大数据管理技术 第二次上机

林汇平 1800013104

项目链接：https://github.com/phoenixrain-pku/BigDataSummer

- 实习要求：每位同学在新概念英语第二册上完成 word count。
- 报告内容：请在报告中详细写明你的实验步骤、技术方法、实习体会等，附上相应的代码段和截图。
- 实习环境：

  虚拟机：Ubuntu 15.1.0 build-13591040

  主机操作系统：Windows 10, 64-bit (Build 17134) 10.0.17134

  内存：4GB

  硬盘：20GB

  CPU：Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz(1992 MHz)

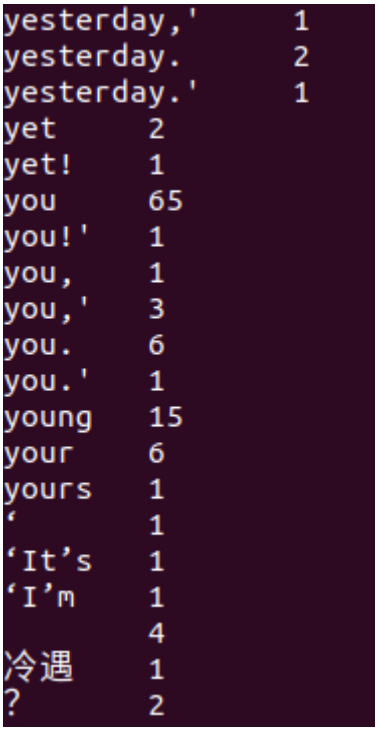
---

1. 效果展示：

   如图为将WordCount程序打包后，在hadoop下执行成功的运行结果。可以看到终端不断更新WordCount程序执行状态。在此之后终端也打印了执行程序时读、写的文件大小。最终的output文件夹可以见压缩包，也可以见项目的Github链接。

```
phoenix@Master:~/myapp$ hadoop jar WordCount2.jar input output
20/07/22 10:17:05 INFO client.RMProxy: Connecting to ResourceManager at Master/192.168.
8.100:8032
20/07/22 10:17:06 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsin
g not performed. Implement the Tool interface and execute your application with ToolRun
ner to remedy this.
20/07/22 10:17:16 INFO input.FileInputFormat: Total input paths to process : 1
20/07/22 10:17:17 INFO mapreduce.JobSubmitter: number of splits:1
20/07/22 10:17:17 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_159538358
7671_0001
20/07/22 10:17:18 INFO impl.YarnClientImpl: Submitted application application_159538358
7671_0001
20/07/22 10:17:18 INFO mapreduce.Job: The url to track the job: http://Master:8088/prox
y/application_1595383587671_0001/
20/07/22 10:17:18 INFO mapreduce.Job: Running job: job_1595383587671_0001
20/07/22 10:17:48 INFO mapreduce.Job: Job job_1595383587671_0001 running in uber mode :
 false
20/07/22 10:17:48 INFO mapreduce.Job:  map 0% reduce 0%
20/07/22 10:18:10 INFO mapreduce.Job:  map 100% reduce 0%
20/07/22 10:18:23 INFO mapreduce.Job:  map 100% reduce 100%
20/07/22 10:18:23 INFO mapreduce.Job: Job job_1595383587671_0001 completed successfully
20/07/22 10:18:23 INFO mapreduce.Job: Counters: 49
```

   使用-get命令将output文件取回本地，并使用cat命令查看。由于没有使用排序，词频统计的结果比较乱。



使用sort命令可以对词频按照从小到大进行排序。如图是部分频数较高的Words。

```
into    45
from    46
up      46
were    47
by      49
out     50
It      51
are     52
this    56
as      65
you     65
his     66
He      70
my      70
be      71
they    71
but     72
has     81
will    82
she     83
not     84
been    86
at      95
on      95
have    97
is      102
for     117
it      127
that    127
The     134
had     144
he      165
in      211
was     225
I       238
of      273
and     274
to      388
a       418
the     818
```

2. 实习过程中遇到的问题与解决方案：

- 在hdfs下突然提示网络不可达，而此处的网络配置和之前的没有区别。尝试在Master和Slave上互ping是也发生类似错误。

```
phoenix@Master:~/myapp$ hdfs dfs -ls
ls: Failed on local exception: java.net.SocketException: 网络不可达;
 Host Details : local host is: "Master/192.168.8.100"; destination h
ost is: "Master":9000;
phoenix@Master:~/myapp$ ping -c 3 Slave
connect: 网络不可达
```

