电子科技大学信息与软件工程学院

实 验 报 告

学　　号 ***

姓　　名 ***

（实验）　课程名称 大数据分析与智能计算

理论教师 汤宇、林迪

实验教师 杨珊

电 子 科 技 大 学

实 验 报 告

学生姓名： * 学号： *** 指导教师：*****

实验地点： 二教 110 实验时间：2023. 10. 10

一、 实验名称：Hadoop 和 Spark 安装配置

二、 实验学时：2 学时

三、 实验目的：

1. 掌握 Hadoop 的环境搭建
2. 掌握 Hadoop、Linux 的基本命令。

四、 实验原理

首先需要创建用户，然后授予用户权利，在 linux 系统上，如果用户没有读写运行的权利将会有许多事务收到限制。然后安装 ssh，安装 ssh 的目的在能在后期做多设备 hadoop 实验时候能够登录到别人的设备上，并且无需密码。运行的 wordCount 代码在 hadoop 的 share 包里，能够检索文本里面单词出现的次数，但这次实验只在单机上运行，所以无论输入和输出都在单机上进行。

五、 实验内容

首先安装 SSH，SSH 为 Secure Shell 的缩写，由 IETF 的网络小组（Network Working Group）所制定；SSH 为建立在应用层基础上的安全协议。安装 SSH 的目的在于为了方便配置集群,能在一台机器上登陆到集群中其他机器上。

然后在 ubuntu 上安装 hadoop，Hadoop 是一个由 Apache 基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。

最后安装 Spark.

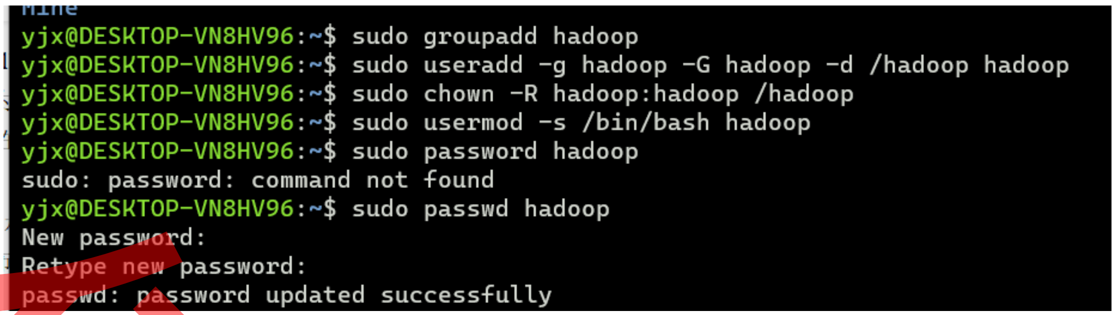
六、 实验设备及环境

VMware Workstation Pro，Ubuntu-20.04.6

七、 实验步骤

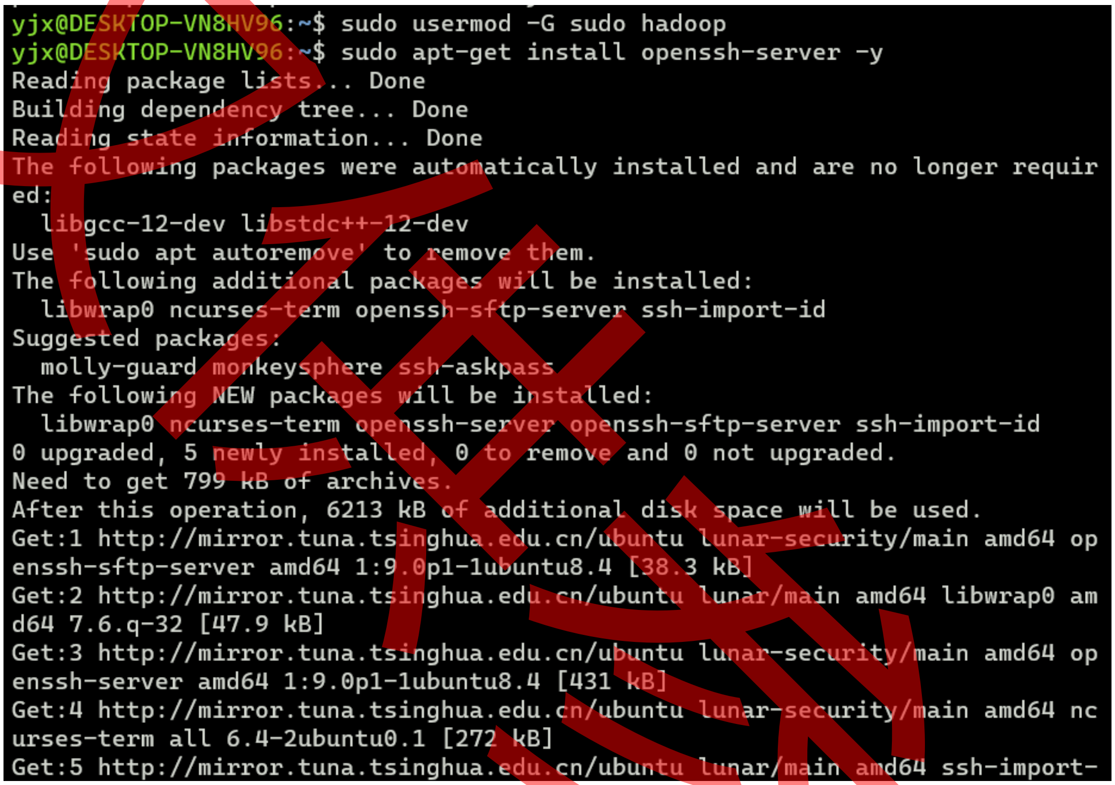
一、Hadoop 安装配置

1. 添加用户及用户组



```
yjx@DESKTOP-VN8HV96:~$ sudo groupadd hadoop
yjx@DESKTOP-VN8HV96:~$ sudo useradd -g hadoop -G hadoop -d /hadoop hadoop
yjx@DESKTOP-VN8HV96:~$ sudo chown -R hadoop:hadoop /hadoop
yjx@DESKTOP-VN8HV96:~$ sudo usermod -s /bin/bash hadoop
yjx@DESKTOP-VN8HV96:~$ sudo password hadoop
sudo: password: command not found
yjx@DESKTOP-VN8HV96:~$ sudo passwd hadoop
New password:
Retype new password:
passwd: password updated successfully
```

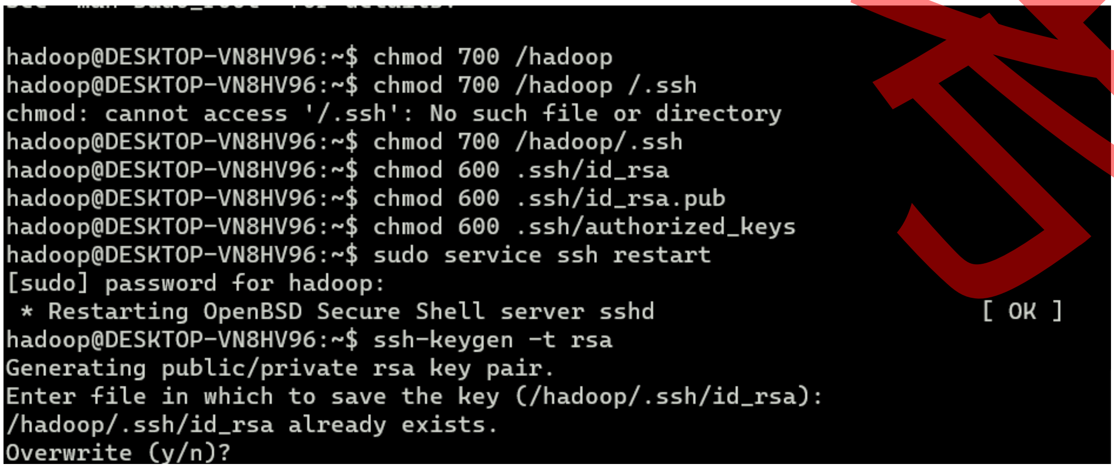
2. 添加 sudo 权限及安装及配置依赖的软件包



```
yjx@DESKTOP-VN8HV96:~$ sudo usermod -G sudo hadoop
yjx@DESKTOP-VN8HV96:~$ sudo apt-get install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer requir
ed:
  libgcc-12-dev libstdc++-12-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libwrap0 ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  libwrap0 ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 799 kB of archives.
After this operation, 6213 kB of additional disk space will be used.
Get:1 http://mirror.tuna.tsinghua.edu.cn/ubuntu lunar-security/main amd64 op
enssh-sftp-server amd64 1:9.0p1-1ubuntu8.4 [38.3 kB]
Get:2 http://mirror.tuna.tsinghua.edu.cn/ubuntu lunar/main amd64 libwrap0 am
d64 7.6.q-32 [47.9 kB]
Get:3 http://mirror.tuna.tsinghua.edu.cn/ubuntu lunar-security/main amd64 op
enssh-server amd64 1:9.0p1-1ubuntu8.4 [431 kB]
Get:4 http://mirror.tuna.tsinghua.edu.cn/ubuntu lunar-security/main amd64 nc
urses-term all 6.4-2ubuntu0.1 [272 kB]
Get:5 http://mirror.tuna.tsinghua.edu.cn/ubuntu lunar/main amd64 ssh-import-
```

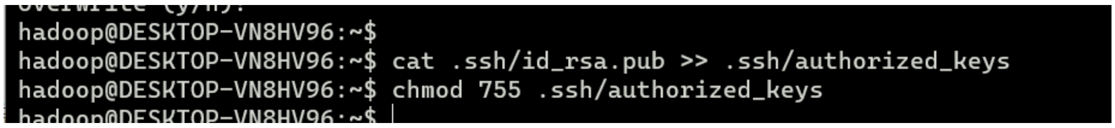
3. 配置 ssh 免密登录

3.1 更改权限



```
hadoop@DESKTOP-VN8HV96:~$ chmod 700 /hadoop
hadoop@DESKTOP-VN8HV96:~$ chmod 700 /hadoop /.ssh
chmod: cannot access '/.ssh': No such file or directory
hadoop@DESKTOP-VN8HV96:~$ chmod 700 /hadoop/.ssh
hadoop@DESKTOP-VN8HV96:~$ chmod 600 .ssh/id_rsa
hadoop@DESKTOP-VN8HV96:~$ chmod 600 .ssh/id_rsa.pub
hadoop@DESKTOP-VN8HV96:~$ chmod 600 .ssh/authorized_keys
hadoop@DESKTOP-VN8HV96:~$ sudo service ssh restart
[sudo] password for hadoop:
 * Restarting OpenBSD Secure Shell server sshd                        [ OK ]
hadoop@DESKTOP-VN8HV96:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/hadoop/.ssh/id_rsa):
/hadoop/.ssh/id_rsa already exists.
Overwrite (y/n)?
```

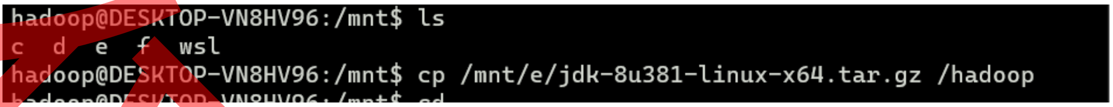
3.2 设置免密



```
hadoop@DESKTOP-VN8HV96:~$
hadoop@DESKTOP-VN8HV96:~$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys
hadoop@DESKTOP-VN8HV96:~$ chmod 755 .ssh/authorized_keys
```

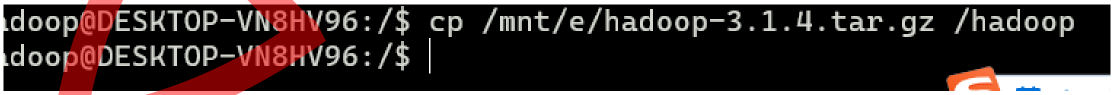
4. 下载安装 Java 和 Hadoop

4.1 下载 jdk1.8 并拷贝到/hadoop 下（注：这里我使用的是 WSL）



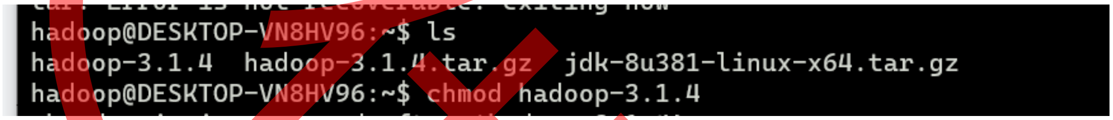
```
hadoop@DESKTOP-VN8HV96:/mnt$ ls
c  d  e  f  wsl
hadoop@DESKTOP-VN8HV96:/mnt$ cp /mnt/e/jdk-8u381-linux-x64.tar.gz /hadoop
```

4.2 下载 Hadoop 并拷贝到/hadoop 下

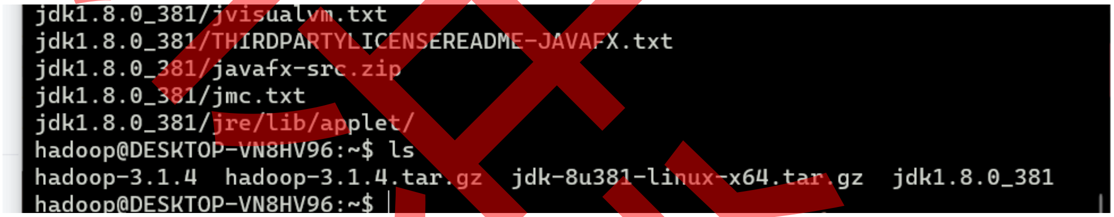


```
doop@DESKTOP-VN8HV96:/$ cp /mnt/e/hadoop-3.1.4.tar.gz /hadoop
doop@DESKTOP-VN8HV96:/$
```

4.3 解压并安装,修改权限



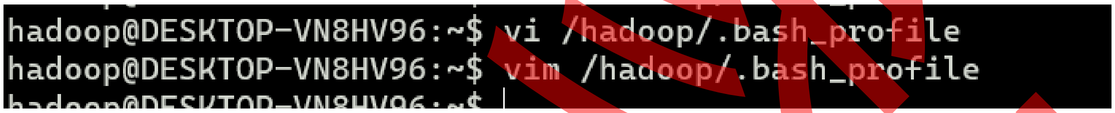
```
hadoop@DESKTOP-VN8HV96:~$ ls
hadoop-3.1.4  hadoop-3.1.4.tar.gz  jdk-8u381-linux-x64.tar.gz
hadoop@DESKTOP-VN8HV96:~$ chmod hadoop-3.1.4
```



```
jdk1.8.0_381/jvisualvm.txt
jdk1.8.0_381/THIRDPARTYLICENSEREADME-JAVAFX.txt
jdk1.8.0_381/javafx-src.zip
jdk1.8.0_381/jmc.txt
jdk1.8.0_381/jre/lib/applet/
hadoop@DESKTOP-VN8HV96:~$ ls
hadoop-3.1.4  hadoop-3.1.4.tar.gz  jdk-8u381-linux-x64.tar.gz  jdk1.8.0_381
hadoop@DESKTOP-VN8HV96:~$
```

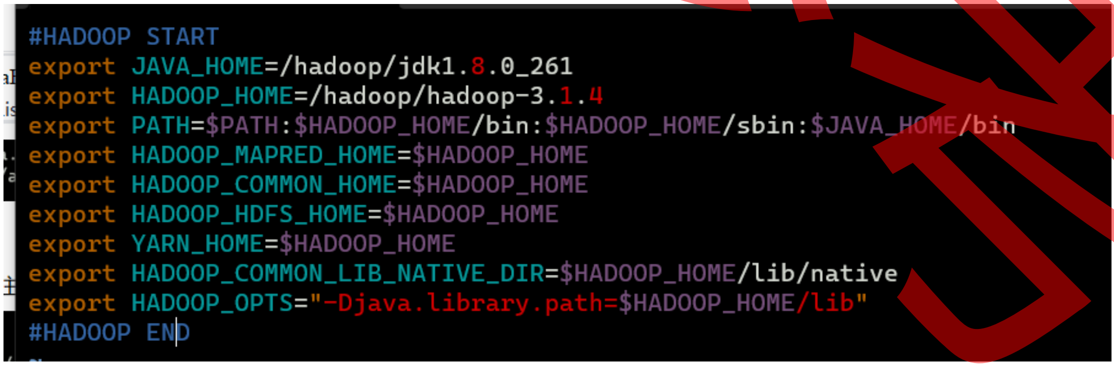
5. 配置 Hadoop 环境变量

5.1 修改配置文件



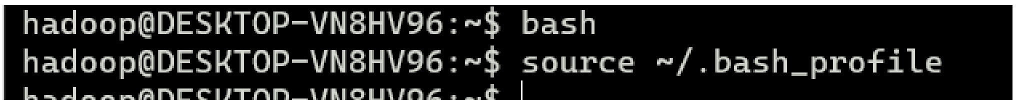
```
hadoop@DESKTOP-VN8HV96:~$ vi /hadoop/.bash_profile
hadoop@DESKTOP-VN8HV96:~$ vim /hadoop/.bash_profile
```

