第4章MapReduce编程入门

(源自:https://biglab.site)

(版本:Ver1.4-20231024)

任务前言

学习目标

1. 掌握如何搭建MapReduce开发环境。
2. 掌握以Eclipse创建MapReduce工程。
3. 理解MapReduce的基本原理及执行流程。
4. 读懂Hadoop官方示例WordCount的源码。
5. 掌握MapReduce编程的基本思路。
6. 理解map函数与reduce函数的处理逻辑。
7. 能够编写MapReduce程序处理简单任务。

任务背景

随着互联网的发展，加入互联网的用户越来越多，互联网的用户规模已不容小视。互联网市场潜力巨大，各大网站的运营商都在采取积极措施，分析用户的特征，根据不同的客户群向其提供差异化的服务，进而达到精准营销的目的。随着一些网站用户的增加，企业越来越难把握用户的需求。为了能更好地满足用户需求，应依据用户的历史浏览记录研究用户的兴趣偏好，分析用户的需求和行为，发现用户的兴趣点，从而将用户分成不同的群体。企业再根据不同的群体提供差异化的服务，改善用户体验。

网站的访问次数分布情况对网站运营商而言也是非常重要的指标之一，网站运营商从数据库中抽取了网站2020年5月至2021年2月用户登录网站的行为日志数据，针对用户访问网站的日志数据，网站运营方提出了两个统计需求。

1. 根据访问时间统计网站每日的总访问次数，按访问日期输出结果。

2. 对统计需求（1）的结果再进行处理，将结果按访问次数进行升序排序

本章将详细讲解使用MapReduce编程解决实际问题。

1. 首先介绍MapReduce开发环境的搭建过程。
2. 接着介绍MapReduce编程原理与执行流程。
3. 结合Hadoop官方的示例源码WordCount介绍MapReduce编程的基本思路与处理逻辑。
4. 最后通过编写MapReduce程序实现竞赛网站每日访问次数的统计，并对统计的结果根据访问次数进行升序排序

软件准备

下载地址：链接: https://pan.baidu.com/s/1DHGRrnjvi4yMY41Se7_vDA 提取码: vbgu

软件列表：

软件	版本	安装包称
IntelliJ IDEA	2018.3.6	ideaIC-2018.3.6.exe
JDK	1.8	jdk-8u281-linux-x64.rpm, jdk-8u281-windows-x64.exe
winutils	3.1.4	winutils.rar
Hadoop	3.1.4	hadoop-3.1.4.tar.gz
Git	2.39.2	Git-2.39.2-64-bit.exe
TortoiseGit	2.14	TortoiseGit-2.14.0.0-64bit.msi
maven	3.8.8	apache-maven-3.8.8-bin.zip

任务4.1在IDEA中搭建MapReduce开发环境

Hadoop框架是基于Java语言开发的，而IntelliJ IDEA是一个常用的Java集成开发工具，因此通常选用IntelliJ IDEA作为MapReduce的编程工具。为了能够成功地进行MapReduce编程，本小节的任务如下。

1. 在本机系统（通常是Windows系统）安装Java。
2. 安装IntelliJ IDEA工具。
3. 在IntelliJ IDEA中创建一个MapReduce工程，并配置MapReduce集成环境

在Windows下安装Java

安装JDK

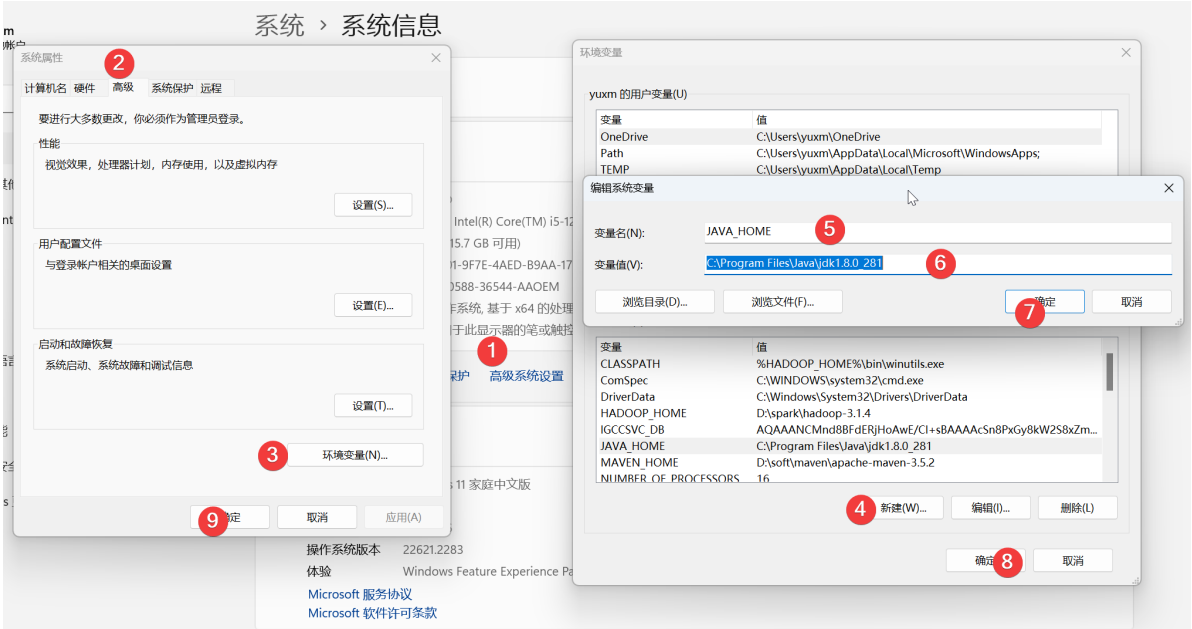
双击JDK安装包jdk-8u281-windows-x64.exe，进入“Java SE 开发工具包”安装向导对话框，单击“下一步”按钮进入安装，然后按默认设置安装直至完成。

添加系统环境变量JAVA_HOME

安装完Java后，需要在Windows系统配置环境变量，只有配置了环境变量，Java编译环境才可以正常使用。在Windows系统配置环境变量的操作步骤如下。右键单击“此电脑”桌面快捷方式，选择“属性”选项，在出现的系统设置窗口中选择“高级系统设置”选项，进入到“系统属性”对话框，单击“环境变量”按钮，弹出“环境变量”对话框，在下方的系统变量中新建：

变量名：JAVA_HOME

变量值：C:\Program Files\Java\jdk1.8.0_281

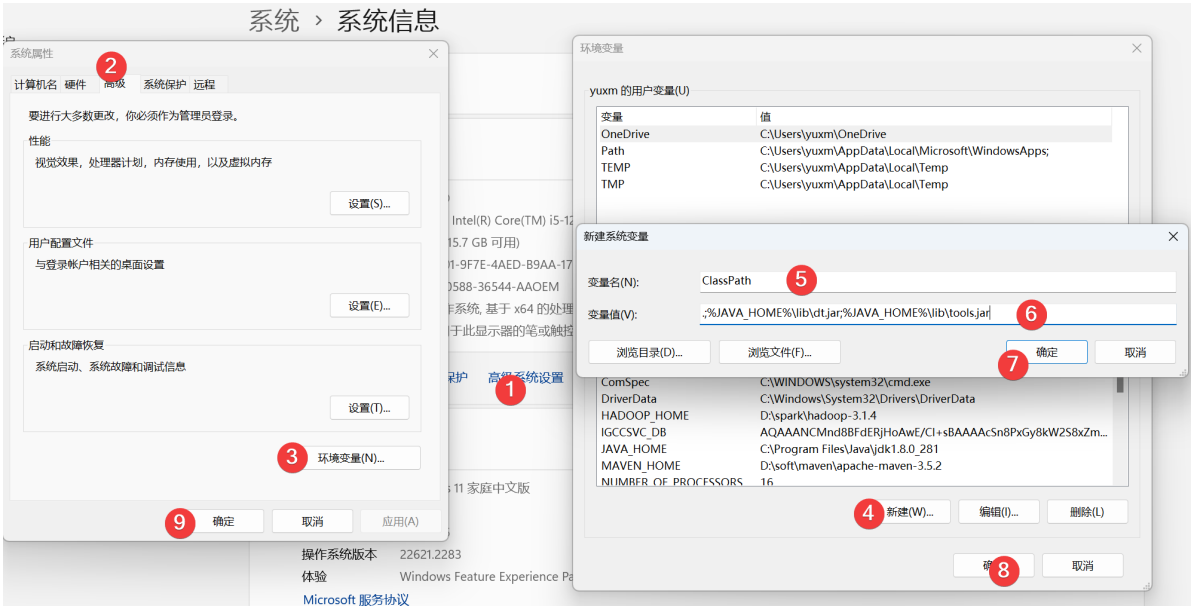


添加系统环境变量ClassPath

进入到“系统属性”对话框，单击“环境变量”按钮，弹出“环境变量”对话框，在下方的系统变量中新建：

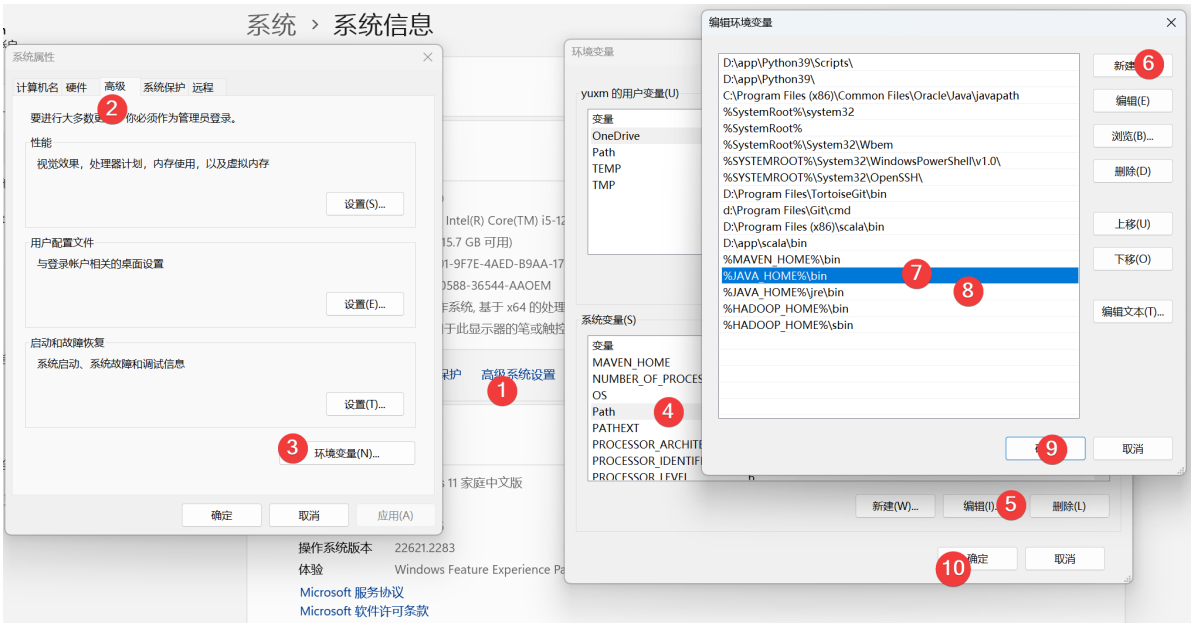
变量名：ClassPath

变量值：.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar



设置系统环境变量PATH

新建： %JAVA_HOME%\bin 和 %JAVA_HOME%\jre\bin

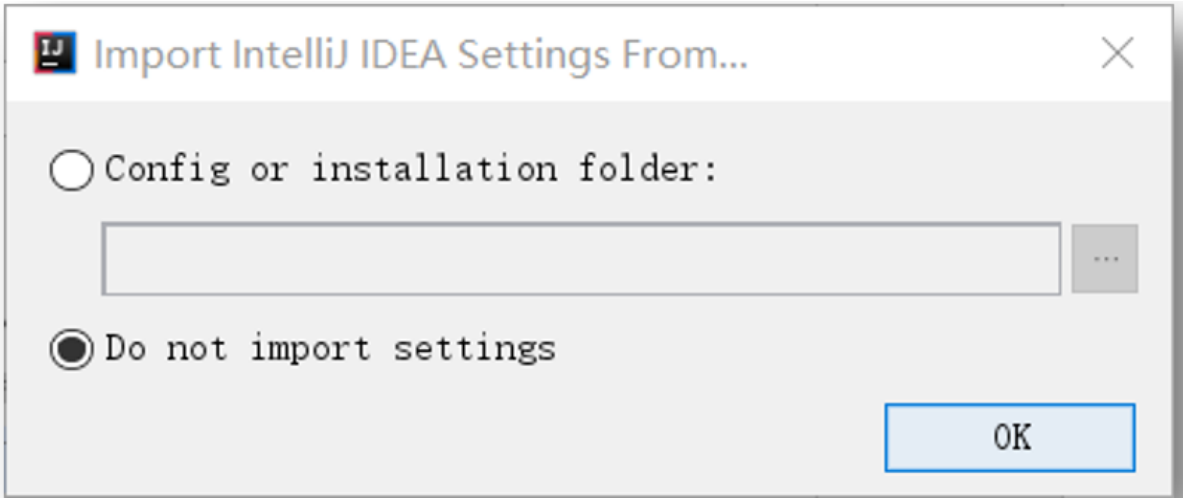


下载与安装IntelliJ IDEA

安装IDEA

使用百度云下载的IntelliJ IDEA，双击运行，使用默认路径安装

双击桌面生成的IDEA工具图标，或从个人计算机的开始菜单中，依次选择“JetBrains”→“IntelliJ IDEA Community Edition 2018.3.6 ”选项，运行IntelliJ IDEA。启动过程中将询问是否导入以前的设定，选择“Do not import settings ”单选按钮，表示不导入，并单击“OK ”按钮进入下一步



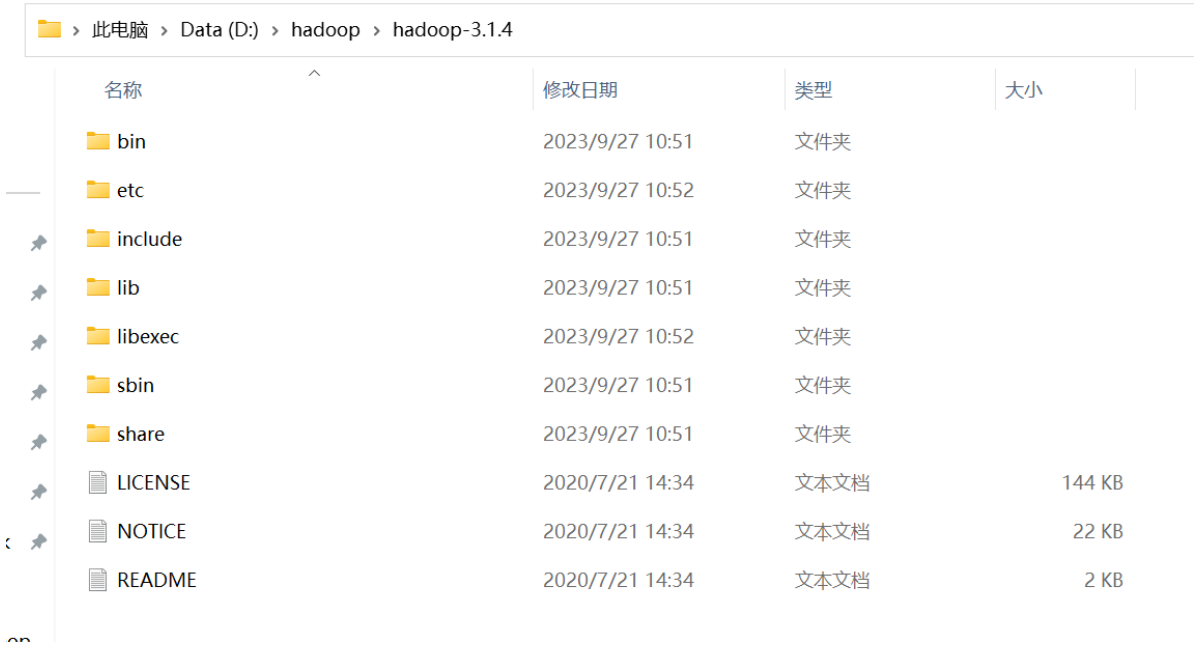
进入下图所示的界面，选择IDEA设计界面的主题，可以选择白色或黑色背景。考虑代码和结果展示的清晰度，单击“Light ”单选按钮，并单击界面左下角的“Skip Remaining and Set Defaults ”按钮，跳过其他设置并采用默认设置。设置完成后，即可进入IDEA的运行界面。

