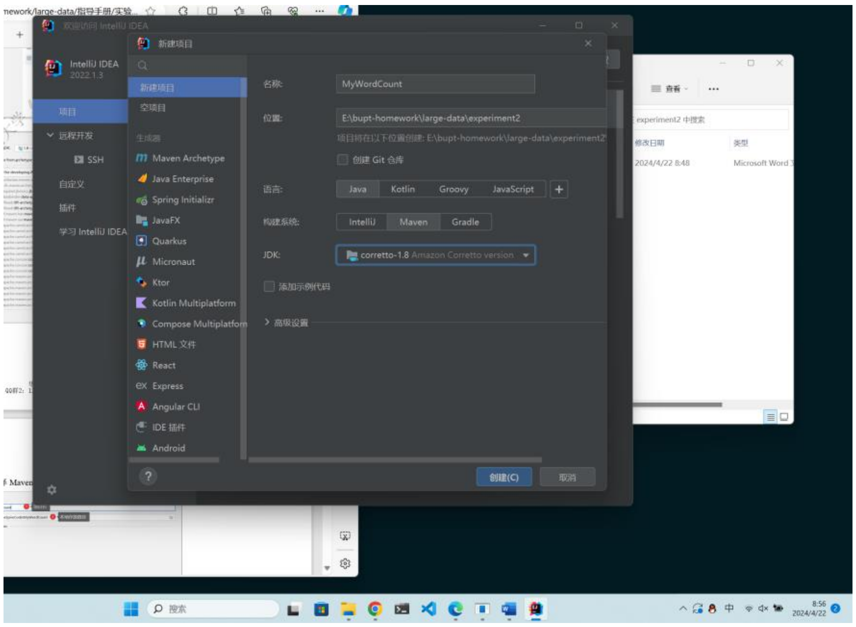
# 实验二——MapReduce 分布式数据处理

姓名：陈朴炎　　　学号：2021211138

## 目录

## 1 实验流程

使用 IDEA 新建一个 java 项目，jdk 选择 1.8



新建项目

名称: MyWordCount

位置: E:\bupt-homework\large-data\experiment2

项目将在以下位置创建: E:\bupt-homework\large-data\experiment2\

☐ 创建 Git 仓库

语言: [✓ Java] Kotlin Groovy JavaScript

构建系统: IntelliJ [✓ Maven] Gradle

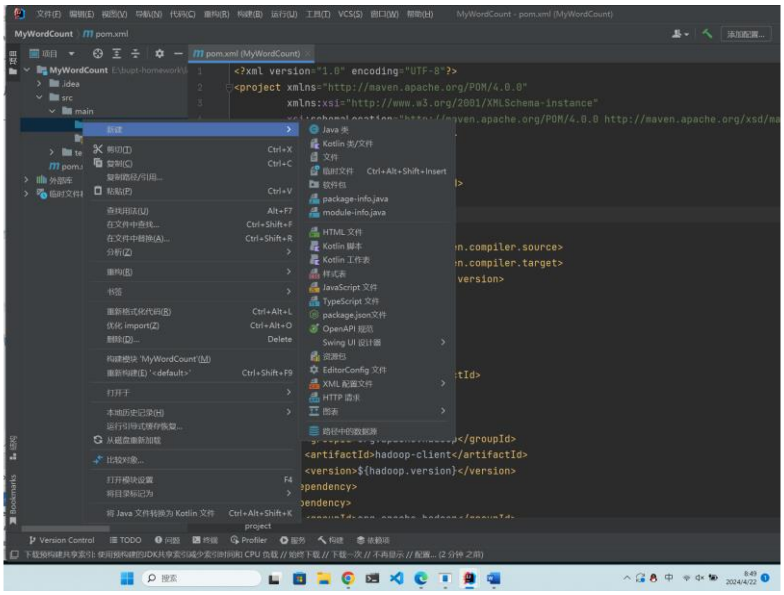
JDK: corretto-1.8 Amazon Corretto version

☐ 添加示例代码

> 高级设置

创建(C) 取消

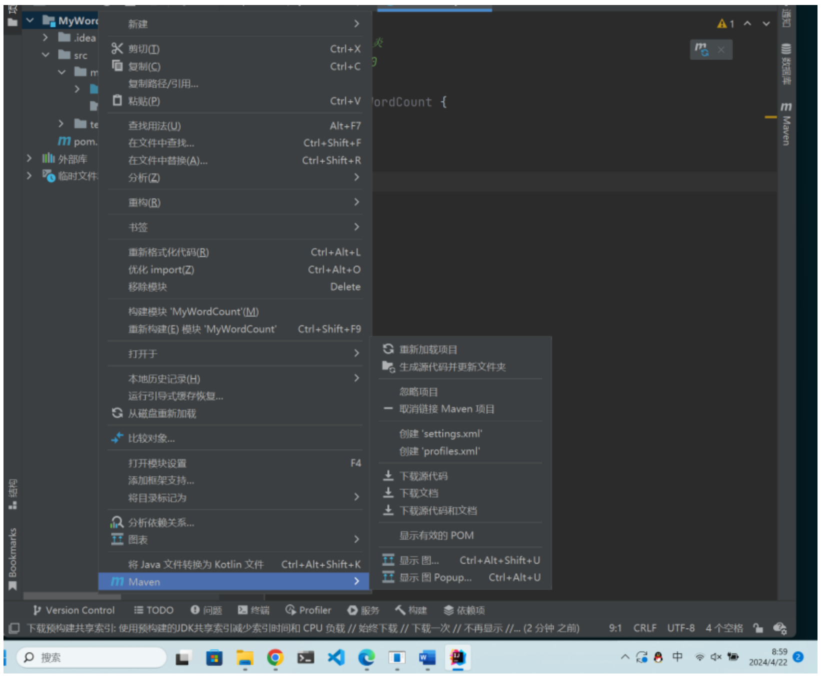
新建一个 MyWordCount 项目，修改里面的 pom.xml 文件，将实验一的配置信息复制到这里面。并新建一个 java 类名为 WordCount



MyWordCount - pom.xml (MyWordCount)

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

修改完 pom.xml 配置文件后，重新构建项目

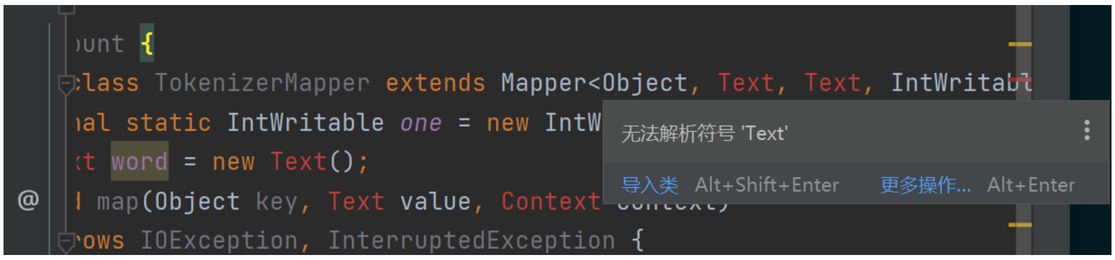


WordCount 类中添加如下代码

这个类的作用是将输入的文本数据切分成单词，并为每个单词输出一个键值对，其中键是单词本身，值是 1，表示该单词出现了一次。

```java
public class WordCount {
    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>{
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

}
```

但是很可惜，会有错误，原因是 Text 类在很多包里面都有，项目没办法自己选择所需要的包，需要我们手动添加。

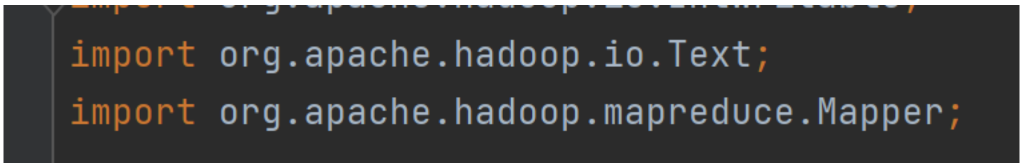


```
›unt {
:lass TokenizerMapper extends Mapper<Object, Text, Text, IntWritabl
ıal static IntWritable one = new IntW
:t word = new Text();
l map(Object key, Text value, Context context)
›ows IOException, InterruptedException {
```

无法解析符号 'Text'

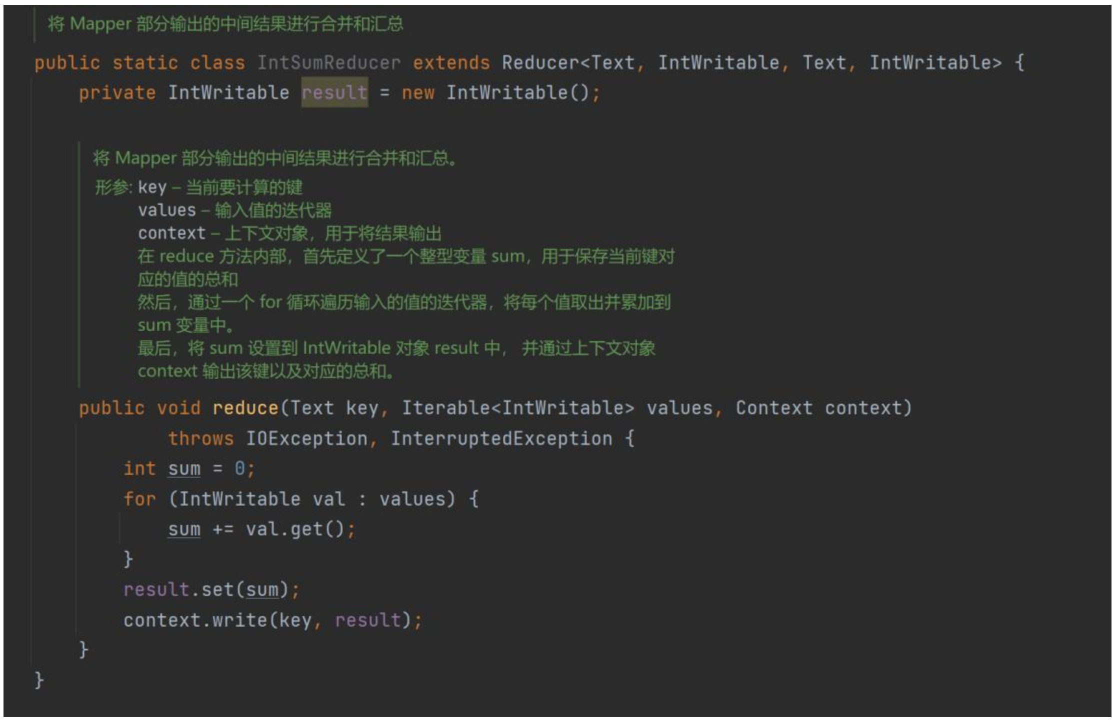
导入类 Alt+Shift+Enter 更多操作... Alt+Enter

导入以下两个包后红色报错消失



```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
```

之后再创建一个 Reducer 对象，继承 hadoop 的 Reducer，用来将 Mapper 的输出结构进行合并和总结，如下图所示。



```
    将 Mapper 部分输出的中间结果进行合并和汇总
public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

        将 Mapper 部分输出的中间结果进行合并和汇总。
        形参: key – 当前要计算的键
              values – 输入值的迭代器
              context – 上下文对象，用于将结果输出
              在 reduce 方法内部，首先定义了一个整型变量 sum，用于保存当前键对
              应的值的总和
              然后，通过一个 for 循环遍历输入的值的迭代器，将每个值取出并累加到
              sum 变量中。
              最后，将 sum 设置到 IntWritable 对象 result 中， 并通过上下文对象
              context 输出该键以及对应的总和。
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

类和函数说明：

在 reduce 函数作为 Reducer 类的主要函数，它的功能是将 Mapper 部分输出的中间结果进行合并和汇总。在 reduce 函数中，首先定义了一个整型变量 sum，用于保存当前键对应的值的总和。然后，通过一个 for 循环遍历输入的值的迭代器，将每个值取出并累加到 sum 变量中。最后，将 sum 设置到

IntWritable 对象 result 中， 并通过上下文对象 context 输出该键以及对应的总和。

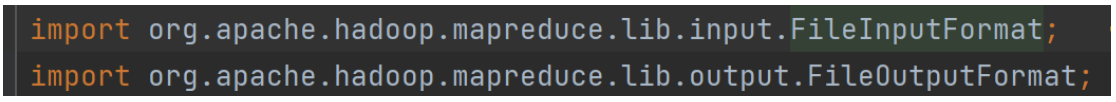
上面两个类编写完成之后，可以开始写 WordCount 的主函数了。

主函数的运行逻辑为：

1. 判断输入的参数格式是否正确
2. 创建 job 对象，并为 job 对象初始化
3. 设置输出的 Key 和 Value
4. 设置输入输出的路径

在主函数中，需要先 new 一个 Configuration 对象。需要注意的是，当前这个 Configuration 对象应该选择 apache.hadoop.conf.Configuration 类。

在第四步的时候，要注意 FileInputFormat 和 FileOutputFormat 应该选择导入的包是以下这两个，不然会报错的。



```
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

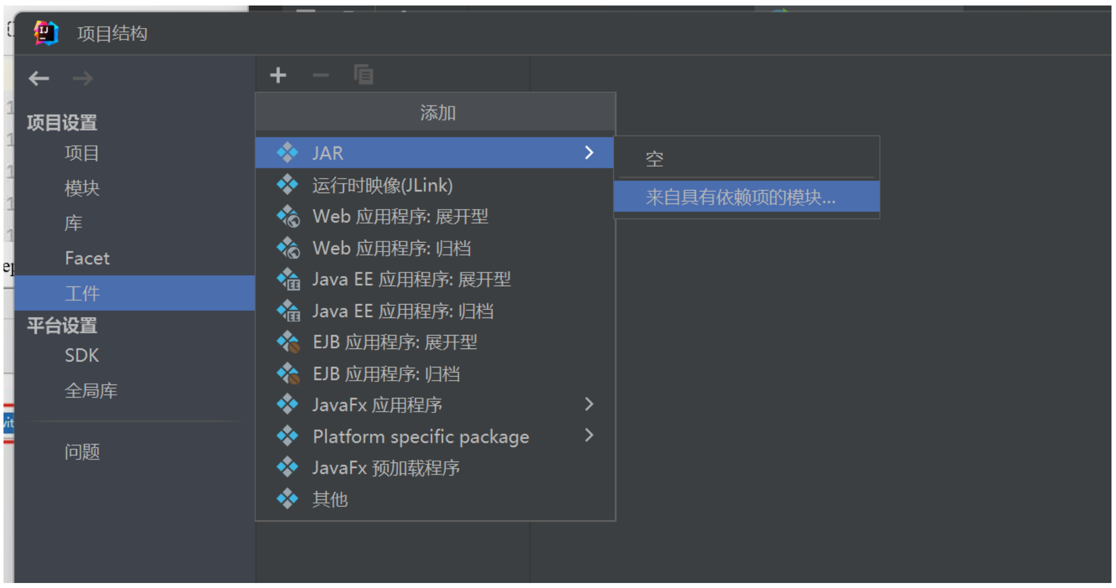
main 函数如下所示：

```
public static void main (String [] args) throws Exception{

    Configuration conf = new Configuration();
    // 限定输出参数必须为 2 个
    String []otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();

    if(otherArgs.length!=2){
        System.err.println("Please usage: word-count <in> <out>");
        System.exit(2);
    }

    // 创建 job 对象
    Job job = new Job(conf, "word count");
```