注意：指导教程缺少第三行，需要自行加上



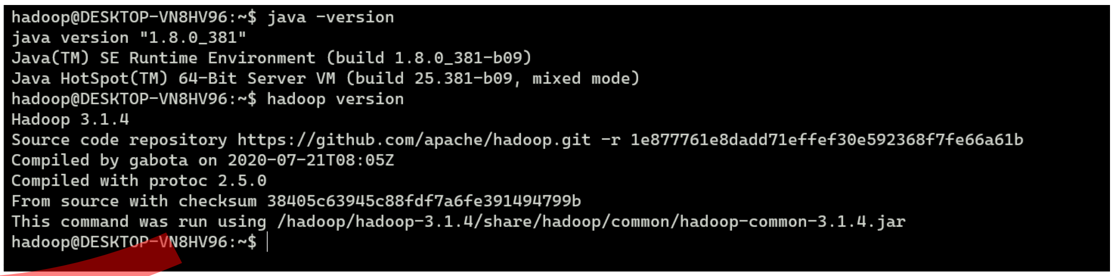
```
#HADOOP START
export JAVA_HOME=/hadoop/jdk1.8.0_261
export HADOOP_HOME=/hadoop/hadoop-3.1.4
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$JAVA_HOME/bin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#HADOOP END
```

5.2 激活新增环境变量



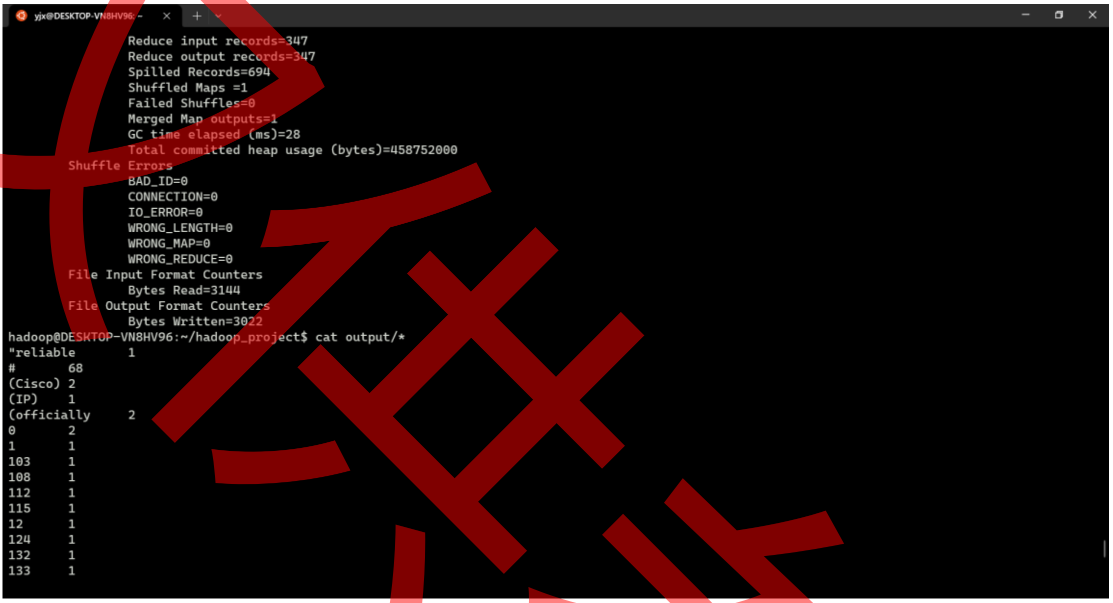
```
hadoop@DESKTOP-VN8HV96:~$ bash
hadoop@DESKTOP-VN8HV96:~$ source ~/.bash_profile
```

5.3 查看 Java 和 Hadoop 的版本信息



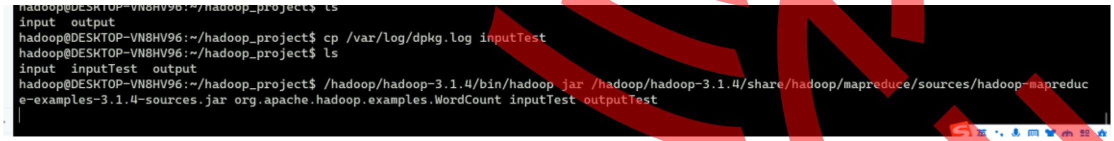
```
hadoop@DESKTOP-VN8HV96:~$ java -version
java version "1.8.0_381"
Java(TM) SE Runtime Environment (build 1.8.0_381-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.381-b09, mixed mode)
hadoop@DESKTOP-VN8HV96:~$ hadoop version
Hadoop 3.1.4
Source code repository https://github.com/apache/hadoop.git -r 1e877761e8dadd71effef30e592368f7fe66a61b
Compiled by gabota on 2020-07-21T08:05Z
Compiled with protoc 2.5.0
From source with checksum 38405c63945c88fdf7a6fe391494799b
This command was run using /hadoop/hadoop-3.1.4/share/hadoop/common/hadoop-common-3.1.4.jar
hadoop@DESKTOP-VN8HV96:~$
```

6. 测试，词频统计

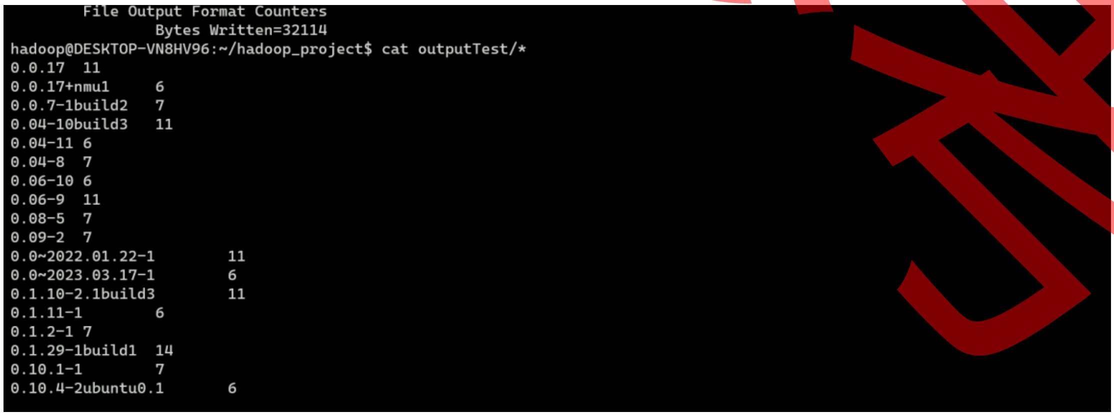


```
                Reduce input records=347
                Reduce output records=347
                Spilled Records=694
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=28
                Total committed heap usage (bytes)=458752000
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=3144
        File Output Format Counters
                Bytes Written=3022
hadoop@DESKTOP-VN8HV96:~/hadoop_project$ cat output/*
"reliable       1
#       68
(Cisco) 2
(IP)    1
(officially     2
0       2
1       1
103     1
108     1
112     1
115     1
12      1
124     1
132     1
133     1
```

八、 实验结果与分析（含重要数据结果分析或核心代码流程分析）



```
hadoop@DESKTOP-VN8HV96:~/hadoop_project$ ls
input  output
hadoop@DESKTOP-VN8HV96:~/hadoop_project$ cp /var/log/dpkg.log inputTest
hadoop@DESKTOP-VN8HV96:~/hadoop_project$ ls
input  inputTest  output
hadoop@DESKTOP-VN8HV96:~/hadoop_project$ /hadoop/hadoop-3.1.4/bin/hadoop jar /hadoop/hadoop-3.1.4/share/hadoop/mapreduce/sources/hadoop-mapreduc
e-examples-3.1.4-sources.jar org.apache.hadoop.examples.WordCount inputTest outputTest
```



```
        File Output Format Counters
                Bytes Written=32114
hadoop@DESKTOP-VN8HV96:~/hadoop_project$ cat outputTest/*
0.0.17  11
0.0.17+nmu1     6
0.0.7-1build2   7
0.04-10build3   11
0.04-11 6
0.04-8  7
0.06-10 6
0.06-9  11
0.08-5  7
0.09-2  7
0.0~2022.01.22-1        11
0.0~2023.03.17-1        6
0.1.10-2.1build3        11
0.1.11-1        6
0.1.2-1 7
0.1.29-1build1  14
0.10.1-1        7
0.10.4-2ubuntu0.1       6
```

/hadoop/hadoop-3.1.4/bin/hadoop jar /hadoop/hadoop-3.1.4/share/hadoop/mapreduce/sources/hadoop-mapreduce-examples-3.1.4-sources.jar

```
org.apache.hadoop.examples.WordCount inputTest outputTest
```

该代码展示了如何使用 Hadoop MapReduce 框架来运行一个简单的 WordCount 示例。对日志文件/var/log/dpkg.log 进行词频统计。

/hadoop/hadoop-3.1.4/bin/hadoop:这是 Hadoop 的命令行工具，用于执行 Hadoop 任务和操作。

Jar：表示运行的任务是一个 Java jar 文件

/hadoop/hadoop-3.1.4/share/hadoop/mapreduce/sources/hadoop-mapreduce-examples-3.1.4-sources.jar：这是 jar 文件的路径，包含了示例带 Hadoop MapReduce 示例代码。

org.apache.hadoop.examples.WordCount ：要运行的 Java 类的全限定名。WordCount 是示例中的一个类，它实现了 WordCount 任务，用于计算文本中各个单词出现的次数。

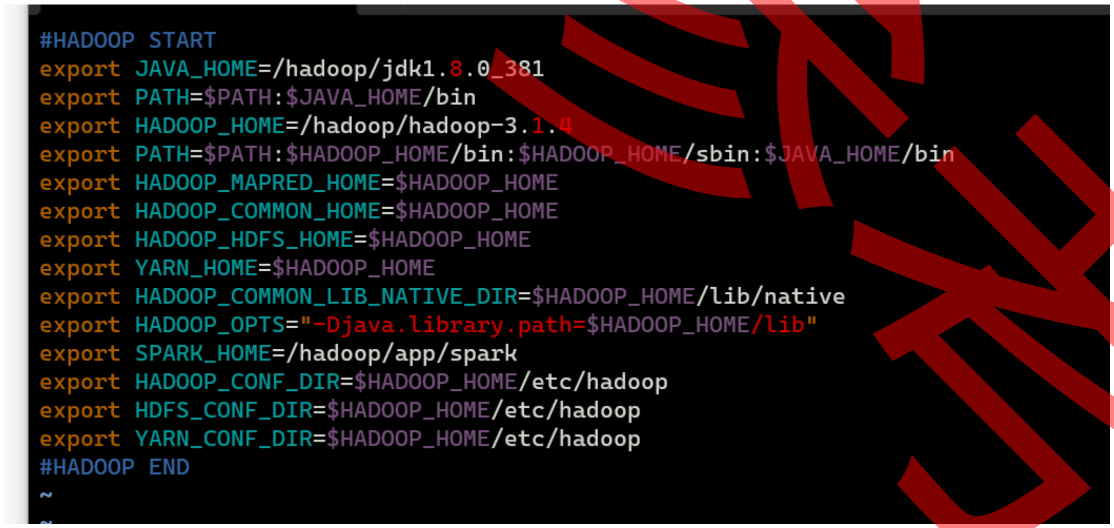
inputTest outputTest：这是我自己命名的输入输出文件。

二、Spark 安装配置

一. hadoop 伪分布式配置

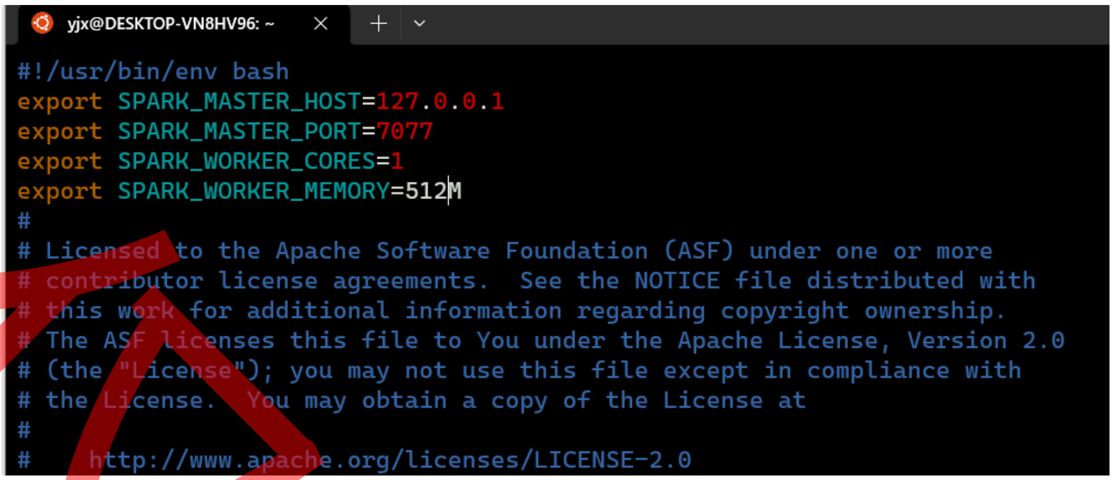
1. 修改配置文件

在 Yarn 上运行 Spark 需要配置 HADOOP_CONF_DIR、YARN_CONF_DIR 和 HDFS_CONF_DIR 环境变量命令

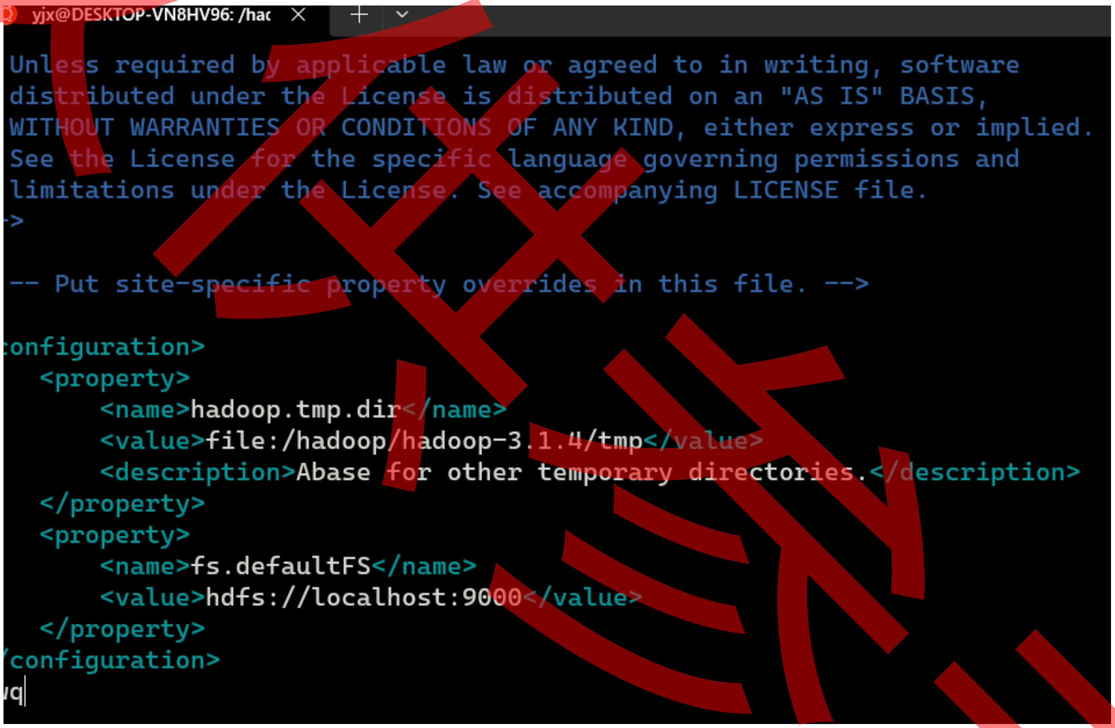


```
#HADOOP START
export JAVA_HOME=/hadoop/jdk1.8.0_381
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=/hadoop/hadoop-3.1.4
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$JAVA_HOME/bin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
export SPARK_HOME=/hadoop/app/spark
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HDFS_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
#HADOOP END
~
~
```