解压hadoop并配置环境变量

解压Hadoop压缩包

从百度网盘下载hadoop-3.1.4.tar.gz 到D:\hadoop目录下

使用winrar将hadoop-3.1.4.tar.gz 解压到当前文件夹下，解压后路径如下：

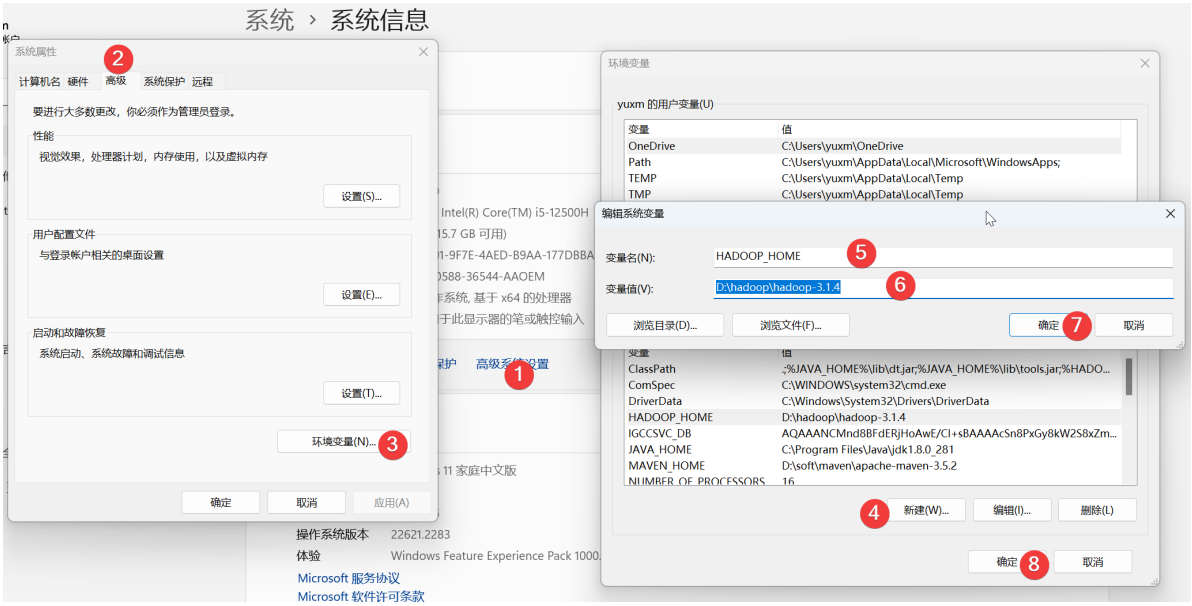


配置环境变量HADOOP_HOME

右击我的电脑-》属性-》高级-》环境变量-》系统变量-》新建

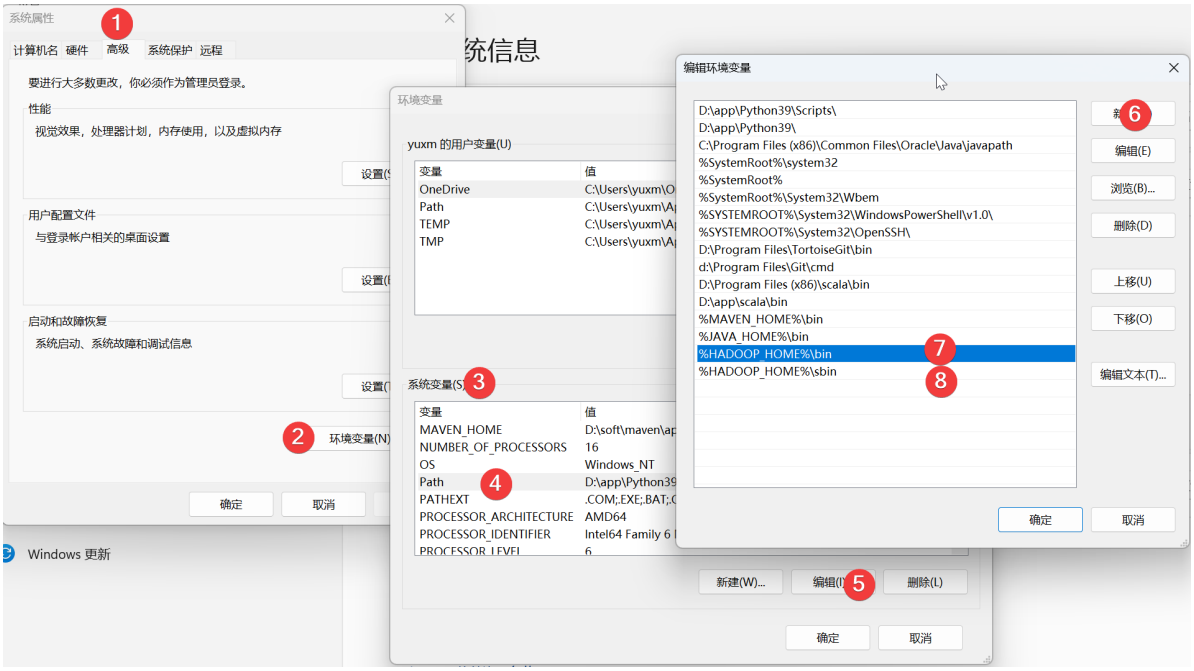
变量名：HADOOP_HOME

变量值：D:\hadoop\hadoop-3.1.4



设置系统环境变量PATH

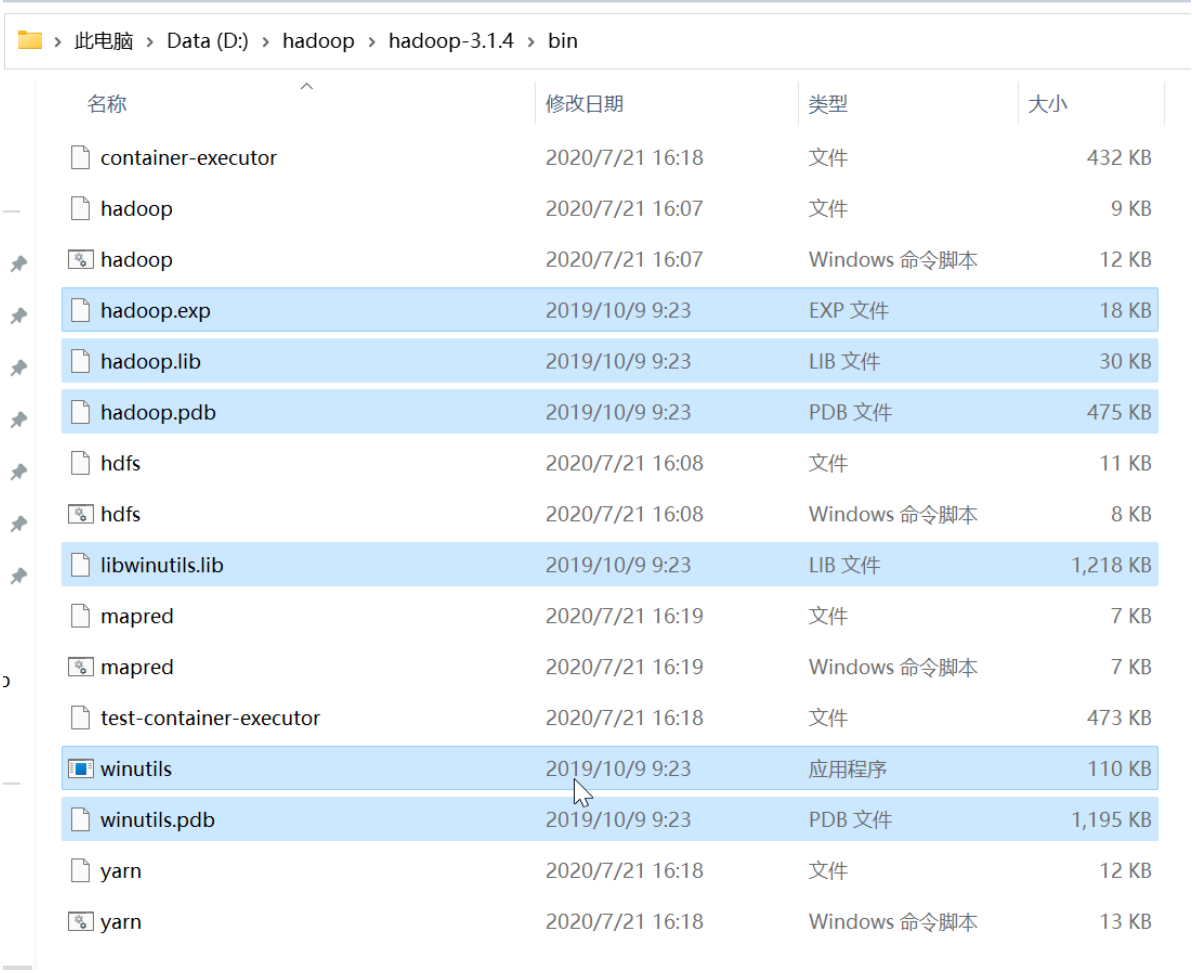
右击我的电脑-》属性-》高级-》环境变量-》系统变量-》修改Path,并添加: %HADOOP_HOME%\bin 和 %HADOOP_HOME%\sbin



配置winutils插件

解压winutils.rar到hadoop的 bin目录下

将从百度网盘下载到的winutils.rar使用winrar解压到D:\hadoop\hadoop-3.1.4\bin 目录下，如下图：解压后文件包括winutils.exe，winutils.pdb,hadoop.exp等

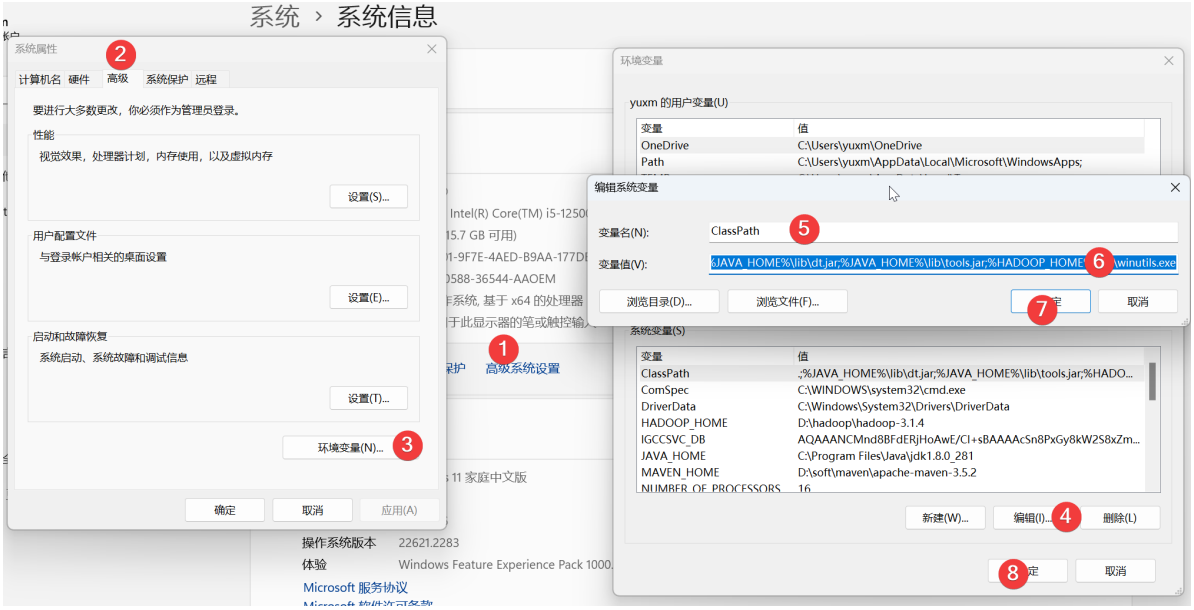


将winutils配置到系统环境变量ClassPath中

进入到“系统属性”对话框，单击“环境变量”按钮，弹出“环境变量”对话框，在下方的系统变量中新建：

变量名：ClassPath

变量值：.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar**;%HADOOP_HOME%\bin\winutils.exe**



安装Maven并配置环境变量

解压maven压缩包

从百度网盘下载apache-maven-3.8.8-bin.zip 到D:\hadoop目录下

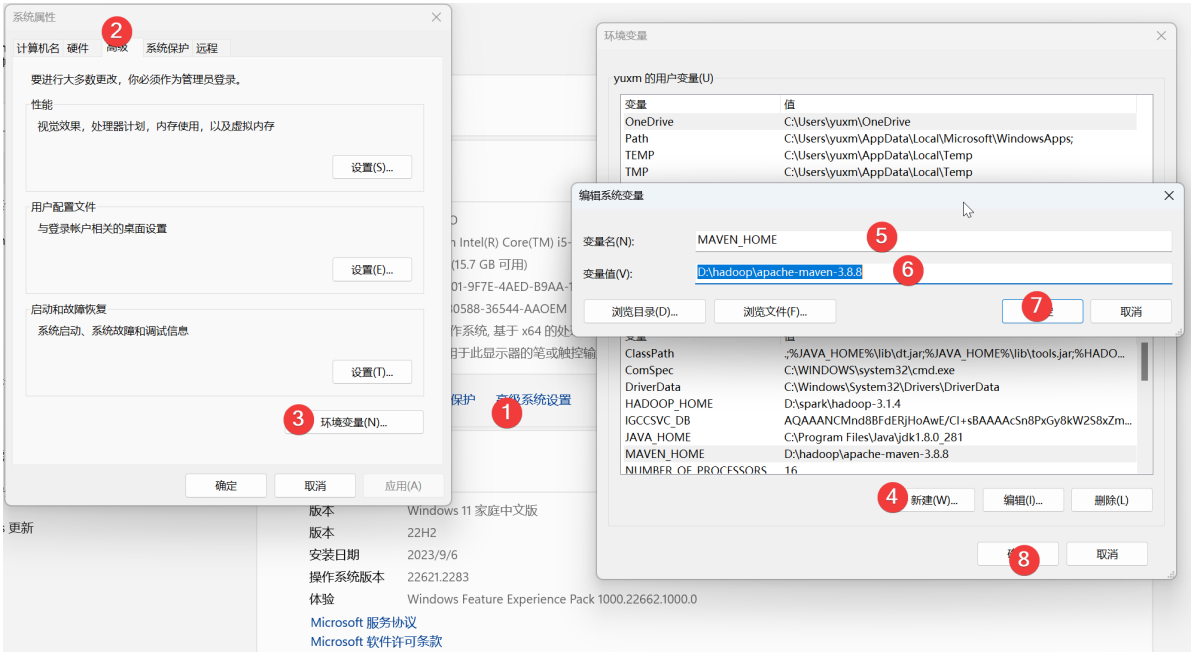
使用winrar将apache-maven-3.8.8-bin.zi 解压到当前文件夹下，解压后路径如下：D:\hadoop\apache-maven-3.8.8

