第9章综合项目实训

CH09实训1：Hadoop综合实训项目

训练要点

- 掌握HDFS文件系统的操作
    1. 掌握hdfs目录创建,文件上传下载
    2. 掌握hdfs的API操作
- 掌握MapReduce的编程
    1. 掌握MapReduce方法和实现
    2. 掌握自定义数据类型
    3. 掌握自定义计数器
    4. 掌握MapReduce 参数的传递
    5. 掌握MapReduce通过使用Combiner,Partitioner来优化的方法
- 掌握MapReduce程序部署和测试
    1. 掌握MapReduce程序打包和运行
    2. 掌握MapReduce程序功能测试
- 应用跨学科知识
    1. 使用Linux的Shell编程
    2. 使用Python数据采集和数据可视化
    3. 应用软件工程项目管理方法

需求说明

1. 本实训充许同学们采集各类题材数据,包括并不限于:商品、音乐、新闻、房产、书籍、招聘
2. 本实训要实现的功能是通过同学采集某类题材数据，采集题材数据到mongodb, 再从mongodb将所有同学采集的同题材数据采集hdfs，进行mapreduce分析，输出分析结果到hdfs，并将结果以可视化方式展现。

实现步骤

1. 数据采集：使用scrapy框架实现某类题材网站的数据采集，存入mongo数据库。
2. 数据分析：1）使用java采集题材数据从mongo到hdfs； 2）使用java对hdfs上的数据进行MR分析；3）使用java将分析结果进行排序。
3. 数据可视：使用python将MR的排序后的结果进行可视化并上传到web服务器。

作业要求

1. 对实训当中要实现的功能进行描述、架构设计、详细设计，整理入实训报告；
2. 对实训过程中源码、操作步骤、运行结果截图，整理入实训报告；
3. 整理上述过程中实现的源码，包括采集的源码，MR分析源码，可视化的源码，全部打包成一个rar文件。

环境准备

采集环境准备

chrome和chromedriver下载

方式一，chrome和chromedriver版本要配套，请从百度网盘上下载：

```
百度网盘下载
链接：https://pan.baidu.com/s/1FF7HHQi9y2kVBXom_Omd_g
提取码：8ft4
在google目录下有chrome的109版本和chromedriver的109版
```