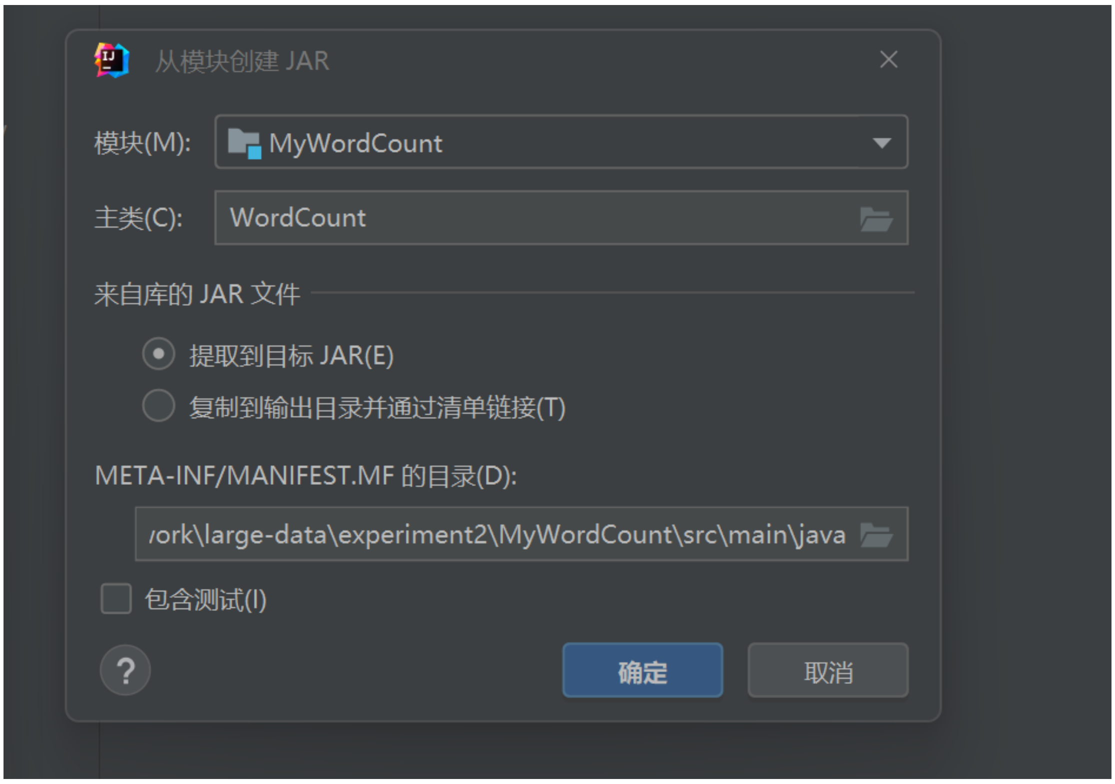
```java

        // 初始化 job 对象
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);

        // 设置输出格式
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        // 设置输入输出
        FileInputFormat.addInputPath(job,new Path(otherArgs[0]));
        FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));

        System.exit(job.waitForCompletion(true)?0:1);
    }
```

接下来应该要打包项目并运行了。

在文件->项目结构中，选择工件，添加，如下图所示



并将主类设置为 WordCount，如下图


从模块创建 JAR

模块(M): MyWordCount

主类(C): WordCount

来自库的 JAR 文件

(● 提取到目标 JAR(E)) ○ 复制到输出目录并通过清单链接(T)

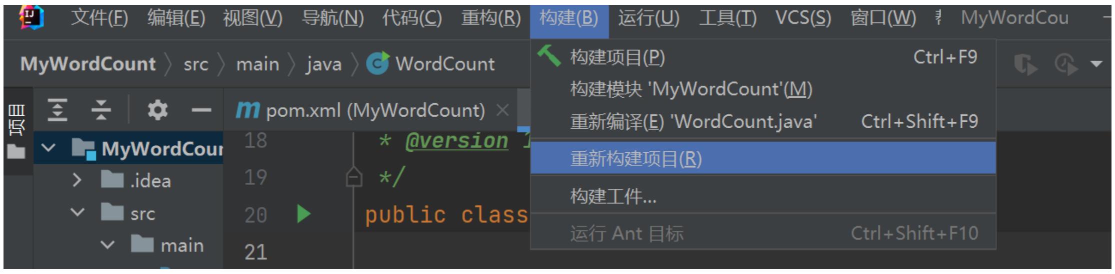
META-INF/MANIFEST.MF 的目录(D): ork\large-data\experiment2\MyWordCount\src\main\java

☐ 包含测试(I)

确定 取消

之后选择应用，确定即可。

选择 构建 -> 重新构建项目

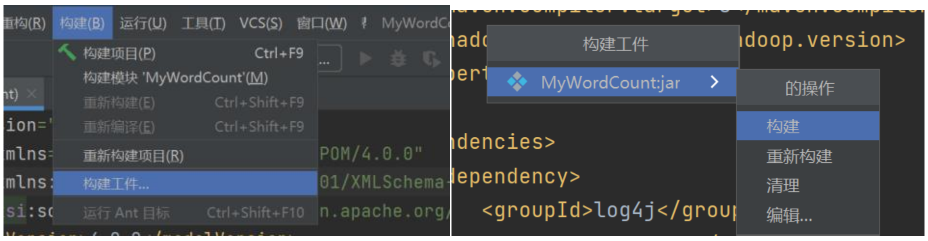

文件(F) 编辑(E) 视图(V) 导航(N) 代码(C) 重构(R) 构建(B) 运行(U) 工具(T) VCS(S) 窗口(W) MyWordCou
MyWordCount 〉 src 〉 main 〉 java 〉 WordCount
pom.xml (MyWordCount)
构建项目(P) Ctrl+F9
构建模块 'MyWordCount'(M)
重新编译(E) 'WordCount.java' Ctrl+Shift+F9
重新构建项目(R)
构建工件...
运行 Ant 目标 Ctrl+Shift+F10
MyWordCount
.idea
src
main

```
 * @version 1
 */
public class
```

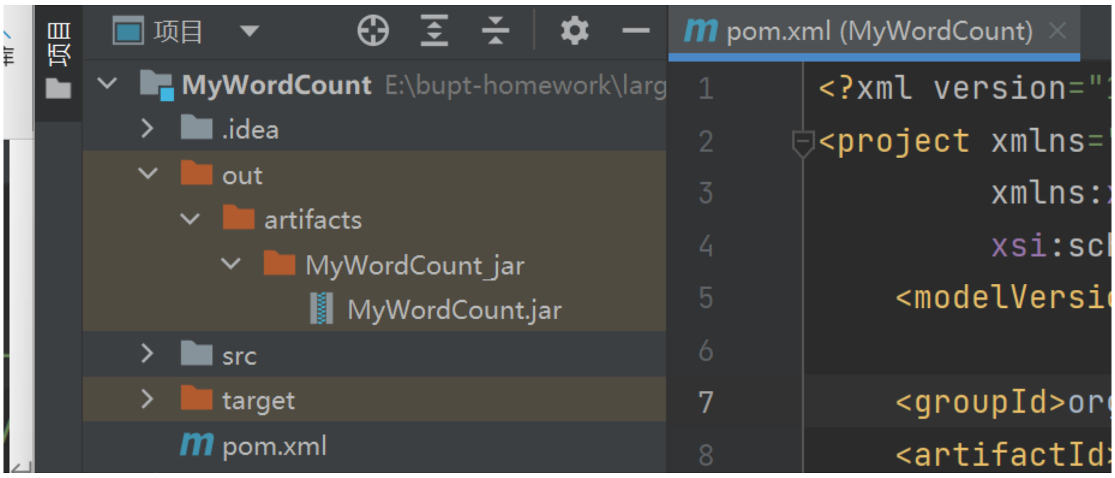
但是这里我没能成功出现 out 包。

原因是没有构建工件。

重新构建工件，如下图所示

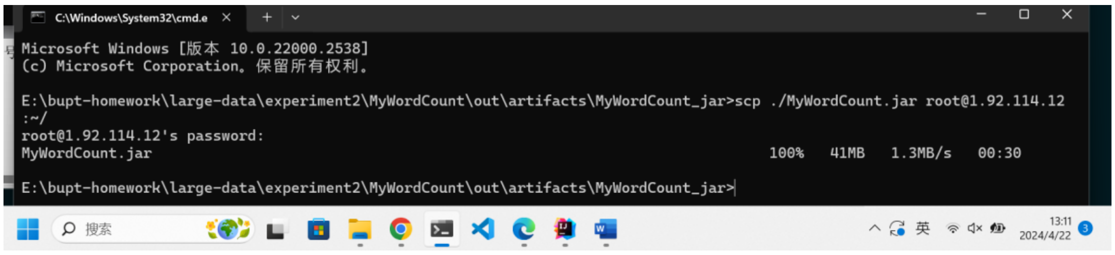

重构(R) 构建(B) 运行(U) 工具(T) VCS(S) 窗口(W) MyWordC
构建项目(P) Ctrl+F9
构建模块 'MyWordCount'(M)
重新构建(E) Ctrl+Shift+F9
重新编译(E) Ctrl+Shift+F9
重新构建项目(R)
构建工件...
运行 Ant 目标 Ctrl+Shift+F10
构建工件
MyWordCount:jar
的操作
构建
重新构建
清理
编辑...

选择构建工件 -> MyWordCount -> 构建，就出现 out 包了

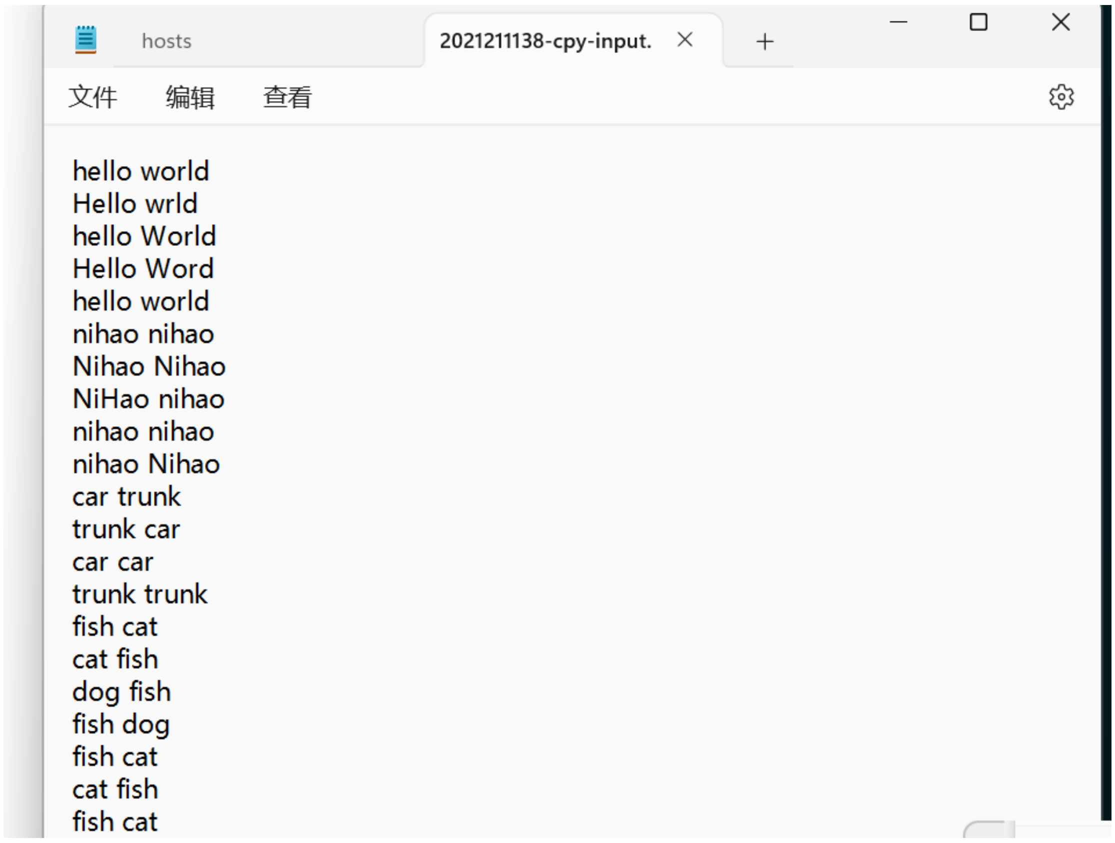


```
<?xml version="
<project xmlns="
         xmlns:
         xsi:sc
    <modelVersi

    <groupId>or
    <artifactId
```

使用 scp 命令将这个 jar 文件发给服务器

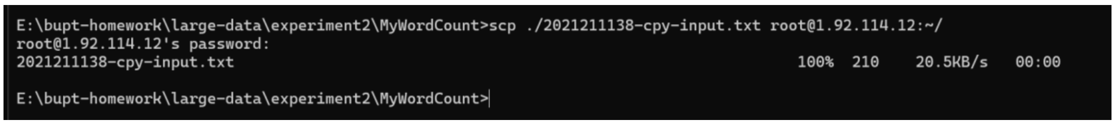


```
Microsoft Windows [版本 10.0.22000.2538]
(c) Microsoft Corporation。保留所有权利。

E:\bupt-homework\large-data\experiment2\MyWordCount\out\artifacts\MyWordCount_jar>scp ./MyWordCount.jar root@1.92.114.12
:~/
root@1.92.114.12's password:
MyWordCount.jar                                                                 100%   41MB   1.3MB/s   00:30

E:\bupt-homework\large-data\experiment2\MyWordCount\out\artifacts\MyWordCount_jar>
```