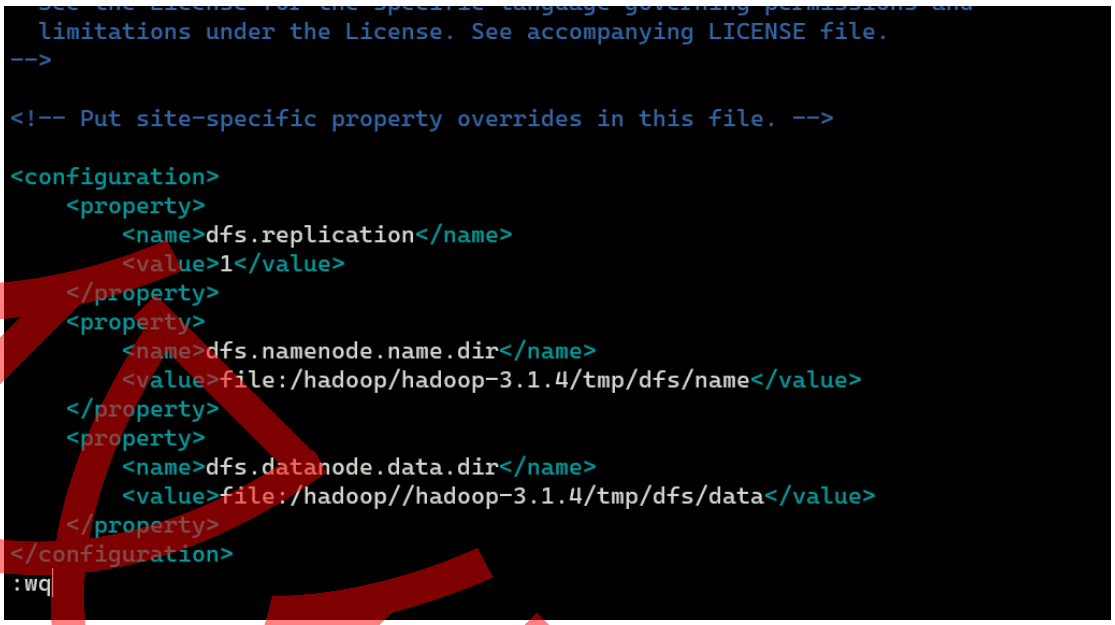
1.1 修改配置文件



```
#!/usr/bin/env bash
export SPARK_MASTER_HOST=127.0.0.1
export SPARK_MASTER_PORT=7077
export SPARK_WORKER_CORES=1
export SPARK_WORKER_MEMORY=512M
#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
```

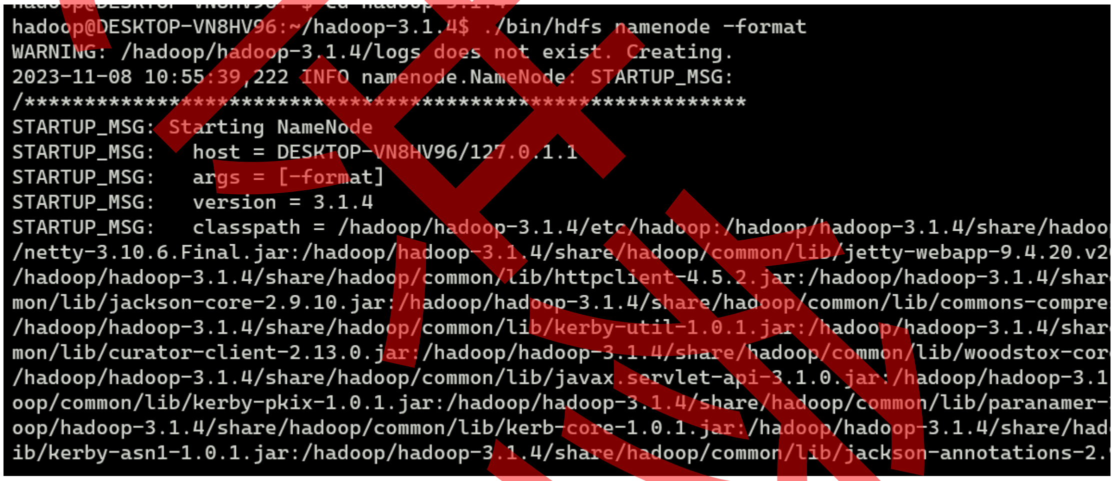


```
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
->

-- Put site-specific property overrides in this file. -->

:onfiguration>
  <property>
      <name>hadoop.tmp.dir</name>
      <value>file:/hadoop/hadoop-3.1.4/tmp</value>
      <description>Abase for other temporary directories.</description>
  </property>
  <property>
      <name>fs.defaultFS</name>
      <value>hdfs://localhost:9000</value>
  </property>
/configuration>
wq
```

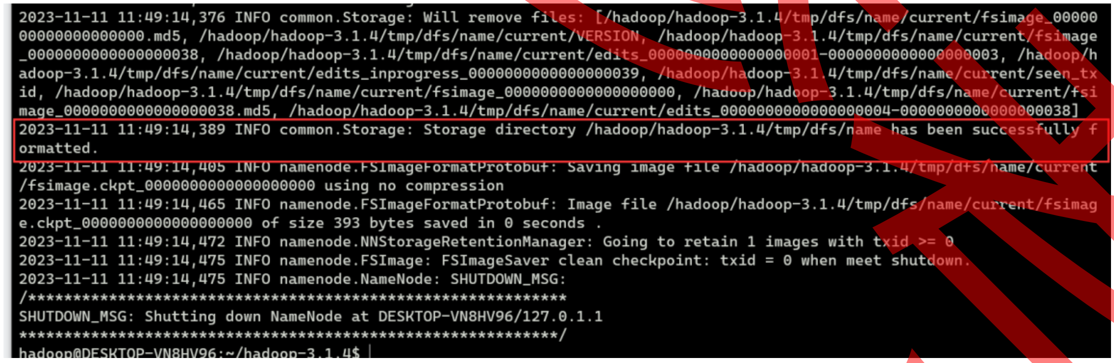


```
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/hadoop/hadoop-3.1.4/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/hadoop//hadoop-3.1.4/tmp/dfs/data</value>
    </property>
</configuration>
:wq
```

1.2namenode 的格式化

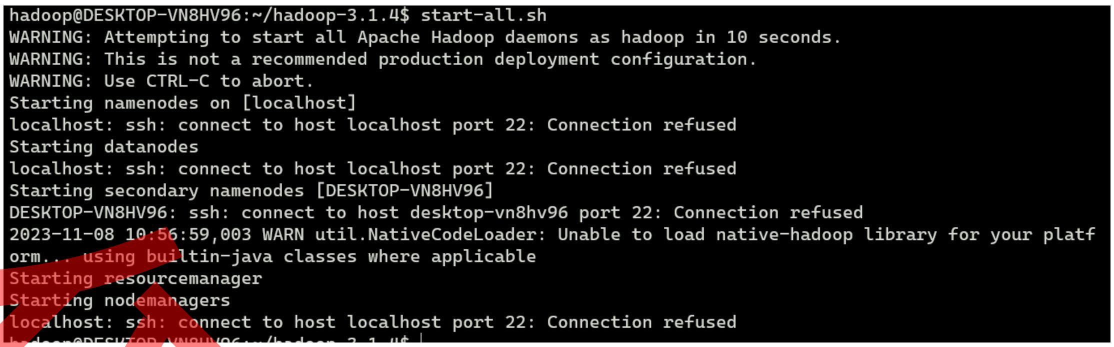


```
hadoop@DESKTOP-VN8HV96:~/hadoop-3.1.4$ ./bin/hdfs namenode -format
WARNING: /hadoop/hadoop-3.1.4/logs does not exist. Creating.
2023-11-08 10:55:39,222 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = DESKTOP-VN8HV96/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.1.4
STARTUP_MSG:   classpath = /hadoop/hadoop-3.1.4/etc/hadoop:/hadoop/hadoop-3.1.4/share/hadoo
/netty-3.10.6.Final.jar:/hadoop/hadoop-3.1.4/share/hadoop/common/lib/jetty-webapp-9.4.20.v2
/hadoop/hadoop-3.1.4/share/hadoop/common/lib/httpclient-4.5.2.jar:/hadoop/hadoop-3.1.4/shar
mon/lib/jackson-core-2.9.10.jar:/hadoop/hadoop-3.1.4/share/hadoop/common/lib/commons-compre
/hadoop/hadoop-3.1.4/share/hadoop/common/lib/kerby-util-1.0.1.jar:/hadoop/hadoop-3.1.4/shar
mon/lib/curator-client-2.13.0.jar:/hadoop/hadoop-3.1.4/share/hadoop/common/lib/woodstox-cor
/hadoop/hadoop-3.1.4/share/hadoop/common/lib/javax.servlet-api-3.1.0.jar:/hadoop/hadoop-3.1
oop/common/lib/kerby-pkix-1.0.1.jar:/hadoop/hadoop-3.1.4/share/hadoop/common/lib/paranamer-
oop/hadoop-3.1.4/share/hadoop/common/lib/kerb-core-1.0.1.jar:/hadoop/hadoop-3.1.4/share/had
ib/kerby-asn1-1.0.1.jar:/hadoop/hadoop-3.1.4/share/hadoop/common/lib/jackson-annotations-2.
```



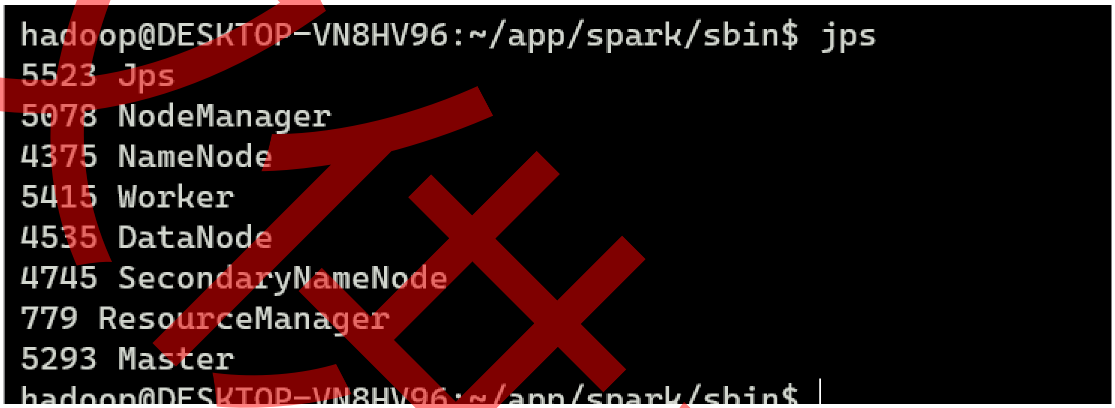
```
2023-11-11 11:49:14,376 INFO common.Storage: Will remove files: [/hadoop/hadoop-3.1.4/tmp/dfs/name/current/fsimage_00000
00000000000000.md5, /hadoop/hadoop-3.1.4/tmp/dfs/name/current/VERSION, /hadoop/hadoop-3.1.4/tmp/dfs/name/current/fsimage
_0000000000000000038, /hadoop/hadoop-3.1.4/tmp/dfs/name/current/edits_0000000000000000001-0000000000000000003, /hadoop/h
adoop-3.1.4/tmp/dfs/name/current/edits_inprogress_0000000000000000039, /hadoop/hadoop-3.1.4/tmp/dfs/name/current/seen_tx
id, /hadoop/hadoop-3.1.4/tmp/dfs/name/current/fsimage_0000000000000000000, /hadoop/hadoop-3.1.4/tmp/dfs/name/current/fsi
mage_0000000000000000038.md5, /hadoop/hadoop-3.1.4/tmp/dfs/name/current/edits_0000000000000000004-0000000000000000038]
2023-11-11 11:49:14,389 INFO common.Storage: Storage directory /hadoop/hadoop-3.1.4/tmp/dfs/name has been successfully f
ormatted.
2023-11-11 11:49:14,405 INFO namenode.FSImageFormatProtobuf: Saving image file /hadoop/hadoop-3.1.4/tmp/dfs/name/current
/fsimage.ckpt_0000000000000000000 using no compression
2023-11-11 11:49:14,465 INFO namenode.FSImageFormatProtobuf: Image file /hadoop/hadoop-3.1.4/tmp/dfs/name/current/fsimag
e.ckpt_0000000000000000000 of size 393 bytes saved in 0 seconds .
2023-11-11 11:49:14,472 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-11-11 11:49:14,475 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid = 0 when meet shutdown.
2023-11-11 11:49:14,475 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-VN8HV96/127.0.1.1
************************************************************/
hadoop@DESKTOP-VN8HV96:~/hadoop-3.1.4$
```

1.3.启动 hadoop（namenode 节点）（start-all.sh 在 sbin 里面）


```
hadoop@DESKTOP-VN8HV96:~/hadoop-3.1.4$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
localhost: ssh: connect to host localhost port 22: Connection refused
Starting datanodes
localhost: ssh: connect to host localhost port 22: Connection refused
Starting secondary namenodes [DESKTOP-VN8HV96]
DESKTOP-VN8HV96: ssh: connect to host desktop-vn8hv96 port 22: Connection refused
2023-11-08 10:56:59,003 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platf
orm... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
localhost: ssh: connect to host localhost port 22: Connection refused
```

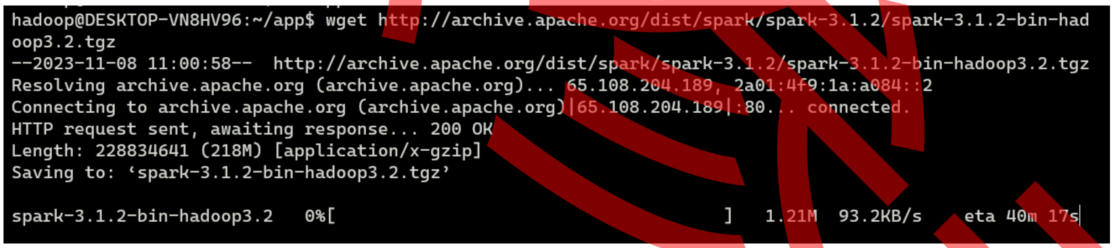
(后续已解决)

1.4 检查是否运行成功

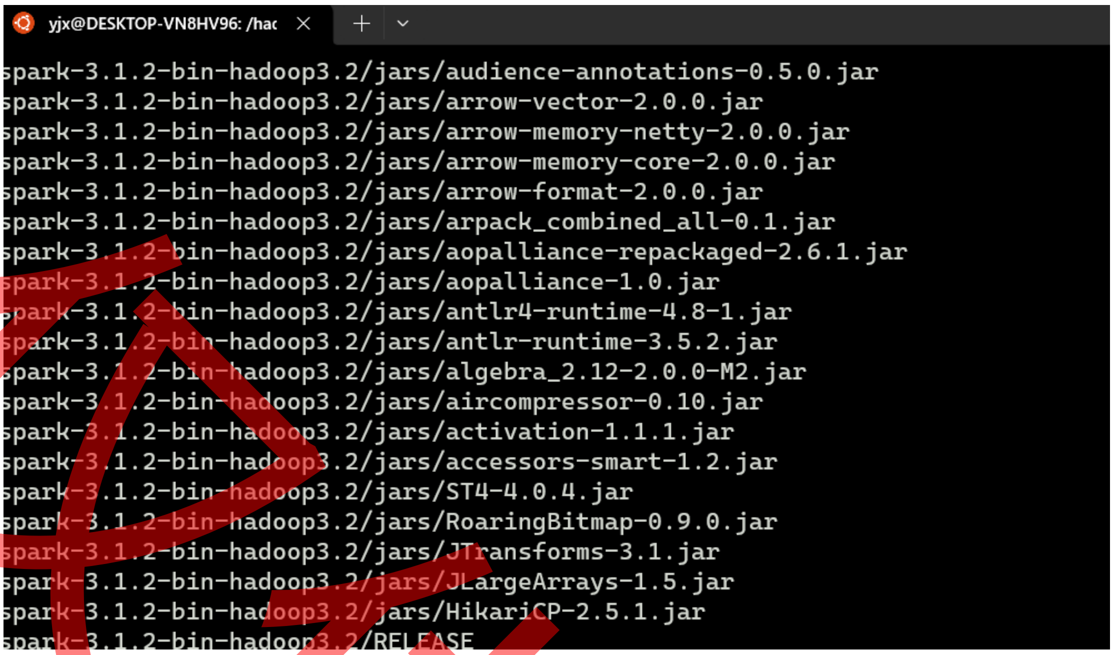

```
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$ jps
5523 Jps
5078 NodeManager
4375 NameNode
5415 Worker
4535 DataNode
4745 SecondaryNameNode
779 ResourceManager
5293 Master
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$
```

2. 安装与配置 Spark

2.1 解压并安装 Spark

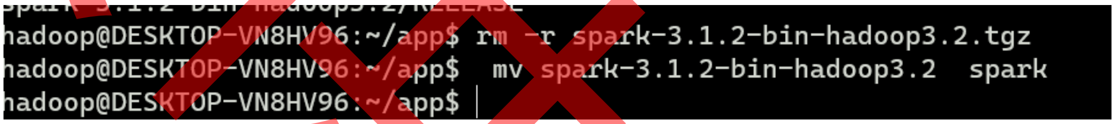

```
hadoop@DESKTOP-VN8HV96:~/app$ wget http://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-had
oop3.2.tgz
--2023-11-08 11:00:58--  http://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 228834641 (218M) [application/x-gzip]
Saving to: ‘spark-3.1.2-bin-hadoop3.2.tgz’

spark-3.1.2-bin-hadoop3.2   0%[                                      ]   1.21M  93.2KB/s    eta 40m 17s
```