配置环境变量MAVEN_HOME

右击我的电脑-》属性-》高级-》环境变量-》系统变量-》新建

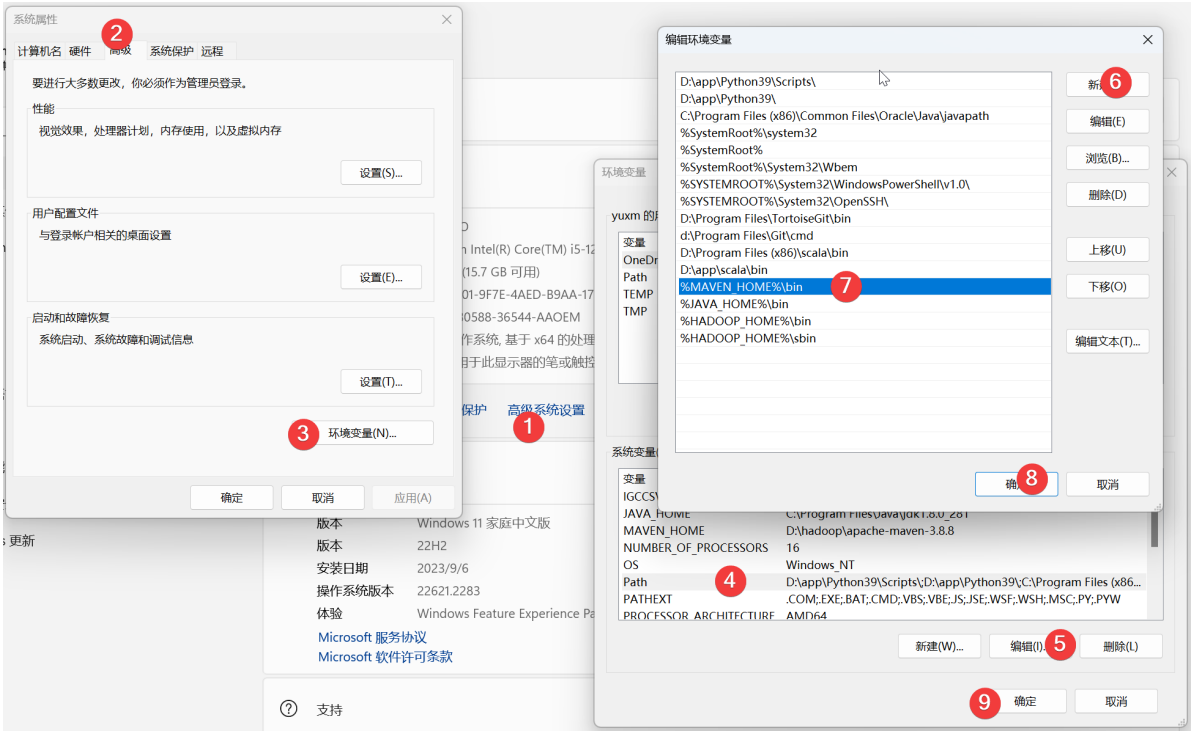
变量名：MAVEN_HOME

变量值：D:\hadoop\apache-maven-3.8.8



设置系统环境变量PATH

右击我的电脑-》属性-》高级-》环境变量-》系统变量-》修改Path,并添加: %MAVEN_HOME%\bin

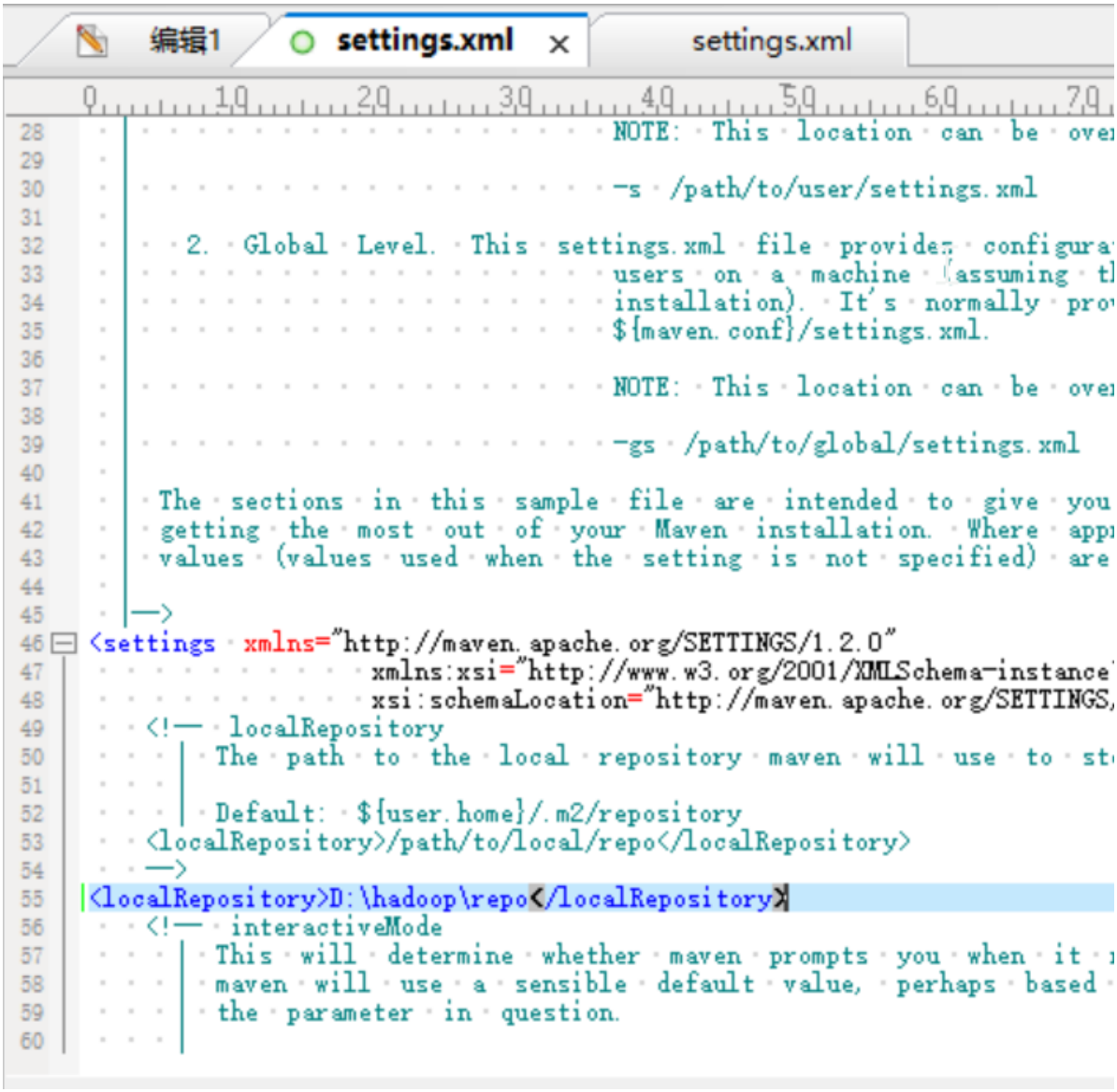


设置MAVEN的settings配置文件

首先，在D:\hadoop目录下创建repo文件夹

然后修改：D:\hadoop\apache-maven-3.8.8\conf\settings.xml，约55行，修改为：

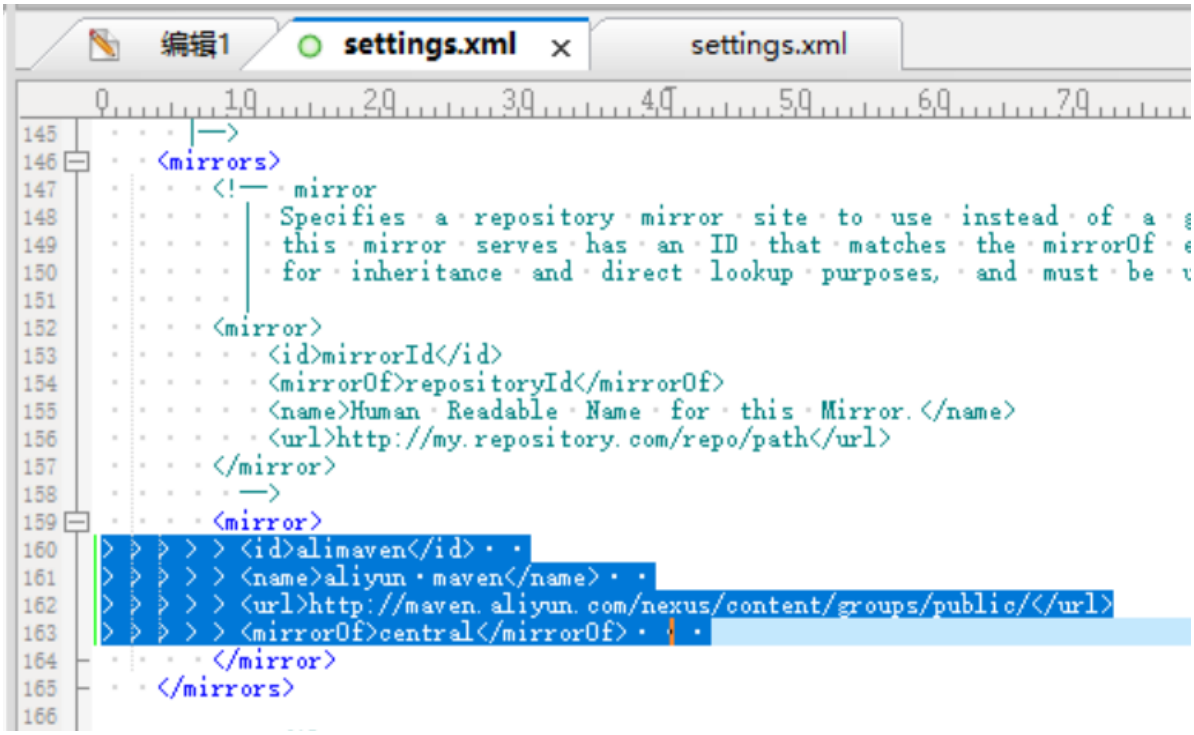
```
<localRepository>D:\hadoop\repo</localRepository>
```



然后，修改160行到164行内容为：

```
    <id>alimaven</id>
    <name>aliyun maven</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
    <mirrorOf>central</mirrorOf>
```

修改后如图：



修改后保存退出。

检查Maven的配置

通过WIN+R，输入cmd，进入命令行，输入 mvn -version ，能正确显示maven的版本号说明正常，如：

```
C:\Users\yuxm>mvn -version
Apache Maven 3.8.8 (4c87b05d9aedce574290d1acc98575ed5eb6cd39)
Maven home: D:\hadoop\apache-maven-3.8.8
Java version: 1.8.0_281, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_281\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\yuxm>
```

新建MapReduce工程

创建Maven项目

安装好IntelliJ IDEA开发工具后，即可在IDEA中创建MapReduce工程。在进入IDEA后，单击“Create New Profile”选项，弹出 “New Project ”对话框，在左侧列表栏中选择“Maven Archetype ”选项，在右侧的工程属性中设置：

Name:Hadoop

Localtion:D:\myname (先在D盘上创建myname目录)

JDK:选择：JDK1.8.0_281

Catalog: 选择：Internal

Archetype:选择：org.apache.maven.archetypes:maven-archetype-quickstart

Version: 选择默认:1.1

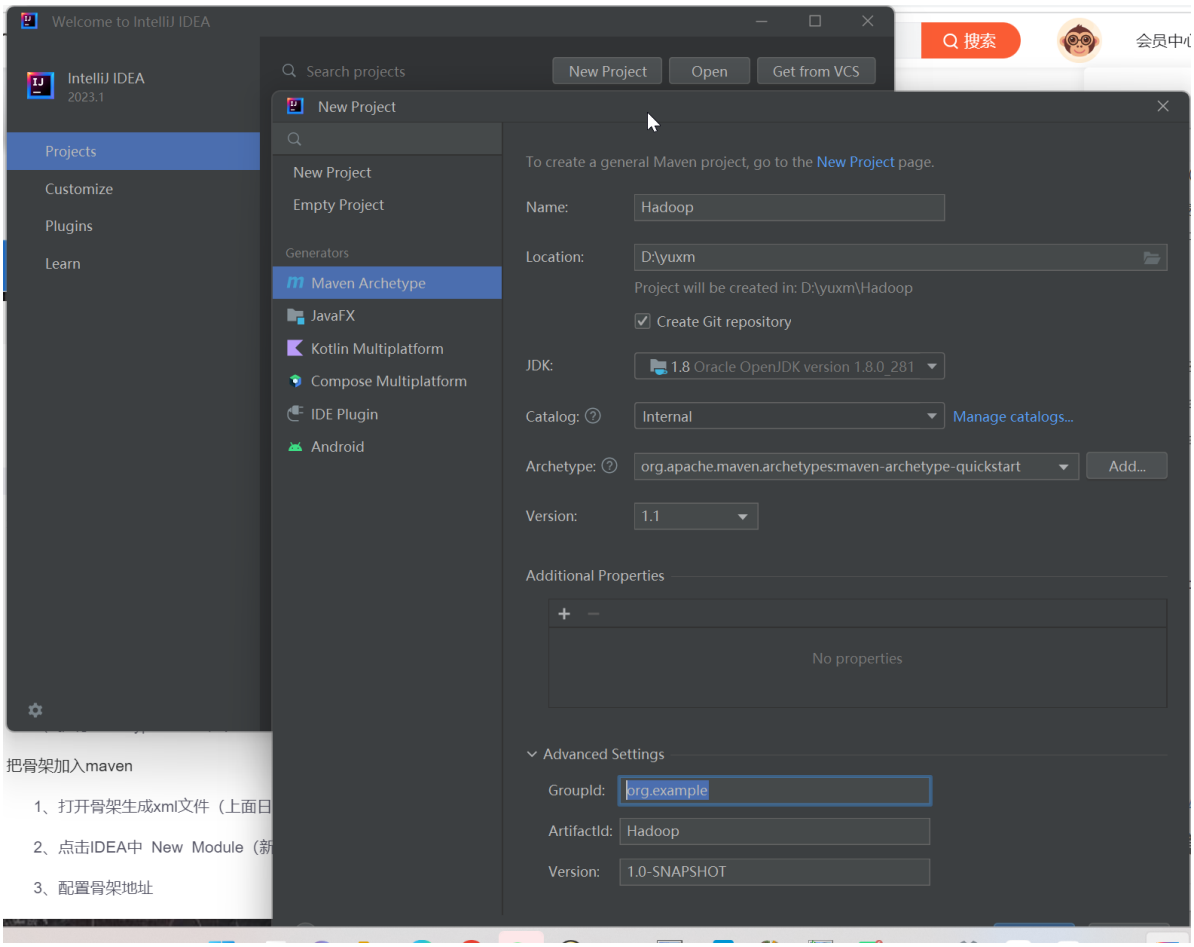
Advanced Settings中

GroupId: 可以修改，或使用默认 org.example

ArtifactId: 使用默认： Hadoop

Versions: 使用默认：1.0-SNAPSHOT

如图：



点击右下方的Create，创建项目

(参考：创建自定义Maven Archetype)

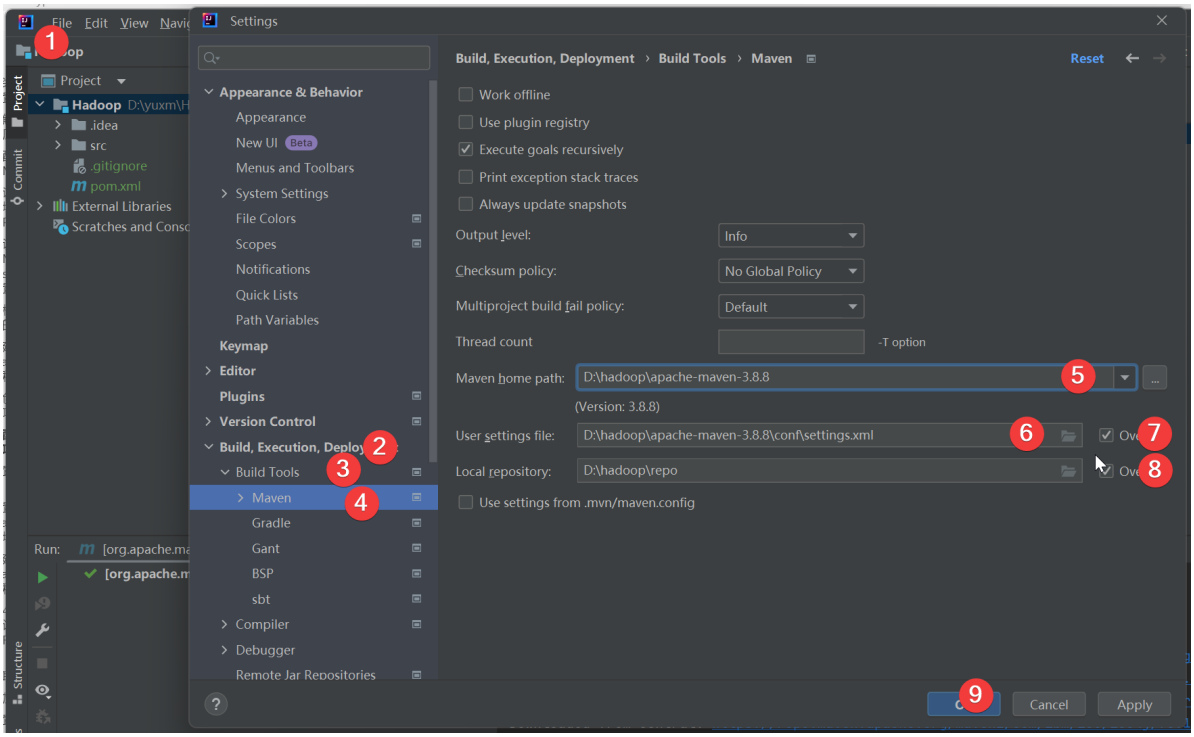
配置Maven项目

进入创建好的项目后，打开菜单：File->Setting,在弹出的设置窗中，选择左侧的树图:Build->Build Tools ->Maven; 然后在右侧中：

Maven home Path:选择：D:\hadoop\apache-maven-3.8.8

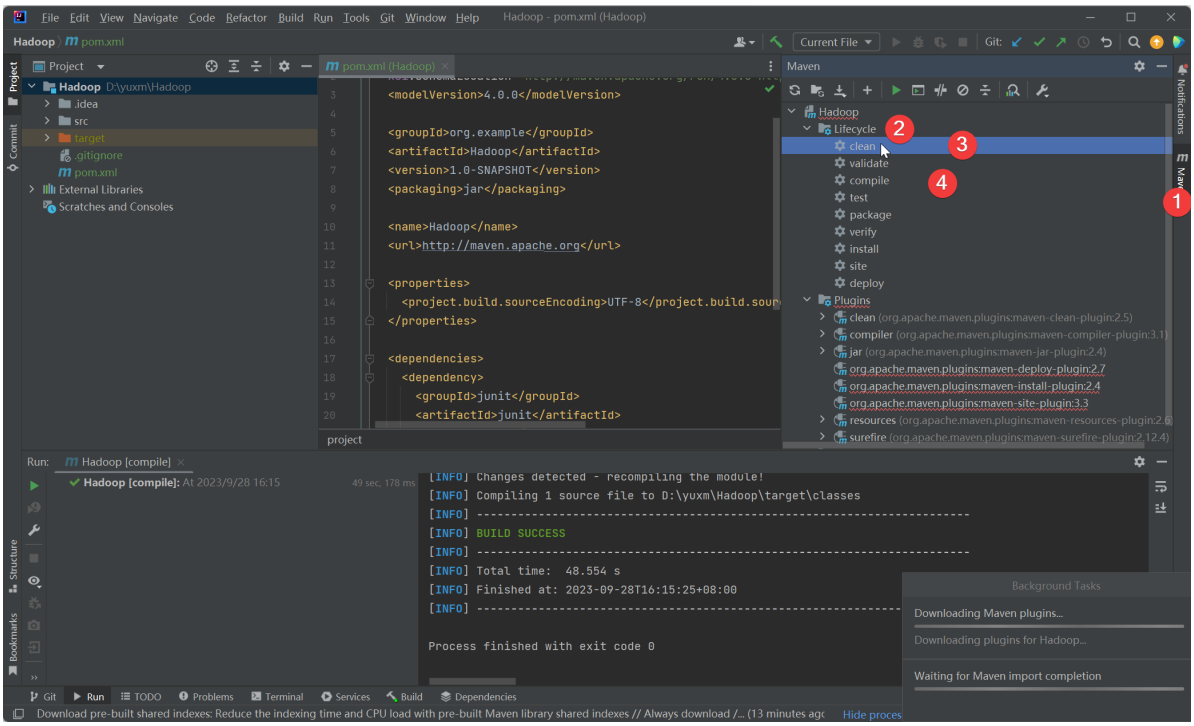
User settings fils:选择：D:\hadoop\apache-maven-3.8.8\conf\settings.xml，选中:Override