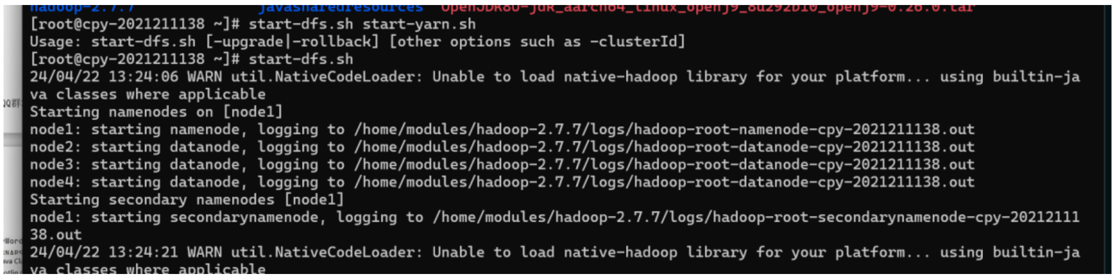
```
hello world
Hello wrld
hello World
Hello Word
hello world
nihao nihao
Nihao Nihao
NiHao nihao
nihao nihao
nihao Nihao
car trunk
trunk car
car car
trunk trunk
fish cat
cat fish
dog fish
fish dog
fish cat
cat fish
fish cat
```

同样的，把这个输入文件传送给服务器

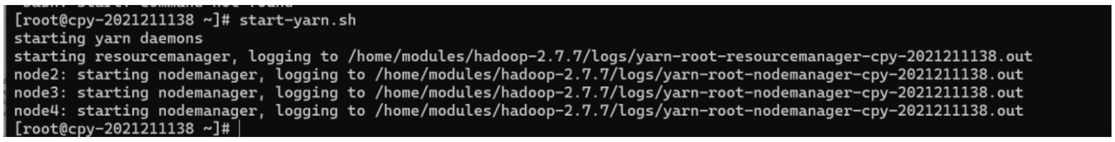


```
E:\bupt-homework\large-data\experiment2\MyWordCount>scp ./2021211138-cpy-input.txt root@1.92.114.12:~/
root@1.92.114.12's password:
2021211138-cpy-input.txt                                                          100%  210    20.5KB/s   00:00

E:\bupt-homework\large-data\experiment2\MyWordCount>
```

连上四台服务器，启动 hadoop 和 yarn

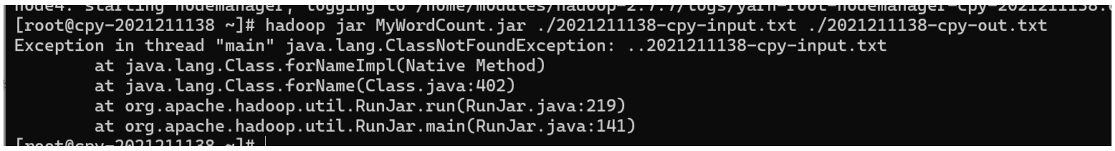


```
[root@cpy-2021211138 ~]# start-dfs.sh start-yarn.sh
Usage: start-dfs.sh [-upgrade|-rollback] [other options such as -clusterId]
[root@cpy-2021211138 ~]# start-dfs.sh
24/04/22 13:24:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-ja
va classes where applicable
Starting namenodes on [node1]
node1: starting namenode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-namenode-cpy-2021211138.out
node2: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
node3: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
node4: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
Starting secondary namenodes [node1]
node1: starting secondarynamenode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-secondarynamenode-cpy-20212111
38.out
24/04/22 13:24:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-ja
va classes where applicable
```



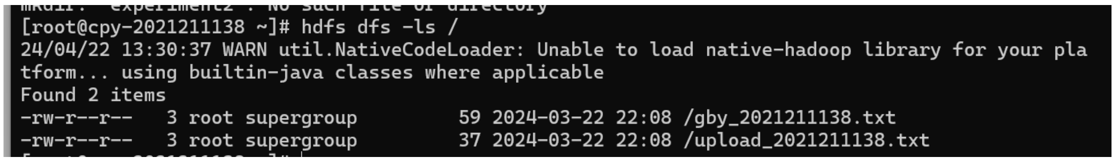
```
[root@cpy-2021211138 ~]# start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-resourcemanager-cpy-2021211138.out
node2: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
node3: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
node4: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
[root@cpy-2021211138 ~]#
```

尝试运行 jar 包，主函数抛出异常



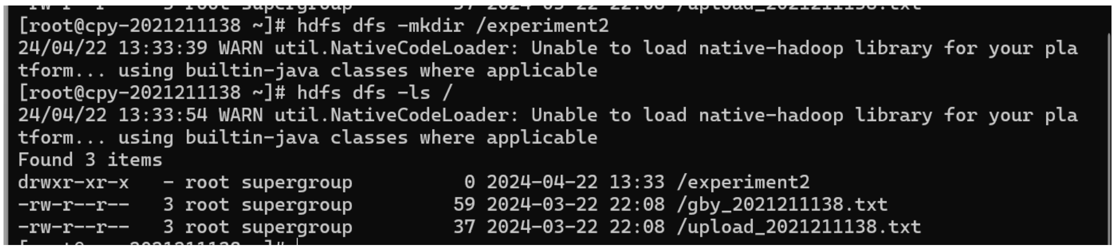
```
[root@cpy-2021211138 ~]# hadoop jar MyWordCount.jar ./2021211138-cpy-input.txt ./2021211138-cpy-out.txt
Exception in thread "main" java.lang.ClassNotFoundException: ..2021211138-cpy-input.txt
        at java.lang.Class.forNameImpl(Native Method)
        at java.lang.Class.forName(Class.java:402)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:219)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:141)
```

我觉得应该是 hdfs 找不到文件导致的



```
[root@cpy-2021211138 ~]# hdfs dfs -ls /
24/04/22 13:30:37 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   3 root supergroup         59 2024-03-22 22:08 /gby_2021211138.txt
-rw-r--r--   3 root supergroup         37 2024-03-22 22:08 /upload_2021211138.txt
```

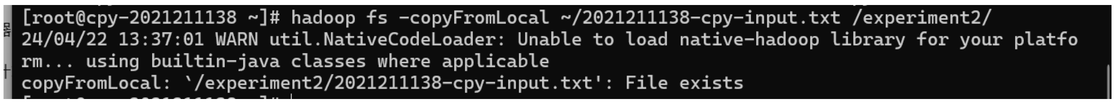
使用 hdfs dfs -ls / 查看 hdfs 系统根目录 / 文件夹下的内容，发现并没有我们想要的输入文件。



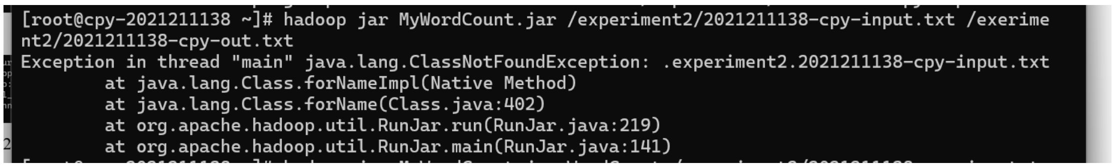
```
[root@cpy-2021211138 ~]# hdfs dfs -mkdir /experiment2
24/04/22 13:33:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
[root@cpy-2021211138 ~]# hdfs dfs -ls /
24/04/22 13:33:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
Found 3 items
drwxr-xr-x   - root supergroup          0 2024-04-22 13:33 /experiment2
-rw-r--r--   3 root supergroup         59 2024-03-22 22:08 /gby_2021211138.txt
-rw-r--r--   3 root supergroup         37 2024-03-22 22:08 /upload_2021211138.txt
```

使用 hdfs dfs -mkdir /experiment2，创建一个 hdfs 文件夹

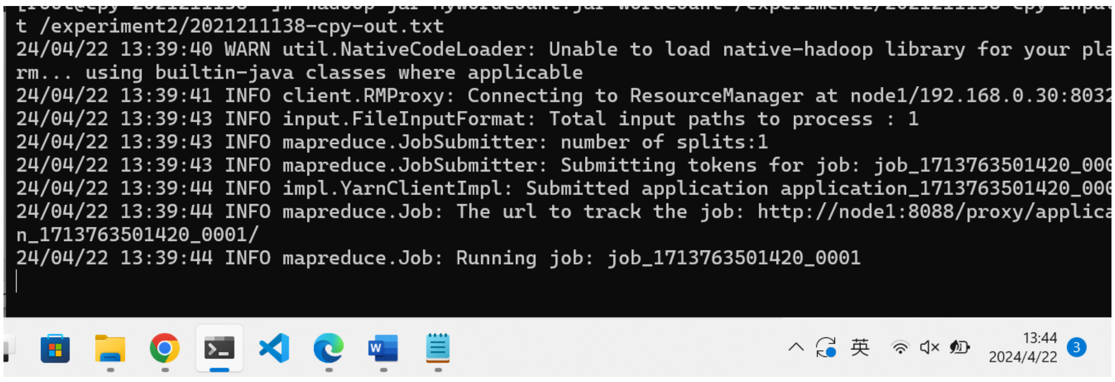
使用 hadoop fs -copyFromLocal <localSrc> <hdfsDst>将本地文件拷到 hdfs 文件系统的路径下


```
[root@cpy-2021211138 ~]# hadoop fs -copyFromLocal ~/2021211138-cpy-input.txt /experiment2/
24/04/22 13:37:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platfo
rm... using builtin-java classes where applicable
copyFromLocal: `/experiment2/2021211138-cpy-input.txt': File exists
```

又报出如下错误

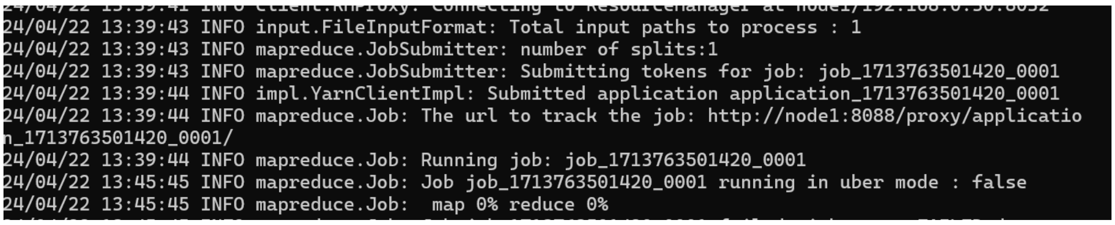

```
[root@cpy-2021211138 ~]# hadoop jar MyWordCount.jar /experiment2/2021211138-cpy-input.txt /exerime
nt2/2021211138-cpy-out.txt
Exception in thread "main" java.lang.ClassNotFoundException: .experiment2.2021211138-cpy-input.txt
        at java.lang.Class.forNameImpl(Native Method)
        at java.lang.Class.forName(Class.java:402)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:219)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:141)
```

原因是 hadoop jar 命令的格式应该是 hadoop jar <JAR 文件路径> <主类> [参数]，而我的命令中缺少了主类的指定。所以应该将命令改为如下所示。


```
t /experiment2/2021211138-cpy-out.txt
24/04/22 13:39:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
rm... using builtin-java classes where applicable
24/04/22 13:39:41 INFO client.RMProxy: Connecting to ResourceManager at node1/192.168.0.30:8032
24/04/22 13:39:43 INFO input.FileInputFormat: Total input paths to process : 1
24/04/22 13:39:43 INFO mapreduce.JobSubmitter: number of splits:1
24/04/22 13:39:43 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1713763501420_000
24/04/22 13:39:44 INFO impl.YarnClientImpl: Submitted application application_1713763501420_000
24/04/22 13:39:44 INFO mapreduce.Job: The url to track the job: http://node1:8088/proxy/applica
n_1713763501420_0001/
24/04/22 13:39:44 INFO mapreduce.Job: Running job: job_1713763501420_0001
```
13:44
2024/4/22

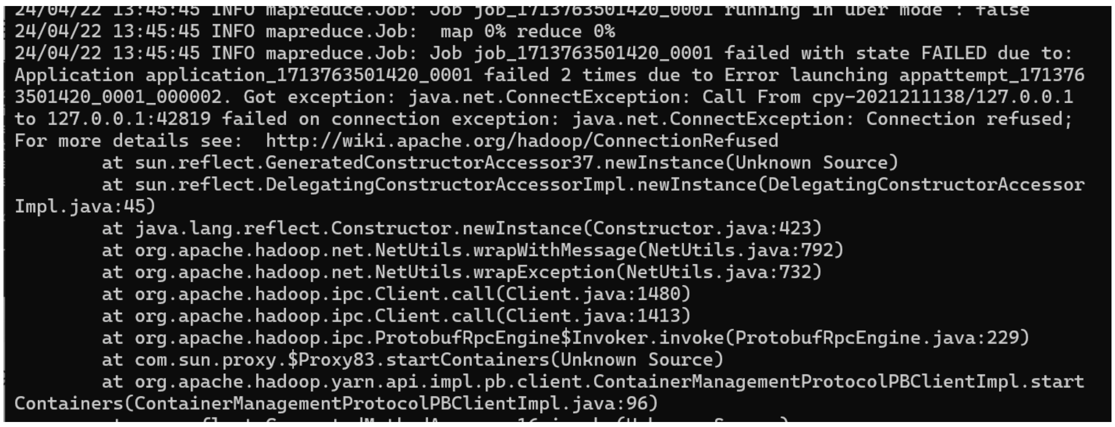
但是可以从时间戳上看到卡住了从 39 分到 44 分，很久都没有动静。

一直到 45 分，又有动静了

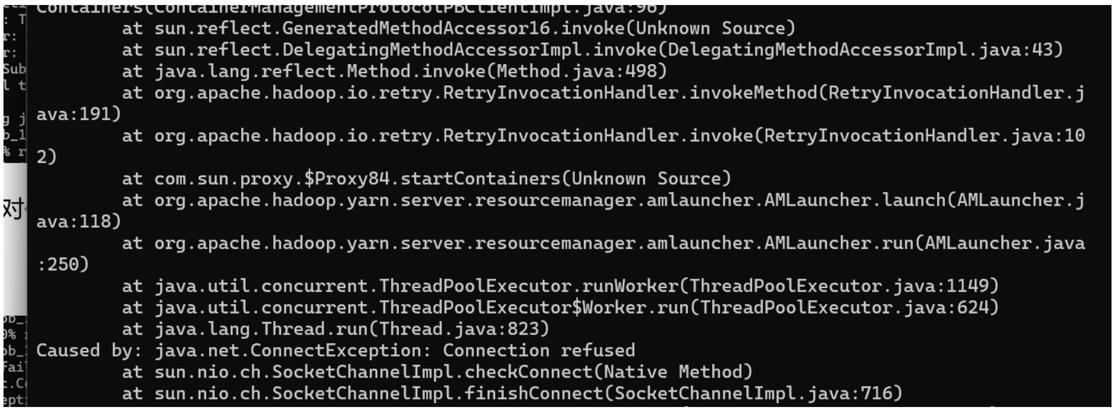

```
24/04/22 13:39:43 INFO input.FileInputFormat: Total input paths to process : 1
24/04/22 13:39:43 INFO mapreduce.JobSubmitter: number of splits:1
24/04/22 13:39:43 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1713763501420_0001
24/04/22 13:39:44 INFO impl.YarnClientImpl: Submitted application application_1713763501420_0001
24/04/22 13:39:44 INFO mapreduce.Job: The url to track the job: http://node1:8088/proxy/applicatio
n_1713763501420_0001/
24/04/22 13:39:44 INFO mapreduce.Job: Running job: job_1713763501420_0001
24/04/22 13:45:45 INFO mapreduce.Job: Job job_1713763501420_0001 running in uber mode : false
24/04/22 13:45:45 INFO mapreduce.Job:  map 0% reduce 0%
```