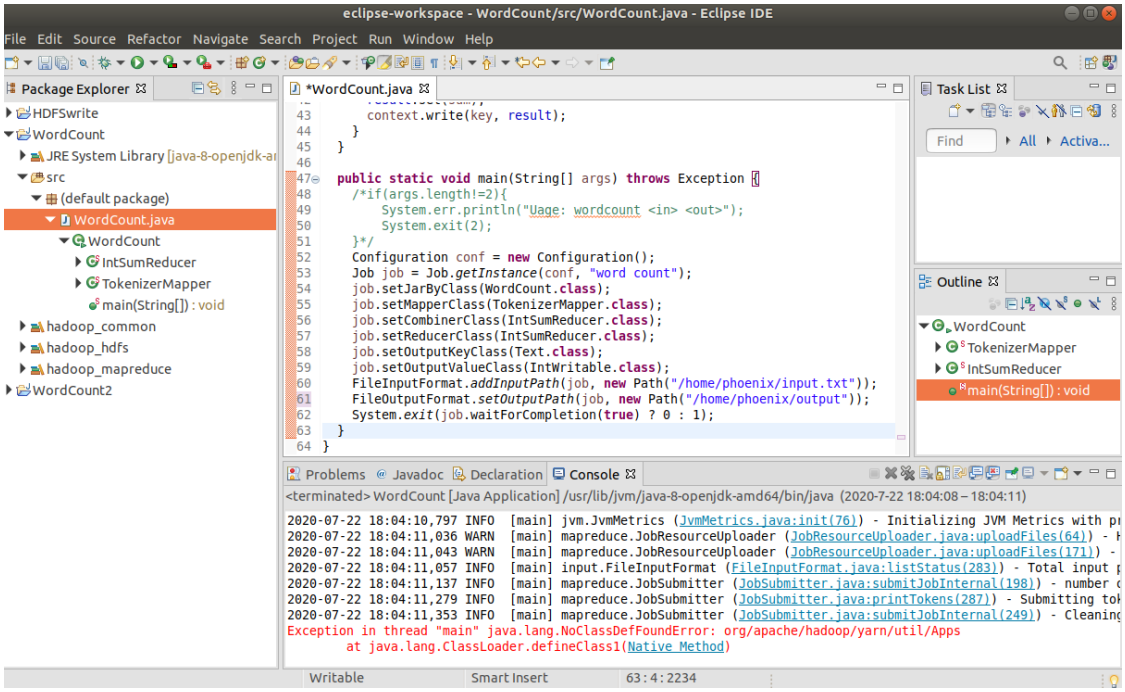
  解决方案：首先检查自己电脑的wifi是否开启。如果已开启，检查Vmware的NAT服务是否开启。两者都开启，且虚拟机网络连接正常后，问题即解决。

- 文件打包成功后在hdfs上执行，有如下报错。

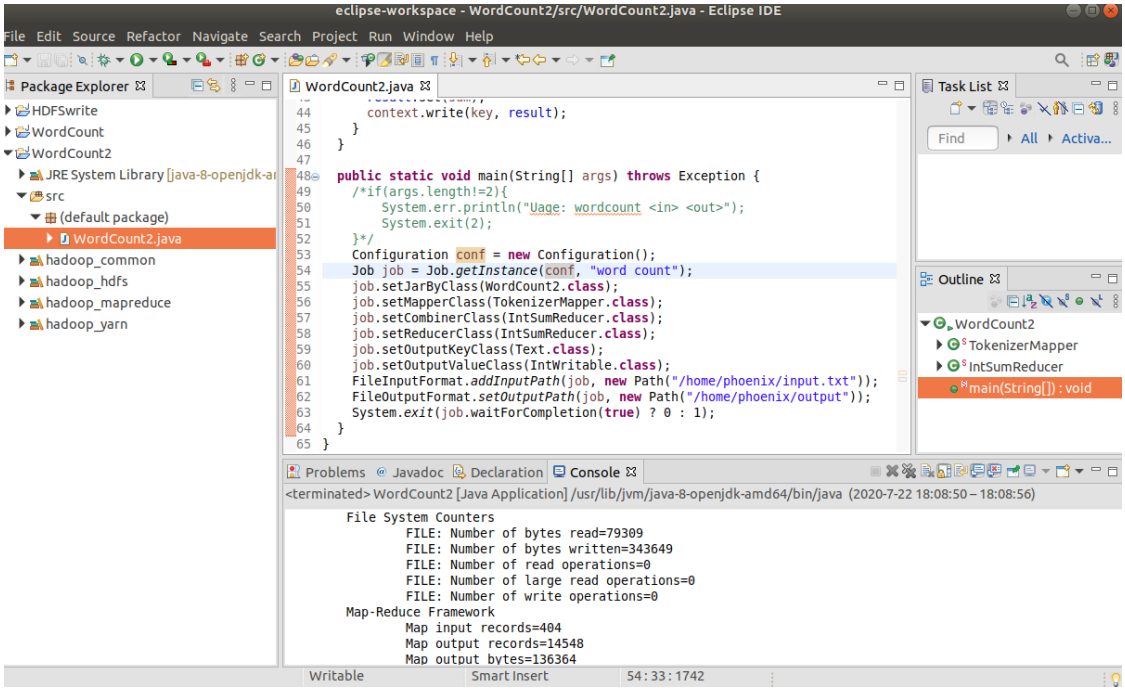
```
phoenix@Master:~/myapp$ hadoop jar WordCount.jar input output
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [rsrc:org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:rsrc:slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/Stati
cLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
20/07/20 12:18:30 INFO client.RMProxy: Connecting to ResourceManager at Master/1
92.168.8.100:8032
20/07/20 12:18:31 INFO ipc.Client: Retrying connect to server: Master/192.168.8.
100:8032. Already tried 0 time(s); retry policy is RetryUpToMaximumCountWithFixe
dSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
20/07/20 12:18:32 INFO ipc.Client: Retrying connect to server: Master/192.168.8.
100:8032. Already tried 1 time(s); retry policy is RetryUpToMaximumCountWithFixe
dSleep(maxRetries=10, sleepTime=1000 MILLISECONDS)
20/07/20 12:18:33 INFO ipc.Client: Retrying connect to server: Master/192.168.8.
100:8032. Already tried 2 time(s); retry policy is RetryUpToMaximumCountWithFixe
```

