个问题，我首先想到的是scala代码出现了问题，检查后果然，有一个低级错误，我把所有的val都写出了var，



**图-5 原始scala代码**

将所有的var修改成val后，我再次执行程序，发现依旧是同样的问题，问题并没有解决。因此我细致的思考了问题出现的原因，并仔细阅读了整个scala代码。由于问题出现是因为分词没有成功，使得统计单词出现次数时是按照字符进行统计，而并没有按照单词统计。因此我重点检查了代码中的

var words:RDD[String] = lines.flatMap((line:String)=>{line.split("")})

由此，我发现了问题所在，在line执行split分词函数时，他的分词依据是””而不是“ ”，我将空字符修改成了空格字符后，spark程序输出了我想要的结果，并解决了这个问题

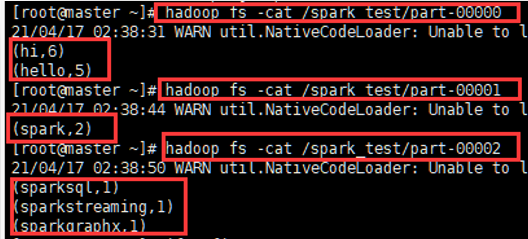


**图-6 修改后scala代码**

**问题思考：**产生这个问题的一个原因就是，首次编写scala程序时，对于语法不熟悉，由于最近一直在使用JavaScript编写相关程序，因此错误的将scala语法中的val写出了JavaScript中常用的var，虽然很快发现了这个问题，但依旧没有找到本质错误。在细致的思考整个问题出现的原因和原理后，我定位到了最终的问题并解决了它。这个问题让我意识到，编写程序代码时不能想当然，而要结合程序实际的要求，编写完成后要review并调试一下代码，看是否是按照解决问题的逻辑和正确性编写的代码，要清楚整个代码在干什么！而不是只是手动敲了敲没有语法错误就结束了，这个问题着实给我敲响了警钟。

**3.执行结果问题**

**问题描述：**执行spark-submit后得到的结果中 part-00000里没有 “(spark，2)” 而是在part-00001中，如图：



**图-7 最终执行问题**

**问题解决：**在咨询了助教后，助教学长热心的回答了问题，原来是因为分布式处理造成的，属于正常现象。至此我的整个实验完成，感谢助教学长的热心解答和帮助！谢谢！

**附 录**

**完整scala代码**

**package** org.example

**import** org.apache.spark.rdd.RDD

**import** org.apache.spark.{SparkConf, SparkContext}

**class** ScalaWordCount {

}

**object** ScalaWordCount{

**def** main(args: Array[String]): Unit={

**val** list = List("hello hi hi spark",

"hello spark hello hi sparksql",

"hello hi hi sparkstreaming",

"hello hi sparkgraphx")

**val** sparkConf = **new** SparkConf().setAppName("word-count").setMaster("yarn")

**val** sc = **new** SparkContext(sparkConf)

**val** lines:RDD[String] = sc.parallelize(list)

**val** words:RDD[String] = lines.flatMap((line:String)=>{line.split(" ")})

**val** wordAndOne:RDD[(String,Int)] = words.map((word:String)=>{(word,1)})

**val** wordAndNum:RDD[(String,Int)] = wordAndOne.reduceByKey((count1:Int,count2:Int)=>{count1+count2})

**val** ret = wordAndNum.sortBy(kv=>kv.\_2,**false**)

print(ret.collect().mkString(","))

ret.saveAsTextFile("hdfs://master:9000/spark\_test")

sc.stop()

}

}