但是 map 0% reduce 0%，显然不对

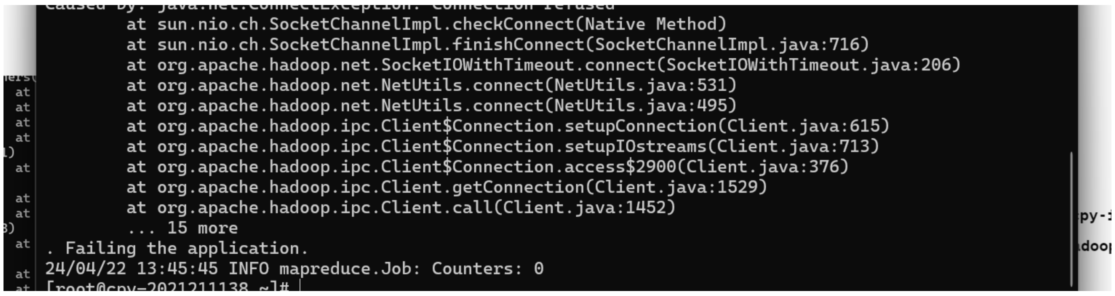
剩下的信息如下图所示



```
24/04/22 13:45:45 INFO mapreduce.Job: Job job_1713763501420_0001 running in uber mode : false
24/04/22 13:45:45 INFO mapreduce.Job:  map 0% reduce 0%
24/04/22 13:45:45 INFO mapreduce.Job: Job job_1713763501420_0001 failed with state FAILED due to:
Application application_1713763501420_0001 failed 2 times due to Error launching appattempt_171376
3501420_0001_000002. Got exception: java.net.ConnectException: Call From cpy-2021211138/127.0.0.1
to 127.0.0.1:42819 failed on connection exception: java.net.ConnectException: Connection refused;
For more details see:  http://wiki.apache.org/hadoop/ConnectionRefused
        at sun.reflect.GeneratedConstructorAccessor37.newInstance(Unknown Source)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessor
Impl.java:45)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
        at org.apache.hadoop.net.NetUtils.wrapWithMessage(NetUtils.java:792)
        at org.apache.hadoop.net.NetUtils.wrapException(NetUtils.java:732)
        at org.apache.hadoop.ipc.Client.call(Client.java:1480)
        at org.apache.hadoop.ipc.Client.call(Client.java:1413)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:229)
        at com.sun.proxy.$Proxy83.startContainers(Unknown Source)
        at org.apache.hadoop.yarn.api.impl.pb.client.ContainerManagementProtocolPBClientImpl.start
Containers(ContainerManagementProtocolPBClientImpl.java:96)
```



```
Containers(ContainerManagementProtocolPBClientImpl.java:96)
        at sun.reflect.GeneratedMethodAccessor16.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.io.retry.RetryInvocationHandler.invokeMethod(RetryInvocationHandler.j
ava:191)
        at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:10
2)
        at com.sun.proxy.$Proxy84.startContainers(Unknown Source)
        at org.apache.hadoop.yarn.server.resourcemanager.amlauncher.AMLauncher.launch(AMLauncher.j
ava:118)
        at org.apache.hadoop.yarn.server.resourcemanager.amlauncher.AMLauncher.run(AMLauncher.java
:250)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:823)
Caused by: java.net.ConnectException: Connection refused
        at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
        at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:716)
```

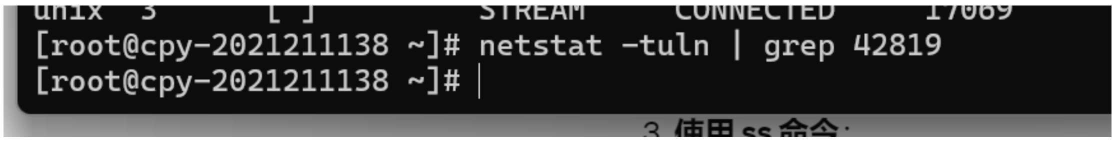


```
Caused by: java.net.ConnectException: Connection refused
        at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
        at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:716)
        at org.apache.hadoop.net.SocketIOWithTimeout.connect(SocketIOWithTimeout.java:206)
        at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:531)
        at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:495)
        at org.apache.hadoop.ipc.Client$Connection.setupConnection(Client.java:615)
        at org.apache.hadoop.ipc.Client$Connection.setupIOstreams(Client.java:713)
        at org.apache.hadoop.ipc.Client$Connection.access$2900(Client.java:376)
        at org.apache.hadoop.ipc.Client.getConnection(Client.java:1529)
        at org.apache.hadoop.ipc.Client.call(Client.java:1452)
        ... 15 more
. Failing the application.
24/04/22 13:45:45 INFO mapreduce.Job: Counters: 0
[root@cpy-2021211138 ~]#
```

从下面这句话中，能得出，连接到 42819 端口失败了。

24/04/22 13:45:45 INFO mapreduce.Job: Job job_1713763501420_0001 failed with state FAILED due to: Application application_1713763501420_0001 failed 2 times due to Error launching appattempt_1713763501420_0001_000002. Got exception: java.net.ConnectException: Call From cpy-2021211138/127.0.0.1 to 127.0.0.1:42819 failed on connection exception: java.net.ConnectException: Connection refused;

之后我使用 netstat 命令，并没有看到 42819 端口打开的信息，我觉得应该是端口没打开导致的。



```
unix  3      [ ]         STREAM     CONNECTED     17069
[root@cpy-2021211138 ~]# netstat -tuln | grep 42819
[root@cpy-2021211138 ~]#
```

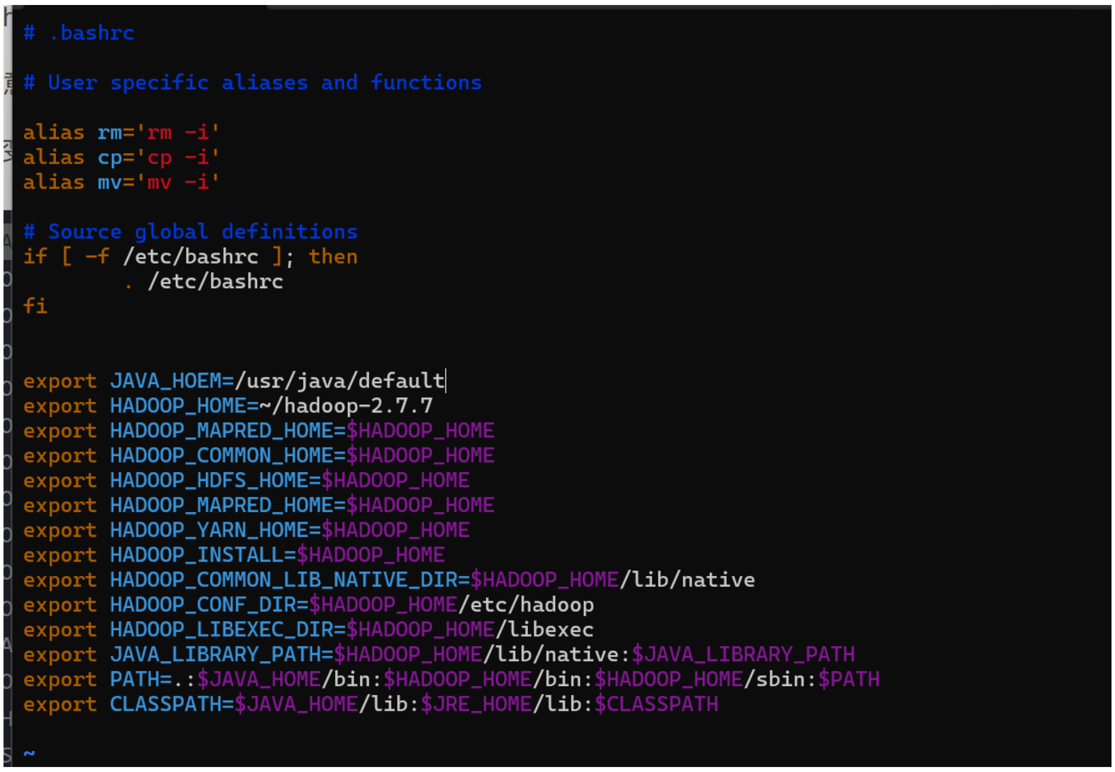
但是重新尝试了很多次每次的端口都不一样，应该不是端口的问题

我查阅了相关资料，有用的如下：

https://blog.csdn.net/u014646662/article/details/82890443

对于我这个问题，应该先配置环境变量

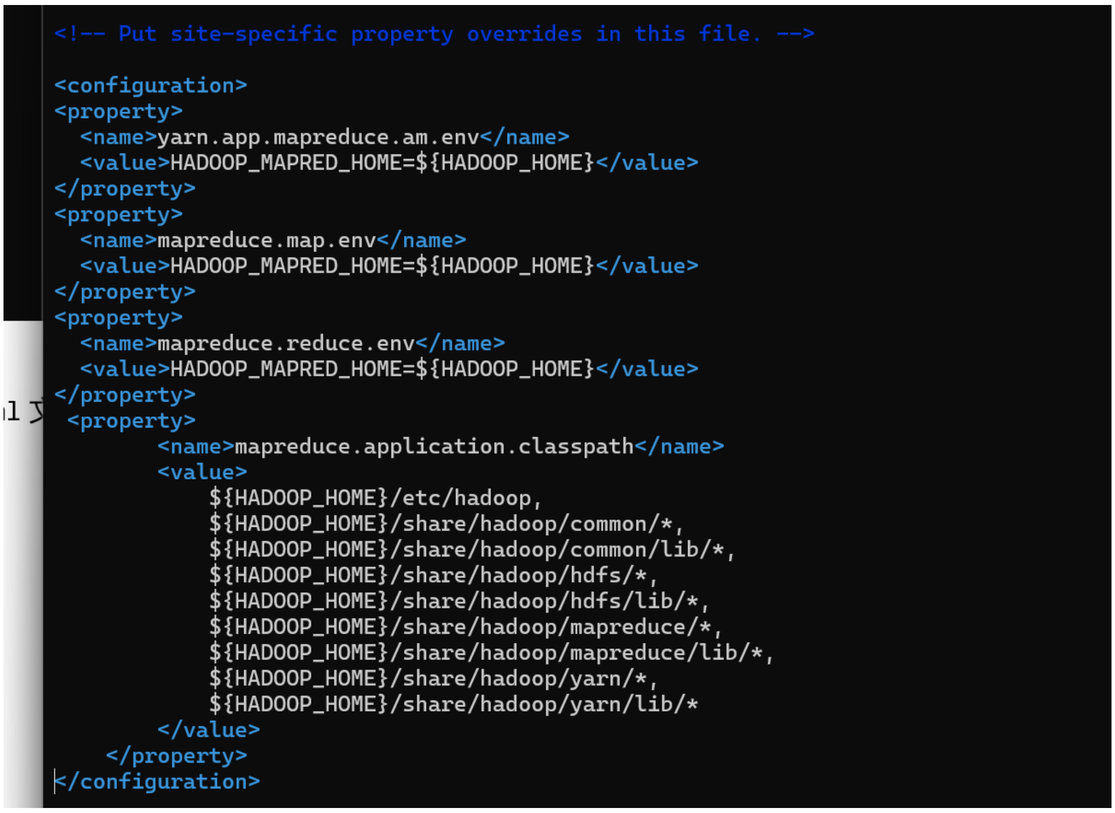
首先输入 vim ~/.bashrc，在最后面添加几行，如下



```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi


export JAVA_HOEM=/usr/java/default
export HADOOP_HOME=~/hadoop-2.7.7
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_LIBEXEC_DIR=$HADOOP_HOME/libexec
export JAVA_LIBRARY_PATH=$HADOOP_HOME/lib/native:$JAVA_LIBRARY_PATH
export PATH=.:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export CLASSPATH=$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH

~
```