Local repository:使用默认:D:\hadoop\repo ， 选中:Override



更新Maven项目

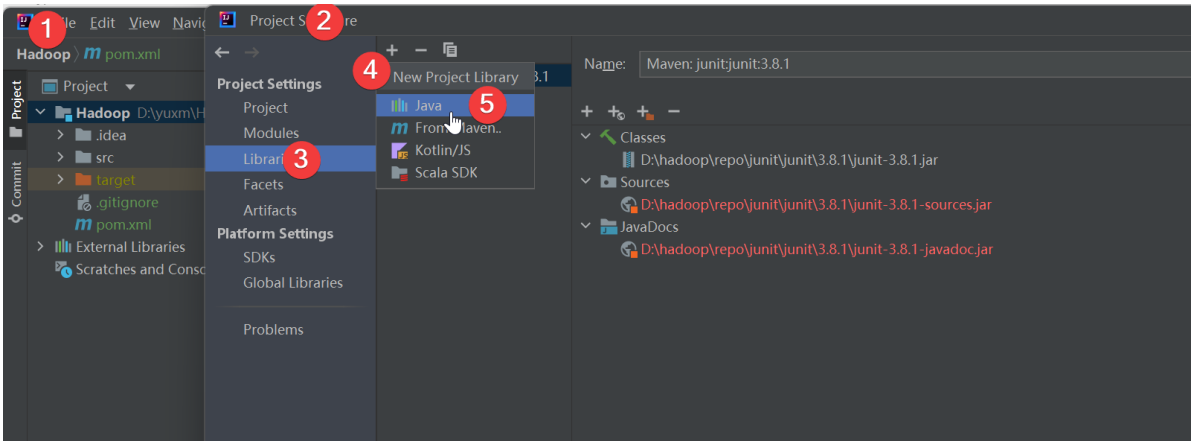
在IDEA右侧方，点击Maven页：展开Hadoop下面的Lifcycle,依次点击clean -> compile ,然后等待更新完成，完成Hadoop树下所有的红色下划线都会消失，就算完成更新。



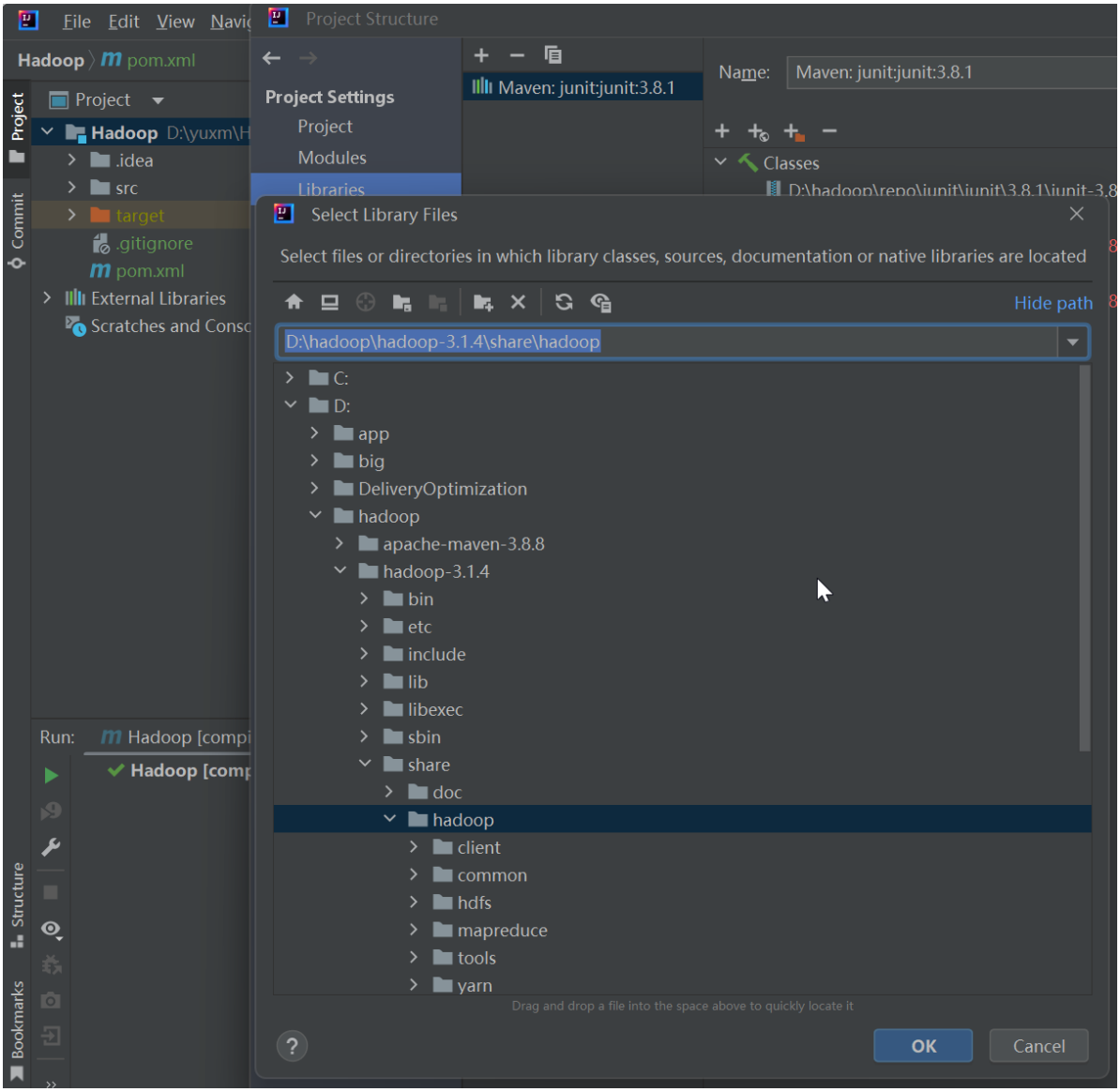
配置MapReduce环境

在Hadoop工程界面中，选择菜单栏中的“File ”→“Project Structure ”命令，也可以直接使用“Ctrl+Alt+Shift+S ”快捷键，打开“Project Structure”的对话框

单击“Project Structure”的对话框左侧的“Libraries ”选项，再右侧单击“+ ”选项，在弹出的选项栏中单击“Java ”选项



在弹出的对话框，选择要添加的jar包，选择我们前面已经解压的Hadoop安装目录:D:\hadoop\hadoop-3.1.4 下的\share\hadoop目录：D:\hadoop\hadoop-3.1.4\share\hadoop ,则将该目录下的全部JAR包导入项目，单击“OK ”按钮进入下一步



需要导入hadoop->share->hadoop下面的各子目录中，以下这些jar包：

```
hadoop-common.jar
hadoop-hdfs.jar
hadoop-mapreduce-client-core.jar
hadoop-yarn-common.jar
hadoop-yarn-api.jar
commons-cli.jar
commons-configuration.jar
commons-lang.jar
commons-logging.jar
log4j.jar
```

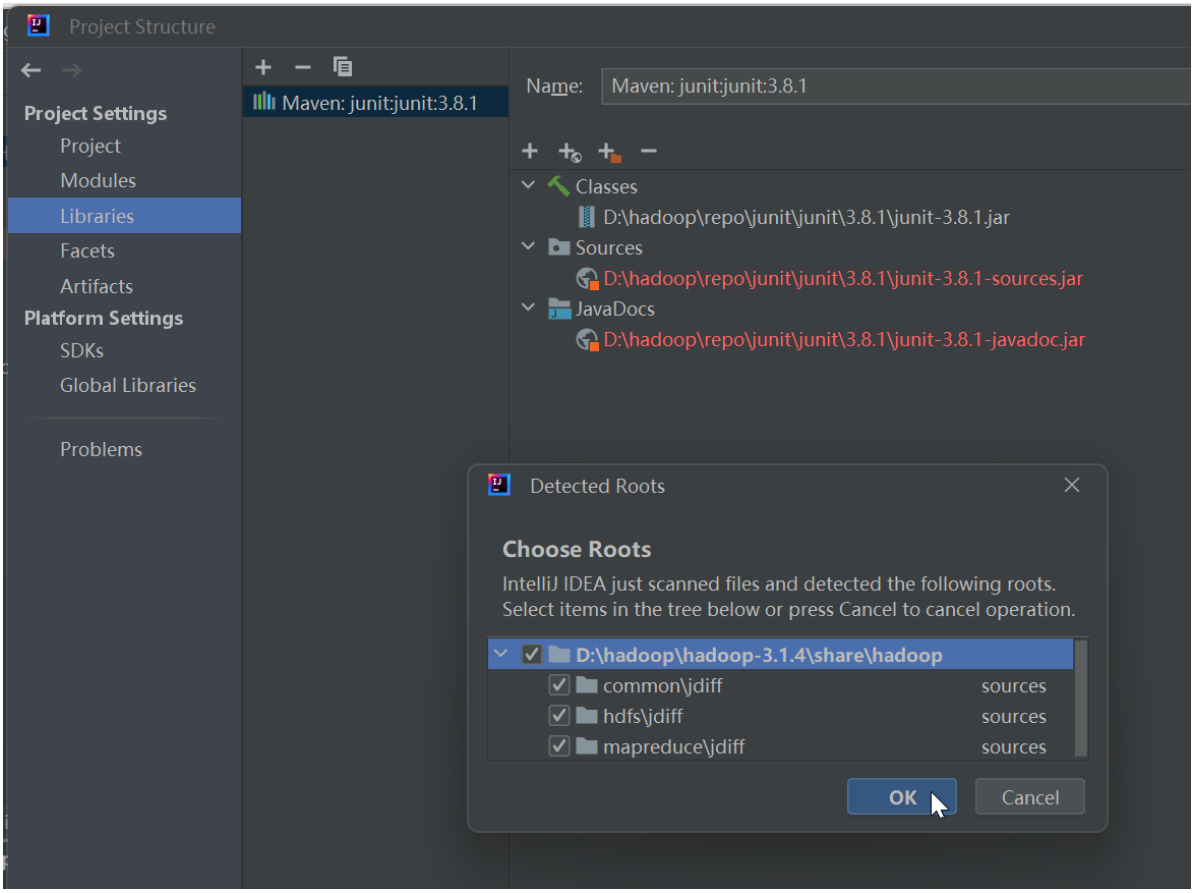
如果需要在IDEA中运行MR程序，需补充添加如下包：

```
# /usr/local/hadoop/share/hadoop/common/lib目录下的
woodstox-core-5.0.3.jar
slf4j-api-1.7.25.jar
slf4j-log4j12-1.7.25.jar
log4j-1.2.17.jar
stax2-api-3.1.4.jar
guava-27.0-jre.jar
commons-collections-3.2.2.jar
htrace-core4-4.1.0-incubating.jar

# share\hadoop\common\lib\目录下的
hadoop-auth-3.1.4.jar

# /share/hadoop/tools/lib 目录下的

```

为了处理log4j版本冲突，删除

```
# \b37066\democount\libs目录下的
slf4j-log4j12-1.7.30.jar
```

点OK后，在弹出的所有提示框中都点OK。



全部JAR包导入后，单击“Apply ”按钮，再单击“OK ”按钮，即可完成MapReduce

任务4.2通过源码初识MapReduce编程

进行MapReduce编程前，需要先掌握MapReduce的基本原理，对MapReduce的核心模块Mapper与Reducer的执行流程有一定的认识。Hadoop官方提供了一些示例源码，十分适合初学者学习。

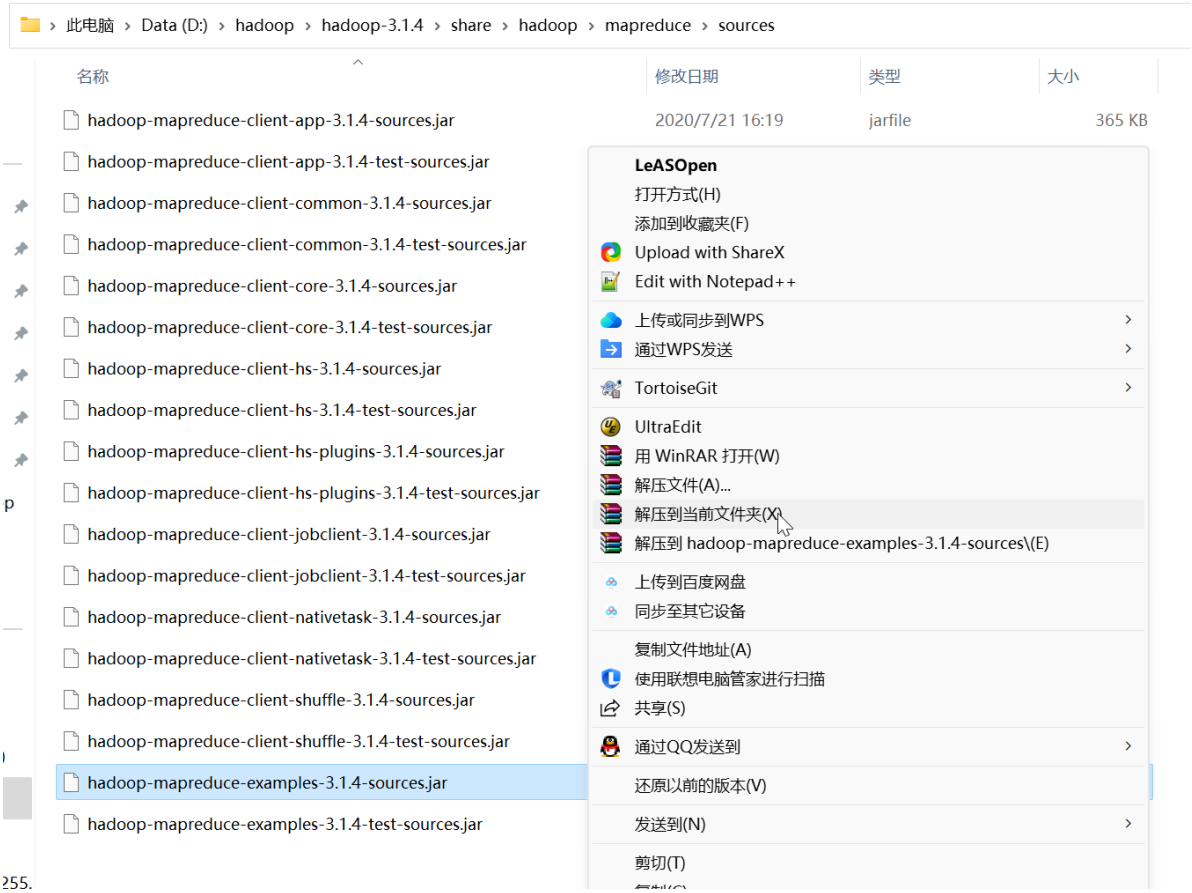
本小节的任务如下。先理解MapReduce的工作原理、核心组成和MapReduce的执行流程。再通过Hadoop官方示例源码WordCount（词频统计）掌握MapReduce的编程方法。