方式二，使用链接下载

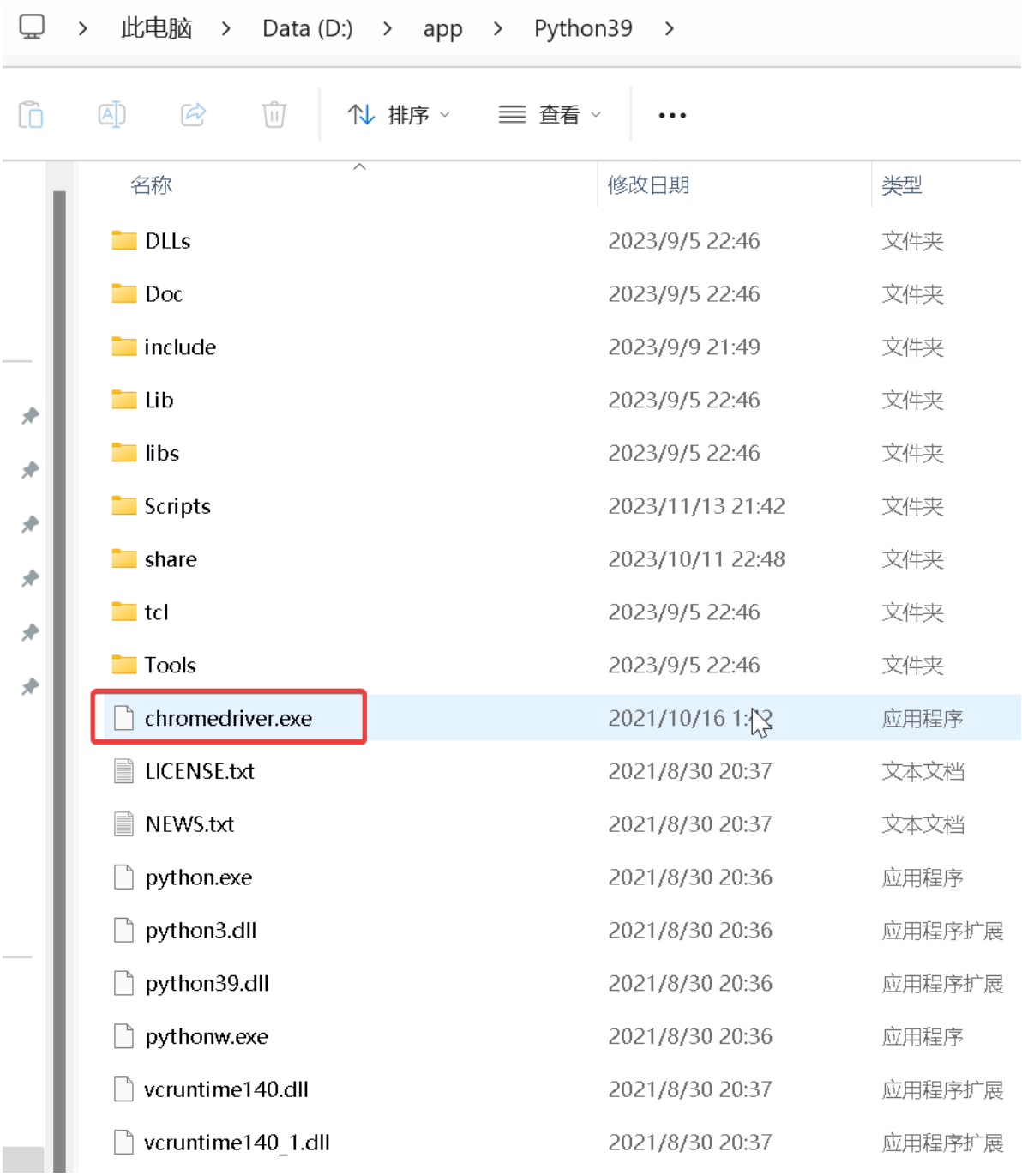
```
链接下载google 109版本
https://dl.google.com/release2/chrome/ad2uwza6rxngw4rvu7lrmj5rvtca_109.0.5414.75/109.0.5414.75_chrome_installer.exe
链接下载chromedriver版本109
https://registry.npmmirror.com/-/binary/chromedriver/109.0.5414.74/chromedriver_win32.zip
```

方式二，使用当前最新的chrome版本，如119,120, 去下载最新的驱动

```
chromedriver最新版本下载：
https://googlechromelabs.github.io/chrome-for-testing/
```