然后在 hadoop-2.7.7/etc/hadoop 中添加一个 mapred-site.xml 文件

文件内容如下：



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

<configuration>
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
 <property>
        <name>mapreduce.application.classpath</name>
        <value>
            ${HADOOP_HOME}/etc/hadoop,
            ${HADOOP_HOME}/share/hadoop/common/*,
            ${HADOOP_HOME}/share/hadoop/common/lib/*,
            ${HADOOP_HOME}/share/hadoop/hdfs/*,
            ${HADOOP_HOME}/share/hadoop/hdfs/lib/*,
            ${HADOOP_HOME}/share/hadoop/mapreduce/*,
            ${HADOOP_HOME}/share/hadoop/mapreduce/lib/*,
            ${HADOOP_HOME}/share/hadoop/yarn/*,
            ${HADOOP_HOME}/share/hadoop/yarn/lib/*
        </value>
    </property>
</configuration>
```

这个文件是为了配置 hadoop 框架的相关参数的

yarn.app.mapreduce.am.env: 这个属性用于配置 MapReduce 应用程序的 ApplicationMaster 环境变量。在这里，通过设置 HADOOP_MAPRED_HOME 为 ${HADOOP_HOME}，将 MapReduce 应用程序的环境变量指向 Hadoop 安装目录。

mapreduce.map.env: 这个属性用于配置 Map Task 执行器的环境变量。同样地，通过设置 HADOOP_MAPRED_HOME 为 ${HADOOP_HOME}，将 Map Task 执行器的环境变量指向 Hadoop 安装目录。

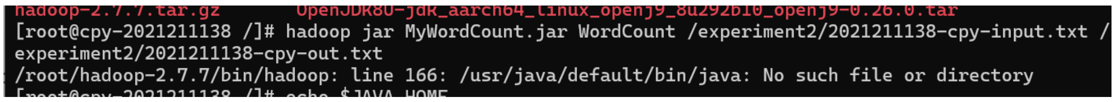
mapreduce.reduce.env: 这个属性用于配置 Reduce Task 执行器的环境变量，同样地，通过设置 HADOOP_MAPRED_HOME 为 ${HADOOP_HOME}，将 Reduce Task 执行器的环境变量指向 Hadoop 安装目录。

mapreduce.application.classpath: 这个属性用于配置 MapReduce

应用程序的类路径，以便能够加载所需的类和库文件。在这里，指定了各种 Hadoop 相关的目录和库文件的路径，以确保 MapReduce 任务能够正常运行。

保存后，执行 source ~/.bashrc 激活新的环境变量

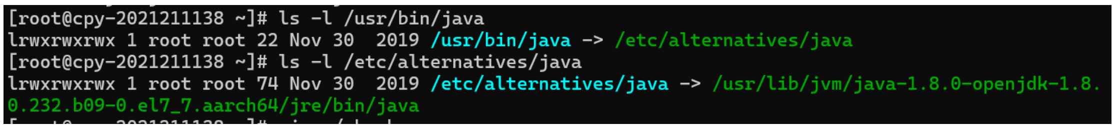
但是这个时候又报错了，如下：



```
hadoop-2.7.7.tar.gz        OpenJDK8U-jdk_aarch64_linux_openj9_8u292b10_openj9-0.26.0.tar
[root@cpy-2021211138 /]# hadoop jar MyWordCount.jar WordCount /experiment2/2021211138-cpy-input.txt /
experiment2/2021211138-cpy-out.txt
/root/hadoop-2.7.7/bin/hadoop: line 166: /usr/java/default/bin/java: No such file or directory
[root@cpy-2021211138 /]# echo $JAVA_HOME
```

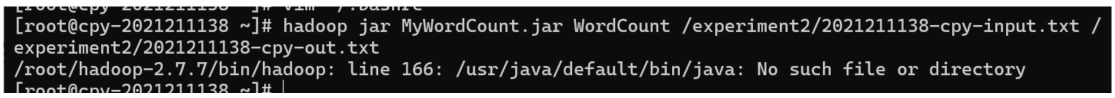
找不到/usr/java/deault/ 路径

我通过如下命令找到了 java 的安装路径



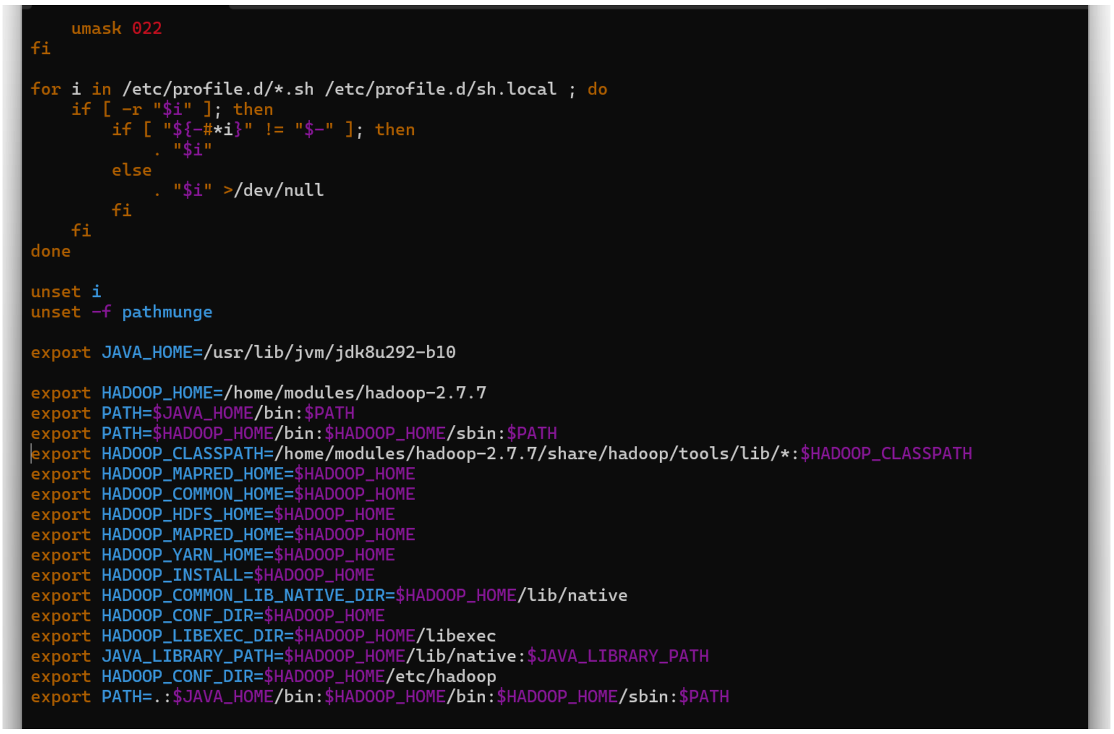
```
[root@cpy-2021211138 ~]# ls -l /usr/bin/java
lrwxrwxrwx 1 root root 22 Nov 30  2019 /usr/bin/java -> /etc/alternatives/java
[root@cpy-2021211138 ~]# ls -l /etc/alternatives/java
lrwxrwxrwx 1 root root 74 Nov 30  2019 /etc/alternatives/java -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.
0.232.b09-0.el7_7.aarch64/jre/bin/java
```

修改了~/.bashrc 的 JAVA_HOME 后，还是不行



```
[root@cpy-2021211138 ~]# vim ~/.bashrc
[root@cpy-2021211138 ~]# hadoop jar MyWordCount.jar WordCount /experiment2/2021211138-cpy-input.txt /
experiment2/2021211138-cpy-out.txt
/root/hadoop-2.7.7/bin/hadoop: line 166: /usr/java/default/bin/java: No such file or directory
[root@cpy-2021211138 ~]#
```

尝试修改 /etc/profile

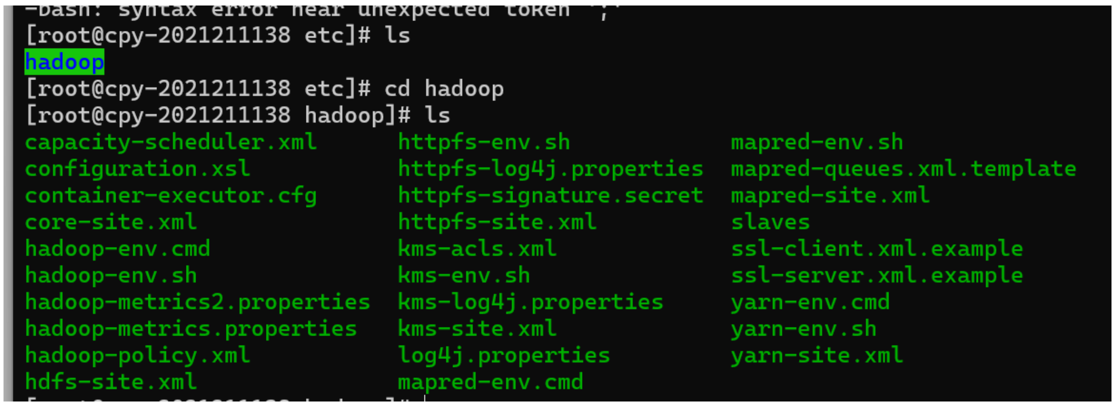


```
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge

export JAVA_HOME=/usr/lib/jvm/jdk8u292-b10

export HADOOP_HOME=/home/modules/hadoop-2.7.7
export PATH=$JAVA_HOME/bin:$PATH
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export HADOOP_CLASSPATH=/home/modules/hadoop-2.7.7/share/hadoop/tools/lib/*:$HADOOP_CLASSPATH
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_CONF_DIR=$HADOOP_HOME
export HADOOP_LIBEXEC_DIR=$HADOOP_HOME/libexec
export JAVA_LIBRARY_PATH=$HADOOP_HOME/lib/native:$JAVA_LIBRARY_PATH
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=.:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

使用 source /etc/profile 重新激活

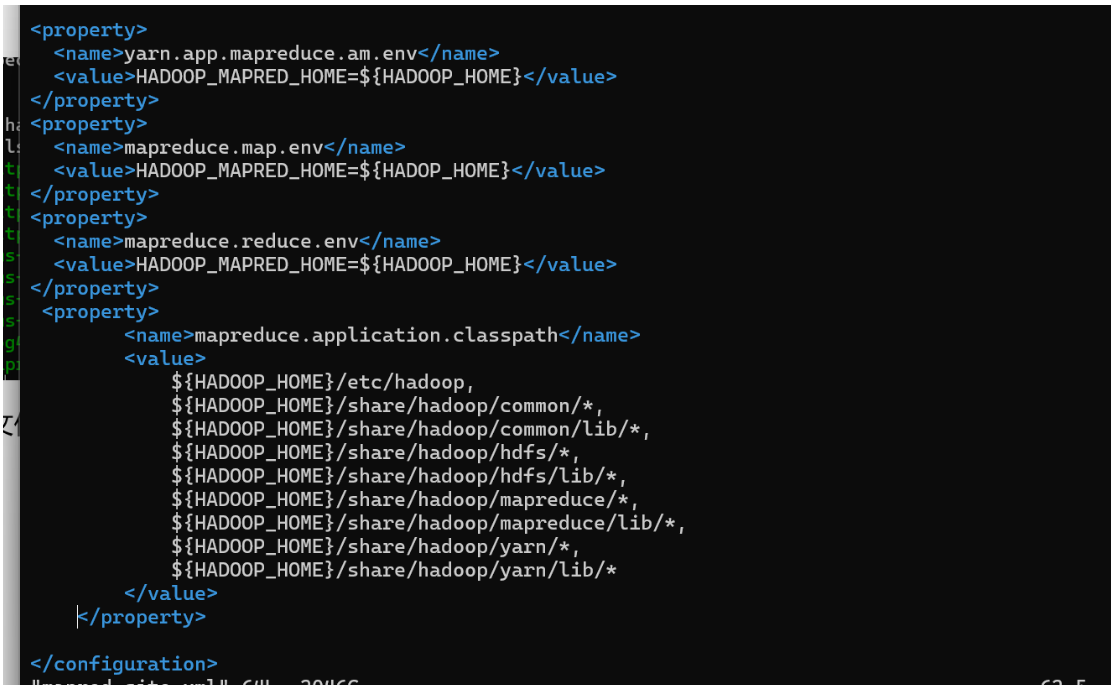
现在进入到 HADOOP_HOME 目录中，如上图我的是/home/modules/hadoop-2.7.7

进入到 etc/hadoop 中



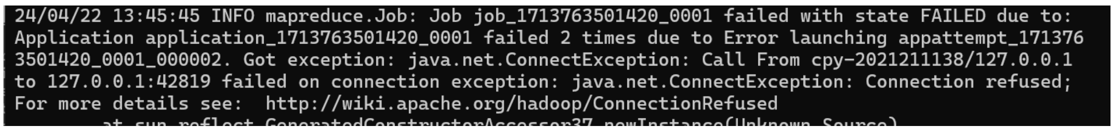
```
-bash: syntax error near unexpected token `;'
[root@cpy-2021211138 etc]# ls
hadoop
[root@cpy-2021211138 etc]# cd hadoop
[root@cpy-2021211138 hadoop]# ls
capacity-scheduler.xml      httpfs-env.sh             mapred-env.sh
configuration.xsl           httpfs-log4j.properties   mapred-queues.xml.template
container-executor.cfg      httpfs-signature.secret   mapred-site.xml
core-site.xml               httpfs-site.xml           slaves
hadoop-env.cmd              kms-acls.xml              ssl-client.xml.example
hadoop-env.sh               kms-env.sh                ssl-server.xml.example
hadoop-metrics2.properties  kms-log4j.properties      yarn-env.cmd
hadoop-metrics.properties   kms-site.xml              yarn-env.sh
hadoop-policy.xml           log4j.properties          yarn-site.xml
hdfs-site.xml               mapred-env.cmd
```

修改 mapred-site.xml 文件，添加如下信息



```
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOP_HOME}</value>
</property>
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
 <property>
        <name>mapreduce.application.classpath</name>
        <value>
            ${HADOOP_HOME}/etc/hadoop,
            ${HADOOP_HOME}/share/hadoop/common/*,
            ${HADOOP_HOME}/share/hadoop/common/lib/*,
            ${HADOOP_HOME}/share/hadoop/hdfs/*,
            ${HADOOP_HOME}/share/hadoop/hdfs/lib/*,
            ${HADOOP_HOME}/share/hadoop/mapreduce/*,
            ${HADOOP_HOME}/share/hadoop/mapreduce/lib/*,
            ${HADOOP_HOME}/share/hadoop/yarn/*,
            ${HADOOP_HOME}/share/hadoop/yarn/lib/*
        </value>
    </property>

</configuration>
```