获取WordCount源码

进入Hadoop目录

进入Hadoop目录D:\hadoop\hadoop-3.1.4\share\hadoop\mapreduce\sources

解压: hadoop-mapreduce-examples-3.1.4-sources.jar 到当前路径下

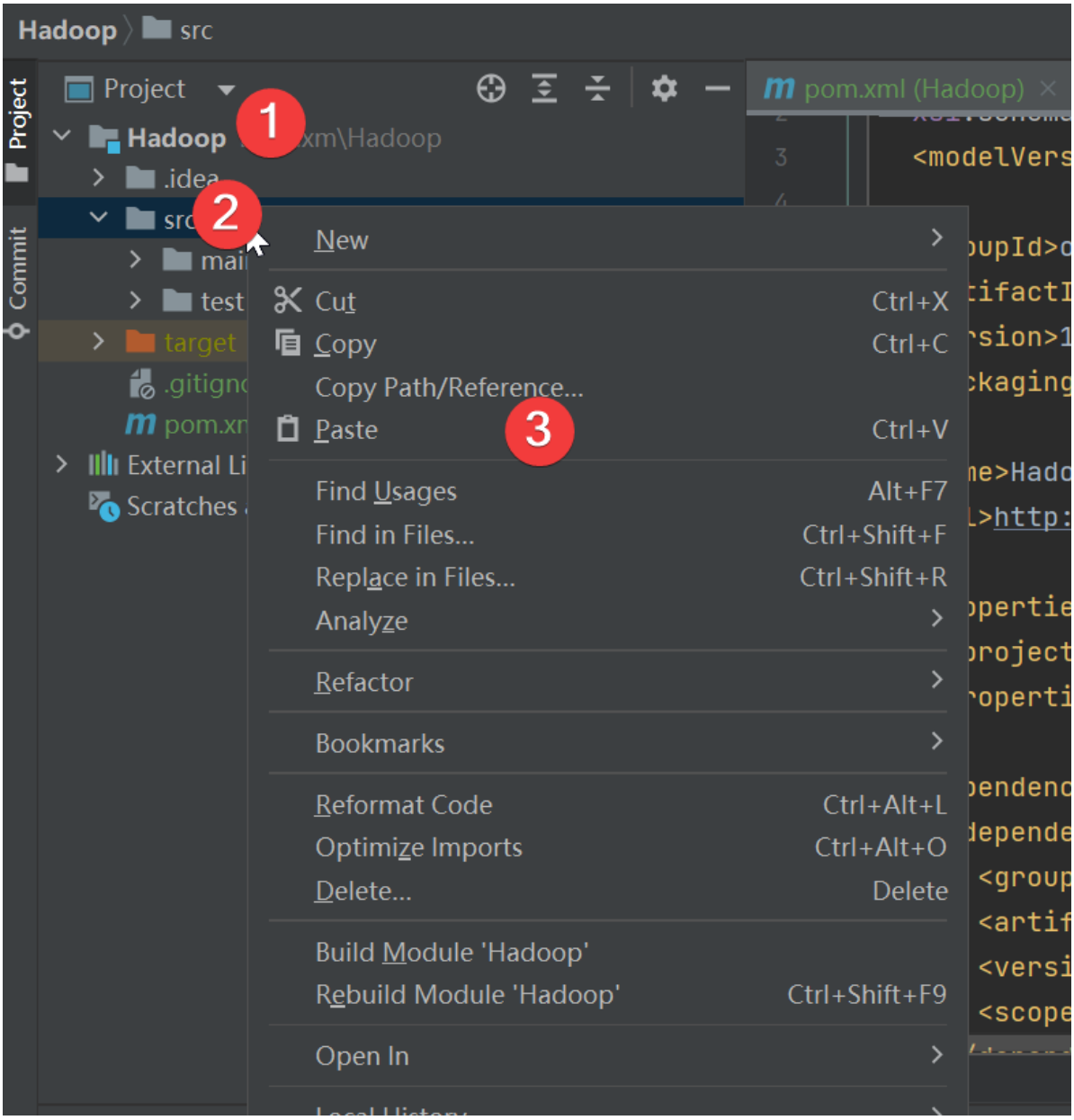


复制WordCount.java 到Hadoop工程

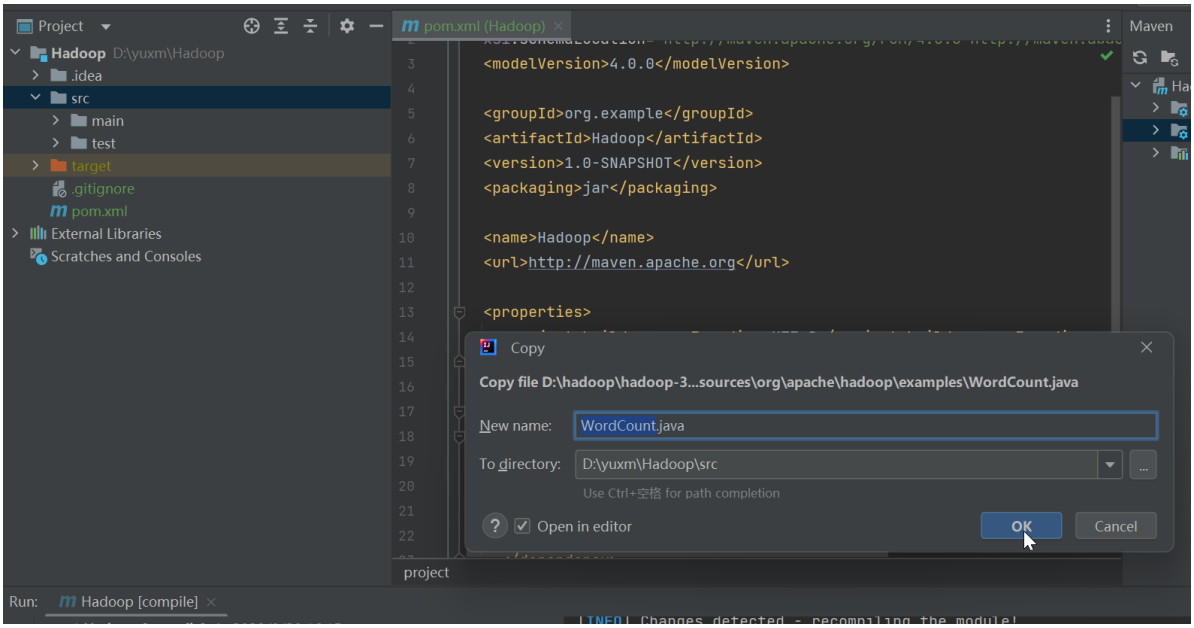
进入解压后的目录：D:\hadoop\hadoop-3.1.4\share\hadoop\mapreduce\sources\org\apache\hadoop\examples

将该WordCount.java 复制到Hadoop工程的src目录下：

1. 选择文件WordCount.java，然后Ctrl+C
2. 打开IDEA中Hadoop项目,右击Hadoop-》scr 目录，在弹出的菜单中选择Paste，将WordCount.java粘贴到src目录下


Hadoop
src
Project
Commit
pom.xml (Hadoop)
1
2
3
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+F7
Find in Files... Ctrl+Shift+F
Replace in Files... Ctrl+Shift+R
Analyze
Refactor
Bookmarks
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Build Module 'Hadoop'
Rebuild Module 'Hadoop' Ctrl+Shift+F9
Open In

在提示框中点击OK

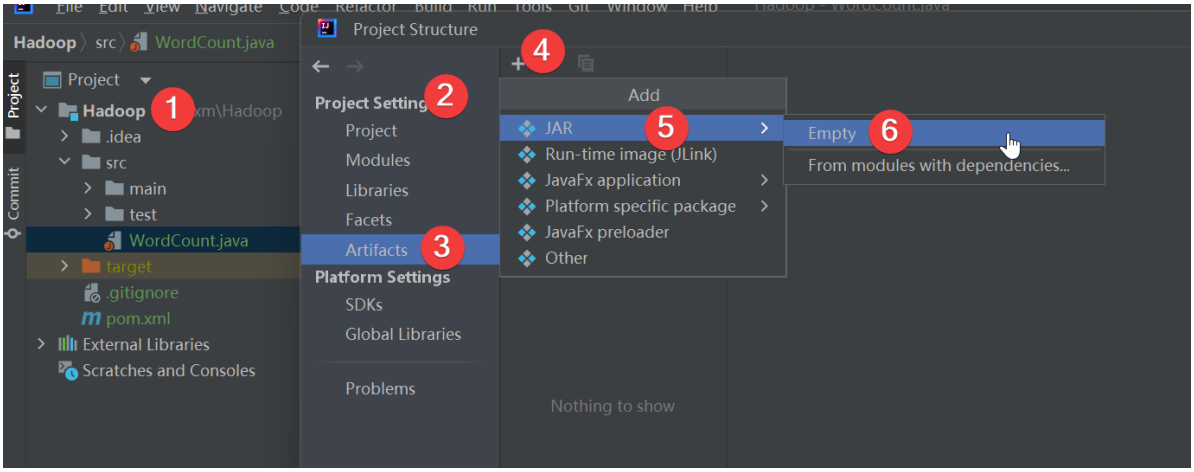

Project
Hadoop D:\yuxm\Hadoop
.idea
src
main
test
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
pom.xml (Hadoop)
<modelVersion>4.0.0</modelVersion>
<groupId>org.example</groupId>
<artifactId>Hadoop</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
<name>Hadoop</name>
<url>http://maven.apache.org</url>
<properties>
Maven
Copy
Copy file D:\hadoop\hadoop-3...sources\org\apache\hadoop\examples\WordCount.java
New name: WordCount.java
To directory: D:\yuxm\Hadoop\src
Use Ctrl+空格 for path completion
Open in editor
OK
Cancel
project
Run: Hadoop [compile]

编译项目

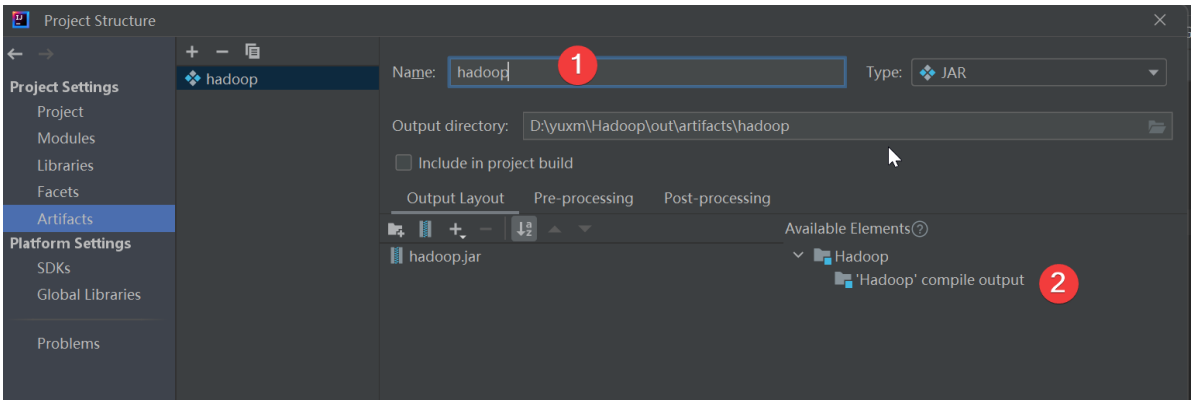
设置项目的输出

在Hadoop工程界面中，选择菜单栏中的“File ”→“Project Structure ”命令，也可以直接使用“Ctrl+Alt+Shift+S ”快捷键，打开“Project Structure”的对话框

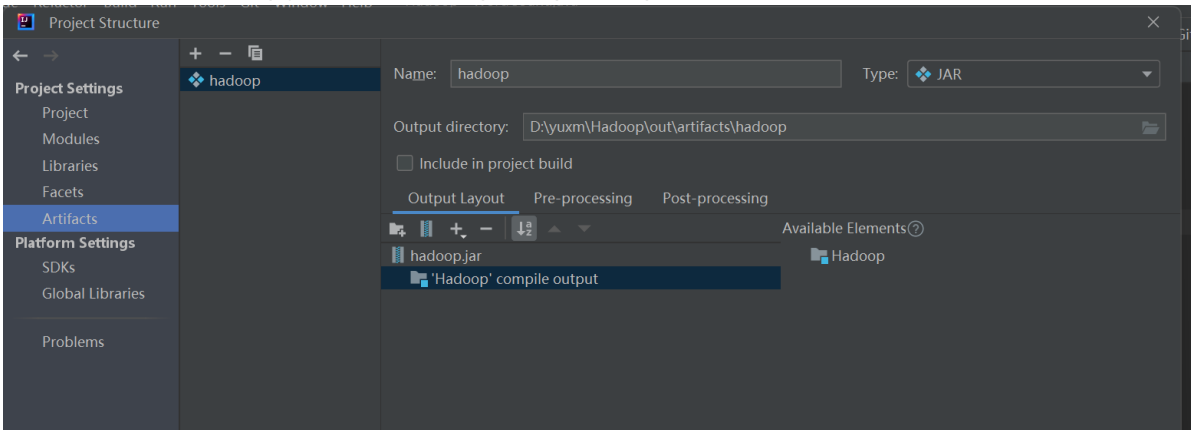
单击“Project Structure”的对话框左侧的“Artifacts”选项，再右侧单击“+ ”选项，在弹出的选项栏中单击“Jar ”选项,选择Empty



然后在弹出的设置框中，设置Name:Hadoop,然后双击Available Elements下的Hadoop compile output,



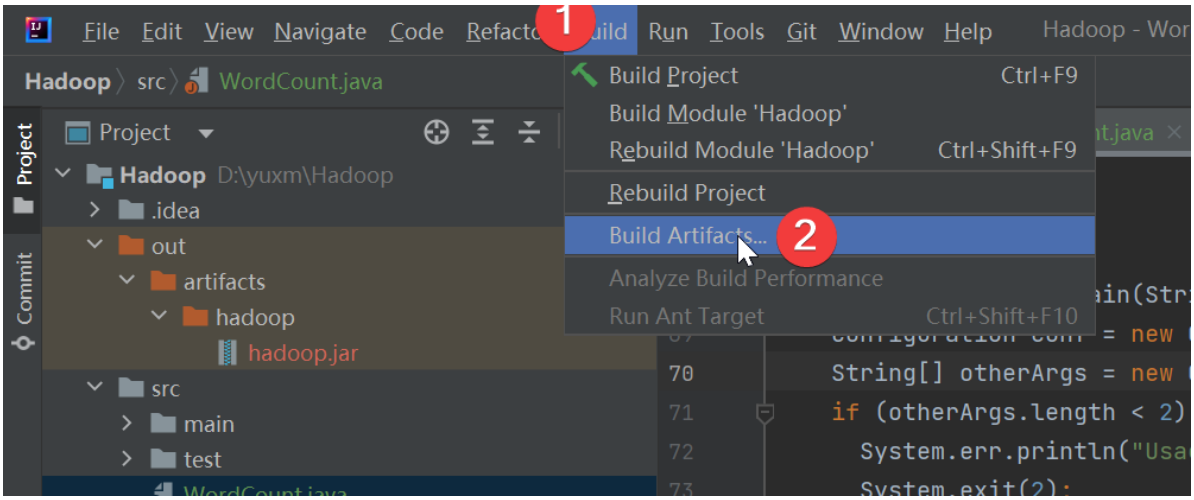
结果会在左侧的Hadoop.jar下出现Hadoop compile output



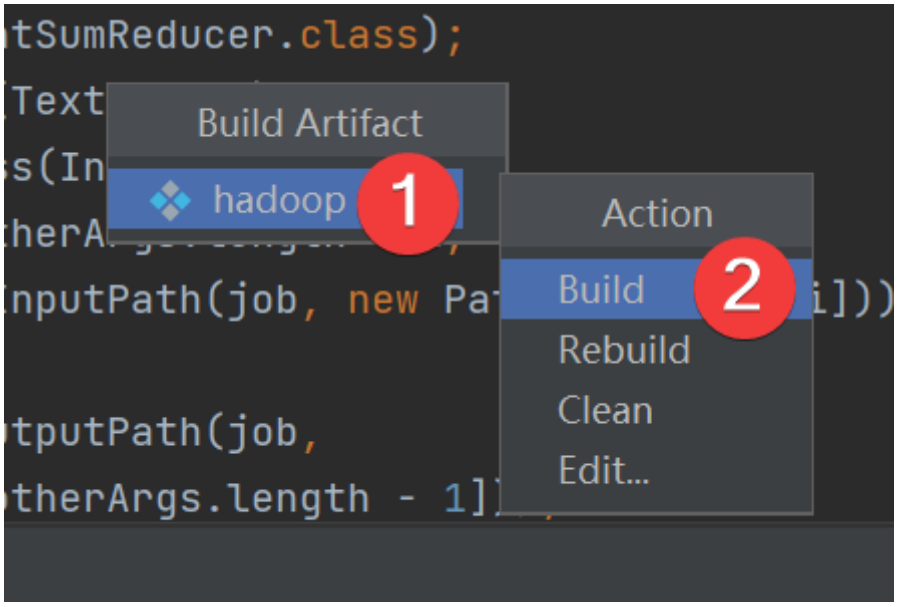
点右下方的OK完成设置

编译项目

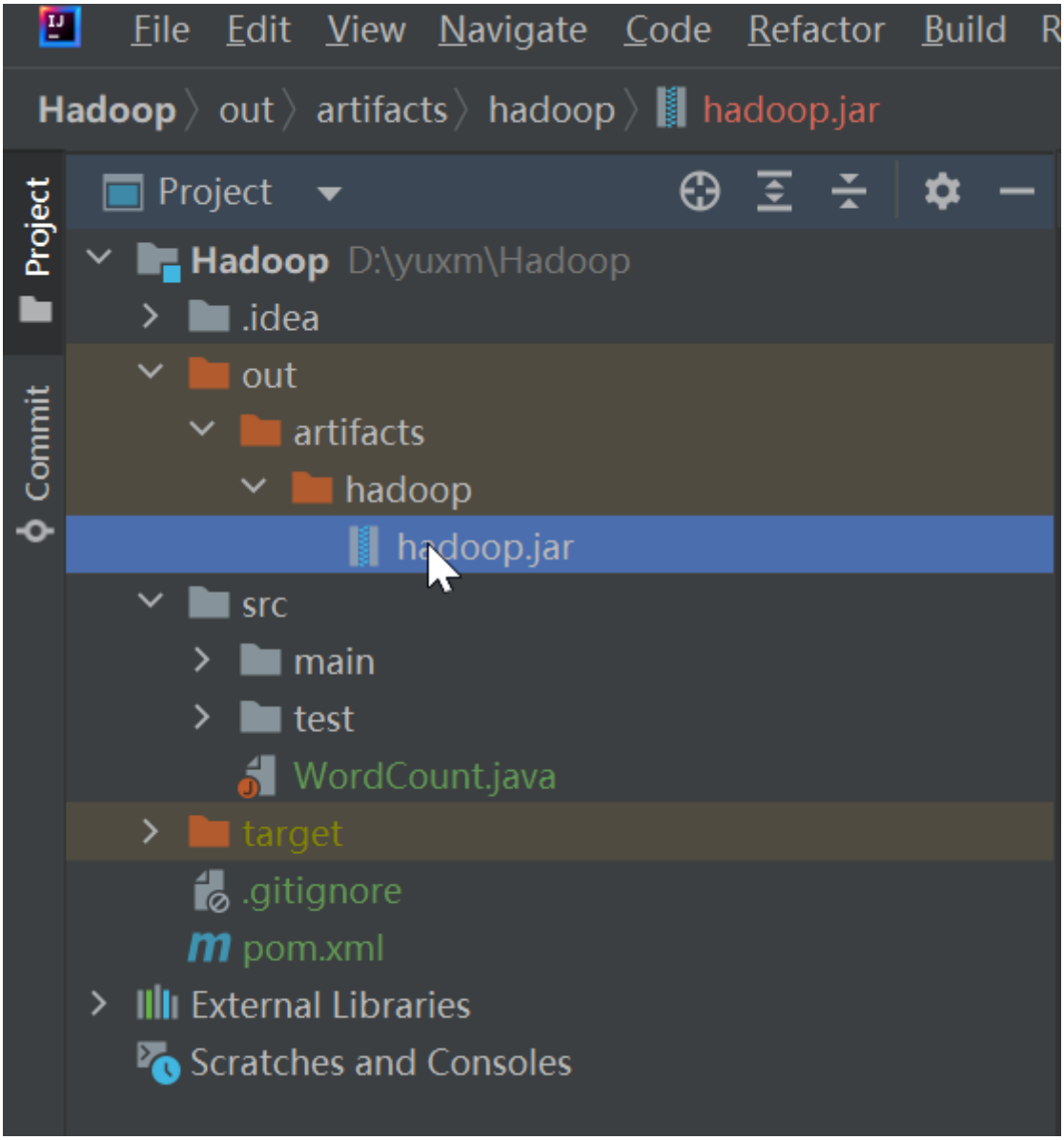
点菜单build->BuildArtifacts...