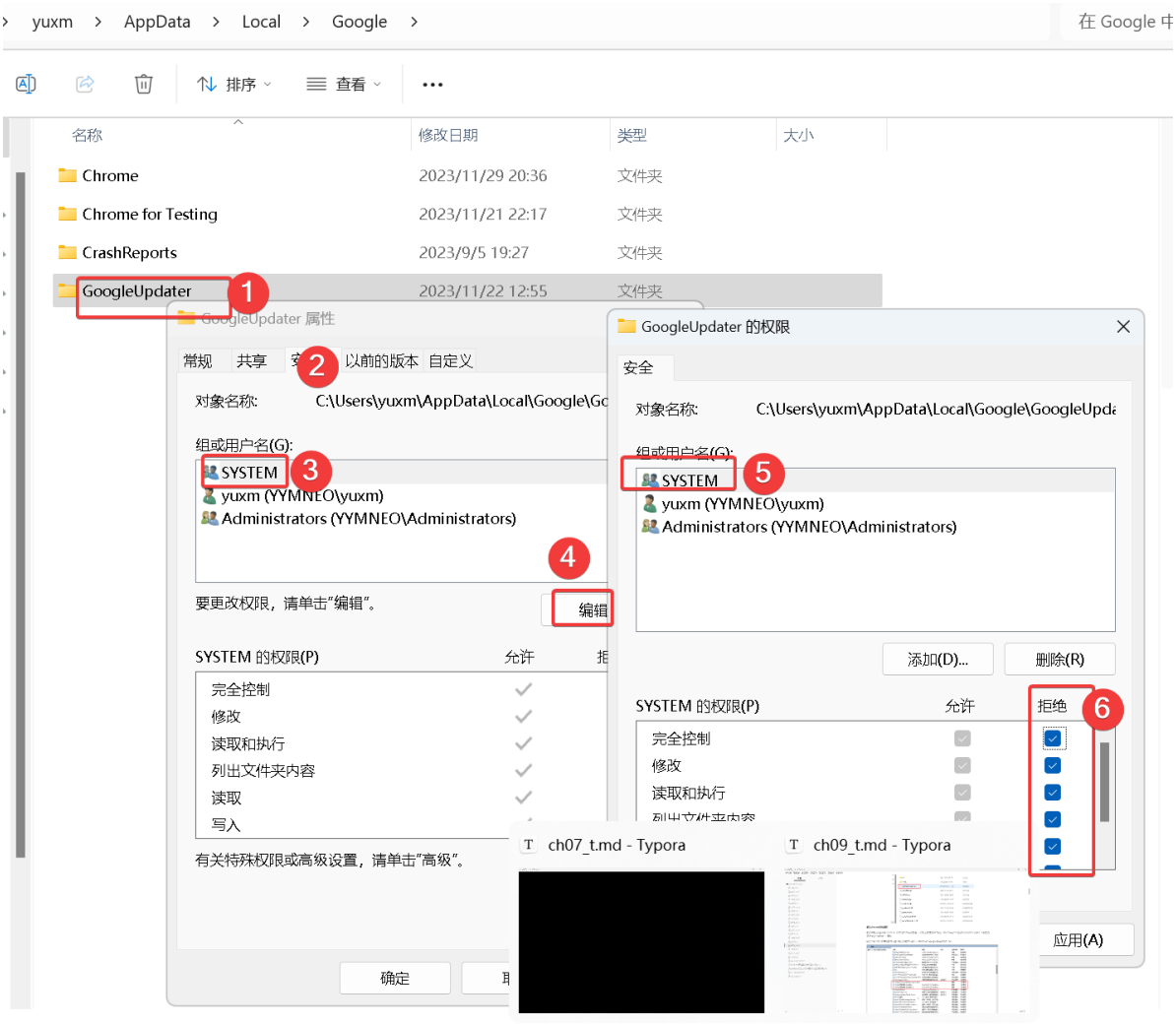
chromedriver.exe部署

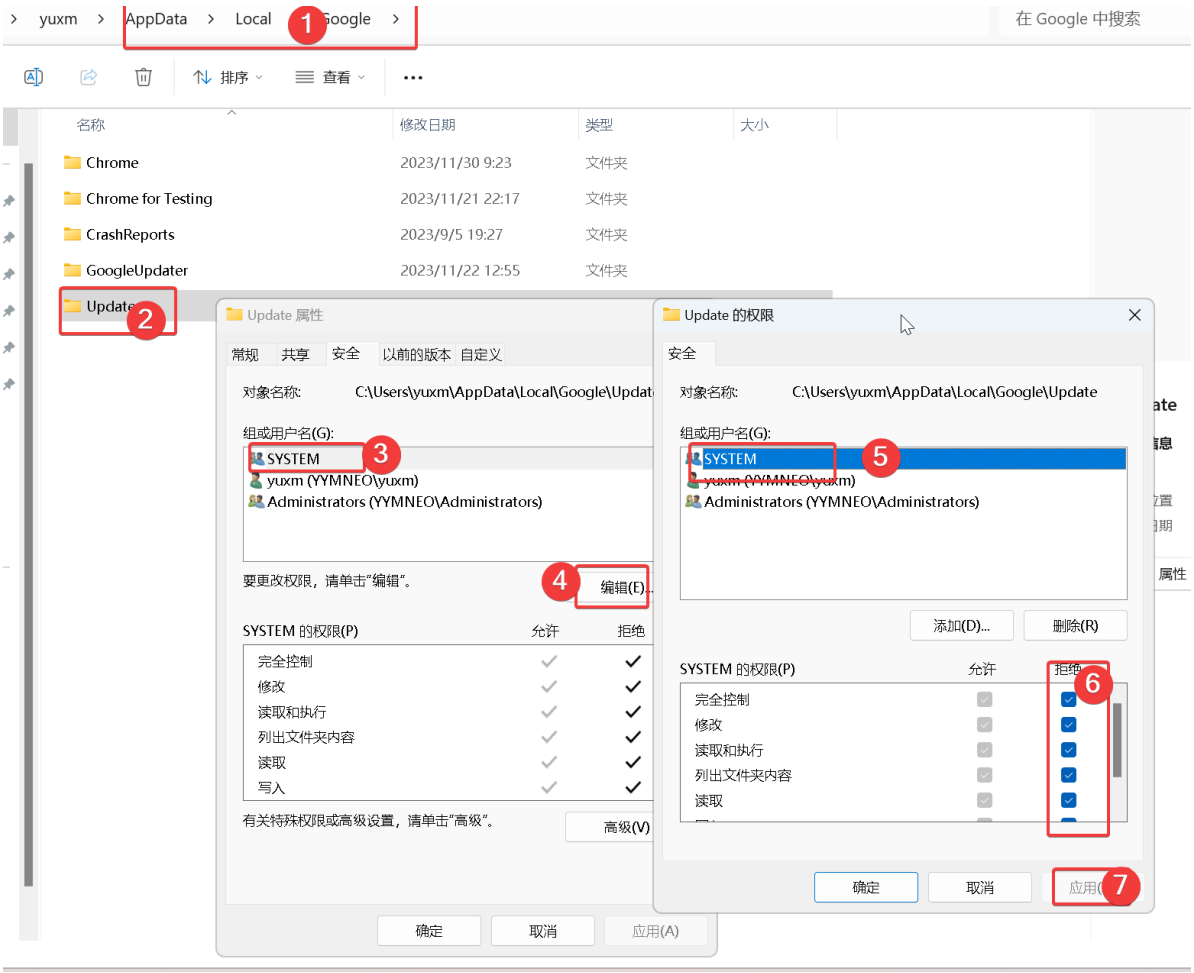
将下载下来的chromedriver_win32_95.zip解压出来的chromedriver.exe复制到python的路径下，如：