解压



```
spark-3.1.2-bin-hadoop3.2/jars/audience-annotations-0.5.0.jar
spark-3.1.2-bin-hadoop3.2/jars/arrow-vector-2.0.0.jar
spark-3.1.2-bin-hadoop3.2/jars/arrow-memory-netty-2.0.0.jar
spark-3.1.2-bin-hadoop3.2/jars/arrow-memory-core-2.0.0.jar
spark-3.1.2-bin-hadoop3.2/jars/arrow-format-2.0.0.jar
spark-3.1.2-bin-hadoop3.2/jars/arpack_combined_all-0.1.jar
spark-3.1.2-bin-hadoop3.2/jars/aopalliance-repackaged-2.6.1.jar
spark-3.1.2-bin-hadoop3.2/jars/aopalliance-1.0.jar
spark-3.1.2-bin-hadoop3.2/jars/antlr4-runtime-4.8-1.jar
spark-3.1.2-bin-hadoop3.2/jars/antlr-runtime-3.5.2.jar
spark-3.1.2-bin-hadoop3.2/jars/algebra_2.12-2.0.0-M2.jar
spark-3.1.2-bin-hadoop3.2/jars/aircompressor-0.10.jar
spark-3.1.2-bin-hadoop3.2/jars/activation-1.1.1.jar
spark-3.1.2-bin-hadoop3.2/jars/accessors-smart-1.2.jar
spark-3.1.2-bin-hadoop3.2/jars/ST4-4.0.4.jar
spark-3.1.2-bin-hadoop3.2/jars/RoaringBitmap-0.9.0.jar
spark-3.1.2-bin-hadoop3.2/jars/JTransforms-3.1.jar
spark-3.1.2-bin-hadoop3.2/jars/JLargeArrays-1.5.jar
spark-3.1.2-bin-hadoop3.2/jars/HikariCP-2.5.1.jar
spark-3.1.2-bin-hadoop3.2/RELEASE
```

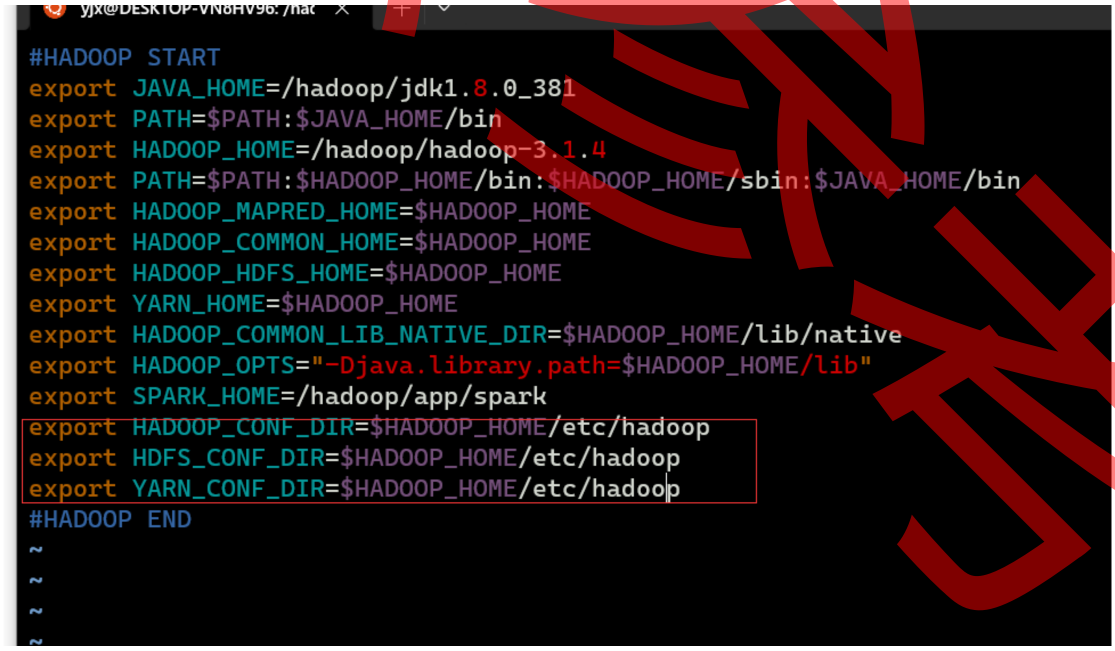
删除安装文件、修改文件名称



```
hadoop@DESKTOP-VN8HV96:~/app$ rm -r spark-3.1.2-bin-hadoop3.2.tgz
hadoop@DESKTOP-VN8HV96:~/app$  mv spark-3.1.2-bin-hadoop3.2  spark
hadoop@DESKTOP-VN8HV96:~/app$
```

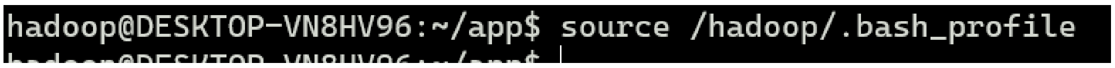
2.2 配置 Hadoop 环境变量

在 Yarn 上运行 Spark 需要配置 HADOOP_CONF_DIR、 YARN_CONF_DIR 和 HDFS_CONF_DIR 环境变量



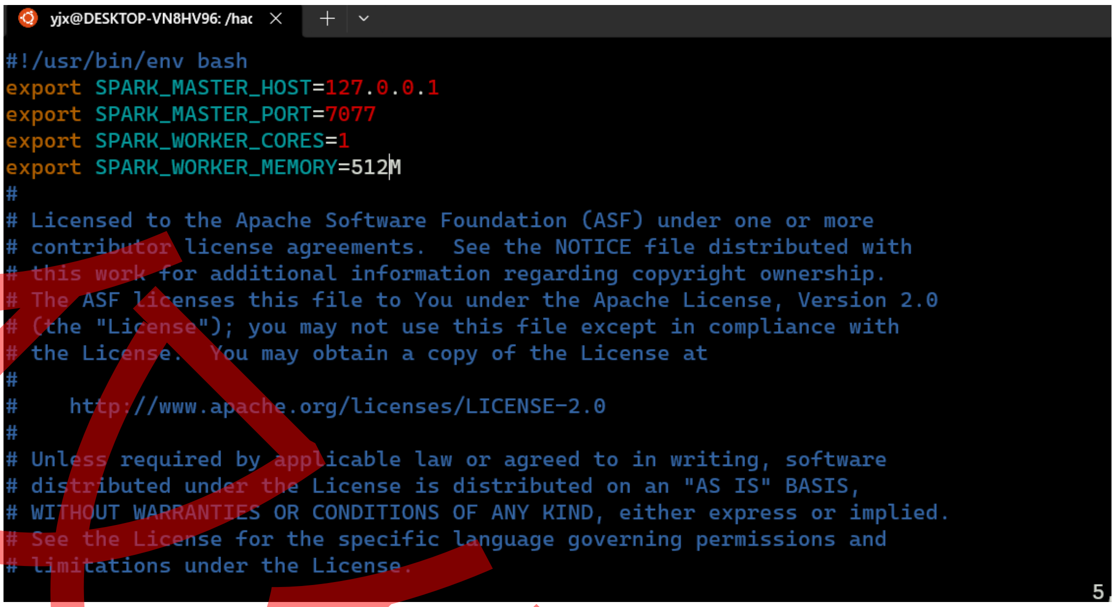
```
#HADOOP START
export JAVA_HOME=/hadoop/jdk1.8.0_381
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=/hadoop/hadoop-3.1.4
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$JAVA_HOME/bin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
export SPARK_HOME=/hadoop/app/spark
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HDFS_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
#HADOOP END
~
~
~
~
```

激活环境变量



```
hadoop@DESKTOP-VN8HV96:~/app$ source /hadoop/.bash_profile
hadoop@DESKTOP-VN8HV96:~/app$
```

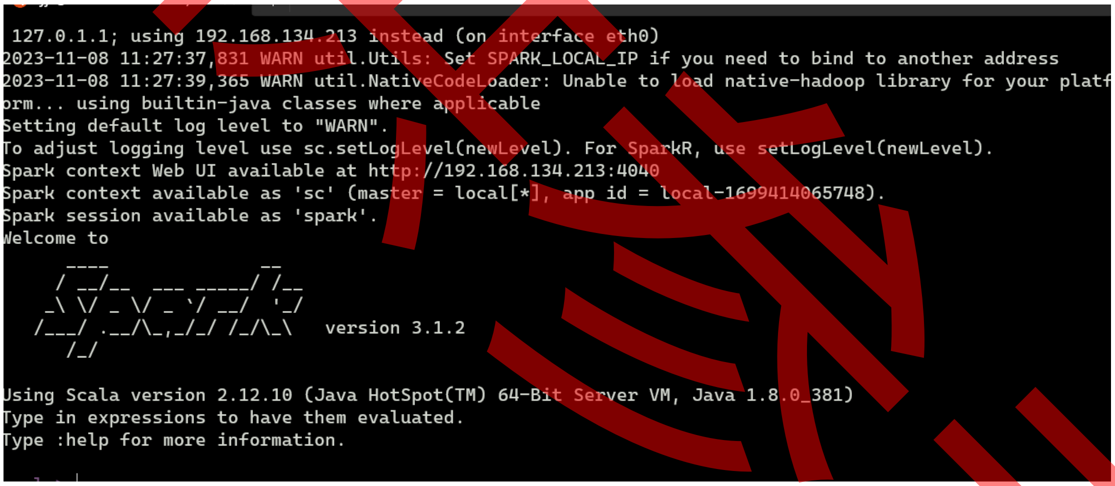
2.3 修改配置文件



```
#!/usr/bin/env bash
export SPARK_MASTER_HOST=127.0.0.1
export SPARK_MASTER_PORT=7077
export SPARK_WORKER_CORES=1
export SPARK_WORKER_MEMORY=512M
#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

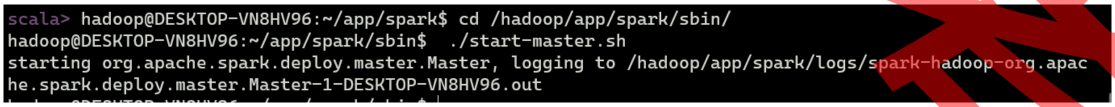
3. Spark 的 启动

3.1 进入 spark-shell



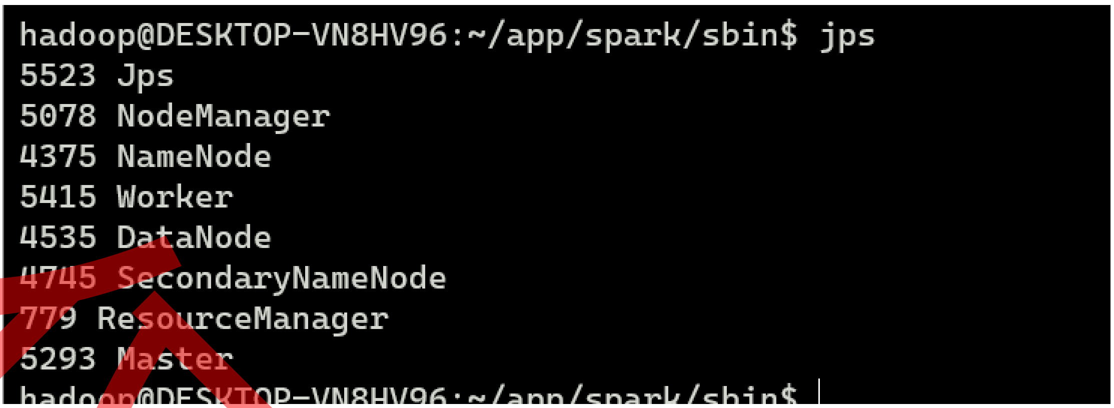
```
 127.0.1.1; using 192.168.134.213 instead (on interface eth0)
2023-11-08 11:27:37,831 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
2023-11-08 11:27:39,365 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platf
orm... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.134.213:4040
Spark context available as 'sc' (master = local[*], app id = local-1699414065748).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

Using Scala version 2.12.10 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_381)
Type in expressions to have them evaluated.
Type :help for more information.
```

3.2 启动 spark



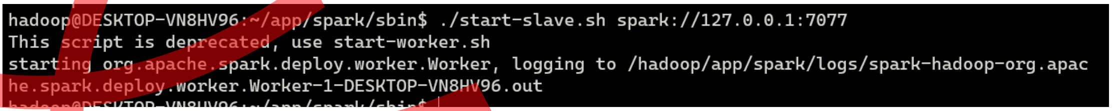
```
scala> hadoop@DESKTOP-VN8HV96:~/app/spark$ cd /hadoop/app/spark/sbin/
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$  ./start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /hadoop/app/spark/logs/spark-hadoop-org.apac
he.spark.deploy.master.Master-1-DESKTOP-VN8HV96.out
```

(1)查看 master 是否启动



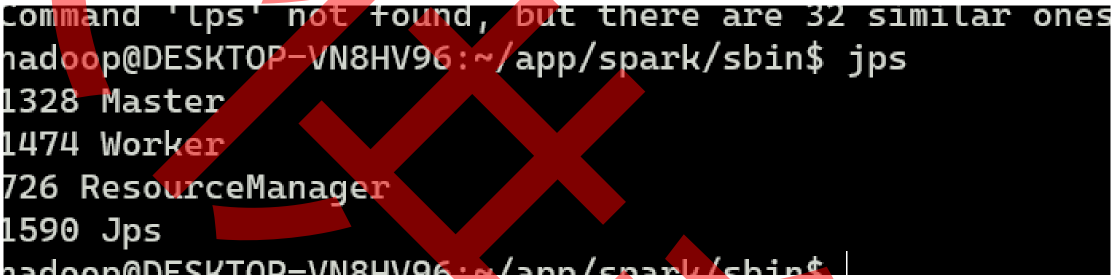
```
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$ jps
5523 Jps
5078 NodeManager
4375 NameNode
5415 Worker
4535 DataNode
4745 SecondaryNameNode
779 ResourceManager
5293 Master
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$
```

(2) 启动 slave



```
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$ ./start-slave.sh spark://127.0.0.1:7077
This script is deprecated, use start-worker.sh
starting org.apache.spark.deploy.worker.Worker, logging to /hadoop/app/spark/logs/spark-hadoop-org.apac
he.spark.deploy.worker.Worker-1-DESKTOP-VN8HV96.out
```

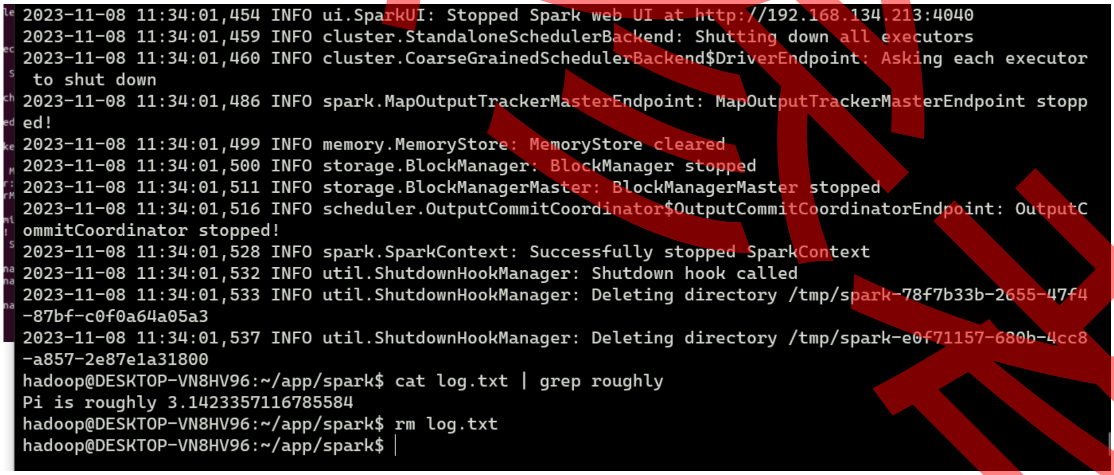
(3) 查看 woker 是否启动



```
Command 'lps' not found, but there are 32 similar ones
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$ jps
1328 Master
1474 Worker
726 ResourceManager
1590 Jps
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$
```

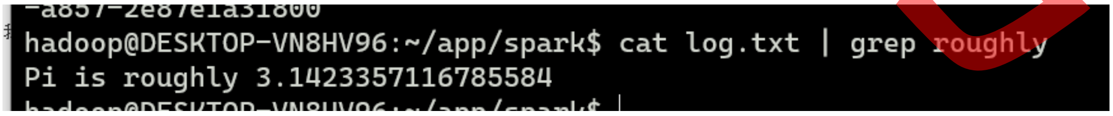
3.3 验证 Spark

（1）运行 pi（π）的实例