弹出菜单：



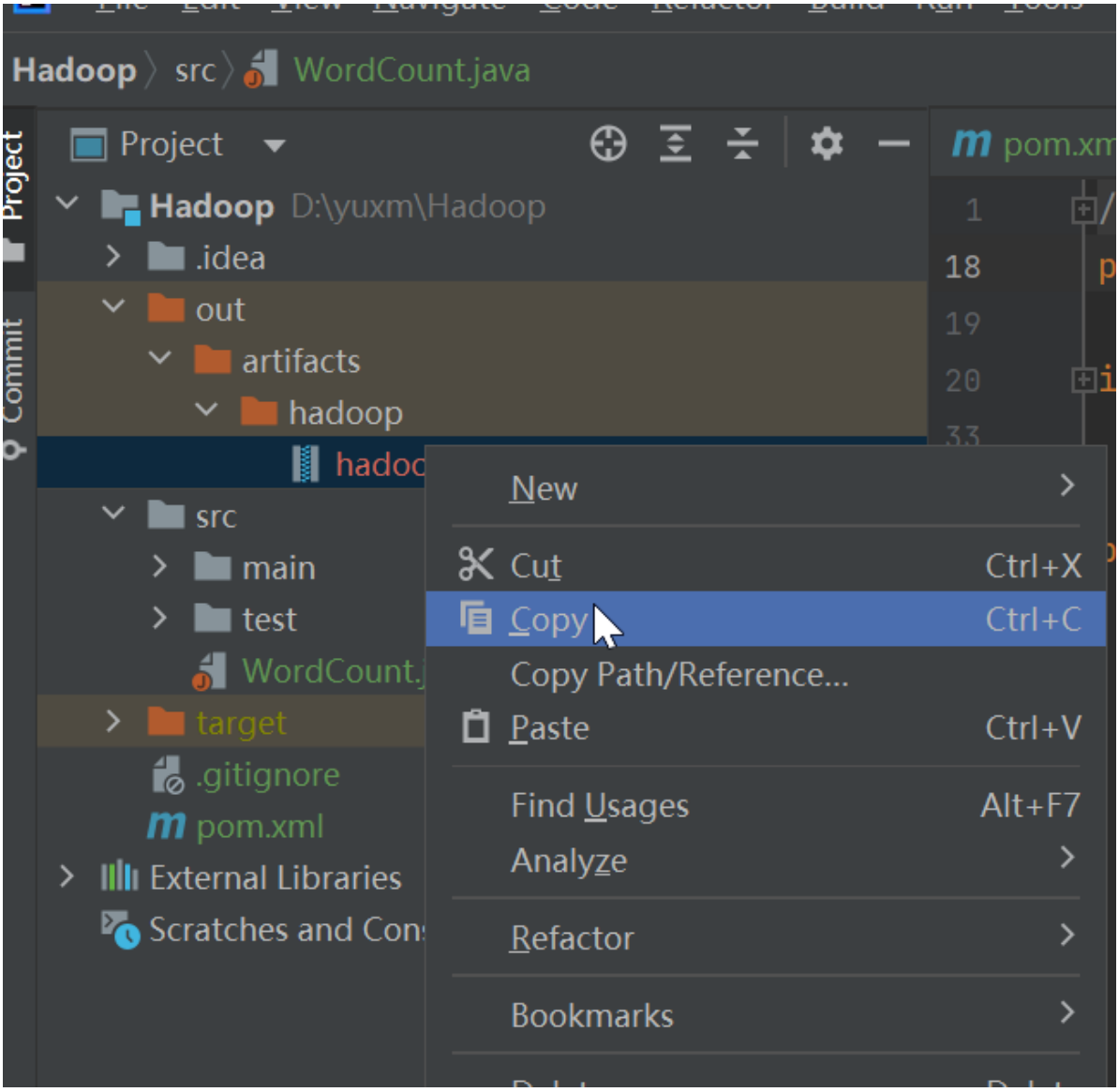
编译完成后，会在左侧源码树图上：out->artifacts-> hadoop下出现hadoop.jar文件，这就是编译后的应用程序：



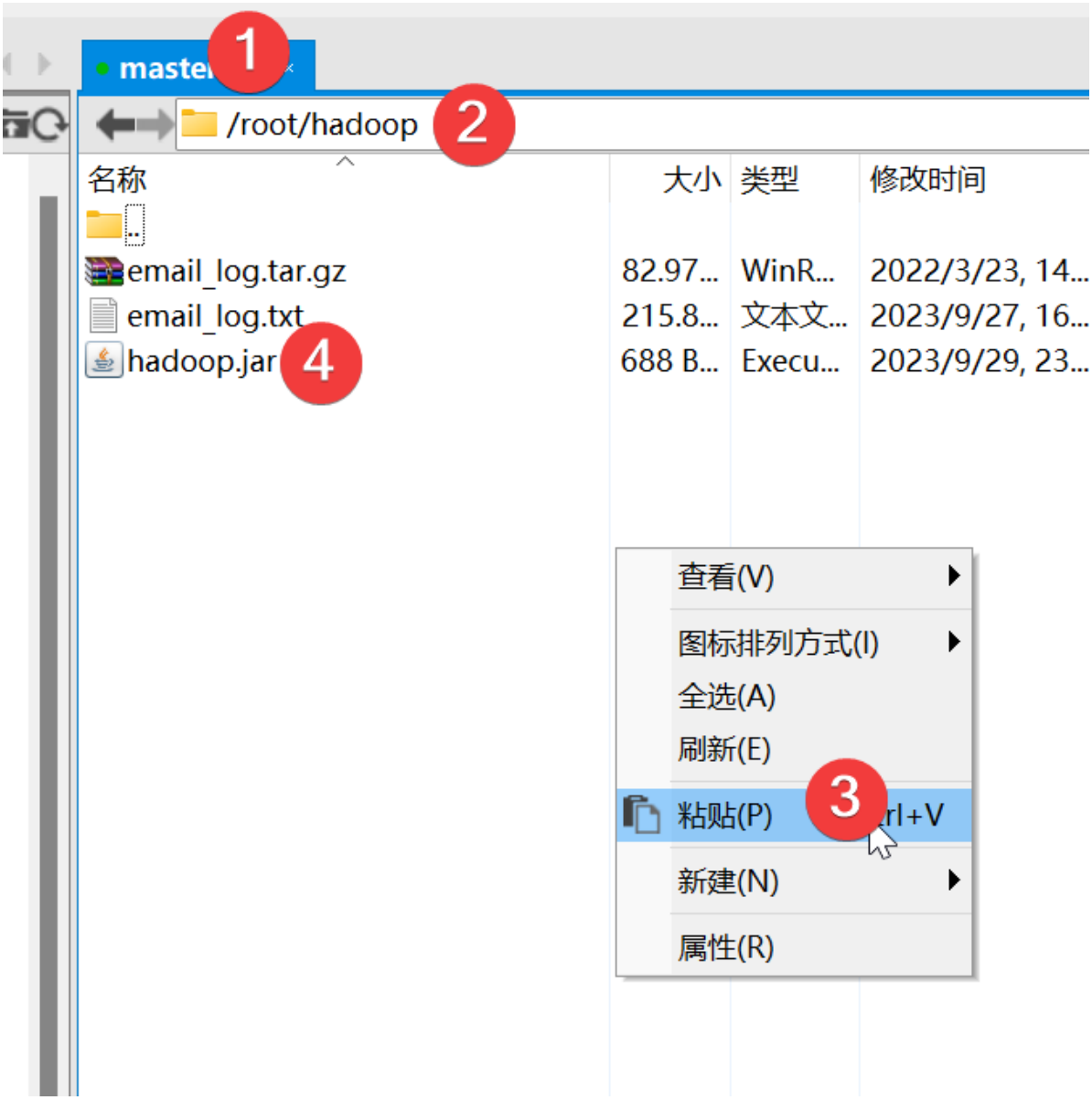
上传文件

上传编译后的文件到集群

右击上图的hadoop.jar,选择菜单“复制”



然后打开xftp工具，连接到master虚拟主机上，进入 /root/hadoop目录，右击空白的地方，在弹出的菜单中选择“粘贴”



运行文件

使用xshell或crt打开master,进入/root/hadoop目录，执行编译后的文件hadoop.jar：

```
cd /root/hadoop
hadoop jar ./hadoop.jar WordCount /user/myname/email_log.txt
/user/myname/output111
```

运行结果如：

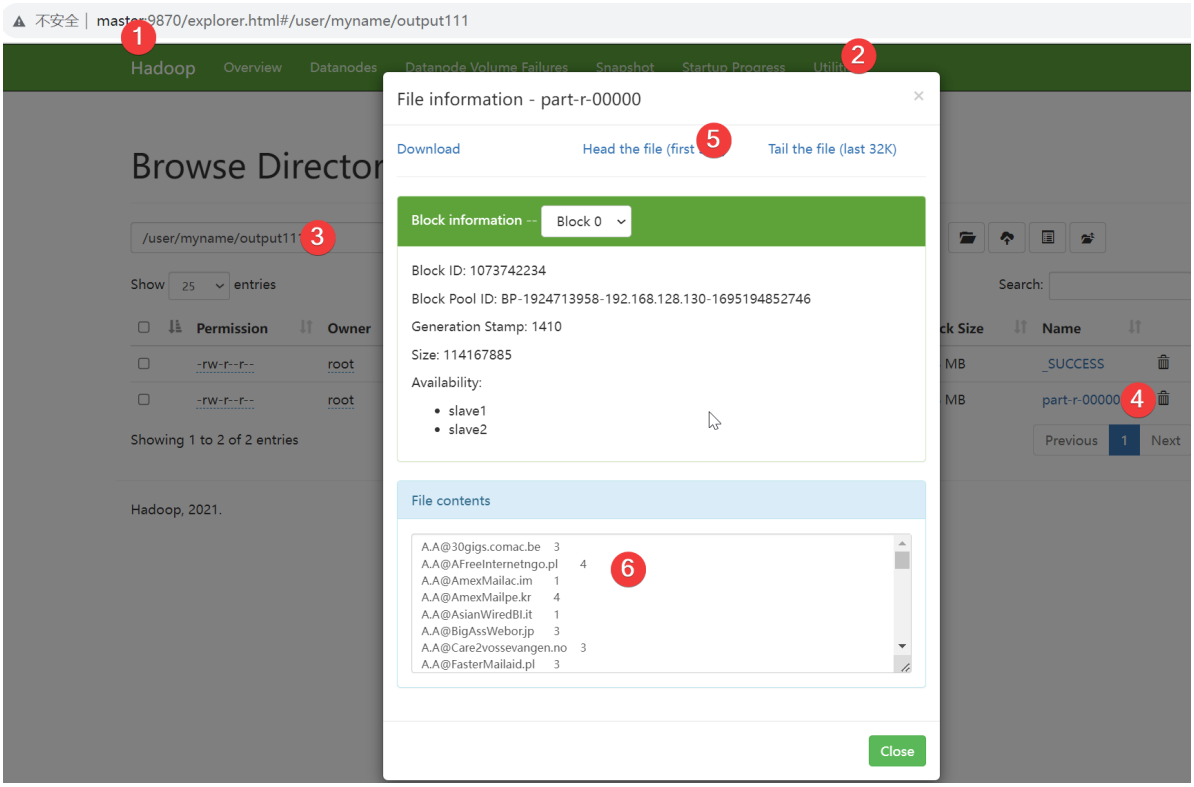
```
[root@master hadoop]# cd /root/hadoop
[root@master hadoop]# hadoop jar ./hadoop.jar
org.apache.hadoop.examples.WordCount /user/myname/email_log.txt
/user/myname/output111
2023-09-29 23:42:44,210 INFO client.RMProxy: Connecting to ResourceManager
at master/192.168.128.130:8032
2023-09-29 23:42:45,198 INFO mapreduce.JobResourceUploader: Disabling
Erasure Coding for path: /tmp/hadoop-
yarn/staging/root/.staging/job_1695864204511_0003
```

```
2023-09-29 23:42:45,646 INFO input.FileInputFormat: Total input files to process : 1
2023-09-29 23:42:45,813 INFO mapreduce.JobSubmitter: number of splits:2
2023-09-29 23:42:46,098 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1695864204511_0003
2023-09-29 23:42:46,101 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-09-29 23:42:46,459 INFO conf.Configuration: resource-types.xml not found
2023-09-29 23:42:46,459 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-09-29 23:42:46,575 INFO impl.YarnClientImpl: Submitted application application_1695864204511_0003
2023-09-29 23:42:46,641 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy/application_1695864204511_0003/
2023-09-29 23:42:46,642 INFO mapreduce.Job: Running job: job_1695864204511_0003
2023-09-29 23:42:56,833 INFO mapreduce.Job: Job job_1695864204511_0003 running in uber mode : false
2023-09-29 23:42:56,835 INFO mapreduce.Job:  map 0% reduce 0%
2023-09-29 23:43:15,063 INFO mapreduce.Job:  map 20% reduce 0%
2023-09-29 23:43:21,131 INFO mapreduce.Job:  map 28% reduce 0%
2023-09-29 23:43:27,206 INFO mapreduce.Job:  map 33% reduce 0%
2023-09-29 23:43:33,268 INFO mapreduce.Job:  map 49% reduce 0%
2023-09-29 23:43:34,296 INFO mapreduce.Job:  map 50% reduce 0%
2023-09-29 23:43:50,459 INFO mapreduce.Job:  map 76% reduce 0%
2023-09-29 23:43:56,523 INFO mapreduce.Job:  map 83% reduce 0%
2023-09-29 23:44:00,565 INFO mapreduce.Job:  map 100% reduce 0%
2023-09-29 23:44:17,719 INFO mapreduce.Job:  map 100% reduce 95%
2023-09-29 23:44:18,732 INFO mapreduce.Job:  map 100% reduce 100%
2023-09-29 23:44:19,760 INFO mapreduce.Job: Job job_1695864204511_0003 completed successfully
2023-09-29 23:44:19,920 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=416431057
                FILE: Number of bytes written=585286627
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=226383989
                HDFS: Number of bytes written=114167885
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=245812
                Total time spent by all reduces in occupied slots (ms)=69040
                Total time spent by all map tasks (ms)=61453
                Total time spent by all reduce tasks (ms)=17260
                Total vcore-milliseconds taken by all map tasks=61453
                Total vcore-milliseconds taken by all reduce tasks=17260
```

```
                Total megabyte-milliseconds taken by all map tasks=125855744
                Total megabyte-milliseconds taken by all reduce
tasks=35348480
        Map-Reduce Framework
                Map input records=8000000
                Map output records=8000000
                Map output bytes=250379675
                Map output materialized bytes=168189616
                Input split bytes=218
                Combine input records=12301355
                Combine output records=9352725
                Reduce input groups=3896706
                Reduce shuffle bytes=168189616
                Reduce input records=5051370
                Reduce output records=3896706
                Spilled Records=17558337
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=1735
                CPU time spent (ms)=77150
                Physical memory (bytes) snapshot=708571136
                Virtual memory (bytes) snapshot=10799960064
                Total committed heap usage (bytes)=522833920
                Peak Map Physical memory (bytes)=208769024
                Peak Map Virtual memory (bytes)=3597885440
                Peak Reduce Physical memory (bytes)=307593216
                Peak Reduce Virtual memory (bytes)=3604189184
        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=226383771
        File Output Format Counters
                Bytes Written=114167885
[root@master hadoop]#
```

查看结果

打开http://master:9870，-》utilities -》Brower the file system，进入目录:/user/myname/output111,查看park-r-00000文件内容：



任务4.3统计网站每日的访问次数

网站的访问次数分布情况对网站运营商而言是十分重要的指标，网站运营商可以根据实际的访问情况总结用户可能感兴趣的内容，调整网站的版块和内容设计。

本小节的任务是通过MapReduce编程实现网站每日的访问量统计。

1. 编写MapReduce程序首先需要考虑Map阶段和Reduce阶段各自的处理逻辑。
2. 再根据处理逻辑编写Mapper模块与Reducer模块的代码。
3. 最后将完整代码编译打包后提交至Hadoop集群运行。

获取测试数据

通过百度网盘下载：

下载地址：链接: https://pan.baidu.com/s/1DHGRrnjvi4yMY41Se7_vDA 提取码: vbgu

软件	大小	版本	安装包称
hadoop_data	99.7M	20230925	hadoop_data.tar.gz
Git	2.39.2	Git-2.39.2-64-bit.exe	
TortoiseGit	2.14	TortoiseGit-2.14.0.0-64bit.msi	

下载hadoop_data.tar.gz后，通过xftp工具上传到master主机的/root/hadoop目录下，然后xshell进入master执行：

```
cd /root/hadoop
tar -zxvf ./hadoop_data.tar.gz
hdfs dfs -put ./hadoop_data/raceData.csv  /user/myname/
```