重新执行 jar 指令，但还是不行。这时候我认真看了一下报错信息

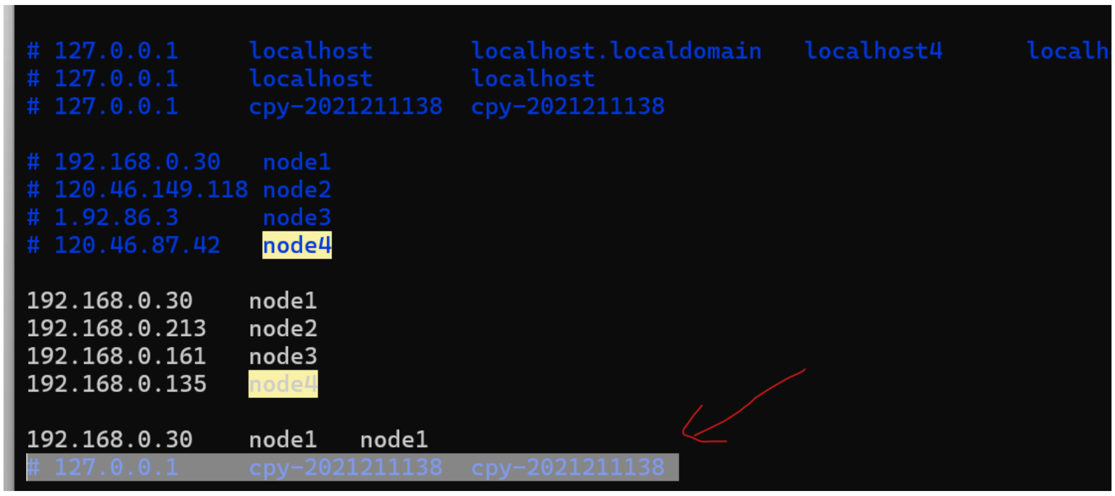


```
24/04/22 13:45:45 INFO mapreduce.Job: Job job_1713763501420_0001 failed with state FAILED due to:
Application application_1713763501420_0001 failed 2 times due to Error launching appattempt_171376
3501420_0001_000002. Got exception: java.net.ConnectException: Call From cpy-2021211138/127.0.0.1
to 127.0.0.1:42819 failed on connection exception: java.net.ConnectException: Connection refused;
For more details see:  http://wiki.apache.org/hadoop/ConnectionRefused
```

它其实是 java 连接错误，在连接到 127.0.0.1 的时候出错了。为什么它会连到 42819 端口呢？我们的 hadoop 服务的端口号并不是这个。我重新回顾了

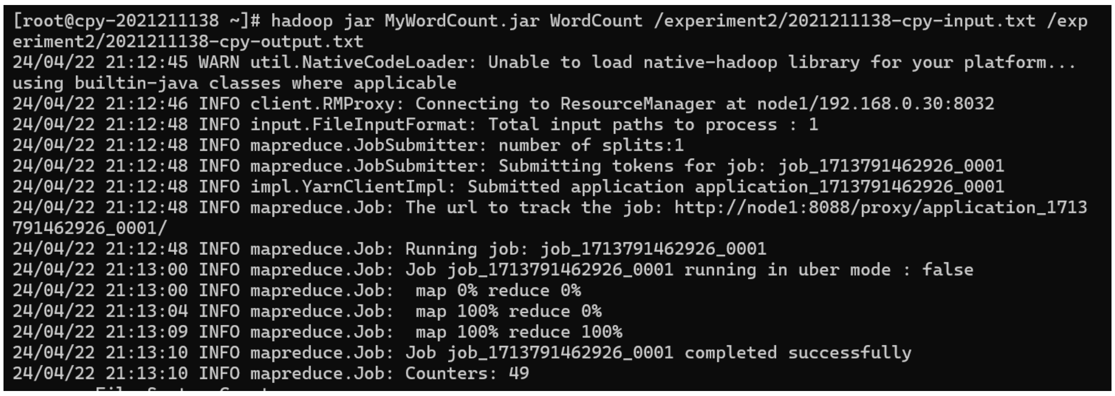
一下这个实验和上一个实验，定位了问题所在：hosts 文件有问题。

打开/etc/hosts 文件

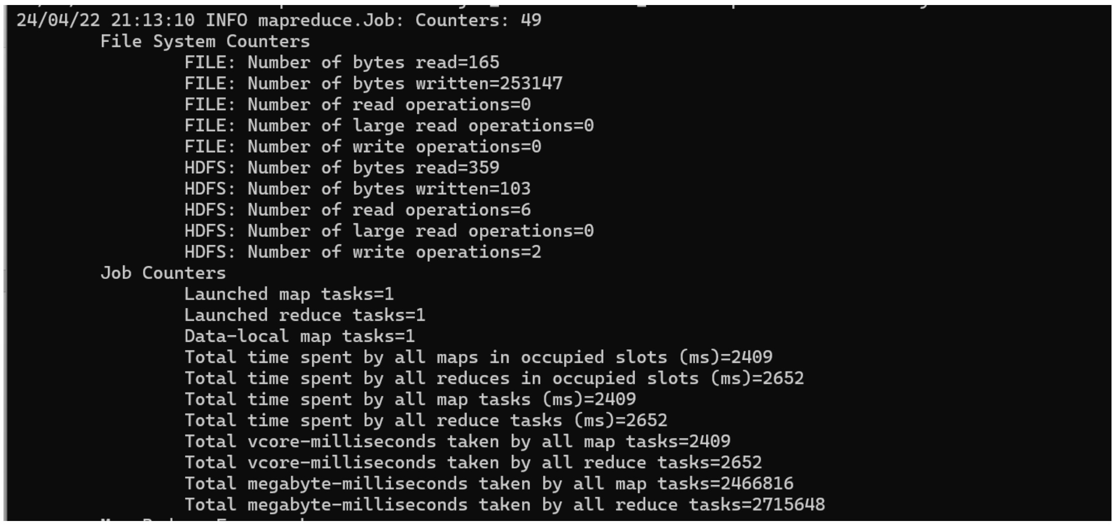


```
# 127.0.0.1     localhost       localhost.localdomain   localhost4      localh
# 127.0.0.1     localhost       localhost
# 127.0.0.1     cpy-2021211138  cpy-2021211138

# 192.168.0.30   node1
# 120.46.149.118 node2
# 1.92.86.3      node3
# 120.46.87.42   node4

192.168.0.30    node1
192.168.0.213   node2
192.168.0.161   node3
192.168.0.135   node4

192.168.0.30    node1   node1
# 127.0.0.1     cpy-2021211138  cpy-2021211138
```

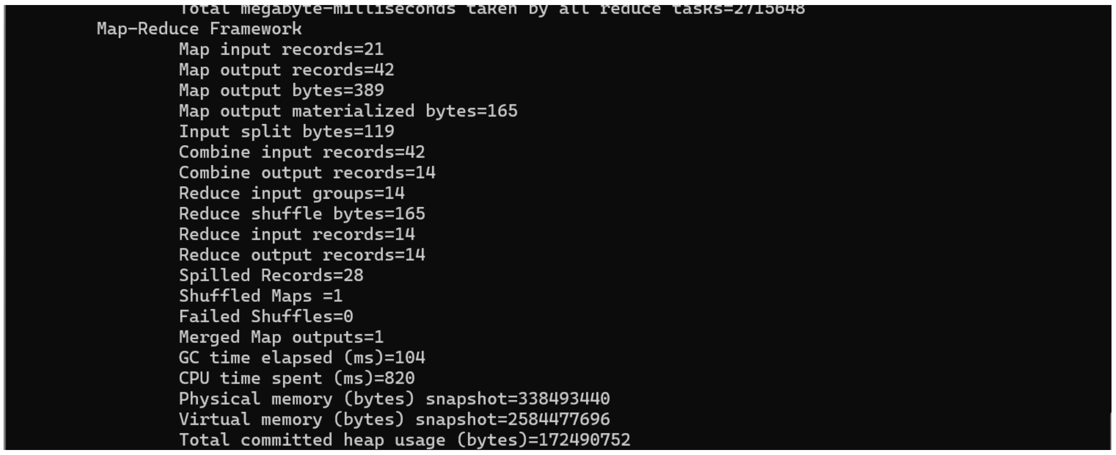
因为服务器每次启动都会自己添加一个 127.0.0.1 的配置，所以它这里自动多了一行127.0.0.1 的配置信息。这也是为什么我始终运行不成功的原因了。把这一行删掉，或者注释掉之后，重新执行 jar 指令。执行过程及结果如下图所示：



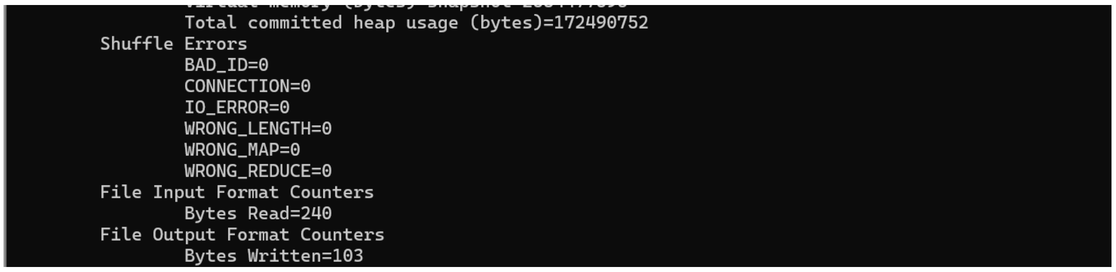
```
[root@cpy-2021211138 ~]# hadoop jar MyWordCount.jar WordCount /experiment2/2021211138-cpy-input.txt /exp
eriment2/2021211138-cpy-output.txt
24/04/22 21:12:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
using builtin-java classes where applicable
24/04/22 21:12:46 INFO client.RMProxy: Connecting to ResourceManager at node1/192.168.0.30:8032
24/04/22 21:12:48 INFO input.FileInputFormat: Total input paths to process : 1
24/04/22 21:12:48 INFO mapreduce.JobSubmitter: number of splits:1
24/04/22 21:12:48 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1713791462926_0001
24/04/22 21:12:48 INFO impl.YarnClientImpl: Submitted application application_1713791462926_0001
24/04/22 21:12:48 INFO mapreduce.Job: The url to track the job: http://node1:8088/proxy/application_1713
791462926_0001/
24/04/22 21:12:48 INFO mapreduce.Job: Running job: job_1713791462926_0001
24/04/22 21:13:00 INFO mapreduce.Job: Job job_1713791462926_0001 running in uber mode : false
24/04/22 21:13:00 INFO mapreduce.Job:  map 0% reduce 0%
24/04/22 21:13:04 INFO mapreduce.Job:  map 100% reduce 0%
24/04/22 21:13:09 INFO mapreduce.Job:  map 100% reduce 100%
24/04/22 21:13:10 INFO mapreduce.Job: Job job_1713791462926_0001 completed successfully
24/04/22 21:13:10 INFO mapreduce.Job: Counters: 49
```



```
24/04/22 21:13:10 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=165
                FILE: Number of bytes written=253147
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=359
                HDFS: Number of bytes written=103
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=2409
                Total time spent by all reduces in occupied slots (ms)=2652
                Total time spent by all map tasks (ms)=2409
                Total time spent by all reduce tasks (ms)=2652
                Total vcore-milliseconds taken by all map tasks=2409
                Total vcore-milliseconds taken by all reduce tasks=2652
                Total megabyte-milliseconds taken by all map tasks=2466816
                Total megabyte-milliseconds taken by all reduce tasks=2715648
```



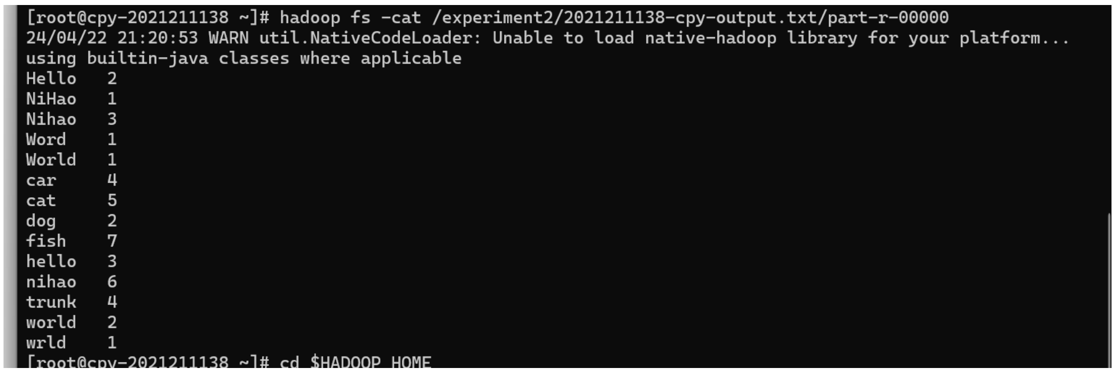
```
                Total megabyte-milliseconds taken by all reduce tasks=2715648
        Map-Reduce Framework
                Map input records=21
                Map output records=42
                Map output bytes=389
                Map output materialized bytes=165
                Input split bytes=119
                Combine input records=42
                Combine output records=14
                Reduce input groups=14
                Reduce shuffle bytes=165
                Reduce input records=14
                Reduce output records=14
                Spilled Records=28
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=104
                CPU time spent (ms)=820
                Physical memory (bytes) snapshot=338493440
                Virtual memory (bytes) snapshot=2584477696
                Total committed heap usage (bytes)=172490752
```



```
                Total committed heap usage (bytes)=172490752
        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=240
        File Output Format Counters
                Bytes Written=103
```

最后执行命令：

hadoop fs -cat /experiment2/2021211138-cpy-output.txt/part-r-00000，结果如下：



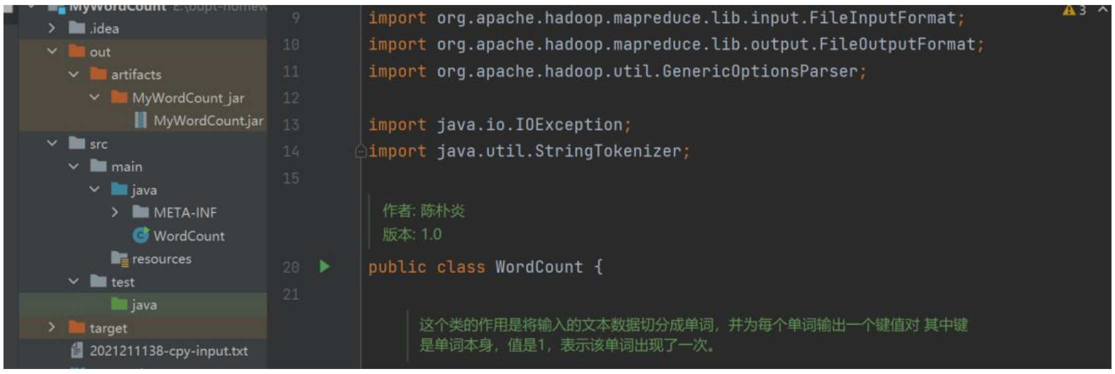
```
[root@cpy-2021211138 ~]# hadoop fs -cat /experiment2/2021211138-cpy-output.txt/part-r-00000
24/04/22 21:20:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
using builtin-java classes where applicable
Hello   2
NiHao   1
Nihao   3
Word    1
World   1
car     4
cat     5
dog     2
fish    7
hello   3
nihao   6
trunk   4
world   2
wrld    1
[root@cpy-2021211138 ~]# cd $HADOOP_HOME
```