```
2023-11-08 11:34:01,454 INFO ui.SparkUI: Stopped Spark web UI at http://192.168.134.213:4040
2023-11-08 11:34:01,459 INFO cluster.StandaloneSchedulerBackend: Shutting down all executors
2023-11-08 11:34:01,460 INFO cluster.CoarseGrainedSchedulerBackend$DriverEndpoint: Asking each executor
 to shut down
2023-11-08 11:34:01,486 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopp
ed!
2023-11-08 11:34:01,499 INFO memory.MemoryStore: MemoryStore cleared
2023-11-08 11:34:01,500 INFO storage.BlockManager: BlockManager stopped
2023-11-08 11:34:01,511 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
2023-11-08 11:34:01,516 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputC
ommitCoordinator stopped!
2023-11-08 11:34:01,528 INFO spark.SparkContext: Successfully stopped SparkContext
2023-11-08 11:34:01,532 INFO util.ShutdownHookManager: Shutdown hook called
2023-11-08 11:34:01,533 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-78f7b33b-2655-47f4
-87bf-c0f0a64a05a3
2023-11-08 11:34:01,537 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-e0f71157-680b-4cc8
-a857-2e87e1a31800
hadoop@DESKTOP-VN8HV96:~/app/spark$ cat log.txt | grep roughly
Pi is roughly 3.1423357116785584
hadoop@DESKTOP-VN8HV96:~/app/spark$ rm log.txt
hadoop@DESKTOP-VN8HV96:~/app/spark$
```

（2）将结果输入到 log.txt 中输出



```
-a857-2e87e1a31800
hadoop@DESKTOP-VN8HV96:~/app/spark$ cat log.txt | grep roughly
Pi is roughly 3.1423357116785584
hadoop@DESKTOP-VN8HV96:~/app/spark$
```

九、 总结及心得体会：

学会了如何配置 Hadoop 的单节点伪分布式环境，包括创建用户、设置权限等。

掌握了 Hadoop 基本命令，可以进行文件操作、查看日志、启动和停止服务等。学会安装和配置 SSH，方便后续搭建集群，能够实现在多台机器之间无密码登录。也实现了 Hadoop 和 Spark 安装，后续进行了 wordcount 实验 ，有助于理解 Hadoop 的基本工作原理和 MapReduce 编程模型。

十、对本实验过程及方法、手段的改进建议：

可以在实验中引入更多的实际应用场景，例如基于 Hadoop 和 Spark 的大数据分析任务，并且流程之间的衔接性不强

报告评分：

指导教师签字：

电 子 科 技 大 学
实　验　报　告

学生姓名： ***　**学号：** ***　**指导教师：** ***

实验地点：第二教学楼 110　　**实验时间：2023.11.8**

一、 实验名称：MapReduce 与 Spark 内存计算的性能对比

二、 实验学时：2 学时

三、 实验目的：

1. 比较 MapReduce 和 Spark 框架的性能： 通过对同一 WordCount 分析程序在 Hadoop MapReduce 和 Spark 框架下的运行，对比两者的性能表现，包括运行时间、资源利用率等方面的指标。
2. 实践 MapReduce 和 Spark 编程： 通过实际操作，学习和熟悉 MapReduce 和 Spark 编程模型，了解它们在大数据处理中的应用。
3. 应用于中文文本的词频统计： 选作部分旨在拓展实验，尝试对中文文本进行词频统计，以了解 MapReduce 和 Spark 在处理非英文文本时的性能和适用性。

四、 实验原理

1. MapReduce 框架： MapReduce 是一种分布式计算框架，它将大规模数据处理任务分为 Map 和 Reduce 两个阶段。Map 阶段负责数据切分和初步处理，Reduce 阶段进行最终的结果合并。Hadoop 是一个实现了 MapReduce 的开源框架。
2. Spark 框架： Spark 是一个快速、通用、可扩展的大数据处理引擎，它支持丰富的数据处理任务。Spark 中的基本抽象是弹性分布式数据集（RDD），它允许并行计算，并提供高层次的 API（如 Spark SQL、Spark Streaming 等）。
3. WordCount 程序： WordCount 是一个经典的大数据处理示例，它统计文本中每个单词出现的次数。该程序在 MapReduce 中通过 Map 和 Reduce 任务实现，而在 Spark 中可以使用 RDD 的转换和操作完成。
4. 性能比较： 通过准备相同大小的文本文件，在两个框架下运行相同的 WordCount 程序，记录运行时间、资源利用情况等指标。比较两者的性能，包括执行速度、扩展性、容错性等方面。

五、 实验内容

使用 Hadoop MapReduce、Spark 框架分别运行 wordcount 分析程序，来对 MapReduce 和 Spark 的性能进行对比。

1） 准备 600-700M 左右的 word.txt 文件
2） 依照步骤完成 MapReduce 代码和运行
3） 依照步骤完成基于 Spark 的代码和运行
4） 比较 Hadoop MapReduce、Spark 框架的运行效率
5） 选作：尝试使用中文文本进行词频统计，并比较性能

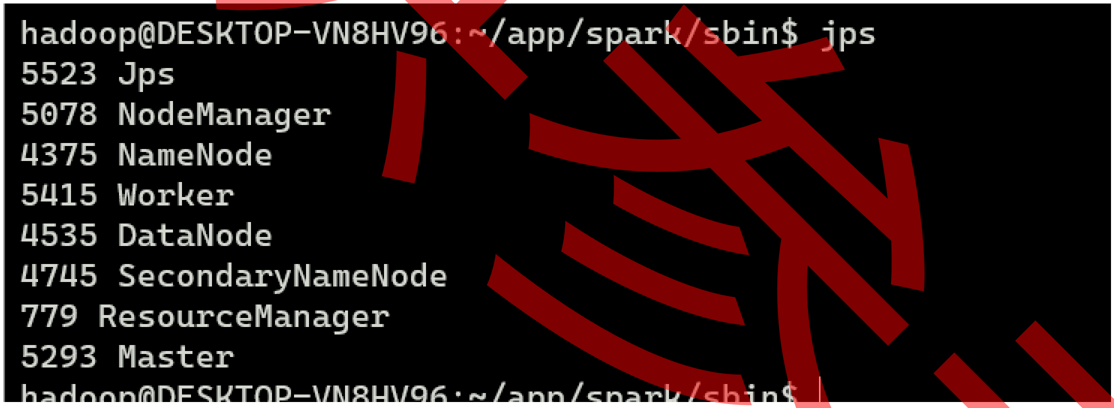
六、 实验设备及环境

- ubuntu-18.04 及以上环境
- jdk-8u261-linux-x64.tar.gz
- Hadoop 3.1.4
- spark-3.0.1-bin-hadoop3.2.tgz

七、 实验步骤

1.实验前准备

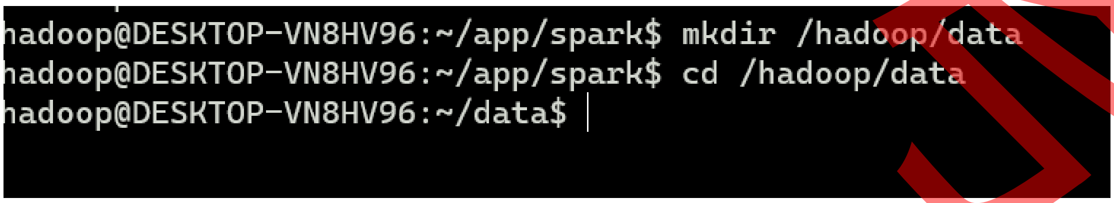
启动 ssh、启动柜 Hadoop、启动 spark



```
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$ jps
5523 Jps
5078 NodeManager
4375 NameNode
5415 Worker
4535 DataNode
4745 SecondaryNameNode
779 ResourceManager
5293 Master
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$
```

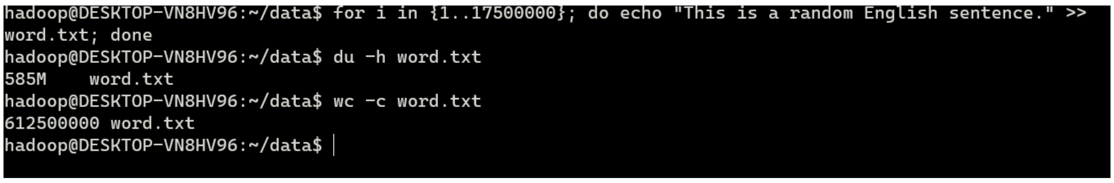
2.将本次实验的数据文件上传到 HDFS 文件系统

2.1 可以建立一个/hadoop/data 目录。

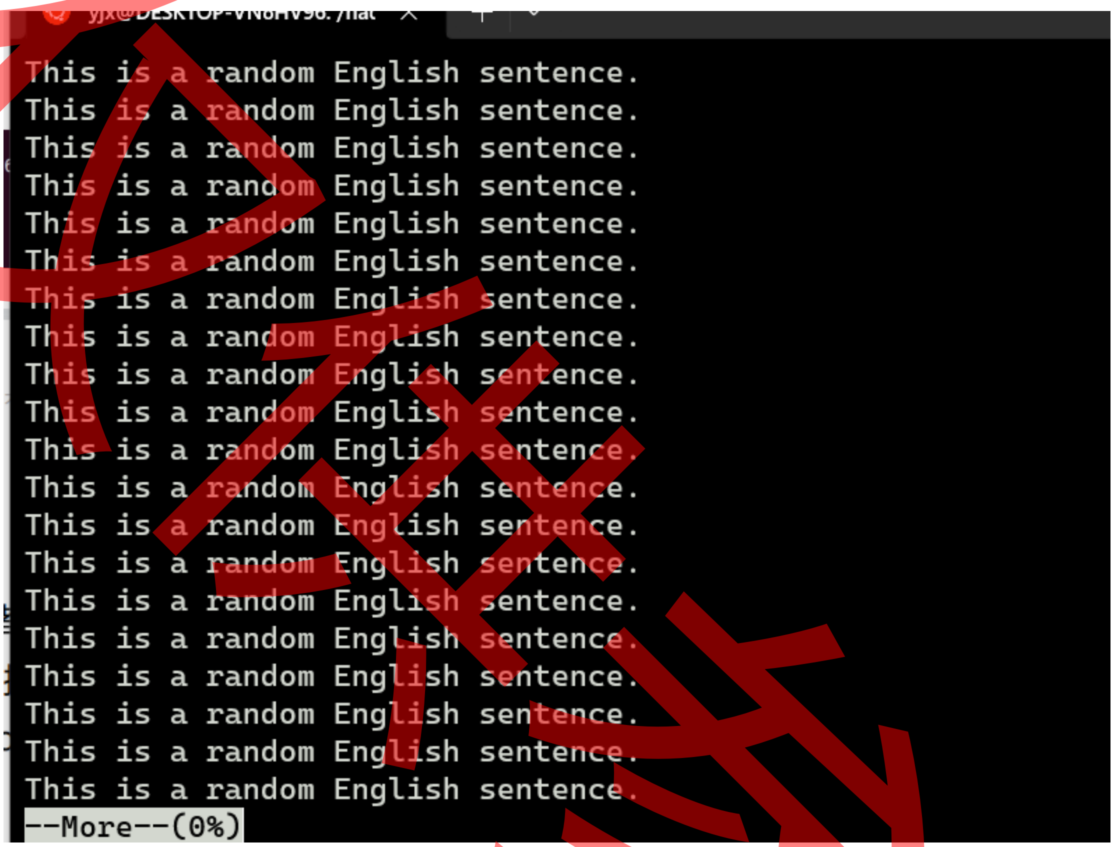


```
hadoop@DESKTOP-VN8HV96:~/app/spark$ mkdir /hadoop/data
hadoop@DESKTOP-VN8HV96:~/app/spark$ cd /hadoop/data
hadoop@DESKTOP-VN8HV96:~/data$
```

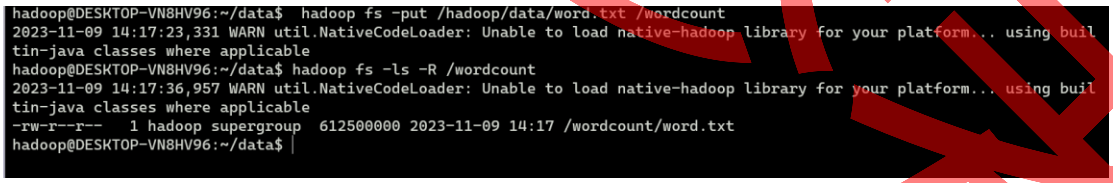
2.2 将 word.txt 上传至该目录下。并查看该目录下将 word.txt 上传至该目录下。并查看该目录下是否有了 word.txt 文件是否有了 word.txt 文件


```
hadoop@DESKTOP-VN8HV96:~/data$ for i in {1..17500000}; do echo "This is a random English sentence." >>
word.txt; done
hadoop@DESKTOP-VN8HV96:~/data$ du -h word.txt
585M    word.txt
hadoop@DESKTOP-VN8HV96:~/data$ wc -c word.txt
612500000 word.txt
hadoop@DESKTOP-VN8HV96:~/data$ |
```

2.3　查看字符集的内容


```
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
This is a random English sentence.
--More--(0%)
```

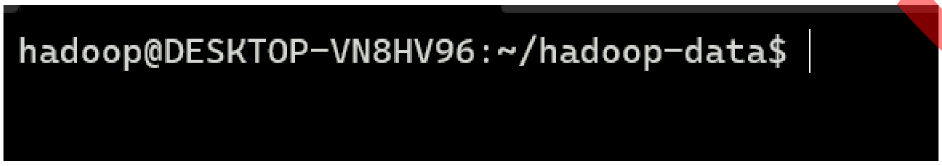
3.将数据文件上传到 HDFS 文件系统