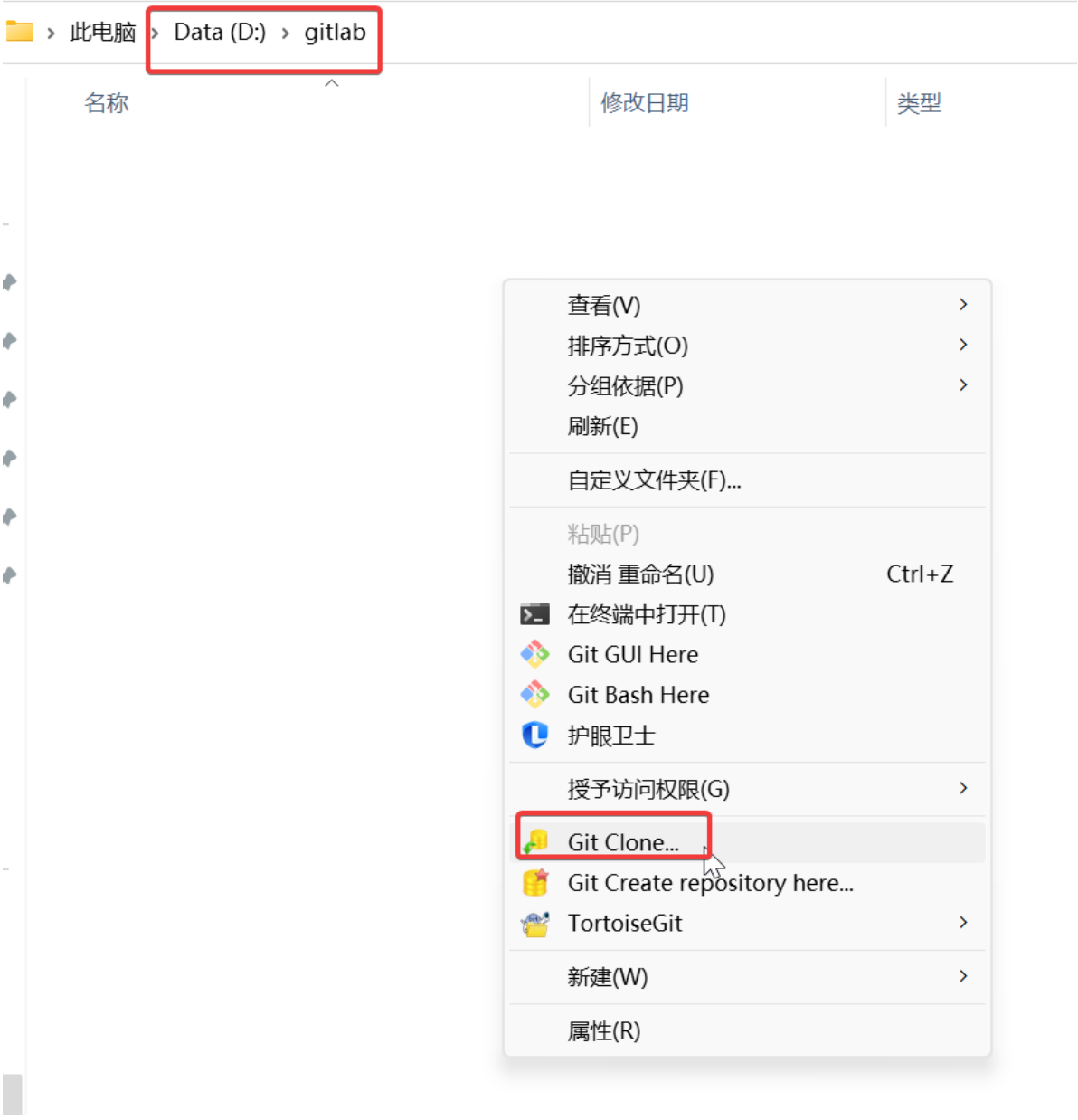
获取源代码

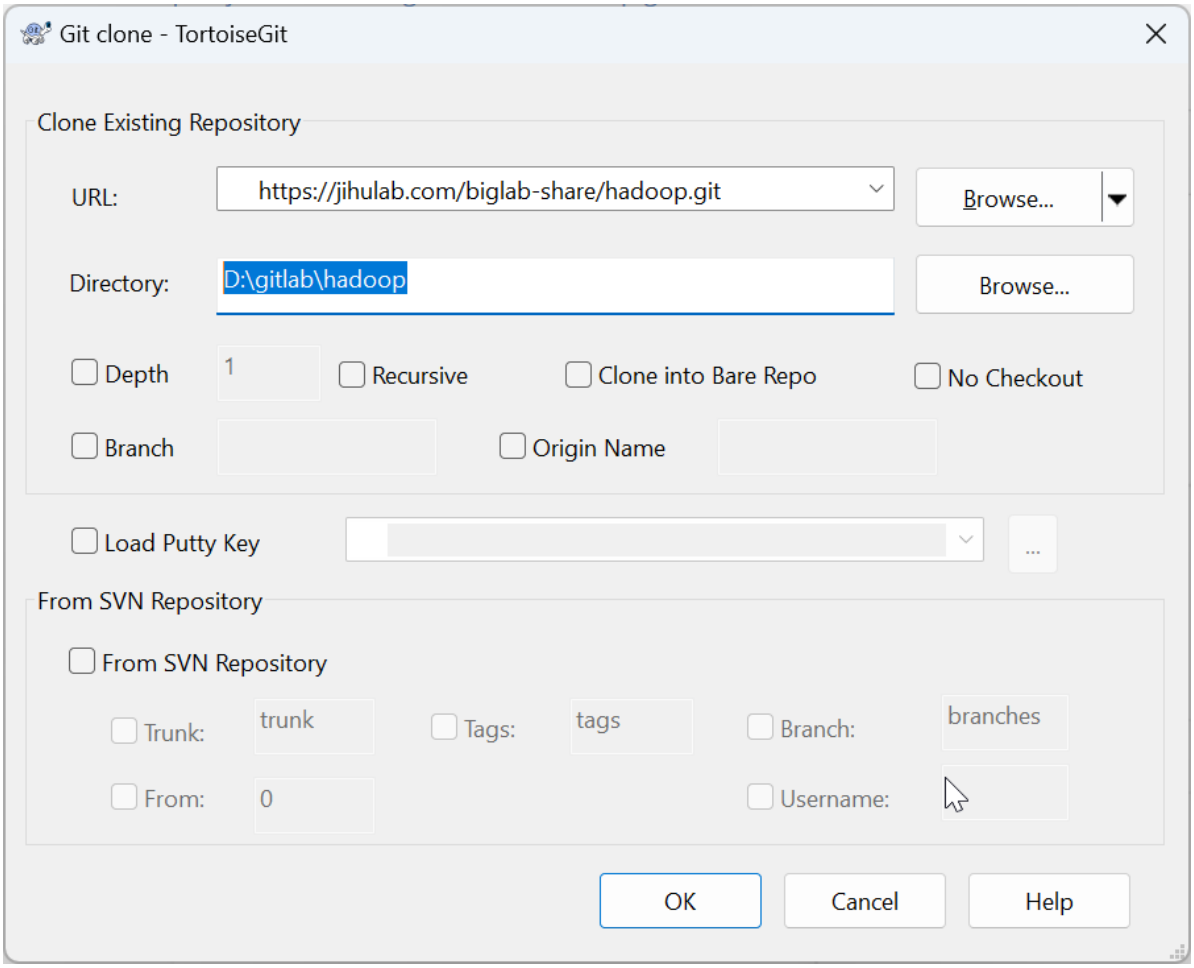
1. 安装从百度网盘下载下来的Git-2.39.2-64-bit.exe 和TortoiseGit-2.14.0.0-64bit.msi
2. 在文件浏览器中，打开D:盘，右击菜单创建目录: d:\gitlab,双击进入d:\gitlab
3. 右击出现菜单，选择Git Clone,在弹出的输入框中粘贴入：

URL: https://jihulab.com/biglab-share/hadoop.git

Directory: D:\gitlab\hadoop

点击OK ,等待完成源代码的下载。





理解源代码

源码类： ch04->dailyAccessCount

Mapper类

```
    public static class MyMap
            extends Mapper<Object, Text, Text, IntWritable> {
        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {
            String line = value.toString();
            String arr[] = line.split(",");
            context.write(new Text(arr[4].substring(0, 9)),
                    new IntWritable(1));
        }
    }
```

Reducer类

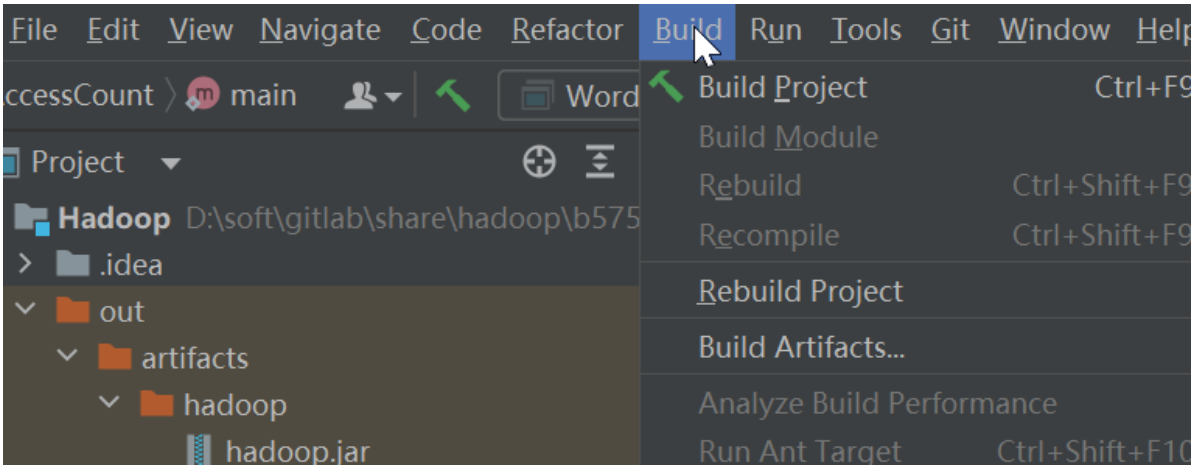
```
    public static class MyReduce
            extends Reducer<Text, IntWritable, Text, IntWritable> {
        public void reduce(Text key, Iterable<IntWritable> values,
                           Context context)
                throws IOException, InterruptedException {
            int count = 0;
            for (IntWritable value : values) {
                count++;
            }
            context.write(key, new IntWritable(count));
        }
    }
```

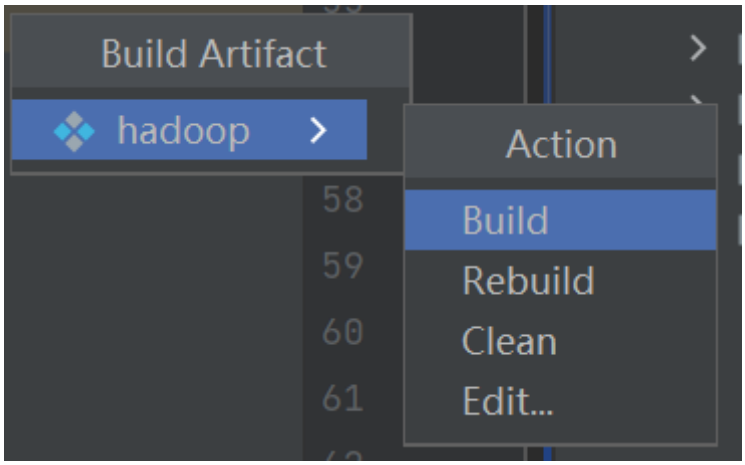
Driver代码

```
public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args)
                .getRemainingArgs();
        if (otherArgs.length < 2) {
            System.err.println("必须输入读取文件路径和输出路径");
            System.exit(2);
        }
        Job job = Job.getInstance(conf, "visits count");
        job.setJarByClass(dailyAccessCount.class);
        job.setMapperClass(MyMap.class);
        job.setReducerClass(MyReduce.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        for (int i = 0; i < otherArgs.length - 1; ++i) {
            FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
        }
        FileOutputFormat.setOutputPath(job,
                new Path(otherArgs[otherArgs.length - 1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
```

编译生成jar包

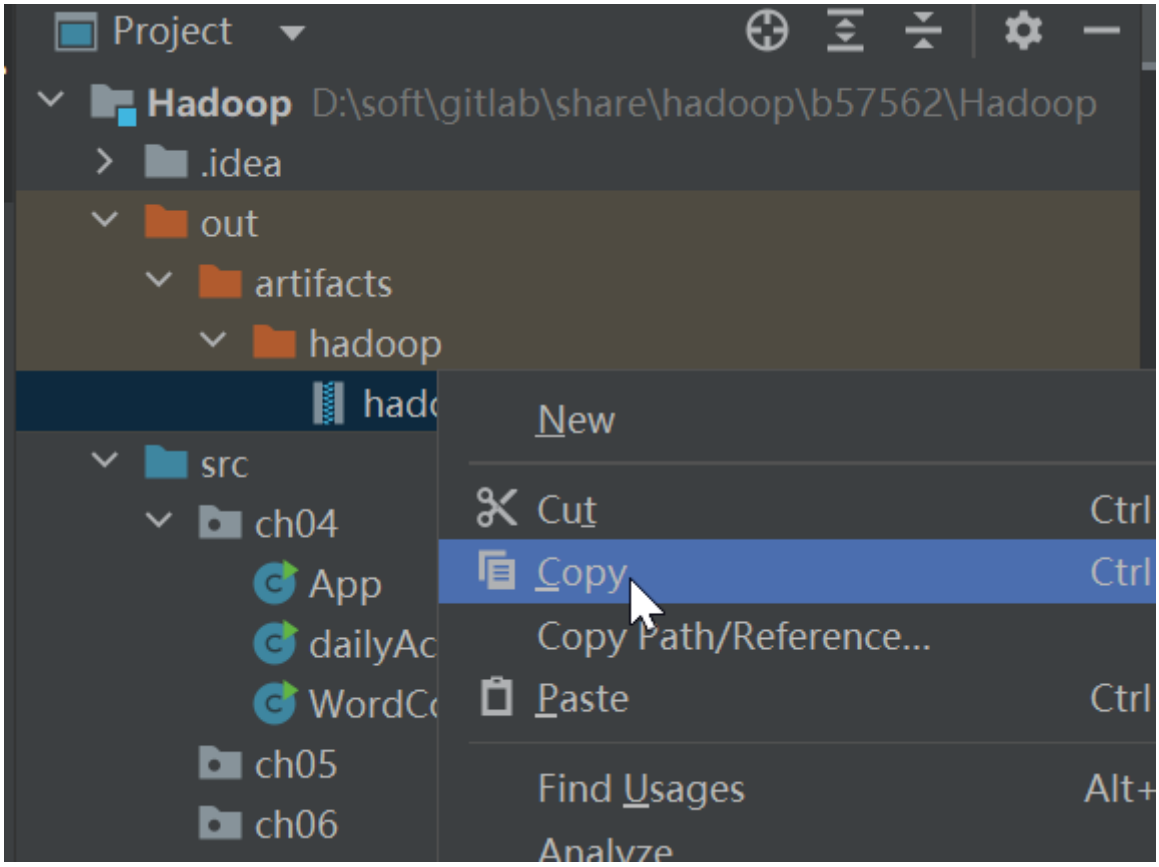
菜单-》Build -》 Build Artifacts -》hadoop-》Build

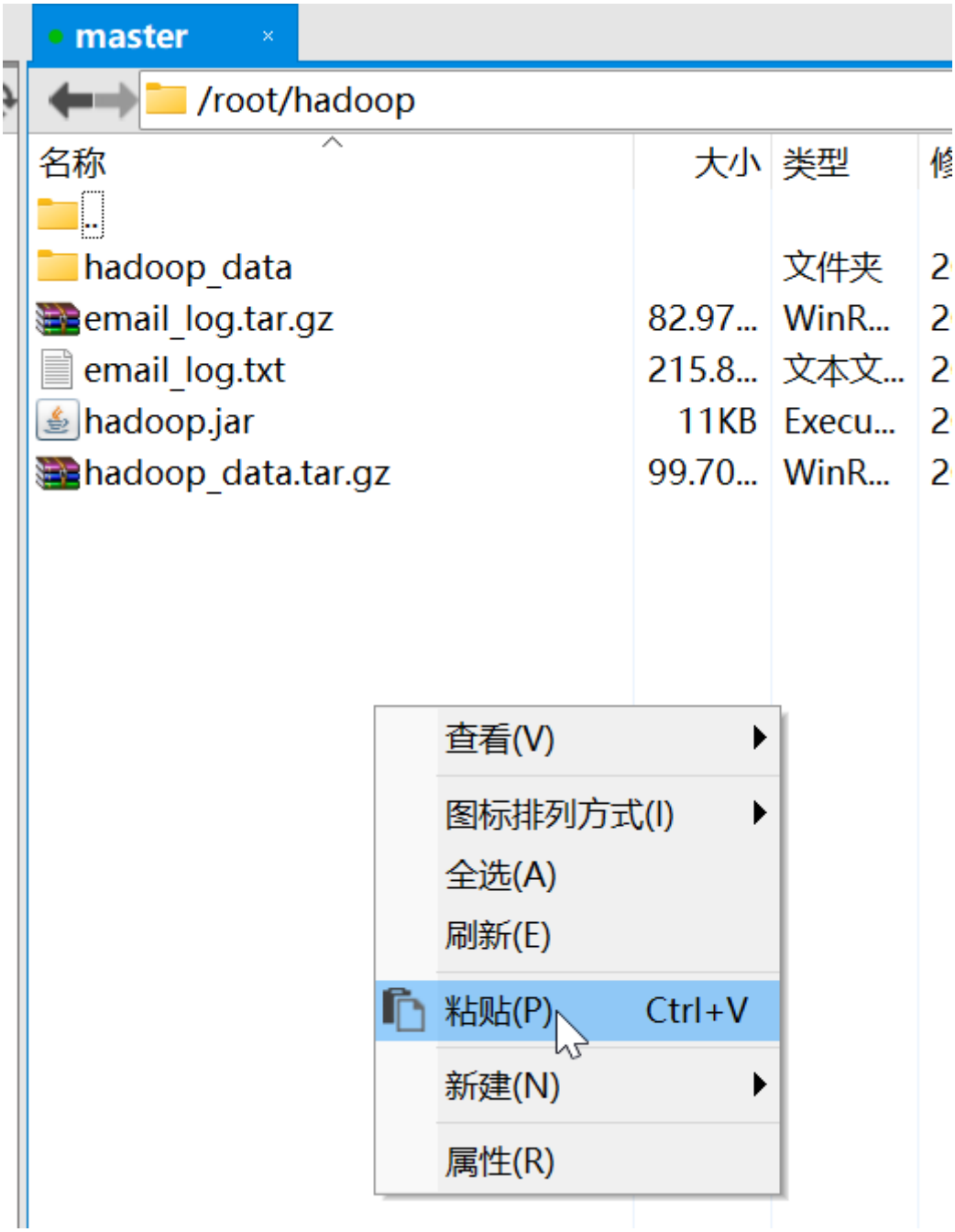




上传jar包到master

1. 在Hadoop项目，左侧树图中-》out-》artifacts -》hadoop -》 hadoop.jar ，右击hadoop.jar，菜单中选择复制
2. 打开xftp，进入master-》root-》hadoop目录，右击选择粘贴