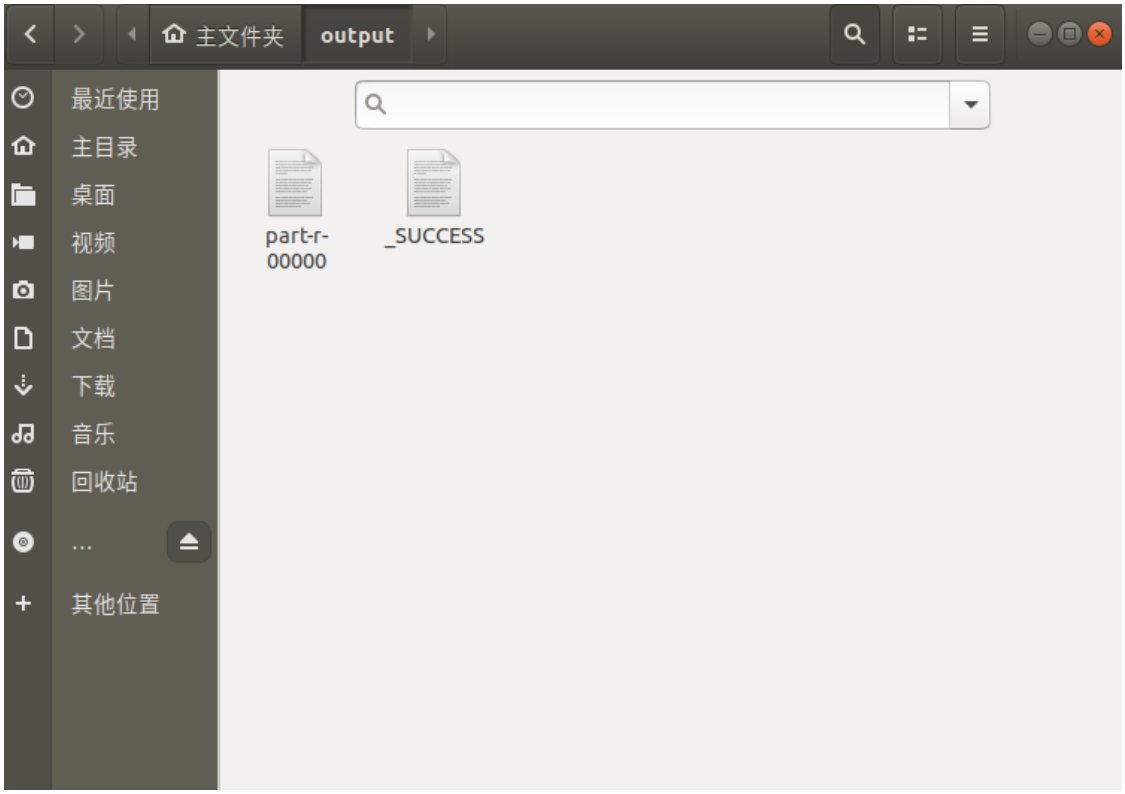
解决方案：

上网查询此问题后并没有得到比较好的答复，看到有人说是java包冲突之类的问题。于是先放弃在hadoop上进行集群，尝试在eclipse上进行local job的测试，对代码进行调整，产生如下报错：



该问题是由于没有导入yarn包所导致的。这是由于第一次安装hadoop与eclipse时，我只按照手册上导入了hdfs包与mapreduce包。因此我建了一个新的java工程，按照导入hdfs包的流程重新导入yarn包，并添加到了user library里面。再次在本地运行后，问题解决，对应路径下也出现了output文件夹。





现在再将该java工程打包并导出，在hadoop即可正确执行，并生成output文件夹了。