```
hadoop@DESKTOP-VN8HV96:~/data$  hadoop fs -put /hadoop/data/word.txt /wordcount
2023-11-09 14:17:23,331 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using buil
tin-java classes where applicable
hadoop@DESKTOP-VN8HV96:~/data$ hadoop fs -ls -R /wordcount
2023-11-09 14:17:36,957 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using buil
tin-java classes where applicable
-rw-r--r--   1 hadoop supergroup  612500000 2023-11-09 14:17 /wordcount/word.txt
hadoop@DESKTOP-VN8HV96:~/data$ |
```

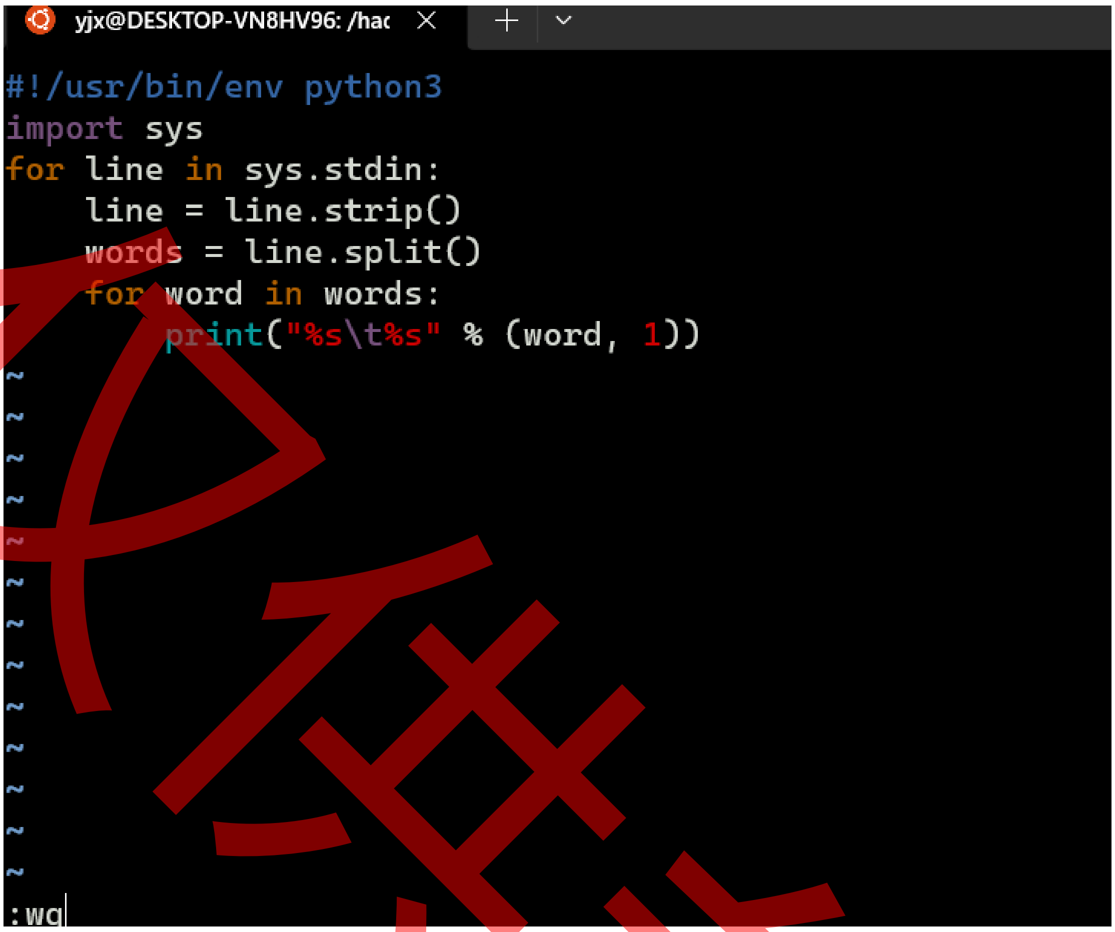
(注：警告可忽略)

4.MapReduce 实现 WordCount 实例（Python）

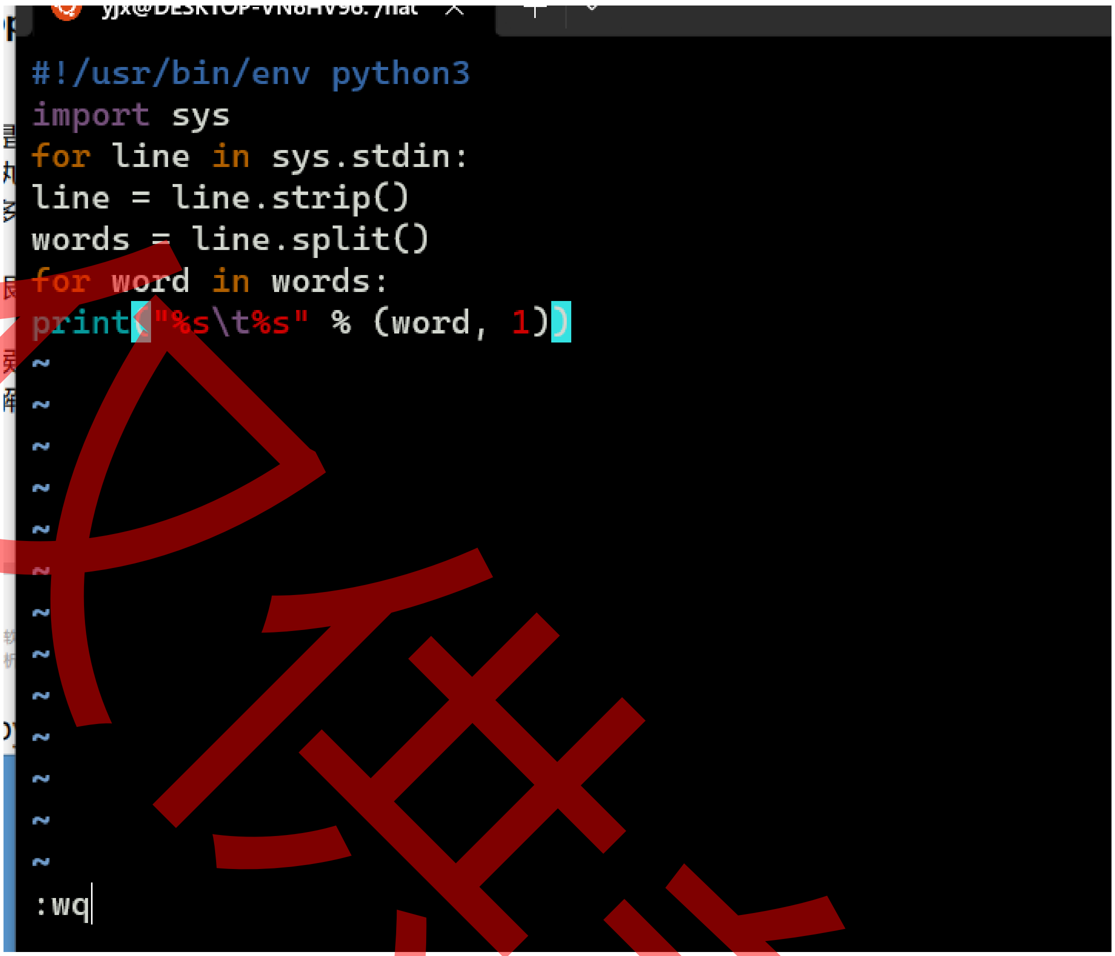
4.1　创建/hadoop/data/mapreduce，并进入到/hadoop/data/mapreduce 目录下


```
hadoop@DESKTOP-VN8HV96:~/hadoop-data$ |
```

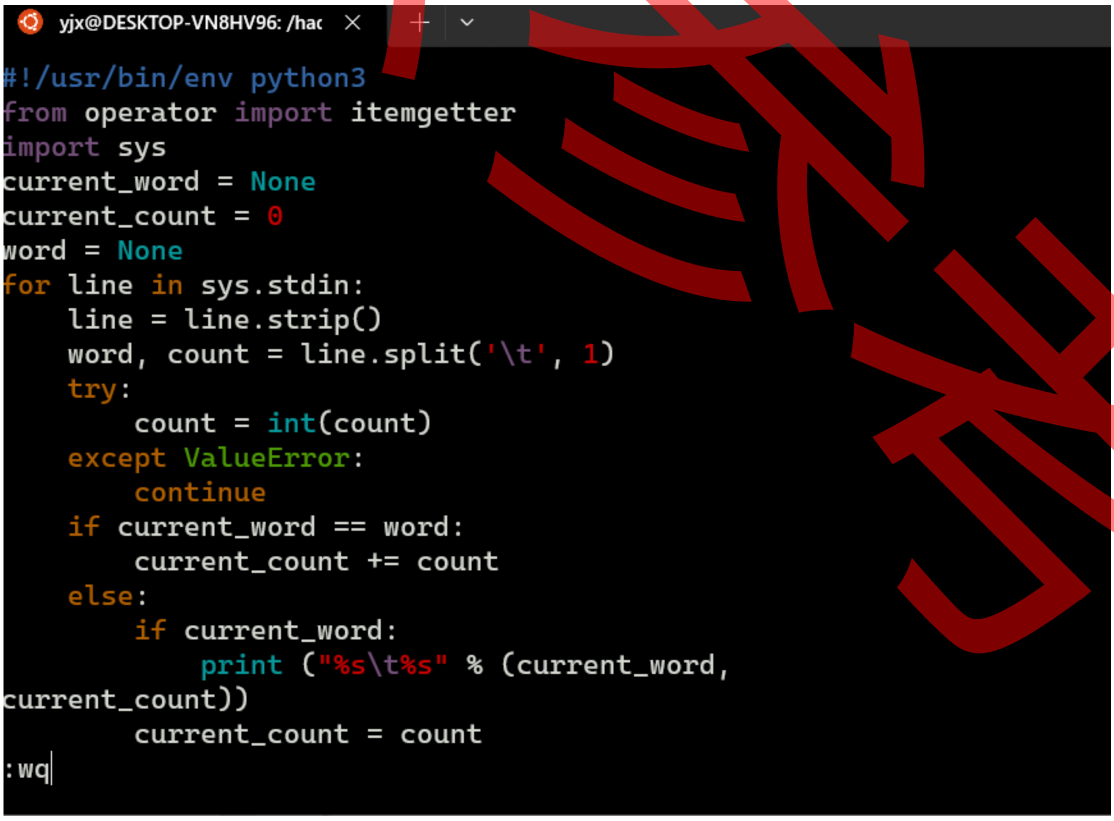
4.2　在这个目录下首先编写 MapReduce WordCount 代码



```
#!/usr/bin/env python3
import sys
for line in sys.stdin:
    line = line.strip()
    words = line.split()
    for word in words:
        print("%s\t%s" % (word, 1))
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

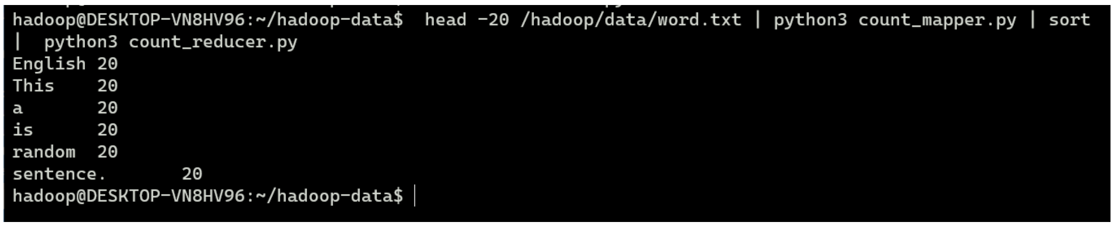


```
#!/usr/bin/env python3
import sys
for line in sys.stdin:
line = line.strip()
words = line.split()
for word in words:
print("%s\t%s" % (word, 1))
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```



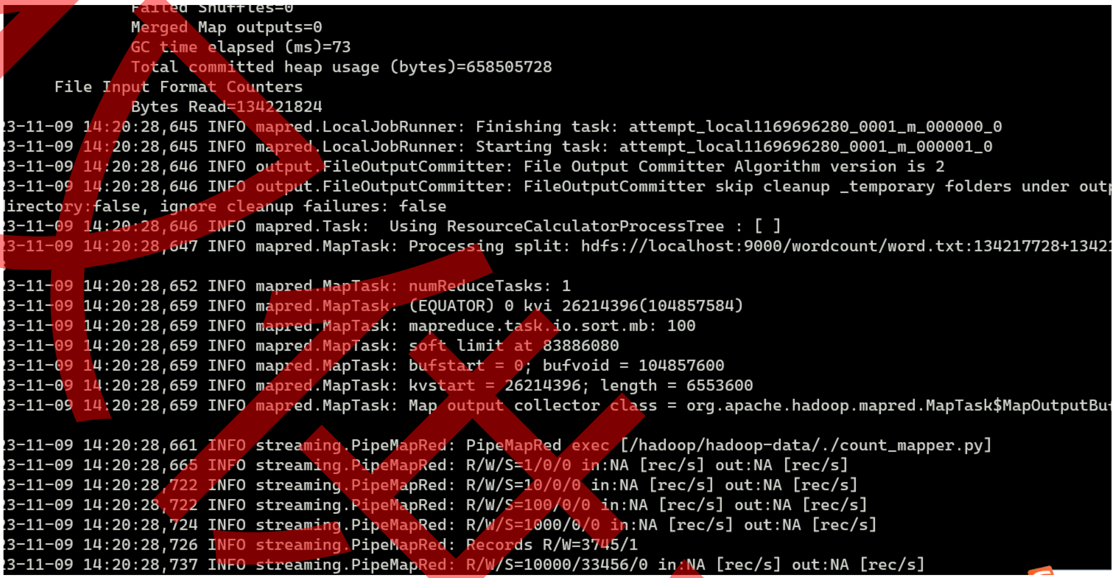
```
#!/usr/bin/env python3
from operator import itemgetter
import sys
current_word = None
current_count = 0
word = None
for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t', 1)
    try:
        count = int(count)
    except ValueError:
        continue
    if current_word == word:
        current_count += count
    else:
        if current_word:
            print ("%s\t%s" % (current_word,
current_count))
        current_count = count
:wq
```

4.3　在本地测试一下 map 和 reduce

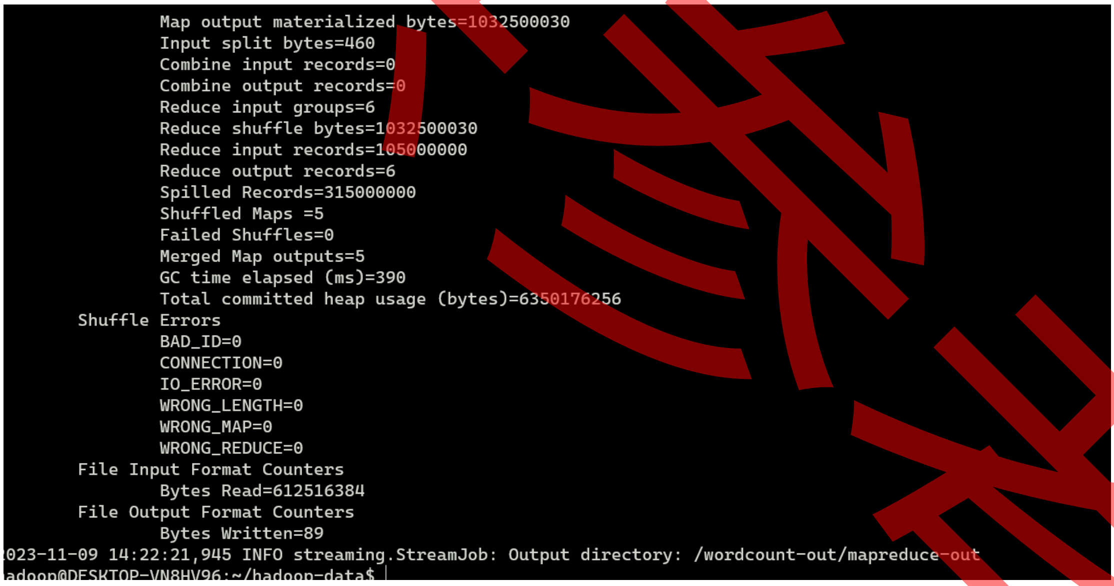


```
hadoop@DESKTOP-VN8HV96:~/hadoop-data$  head -20 /hadoop/data/word.txt | python3 count_mapper.py | sort
|  python3 count_reducer.py
English 20
This    20
a       20
is      20
random  20
sentence.       20
hadoop@DESKTOP-VN8HV96:~/hadoop-data$
```

4.4　运行该实例

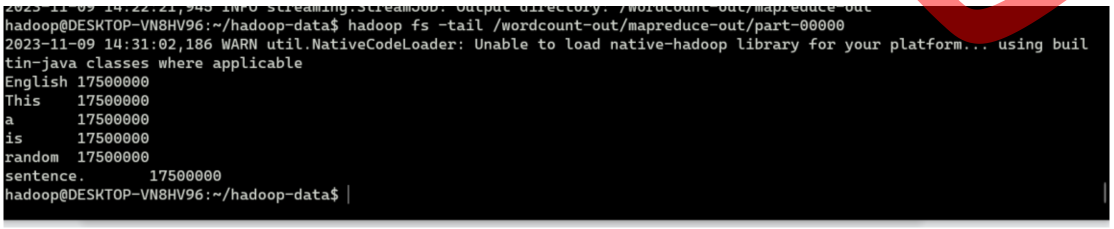


```
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=73
                Total committed heap usage (bytes)=658505728
        File Input Format Counters
                Bytes Read=134221824
23-11-09 14:20:28,645 INFO mapred.LocalJobRunner: Finishing task: attempt_local1169696280_0001_m_000000_0
23-11-09 14:20:28,645 INFO mapred.LocalJobRunner: Starting task: attempt_local1169696280_0001_m_000001_0
23-11-09 14:20:28,646 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
23-11-09 14:20:28,646 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under outp
irectory:false, ignore cleanup failures: false
23-11-09 14:20:28,646 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
23-11-09 14:20:28,647 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/wordcount/word.txt:134217728+13421

23-11-09 14:20:28,652 INFO mapred.MapTask: numReduceTasks: 1
23-11-09 14:20:28,659 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
23-11-09 14:20:28,659 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
23-11-09 14:20:28,659 INFO mapred.MapTask: soft limit at 83886080
23-11-09 14:20:28,659 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
23-11-09 14:20:28,659 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
23-11-09 14:20:28,659 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBu

23-11-09 14:20:28,661 INFO streaming.PipeMapRed: PipeMapRed exec [/hadoop/hadoop-data/./count_mapper.py]
23-11-09 14:20:28,665 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/s] out:NA [rec/s]
23-11-09 14:20:28,722 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [rec/s] out:NA [rec/s]
23-11-09 14:20:28,722 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [rec/s] out:NA [rec/s]
23-11-09 14:20:28,724 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [rec/s] out:NA [rec/s]
23-11-09 14:20:28,726 INFO streaming.PipeMapRed: Records R/W=3745/1
23-11-09 14:20:28,737 INFO streaming.PipeMapRed: R/W/S=10000/33456/0 in:NA [rec/s] out:NA [rec/s]
```