实验就顺利完成了。

## 2 实验结果与分析

其实实验结果与分析已经在 1 实验流程里都有了。这里我再次放一下。

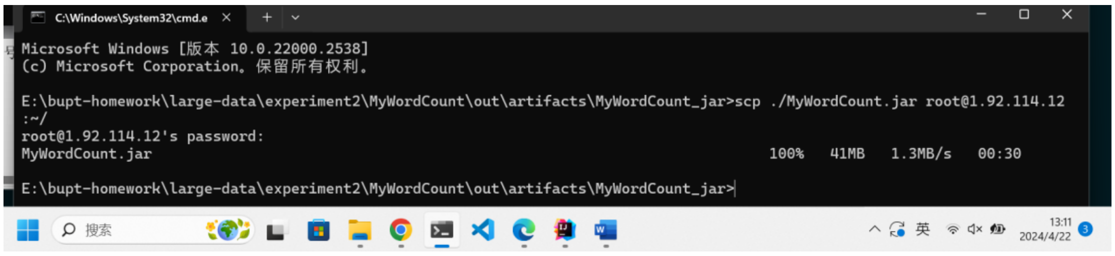
### 2.1 jar 包生成图



```java
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

import java.io.IOException;
import java.util.StringTokenizer;

  作者: 陈朴炎
  版本: 1.0
public class WordCount {

     这个类的作用是将输入的文本数据切分成单词，并为每个单词输出一个键值对 其中键
     是单词本身，值是1，表示该单词出现了一次。
```

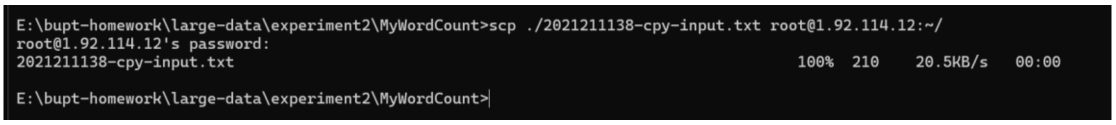
### 2.2 执行命令截图

1. 发送 jar 包



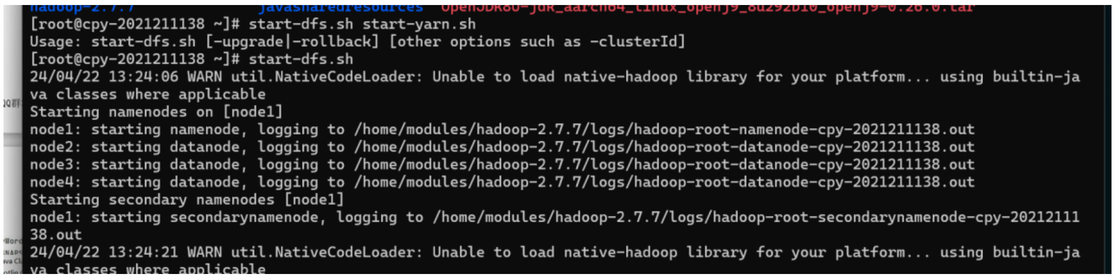
```
Microsoft Windows [版本 10.0.22000.2538]
(c) Microsoft Corporation。保留所有权利。

E:\bupt-homework\large-data\experiment2\MyWordCount\out\artifacts\MyWordCount_jar>scp ./MyWordCount.jar root@1.92.114.12
:~/
root@1.92.114.12's password:
MyWordCount.jar                                                      100%   41MB   1.3MB/s   00:30

E:\bupt-homework\large-data\experiment2\MyWordCount\out\artifacts\MyWordCount_jar>
```

2. 发送 input 文件



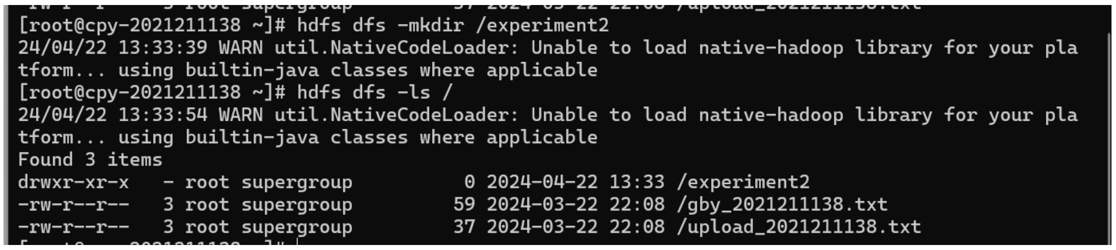
```
E:\bupt-homework\large-data\experiment2\MyWordCount>scp ./2021211138-cpy-input.txt root@1.92.114.12:~/
root@1.92.114.12's password:
2021211138-cpy-input.txt                                                   100%  210    20.5KB/s   00:00

E:\bupt-homework\large-data\experiment2\MyWordCount>
```

3．启动 hadoop 集群



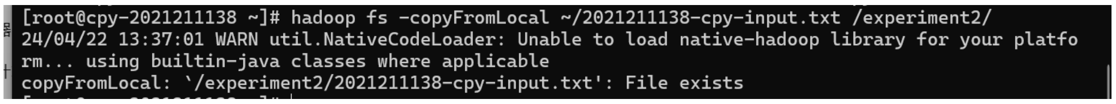
```
[root@cpy-2021211138 ~]# start-dfs.sh start-yarn.sh
Usage: start-dfs.sh [-upgrade|-rollback] [other options such as -clusterId]
[root@cpy-2021211138 ~]# start-dfs.sh
24/04/22 13:24:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-ja
va classes where applicable
Starting namenodes on [node1]
node1: starting namenode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-namenode-cpy-2021211138.out
node2: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
node3: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
node4: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
Starting secondary namenodes [node1]
node1: starting secondarynamenode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-secondarynamenode-cpy-20212111
38.out
24/04/22 13:24:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-ja
va classes where applicable
```

4．查看 hdfs 根目录，新建目录 experiment2



```
[root@cpy-2021211138 ~]# hdfs dfs -mkdir /experiment2
24/04/22 13:33:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
[root@cpy-2021211138 ~]# hdfs dfs -ls /
24/04/22 13:33:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pla
tform... using builtin-java classes where applicable
Found 3 items
drwxr-xr-x   - root supergroup          0 2024-04-22 13:33 /experiment2
-rw-r--r--   3 root supergroup         59 2024-03-22 22:08 /gby_2021211138.txt
-rw-r--r--   3 root supergroup         37 2024-03-22 22:08 /upload_2021211138.txt
```

5．将输入文件放入到新建的目录下

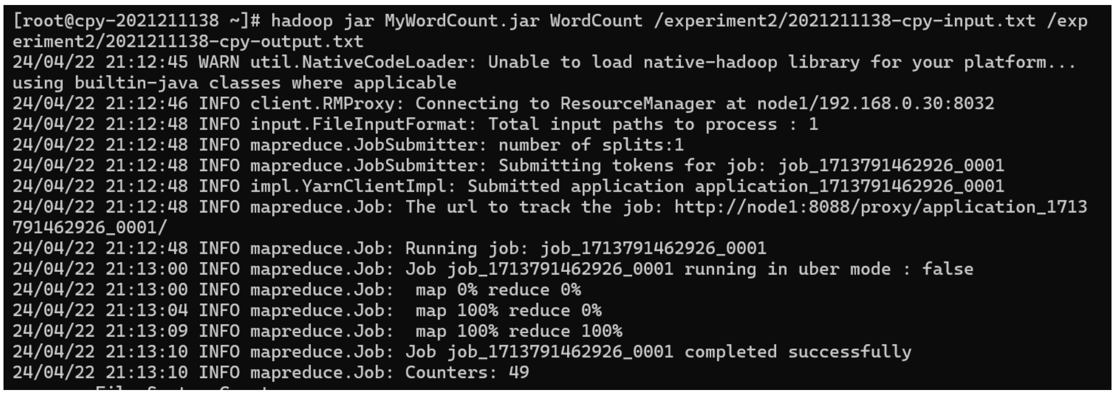


```
[root@cpy-2021211138 ~]# hadoop fs -copyFromLocal ~/2021211138-cpy-input.txt /experiment2/
24/04/22 13:37:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platfo
rm... using builtin-java classes where applicable
copyFromLocal: `/experiment2/2021211138-cpy-input.txt': File exists
```

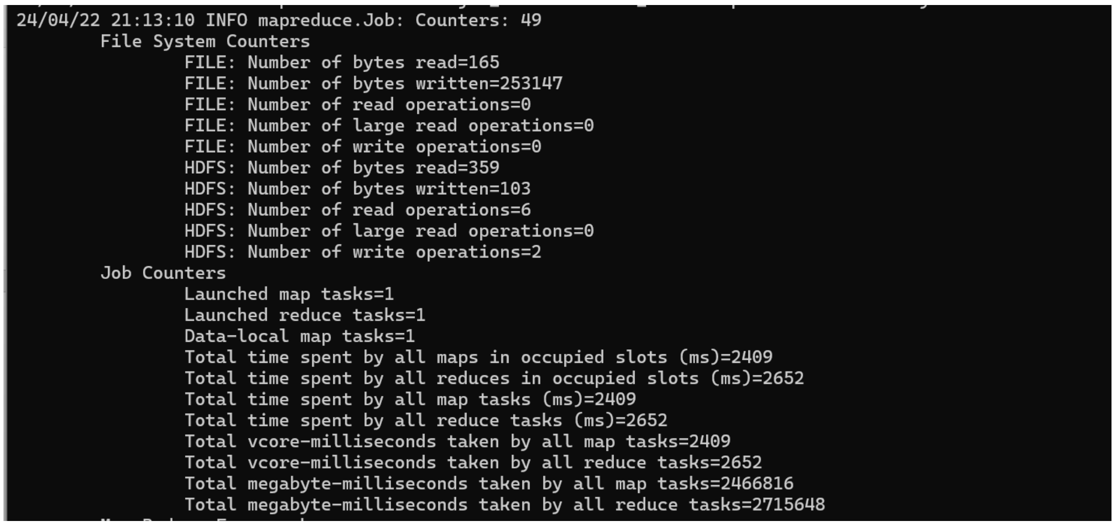
## 2.3 步骤 8 执行命令结果图

第 8 步这里实验指导手册里命令格式漏了一个好像，应该如下所示：

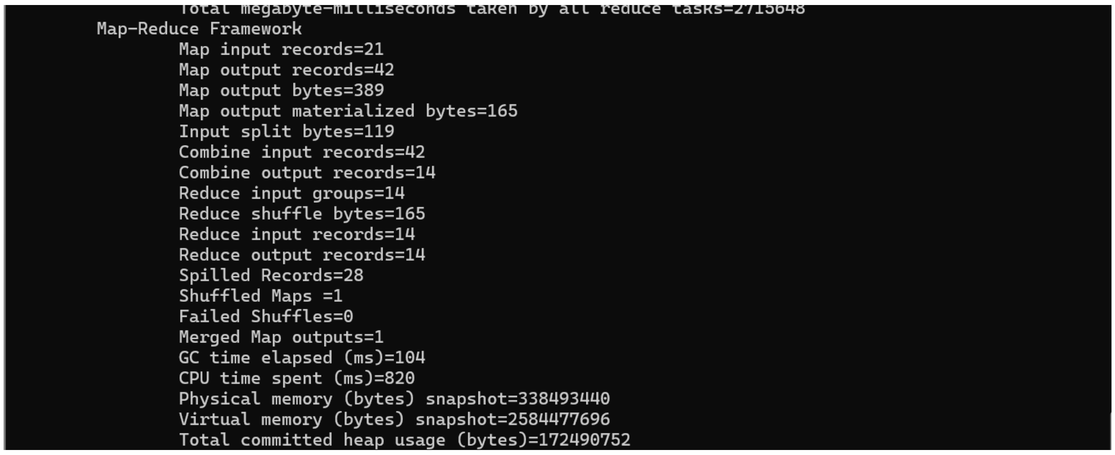
hadoop jar <jar 包> <main 函数的入口类> <input 文件或路径> <output 路径>。



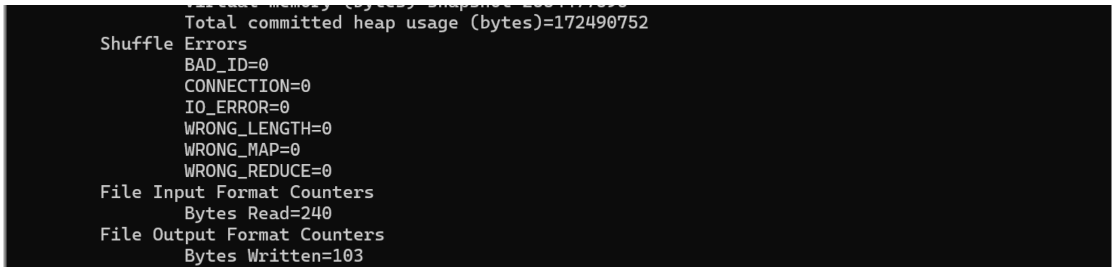
```
[root@cpy-2021211138 ~]# hadoop jar MyWordCount.jar WordCount /experiment2/2021211138-cpy-input.txt /exp
eriment2/2021211138-cpy-output.txt
24/04/22 21:12:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
using builtin-java classes where applicable
24/04/22 21:12:46 INFO client.RMProxy: Connecting to ResourceManager at node1/192.168.0.30:8032
24/04/22 21:12:48 INFO input.FileInputFormat: Total input paths to process : 1
24/04/22 21:12:48 INFO mapreduce.JobSubmitter: number of splits:1
24/04/22 21:12:48 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1713791462926_0001
24/04/22 21:12:48 INFO impl.YarnClientImpl: Submitted application application_1713791462926_0001
24/04/22 21:12:48 INFO mapreduce.Job: The url to track the job: http://node1:8088/proxy/application_1713
791462926_0001/
24/04/22 21:12:48 INFO mapreduce.Job: Running job: job_1713791462926_0001
24/04/22 21:13:00 INFO mapreduce.Job: Job job_1713791462926_0001 running in uber mode : false
24/04/22 21:13:00 INFO mapreduce.Job:  map 0% reduce 0%
24/04/22 21:13:04 INFO mapreduce.Job:  map 100% reduce 0%
24/04/22 21:13:09 INFO mapreduce.Job:  map 100% reduce 100%
24/04/22 21:13:10 INFO mapreduce.Job: Job job_1713791462926_0001 completed successfully
24/04/22 21:13:10 INFO mapreduce.Job: Counters: 49
```



```
24/04/22 21:13:10 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=165
                FILE: Number of bytes written=253147
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=359
                HDFS: Number of bytes written=103
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=2409
                Total time spent by all reduces in occupied slots (ms)=2652
                Total time spent by all map tasks (ms)=2409
                Total time spent by all reduce tasks (ms)=2652
                Total vcore-milliseconds taken by all map tasks=2409
                Total vcore-milliseconds taken by all reduce tasks=2652
                Total megabyte-milliseconds taken by all map tasks=2466816
                Total megabyte-milliseconds taken by all reduce tasks=2715648
```