运行jar

使用crt或xshell进入master

```
cd /root/hadoop
hadoop  jar hadoop.jar ch04.dailyAccessCount /user/myname/raceData.csv
/user/myname/output_AccessCount
```

运行结果如：

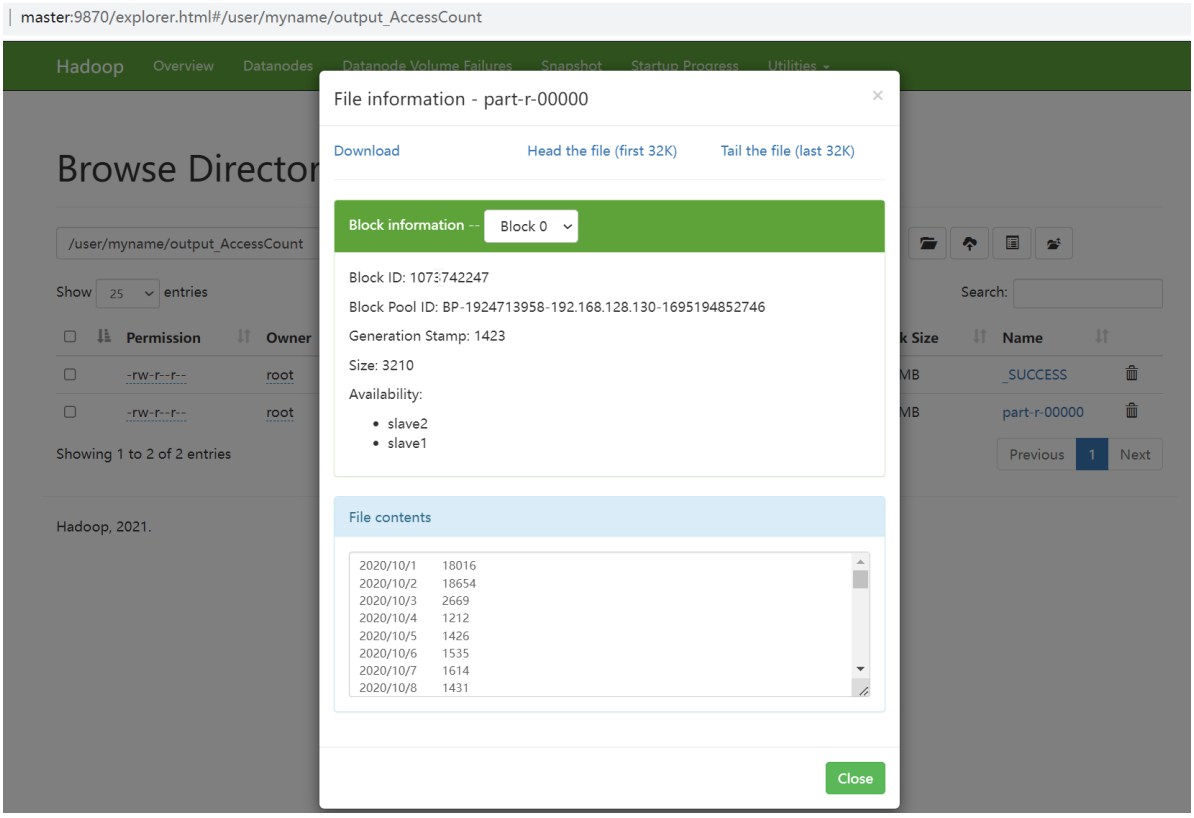
```
[root@master hadoop]# hadoop  jar hadoop.jar ch04.dailyAccessCount
/user/myname/raceData.csv /user/myname/output_AccessCount
2023-09-30 22:40:55,557 INFO client.RMProxy: Connecting to ResourceManager
at master/192.168.128.130:8032
2023-09-30 22:40:57,087 INFO mapreduce.JobResourceUploader: Disabling
Erasure Coding for path: /tmp/hadoop-
yarn/staging/root/.staging/job_1696004849840_0001
```

```
2023-09-30 22:40:57,705 INFO input.FileInputFormat: Total input files to process : 1
2023-09-30 22:40:57,934 INFO mapreduce.JobSubmitter: number of splits:1
2023-09-30 22:40:58,402 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1696004849840_0001
2023-09-30 22:40:58,408 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-09-30 22:40:58,807 INFO conf.Configuration: resource-types.xml not found
2023-09-30 22:40:58,807 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-09-30 22:40:59,628 INFO impl.YarnClientImpl: Submitted application application_1696004849840_0001
2023-09-30 22:40:59,728 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy/application_1696004849840_0001/
2023-09-30 22:40:59,730 INFO mapreduce.Job: Running job: job_1696004849840_0001
2023-09-30 22:41:17,331 INFO mapreduce.Job: Job job_1696004849840_0001 running in uber mode : false
2023-09-30 22:41:17,334 INFO mapreduce.Job:  map 0% reduce 0%
2023-09-30 22:41:32,591 INFO mapreduce.Job:  map 100% reduce 0%
2023-09-30 22:41:40,682 INFO mapreduce.Job:  map 100% reduce 100%
2023-09-30 22:41:41,715 INFO mapreduce.Job: Job job_1696004849840_0001 completed successfully
2023-09-30 22:41:41,901 INFO mapreduce.Job: Counters: 53
        File System Counters
                FILE: Number of bytes read=6233654
                FILE: Number of bytes written=12910773
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=51641009
                HDFS: Number of bytes written=3210
                HDFS: Number of read operations=8
                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)=49260
                Total time spent by all reduces in occupied slots (ms)=23328
                Total time spent by all map tasks (ms)=12315
                Total time spent by all reduce tasks (ms)=5832
                Total vcore-milliseconds taken by all map tasks=12315
                Total vcore-milliseconds taken by all reduce tasks=5832
                Total megabyte-milliseconds taken by all map tasks=25221120
                Total megabyte-milliseconds taken by all reduce tasks=11943936
        Map-Reduce Framework
                Map input records=389603
                Map output records=389603
                Map output bytes=5454442
                Map output materialized bytes=6233654
                Input split bytes=108
```

```
                Combine input records=0
                Combine output records=0
                Reduce input groups=224
                Reduce shuffle bytes=6233654
                Reduce input records=389603
                Reduce output records=224
                Spilled Records=779206
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=522
                CPU time spent (ms)=9040
                Physical memory (bytes) snapshot=320536576
                Virtual memory (bytes) snapshot=7196209152
                Total committed heap usage (bytes)=148172800
                Peak Map Physical memory (bytes)=206639104
                Peak Map Virtual memory (bytes)=3594711040
                Peak Reduce Physical memory (bytes)=113897472
                Peak Reduce Virtual memory (bytes)=3601498112
        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=51640901
        File Output Format Counters
                Bytes Written=3210
[root@master hadoop]#
```

在 HDFS 查看运行结果：

根据crt命令行的运行命令，输出的文件在目录下：/user/myname/output_AccessCount



任务4.4将网站每日访问量根据访问次数进行升序排序

任务描述

在上一小节中虽然已实现了网站每日访问次数的统计，并将输出结果保存至HDFS，但是网站每日的访问次数是根据日期进行升序排序的，不能直观地看出访问次数大致的分布情况。

本小节的任务如下。

1. 读取/Tipdm/Hadoop/MapReduce/Result/dailyAccessCount目录中的输出结果。
2. 按照访问次数进行升序排序。
3. 将排序后的结果存储至HDFS。

实验数据

是在4.3统计网站每日的访问次数基础上，在其输出结果文件基础进行排序处理，所以不用额外准备测试数据。

所以待分析文件为： /user/myname/output_AccessCount

理解源代码

源代码的下载，已经上4.3节完成，所以后面的源码就不再另外下载。

源码类： ch04->accessTimesSort

Mapper类

```
    public static class MyMap extends Mapper<Object, Text, IntWritable,
Text> {
        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {
            String line = value.toString();
            //指定tab为分隔符
            String arr[] = line.split("\t");
            //key：统计结果， value：日期
            context.write(new IntWritable(Integer.parseInt(arr[1])),
                    new Text(arr[0]));
        }
    }
```

Reducer类

```
    public static class MyReduce extends Reducer<IntWritable, Text, Text,
IntWritable> {
        public void reduce(IntWritable key, Iterable<Text> values,
                           Context context)
                throws IOException, InterruptedException {
            for (Text value : values) {
                context.write(value, key);
            }
        }
    }
```

Driver代码

```
 public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args)
                .getRemainingArgs();
        if (otherArgs.length < 2) {
            System.err.println("必须输入读取文件路径和输出路径");
            System.exit(2);
        }
        Job job = Job.getInstance(conf, "Visits Sort");
        job.setJarByClass(accessTimesSort.class);
        job.setMapperClass(MyMap.class);
        job.setReducerClass(MyReduce.class);
        job.setMapOutputKeyClass(IntWritable.class);
        job.setMapOutputValueClass(Text.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        for (int i = 0; i < otherArgs.length - 1; ++i) {
            FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
        }
        FileOutputFormat.setOutputPath(job,
```

```
                new Path(otherArgs[otherArgs.length - 1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }ja
```

编译生成jar包

菜单-》Build -》 Build Artifacts -》hadoop-》Build， 参考4.3节编译生成jar包图例

上传jar包到master

1. 在Hadoop项目，左侧树图中-》out-》artifacts -》hadoop -》 hadoop.jar ，右击hadoop.jar，菜单中选择复制
2. 打开xftp，进入master-》root-》hadoop目录，右击选择粘贴
3. 参考4.3节上传jar包到master

运行jar

使用crt或xshell进入master

```
cd /root/hadoop
hadoop  jar hadoop.jar ch04.accessTimesSort /user/myname/output_AccessCount
/user/myname/output_AccessTimeSort
```

运行结果如：

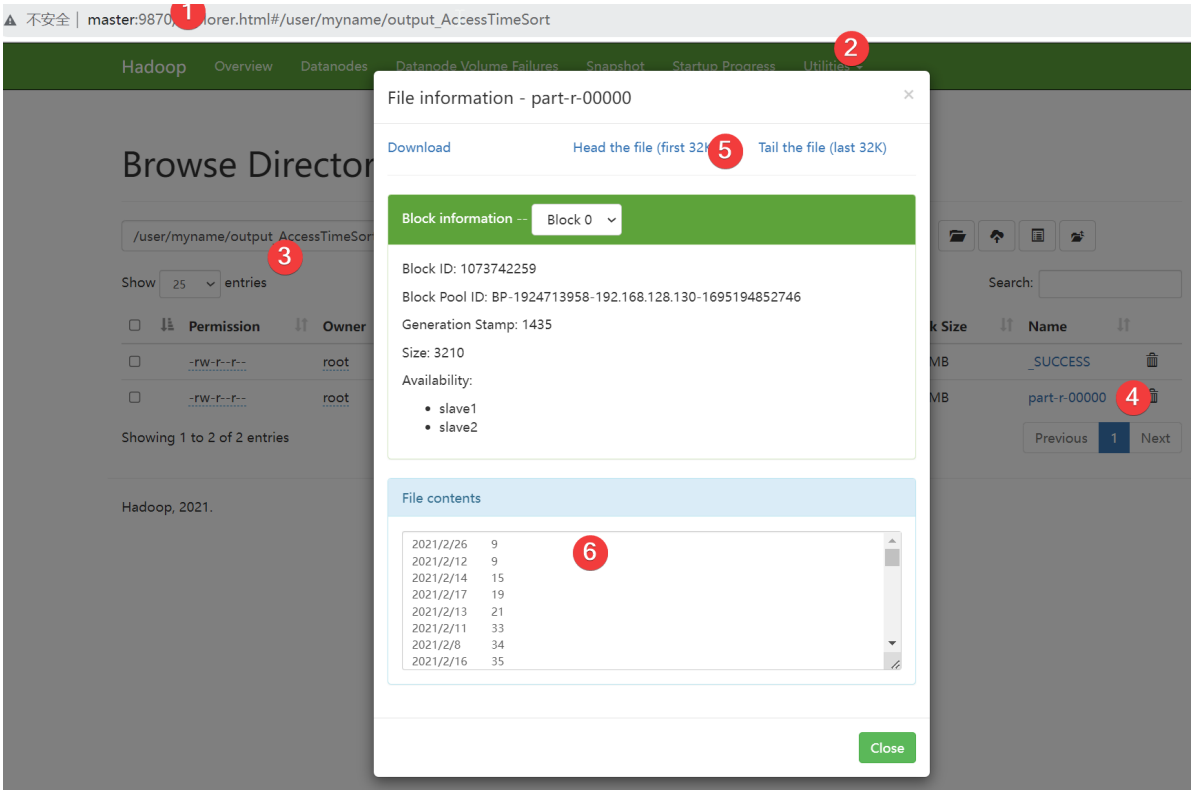
```
[root@master hadoop]# hadoop  jar hadoop.jar ch04.accessTimesSort
/user/myname/output_AccessCount /user/myname/output_AccessTimeSort
2023-10-01 11:54:30,995 INFO client.RMProxy: Connecting to ResourceManager
at master/192.168.128.130:8032
2023-10-01 11:54:32,329 INFO mapreduce.JobResourceUploader: Disabling
Erasure Coding for path: /tmp/hadoop-
yarn/staging/root/.staging/job_1696004849840_0002
2023-10-01 11:54:32,907 INFO input.FileInputFormat: Total input files to
process : 1
2023-10-01 11:54:33,101 INFO mapreduce.JobSubmitter: number of splits:1
2023-10-01 11:54:33,433 INFO mapreduce.JobSubmitter: Submitting tokens for
job: job_1696004849840_0002
2023-10-01 11:54:33,435 INFO mapreduce.JobSubmitter: Executing with tokens:
[]
2023-10-01 11:54:33,870 INFO conf.Configuration: resource-types.xml not
found
2023-10-01 11:54:33,870 INFO resource.ResourceUtils: Unable to find
'resource-types.xml'.
2023-10-01 11:54:34,018 INFO impl.YarnClientImpl: Submitted application
application_1696004849840_0002
2023-10-01 11:54:34,107 INFO mapreduce.Job: The url to track the job:
http://master:8088/proxy/application_1696004849840_0002/
2023-10-01 11:54:34,109 INFO mapreduce.Job: Running job:
job_1696004849840_0002
2023-10-01 11:54:46,491 INFO mapreduce.Job: Job job_1696004849840_0002
running in uber mode : false
2023-10-01 11:54:46,494 INFO mapreduce.Job:  map 0% reduce 0%
```

```
2023-10-01 11:54:55,681 INFO mapreduce.Job:  map 100% reduce 0%
2023-10-01 11:55:02,758 INFO mapreduce.Job:  map 100% reduce 100%
2023-10-01 11:55:02,780 INFO mapreduce.Job: Job job_1696004849840_0002 completed successfully
2023-10-01 11:55:02,976 INFO mapreduce.Job: Counters: 53
        File System Counters
                FILE: Number of bytes read=3590
                FILE: Number of bytes written=451411
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3337
                HDFS: Number of bytes written=3210
                HDFS: Number of read operations=8
                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)=25848
                Total time spent by all reduces in occupied slots (ms)=19936
                Total time spent by all map tasks (ms)=6462
                Total time spent by all reduce tasks (ms)=4984
                Total vcore-milliseconds taken by all map tasks=6462
                Total vcore-milliseconds taken by all reduce tasks=4984
                Total megabyte-milliseconds taken by all map tasks=13234176
                Total megabyte-milliseconds taken by all reduce tasks=10207232
        Map-Reduce Framework
                Map input records=224
                Map output records=224
                Map output bytes=3136
                Map output materialized bytes=3590
                Input split bytes=127
                Combine input records=0
                Combine output records=0
                Reduce input groups=212
                Reduce shuffle bytes=3590
                Reduce input records=224
                Reduce output records=224
                Spilled Records=448
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=291
                CPU time spent (ms)=3360
                Physical memory (bytes) snapshot=315863040
                Virtual memory (bytes) snapshot=7196200960
                Total committed heap usage (bytes)=140865536
                Peak Map Physical memory (bytes)=206155776
                Peak Map Virtual memory (bytes)=3594711040
                Peak Reduce Physical memory (bytes)=109707264
                Peak Reduce Virtual memory (bytes)=3601489920
        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=3210
        File Output Format Counters
                Bytes Written=3210
[root@master hadoop]#
```

在HDFS查看运行结果：

根据crt命令行的运行命令，输出的文件在目录下：/user/myname/output_AccessTimeSort



GIT源码下载参考：

使用方法

1. 安装Git-2.39.2-64-bit.exe和TortoiseGit-2.14.0.0-64bit.msi
2. 进入放置源码的路径，如D://hadoop
3. 右击空白，在菜单中选择git clone
4. 在url中粘贴： https://jihulab.com/biglab-share/hadoop.git
5. 确认，下载源码

GIT安装视频

学习视频

版本历史

Ver1.0-20230612

- 初始版本

Ver1.1-20230613

- 更新eclipse版本为2023.03

Ver1.2-20230828

- 增加markdown版本

Ver1.3-20231001

- 修改更新章节内容，案例更新

Ver1.4-20231024

- 修改更新实训1内容，案例更新

上一章节 下一章节