```
                Map output materialized bytes=1032500030
                Input split bytes=460
                Combine input records=0
                Combine output records=0
                Reduce input groups=6
                Reduce shuffle bytes=1032500030
                Reduce input records=105000000
                Reduce output records=6
                Spilled Records=315000000
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=390
                Total committed heap usage (bytes)=6350176256
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=612516384
        File Output Format Counters
                Bytes Written=89
2023-11-09 14:22:21,945 INFO streaming.StreamJob: Output directory: /wordcount-out/mapreduce-out
adoop@DESKTOP-VN8HV96:~/hadoop-data$
```

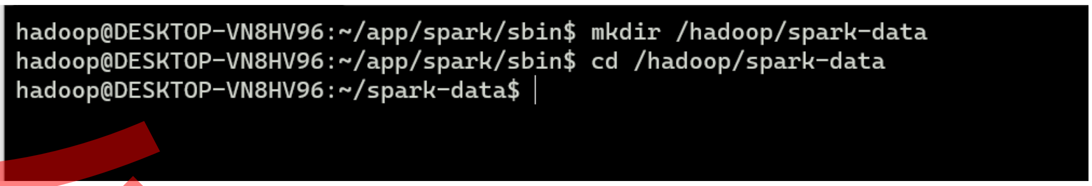
4.5　运行该实例



```
2023-11-09 14:22:21,945 INFO streaming.StreamJob: Output directory: /wordcount-out/mapreduce-out
hadoop@DESKTOP-VN8HV96:~/hadoop-data$ hadoop fs -tail /wordcount-out/mapreduce-out/part-00000
2023-11-09 14:31:02,186 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using buil
tin-java classes where applicable
English 17500000
This    17500000
a       17500000
is      17500000
random  17500000
sentence.       17500000
hadoop@DESKTOP-VN8HV96:~/hadoop-data$
```

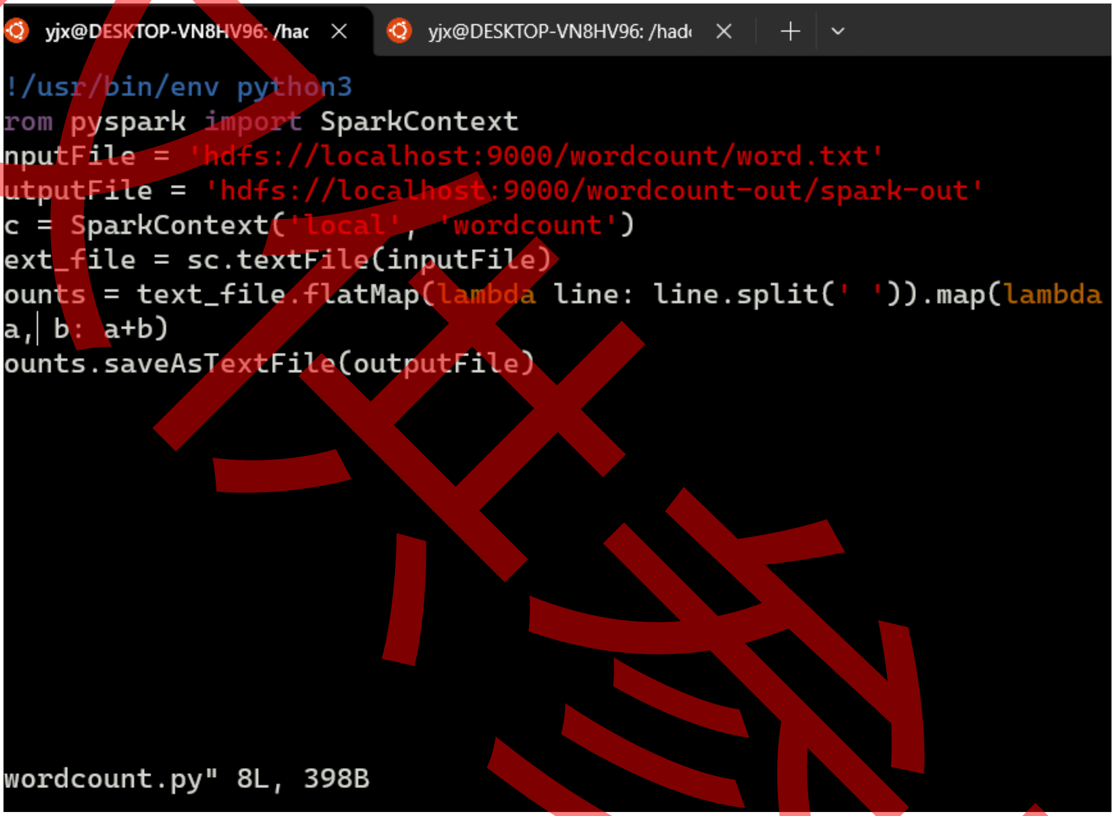
5.Spark 实现 WordCount 实例

5.1　创建/hadoop/data/spark，并进入到/hadoop/data/spark 目录下



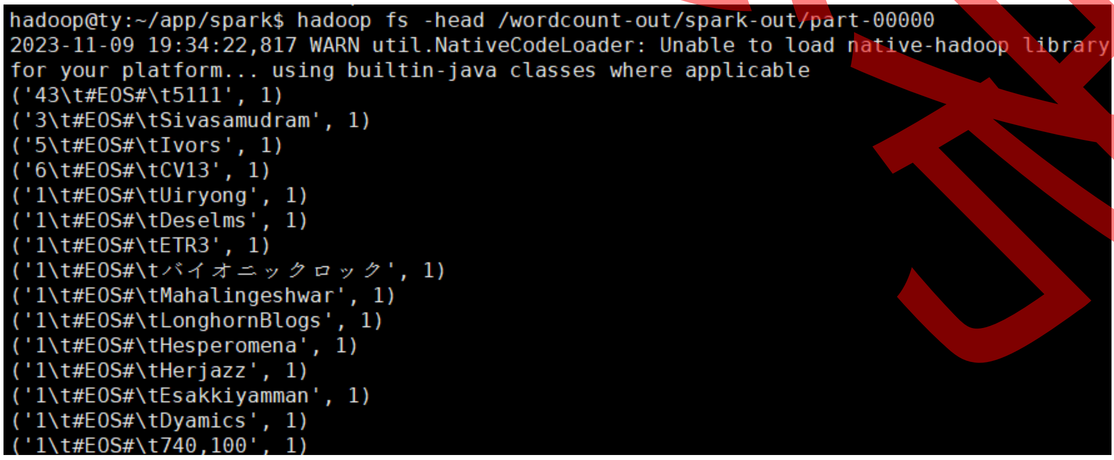
```
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$ mkdir /hadoop/spark-data
hadoop@DESKTOP-VN8HV96:~/app/spark/sbin$ cd /hadoop/spark-data
hadoop@DESKTOP-VN8HV96:~/spark-data$
```

5.2　编写 Spark WordCount 代码



```
!/usr/bin/env python3
rom pyspark import SparkContext
nputFile = 'hdfs://localhost:9000/wordcount/word.txt'
utputFile = 'hdfs://localhost:9000/wordcount-out/spark-out'
c = SparkContext('local', 'wordcount')
ext_file = sc.textFile(inputFile)
ounts = text_file.flatMap(lambda line: line.split(' ')).map(lambda
a, b: a+b)
ounts.saveAsTextFile(outputFile)




wordcount.py" 8L, 398B
```

5.3　运行该实例查看运行结果

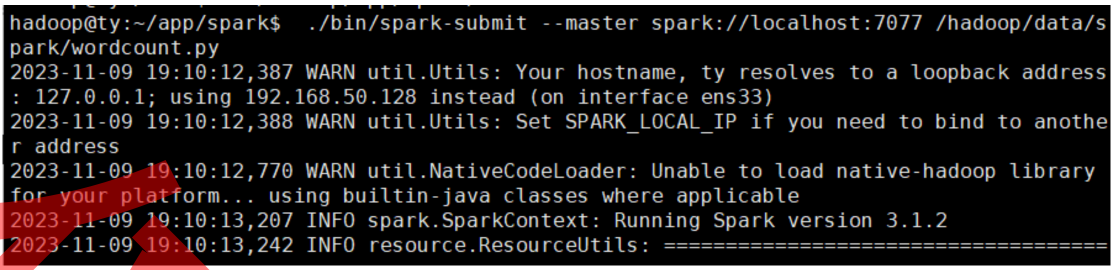


```
hadoop@ty:~/app/spark$ hadoop fs -head /wordcount-out/spark-out/part-00000
2023-11-09 19:34:22,817 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
('43\t#EOS#\t5111', 1)
('3\t#EOS#\tSivasamudram', 1)
('5\t#EOS#\tIvors', 1)
('6\t#EOS#\tCV13', 1)
('1\t#EOS#\tUiryong', 1)
('1\t#EOS#\tDeselms', 1)
('1\t#EOS#\tETR3', 1)
('1\t#EOS#\tバイオニックロック', 1)
('1\t#EOS#\tMahalingeshwar', 1)
('1\t#EOS#\tLonghornBlogs', 1)
('1\t#EOS#\tHesperomena', 1)
('1\t#EOS#\tHerjazz', 1)
('1\t#EOS#\tEsakkiyamman', 1)
('1\t#EOS#\tDyamics', 1)
('1\t#EOS#\t740,100', 1)
```

八、　实验结果与分析（含重要数据结果分析或核心代码流程分析）

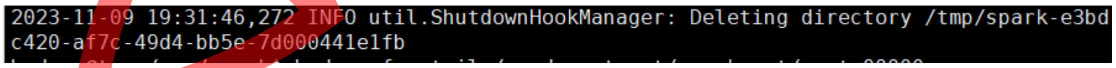
1.Spark 计算时间

1.1Spark 启动时间



```
hadoop@ty:~/app/spark$  ./bin/spark-submit --master spark://localhost:7077 /hadoop/data/s
park/wordcount.py
2023-11-09 19:10:12,387 WARN util.Utils: Your hostname, ty resolves to a loopback address
: 127.0.0.1; using 192.168.50.128 instead (on interface ens33)
2023-11-09 19:10:12,388 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to anothe
r address
2023-11-09 19:10:12,770 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
2023-11-09 19:10:13,207 INFO spark.SparkContext: Running Spark version 3.1.2
2023-11-09 19:10:13,242 INFO resource.ResourceUtils: ====================================
```

1.2 Spark 结束时间

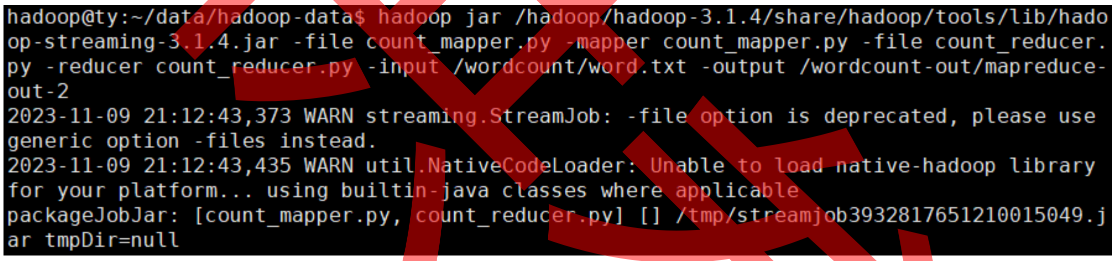


```
2023-11-09 19:31:46,272 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-e3bd
c420-af7c-49d4-bb5e-7d000441e1fb
```

1.3 Spark 总耗时：20m34s=1234s

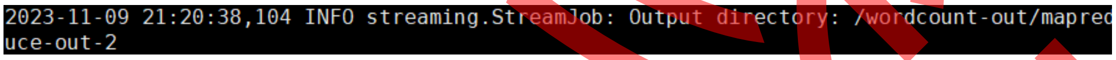
2. Hadoop 计算时间

2.1 Hadoop 启动时间



```
hadoop@ty:~/data/hadoop-data$ hadoop jar /hadoop/hadoop-3.1.4/share/hadoop/tools/lib/hado
op-streaming-3.1.4.jar -file count_mapper.py -mapper count_mapper.py -file count_reducer.
py -reducer count_reducer.py -input /wordcount/word.txt -output /wordcount-out/mapreduce-
out-2
2023-11-09 21:12:43,373 WARN streaming.StreamJob: -file option is deprecated, please use
generic option -files instead.
2023-11-09 21:12:43,435 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
packageJobJar: [count_mapper.py, count_reducer.py] [] /tmp/streamjob3932817651210015049.j
ar tmpDir=null
```

2.2 Hadoop 结束时间



```
2023-11-09 21:20:38,104 INFO streaming.StreamJob: Output directory: /wordcount-out/mapred
uce-out-2
```

2.3 Hadoop 总耗时：7m55s=475s

3.对比

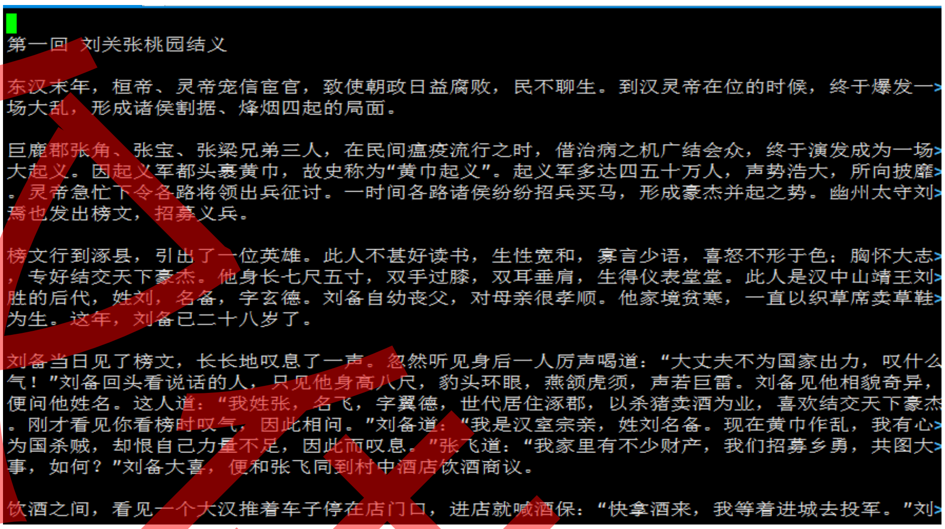
得到 Hadoop 总耗时 475s < Spark 总耗时 1234s，故 Hadoop 计算速度更快。

补充：加分内容：Word.txt 使用中文文本，在 hadoop 和 Spark 进行计算之前进行中文分词再做计算

1. word.txt 使用中文文本并进行中文分词：

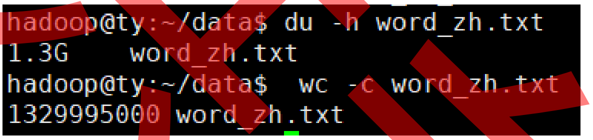
1.1 文本文件信息

文本文件来自《三国演义》:https://sanguo.5000yan.com/xiazai/sanguoyanyi.zip



```
第一回 刘关张桃园结义

东汉末年，桓帝、灵帝宠信宦官，致使朝政日益腐败，民不聊生。到汉灵帝在位的时候，终于爆发一>
场大乱，形成诸侯割据、烽烟四起的局面。

巨鹿郡张角、张宝、张梁兄弟三人，在民间瘟疫流行之时，借治病之机广结会众，终于演发成为一场>
大起义。因起义军都头裹黄巾，故史称为"黄巾起义"。起义军多达四五十万人，声势浩大，所向披靡>
。灵帝急忙下令各路将领出兵征讨。一时间各路诸侯纷纷招兵买马，形成豪杰并起之势。幽州太守刘>
焉也发出榜文，招募义兵。

榜文行到涿县，引出了一位英雄。此人不甚好读书，生性宽和，寡言少语，喜怒不形于色；胸怀大志>
，专好结交天下豪杰。他身长七尺五寸，双手过膝，双耳垂肩，生得仪表堂堂。此人是汉中山靖王刘>
胜的后代，姓刘，名备，字玄德。刘备自幼丧父，对母亲很孝顺。他家境贫寒，一直以织草席卖草鞋>
为生。这年，刘备已二十八岁了。

刘备当日见了榜文，长长地叹息了一声。忽然听见身后一人厉声喝道："大丈夫不为国家出力，叹什么
气！"刘备回头看说话的人，只见他身高八尺，豹头环眼，燕颔虎须，声若巨雷。刘备见他相貌奇异，
便问他姓名。这人道："我姓张，名飞，字翼德，世代居住涿郡，以杀猪卖酒为业，喜欢结交天下豪杰
。刚才看见你看榜时叹气，因此相问。"刘备道："我是汉室宗亲，姓刘名备。现在黄巾作乱，我有心>
为国杀贼，却恨自己力量不足，因此而叹息。"张飞道："我家里有不少财产，我们招募乡勇，共图大>
事，如何？"刘备大喜，便和张飞同到村中酒店饮酒商议。