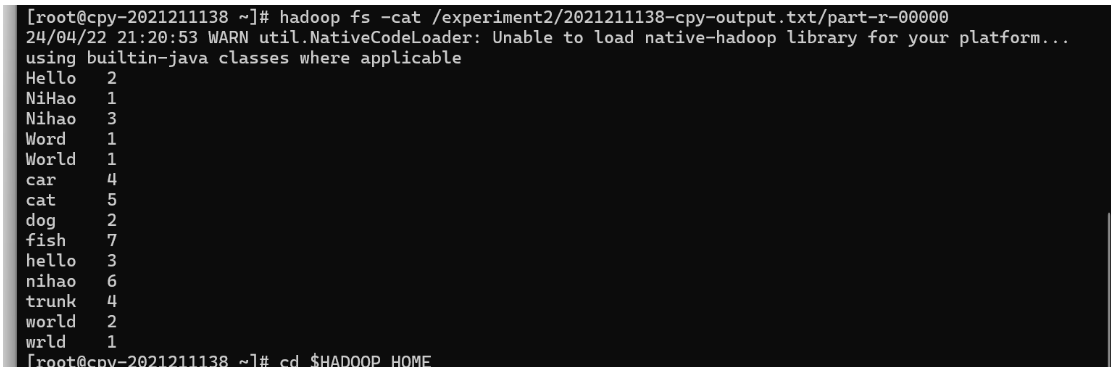
```
        Map-Reduce Framework
                Map input records=21
                Map output records=42
                Map output bytes=389
                Map output materialized bytes=165
                Input split bytes=119
                Combine input records=42
                Combine output records=14
                Reduce input groups=14
                Reduce shuffle bytes=165
                Reduce input records=14
                Reduce output records=14
                Spilled Records=28
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=104
                CPU time spent (ms)=820
                Physical memory (bytes) snapshot=338493440
                Virtual memory (bytes) snapshot=2584477696
                Total committed heap usage (bytes)=172490752
```



```
                Total committed heap usage (bytes)=172490752
```

```
        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=240
        File Output Format Counters
                Bytes Written=103
```

## 2.4 步骤 9 执行命令结果图

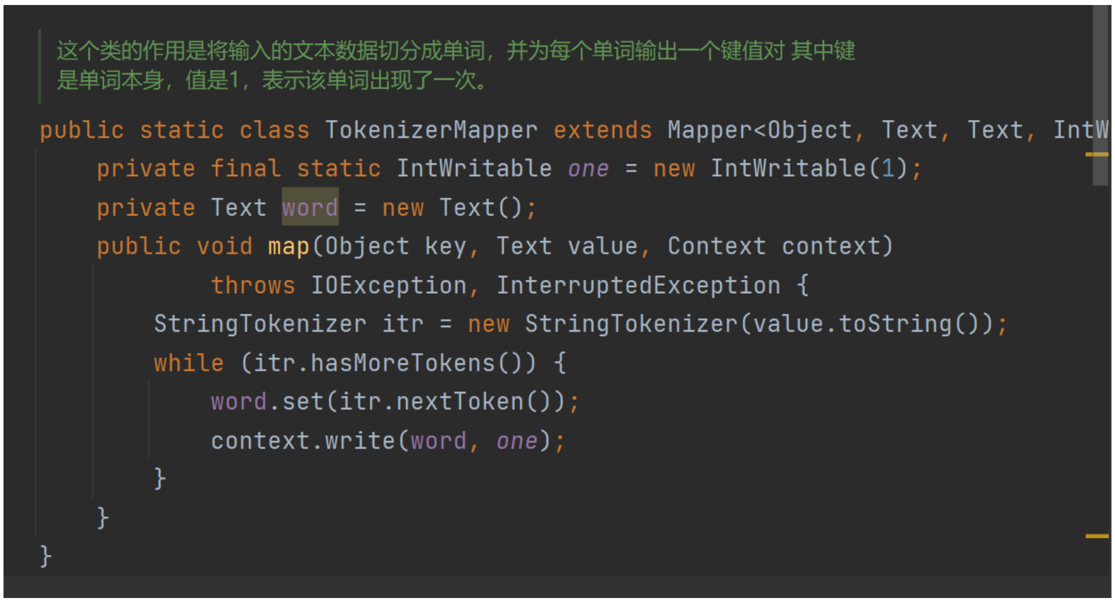


```
[root@cpy-2021211138 ~]# hadoop fs -cat /experiment2/2021211138-cpy-output.txt/part-r-00000
24/04/22 21:20:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
using builtin-java classes where applicable
Hello   2
NiHao   1
Nihao   3
Word    1
World   1
car     4
cat     5
dog     2
fish    7
hello   3
nihao   6
trunk   4
world   2
wrld    1
[root@cpy-2021211138 ~]# cd $HADOOP_HOME
```

## 2.5 提交 jar 包（另附件）

## 2.6 解释 WordCount 程序代码

类 TokenizerMapper:这个类的作用是将输入的文本数据切分成单词,并为每个单词输出一个键值对 其中键是单词本身，值是 1，表示该单词出现了一次。



```
这个类的作用是将输入的文本数据切分成单词，并为每个单词输出一个键值对 其中键
是单词本身，值是1，表示该单词出现了一次。
public static class TokenizerMapper extends Mapper<Object, Text, Text, IntW
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    public void map(Object key, Text value, Context context)
            throws IOException, InterruptedException {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
        }
    }
}
```

类 IntSumReducer: 将 Mapper 输出的中间结果进行合并和汇总

Reduce 函数:

形参:

key – 当前要计算的键

values – 输入值的迭代器

context – 上下文对象，用于将结果输出

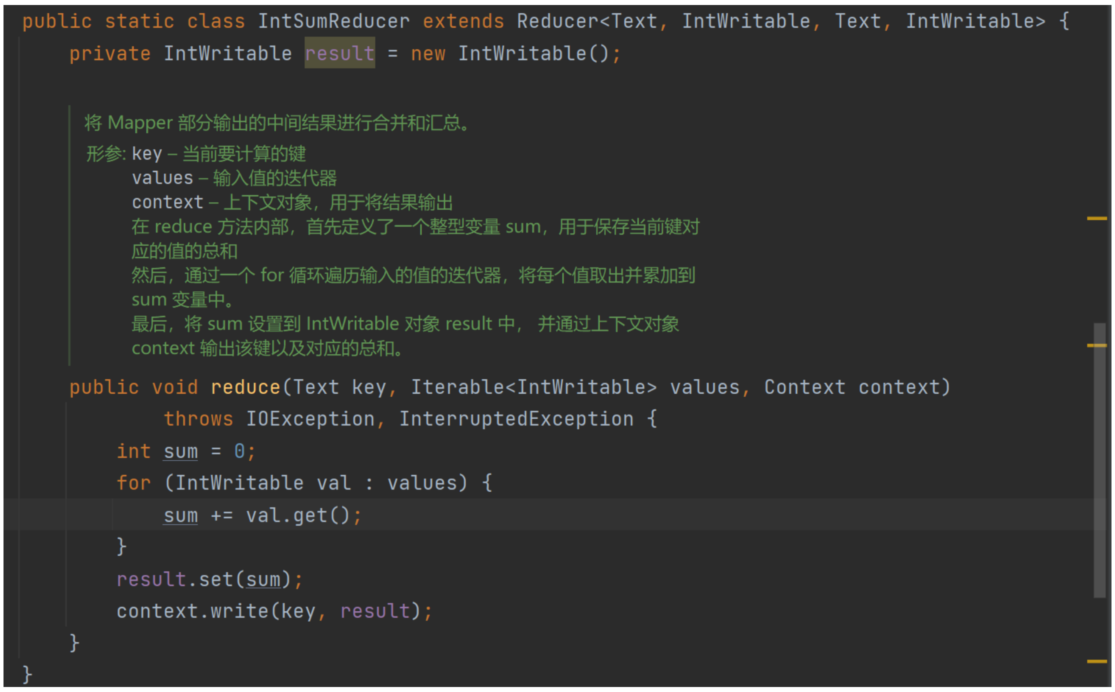
函数概述:

在 reduce 方法内部，首先定义了一个整型变量 sum，用于保存当前键对应的值的总和

然后，通过一个 for 循环遍历输入的值的迭代器，将每个值取出并累加到

sum 变量中。

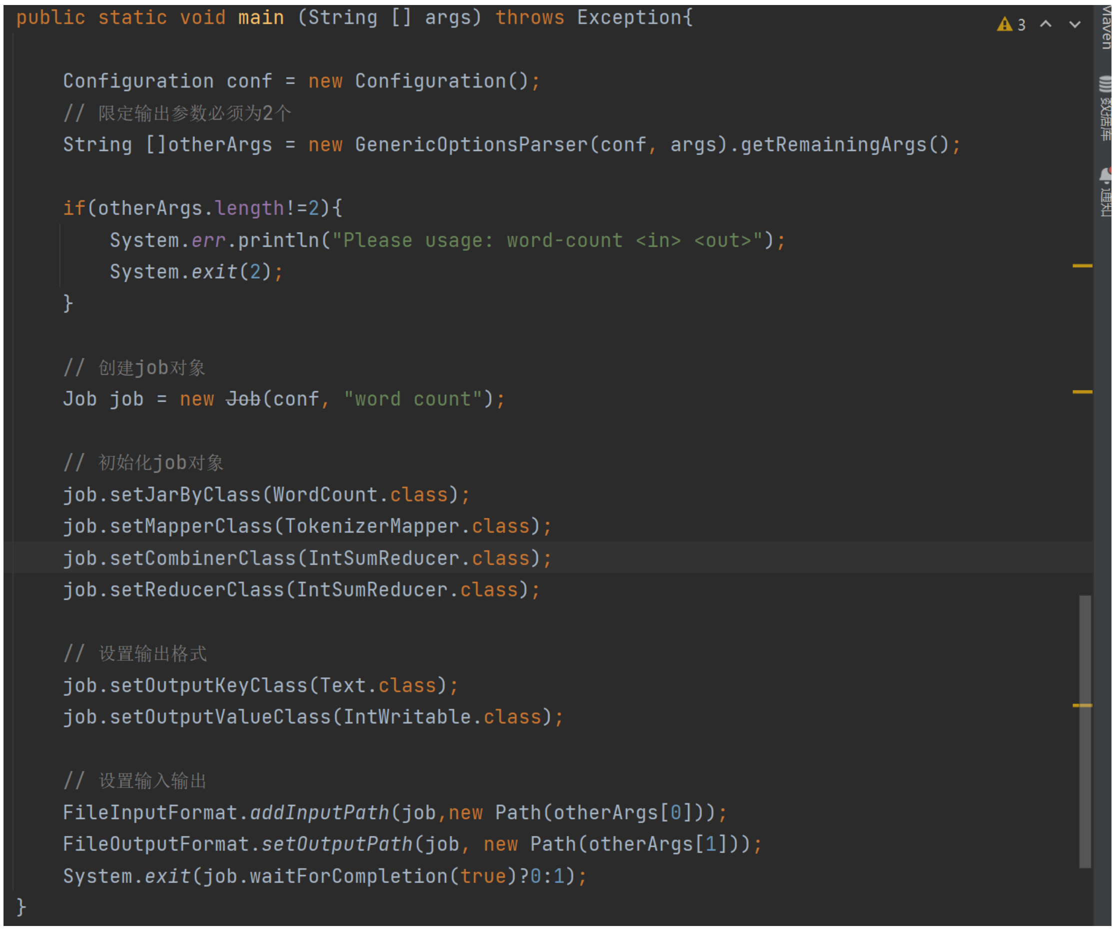
  最后，将 sum 设置到 IntWritable 对象 result 中， 并通过上下文对象 context 输出该键以及对应的总和。



```java
public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

    将 Mapper 部分输出的中间结果进行合并和汇总。
    形参: key – 当前要计算的键
          values – 输入值的迭代器
          context – 上下文对象，用于将结果输出
          在 reduce 方法内部，首先定义了一个整型变量 sum，用于保存当前键对
          应的值的总和
          然后，通过一个 for 循环遍历输入的值的迭代器，将每个值取出并累加到
          sum 变量中。
          最后，将 sum 设置到 IntWritable 对象 result 中， 并通过上下文对象
          context 输出该键以及对应的总和。

    public void reduce(Text key, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

主函数的运行逻辑为：

1. 判断输入的参数格式是否正确
2. 创建 job 对象，并为 job 对象初始化
3. 设置输出的 Key 和 Value
4. 设置输入输出的路径



```java
public static void main (String [] args) throws Exception{

    Configuration conf = new Configuration();
    // 限定输出参数必须为2个
    String []otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();

    if(otherArgs.length!=2){
        System.err.println("Please usage: word-count <in> <out>");
        System.exit(2);
    }

    // 创建job对象
    Job job = new ~~Job~~(conf, "word count");

    // 初始化job对象
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);

    // 设置输出格式
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    // 设置输入输出
    FileInputFormat.addInputPath(job,new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true)?0:1);
}
```