饮酒之间，看见一个大汉推着车子停在店门口，进店就喊酒保："快拿酒来，我等着进城去投军。"刘>
```

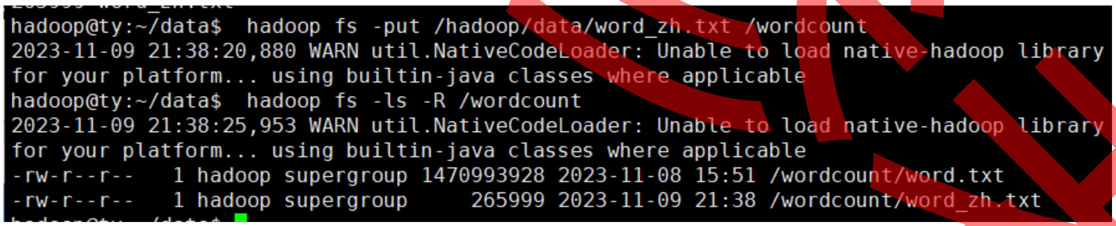
查看数据集大小和数据集字符数：



```
hadoop@ty:~/data$ du -h word_zh.txt
1.3G    word_zh.txt
hadoop@ty:~/data$  wc -c word_zh.txt
1329995000 word_zh.txt
```

2. 上传至 HDFS 文件系统

2.1 上传至 HDFS 文件系统



```
hadoop@ty:~/data$  hadoop fs -put /hadoop/data/word_zh.txt /wordcount
2023-11-09 21:38:20,880 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
hadoop@ty:~/data$  hadoop fs -ls -R /wordcount
2023-11-09 21:38:25,953 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
-rw-r--r--   1 hadoop supergroup 1470993928 2023-11-08 15:51 /wordcount/word.txt
-rw-r--r--   1 hadoop supergroup     265999 2023-11-09 21:38 /wordcount/word_zh.txt
```

2.2MapReduce 实现 wordcount 实例

（1）编写 Map 阶段代码：count_mapper_zh.py，使用 jieba 分词器进行中文分词

```
#!/usr/bin/env python3

import sys

import jieba



for line in sys.stdin:
    line = line.strip()
    words = jieba.cut(line, cut_all=False)  # 进行中文分词
    for word in words:
        print("%s\t%s" % (word, 1))
```

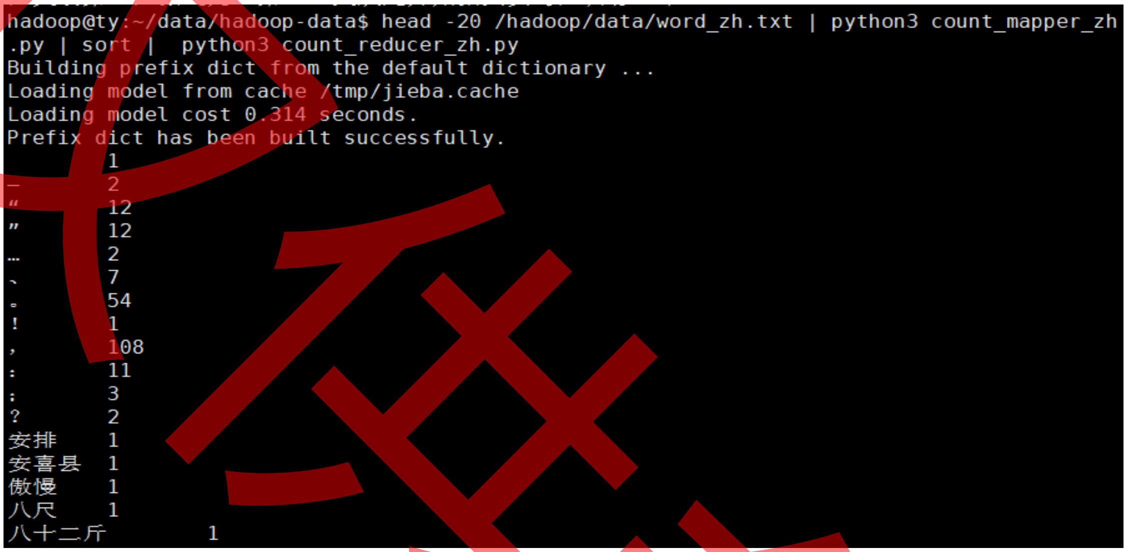
代码 3 count_mapper_zh.py 代码

（2）编写 Reducer 阶段代码：count_reducer_zh.py

```
#!/usr/bin/env python3
from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

for line in sys.stdin:
    line = line.strip()
    if '\t' not in line:
        continue
    word, count = line.split('\t', 1)
    try:
        count = int(count)
    except ValueError:
        continue

    if current_word == word:
        current_count += count
    else:
        if current_word:
            print ("%s\t%s" % (current_word, current_count))
        current_count = count
        current_word = word
if current_word == word:
    print ("%s\t%s" % (current_word, current_count))
```

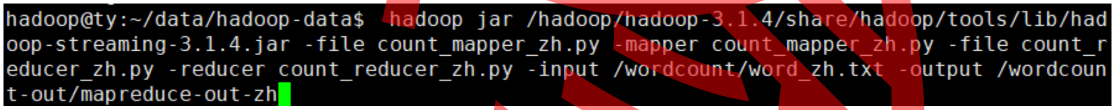
代码 4 count_reducer_zh.py 代码

2.3 本地测试 Map 和 Reduce



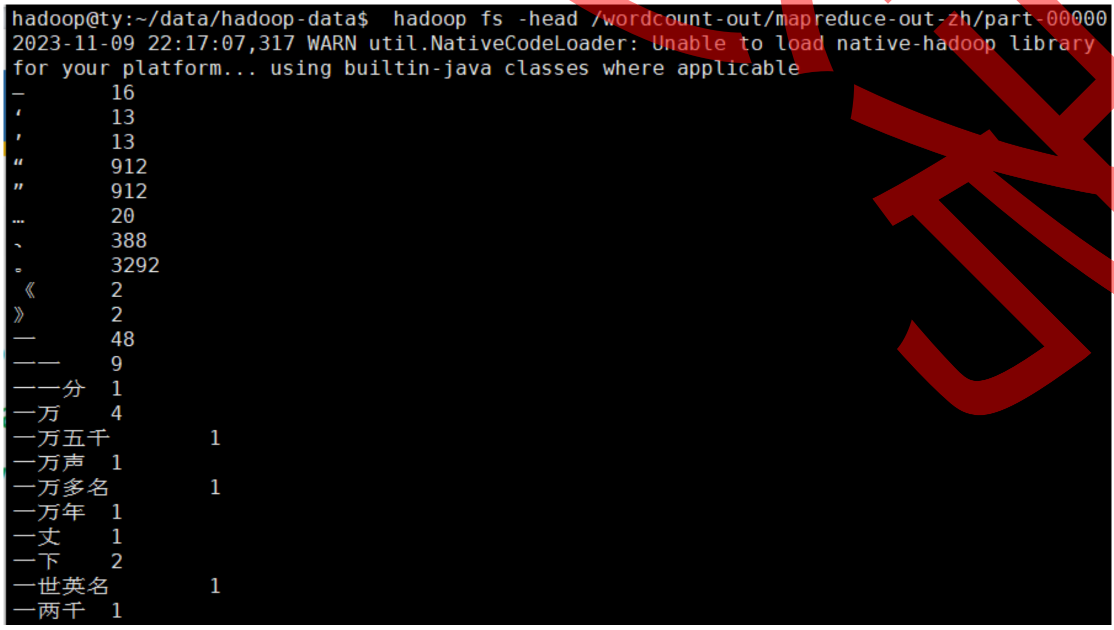
```
hadoop@ty:~/data/hadoop-data$ head -20 /hadoop/data/word_zh.txt | python3 count_mapper_zh
.py | sort |  python3 count_reducer_zh.py
Building prefix dict from the default dictionary ...
Loading model from cache /tmp/jieba.cache
Loading model cost 0.314 seconds.
Prefix dict has been built successfully.
        1
–       2
“       12
”       12
…       2
、      7
。      54
！      1
，      108
：      11
；      3
？      2
安排    1
安喜县  1
傲慢    1
八尺    1
八十二斤        1
```

（1）运行实例



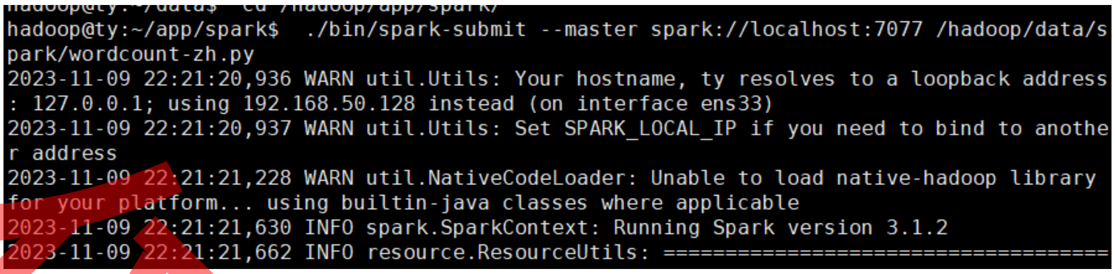
```
hadoop@ty:~/data/hadoop-data$  hadoop jar /hadoop/hadoop-3.1.4/share/hadoop/tools/lib/had
oop-streaming-3.1.4.jar -file count_mapper_zh.py -mapper count_mapper_zh.py -file count_r
educer_zh.py -reducer count_reducer_zh.py -input /wordcount/word_zh.txt -output /wordcoun
t-out/mapreduce-out-zh
```

（2）查看结果



```
hadoop@ty:~/data/hadoop-data$  hadoop fs -head /wordcount-out/mapreduce-out-zh/part-00000
2023-11-09 22:17:07,317 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
–       16
‘       13
’       13
“       912
”       912
…       20
、      388
。      3292
《      2
》      2
—       48
——      9
——分    1
一万    4
一万五千        1
一万声  1
一万多名        1
一万年  1
一丈    1
一下    2
一世英名        1
一两千  1
```

3. Spark 实现 wordcount 实例



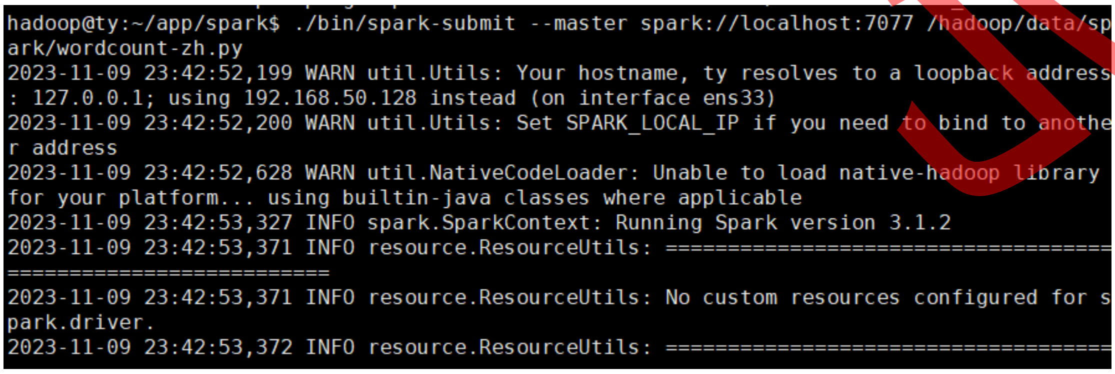
```
hadoop@ty:~/data$  cd /hadoop/app/spark/
hadoop@ty:~/app/spark$  ./bin/spark-submit --master spark://localhost:7077 /hadoop/data/s
park/wordcount-zh.py
2023-11-09 22:21:20,936 WARN util.Utils: Your hostname, ty resolves to a loopback address
: 127.0.0.1; using 192.168.50.128 instead (on interface ens33)
2023-11-09 22:21:20,937 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to anothe
r address
2023-11-09 22:21:21,228 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
2023-11-09 22:21:21,630 INFO spark.SparkContext: Running Spark version 3.1.2
2023-11-09 22:21:21,662 INFO resource.ResourceUtils: ==================================
```

3.1 编写 Spark wordcount 代码：wordcount-zh.py

```
#!/usr/bin/env python3
from pyspark import SparkContext
import jieba

inputFile = ’hdfs://localhost:9000/wordcount/word_zh.txt’
outputFile = ’hdfs://localhost:9000/wordcount-out/spark-out-zh’
sc = SparkContext(’local’, ’wordcount’)
text_file = sc.textFile(inputFile)

# 使用中文分词对每行进行分词操作
word_counts = text_file.flatMap(lambda line: jieba.lcut(line)).map
(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)

word_counts.saveAsTextFile(outputFile)
```

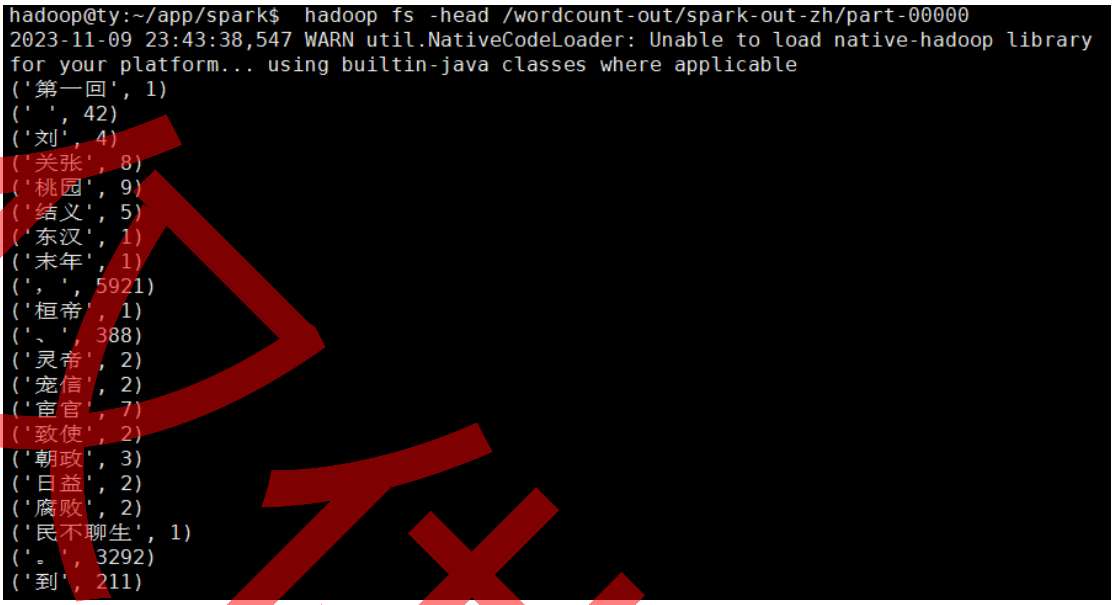
代码 5 wordcount-zh.py 代码

（1）运行实例



```
hadoop@ty:~/app/spark$ ./bin/spark-submit --master spark://localhost:7077 /hadoop/data/sp
ark/wordcount-zh.py
2023-11-09 23:42:52,199 WARN util.Utils: Your hostname, ty resolves to a loopback address
: 127.0.0.1; using 192.168.50.128 instead (on interface ens33)
2023-11-09 23:42:52,200 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to anothe
r address
2023-11-09 23:42:52,628 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
2023-11-09 23:42:53,327 INFO spark.SparkContext: Running Spark version 3.1.2
2023-11-09 23:42:53,371 INFO resource.ResourceUtils: ==================================
==========================
2023-11-09 23:42:53,371 INFO resource.ResourceUtils: No custom resources configured for s
park.driver.
2023-11-09 23:42:53,372 INFO resource.ResourceUtils: ==================================
```

（2）查看结果



```
hadoop@ty:~/app/spark$  hadoop fs -head /wordcount-out/spark-out-zh/part-00000
2023-11-09 23:43:38,547 WARN util.NativeCodeLoader: Unable to load native-hadoop library
for your platform... using builtin-java classes where applicable
('第一回', 1)
(' ', 42)
('刘', 4)
('关张', 8)
('桃园', 9)
('结义', 5)
('东汉', 1)
('末年', 1)
('，', 5921)
('桓帝', 1)
('、', 388)
('灵帝', 2)
('宠信', 2)
('宦官', 7)
('致使', 2)
('朝政', 3)
('日益', 2)
('腐败', 2)
('民不聊生', 1)
('。', 3292)
('到', 211)
```

九、 总结及心得体会：

通过实验，深入了解了 MapReduce 和 Spark 这两个大数据处理框架。了解了它们的基本原理、编程模型以及应用场景，对分布式计算有了更加全面的认识。在实验中，通过编写 WordCount 程序，深入理解了 MapReduce 和 Spark 的编程模型。通过实际的代码编写和运行，对大数据处理中的分布式计算任务有了更深刻的理解。

选作部分的尝试对中文文本进行词频统计，是对框架在处理非英文文本上的一次拓展。这种尝试不仅拓宽了应用范围，还加深了对框架适用性的理解。

十、对本实验过程及方法、手段的改进建议：

在性能比较部分，可以加入更深入的性能分析，例如资源利用率、执行时间的详细对比等，以便更全面地了解 MapReduce 和 Spark 在不同场景下的表现

报告评分：

指导教师签字：