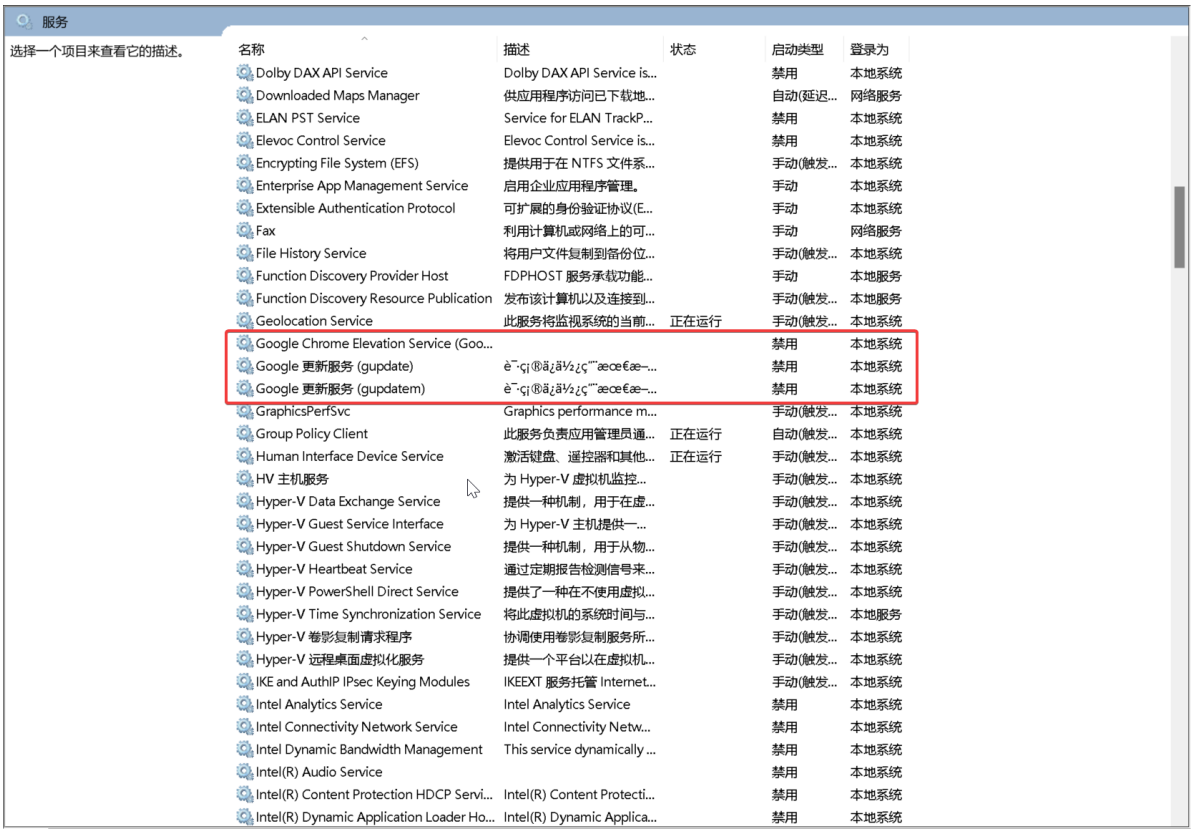
禁止chrome自动更新

右击桌面上的google chrome ->打开文件所在的目录 -> 退到上两级目录（如:C:\Users\yuxm\AppData\Local\Google）->分别右击目录:GoogleUpdater 和目录:Update ->属性->安全->编辑->拒绝->应用,如下图:

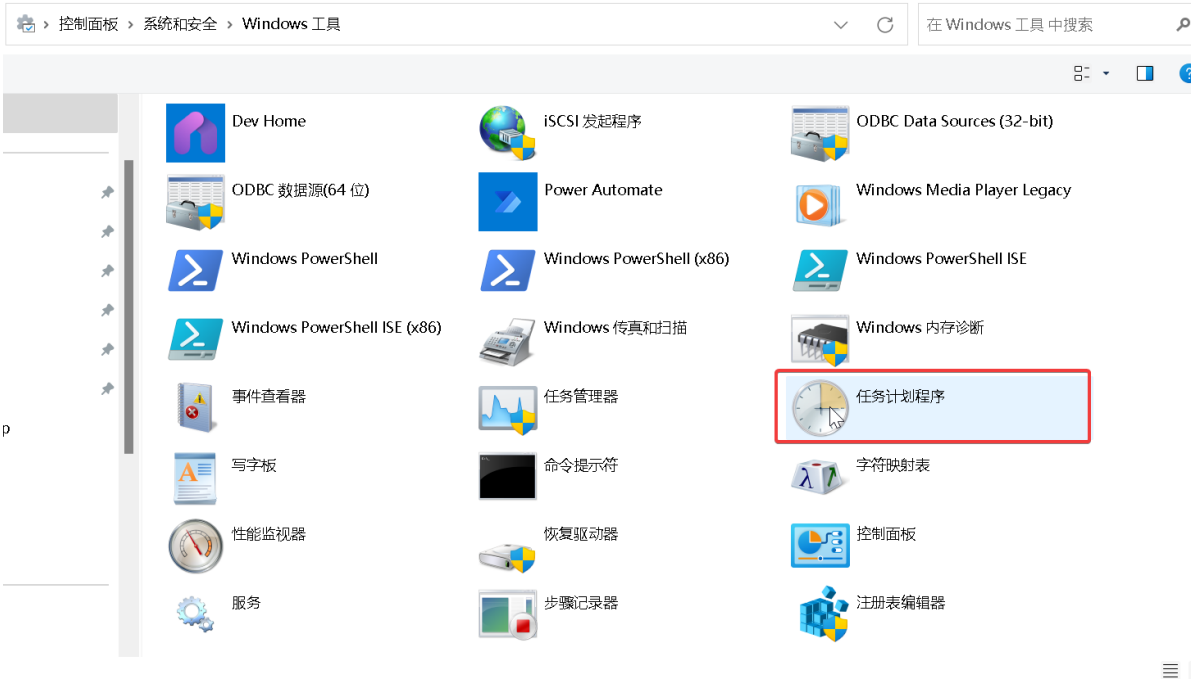

yuxm › AppData › Local › Google ›
在 Google 中
排序
查看
名称
修改日期
类型
大小
Chrome
2023/11/29 20:36
文件夹
Chrome for Testing
2023/11/21 22:17
文件夹
CrashReports
2023/9/5 19:27
文件夹
GoogleUpdater
2023/11/22 12:55
文件夹
1
GoogleUpdater 属性
常规
共享
以前的版本
自定义
2
对象名称:
C:\Users\yuxm\AppData\Local\Google\Gc
组或用户名(G):
SYSTEM
3
yuxm (YYMNEO\yuxm)
Administrators (YYMNEO\Administrators)
4
要更改权限，请单击"编辑"。
编辑
SYSTEM 的权限(P)
允许
完全控制
修改
读取和执行
列出文件夹内容
读取
写入
有关特殊权限或高级设置，请单击"高级"。
确定
GoogleUpdater 的权限
安全
对象名称:
C:\Users\yuxm\AppData\Local\Google\GoogleUpda
组或用户名(G):
SYSTEM
5
yuxm (YYMNEO\yuxm)
Administrators (YYMNEO\Administrators)
添加(D)...
删除(R)
SYSTEM 的权限(P)
允许
拒绝
6
完全控制
修改
读取和执行
列出文件夹内容
读取
应用(A)
ch07_t.md - Typora
ch09_t.md - Typora


yuxm › AppData › Local › Google ›
1
在 Google 中搜索
排序
查看
名称
修改日期
类型
大小
Chrome
2023/11/30 9:23
文件夹
Chrome for Testing
2023/11/21 22:17
文件夹
CrashReports
2023/9/5 19:27
文件夹
GoogleUpdater
2023/11/22 12:55
文件夹
Update
2
Update 属性
常规
共享
安全
以前的版本
自定义
对象名称:
C:\Users\yuxm\AppData\Local\Google\Updat
组或用户名(G):
SYSTEM
3
yuxm (YYMNEO\yuxm)
Administrators (YYMNEO\Administrators)
4
要更改权限，请单击"编辑"。
编辑(E)...
SYSTEM 的权限(P)
允许
拒绝
完全控制
修改
读取和执行
列出文件夹内容
读取
写入
有关特殊权限或高级设置，请单击"高级"。
高级(V)
确定
取消
应用(A)
Update 的权限
安全
对象名称:
C:\Users\yuxm\AppData\Local\Google\Update
组或用户名(G):
SYSTEM
5
yuxm (YYMNEO\yuxm)
Administrators (YYMNEO\Administrators)
添加(D)...
删除(R)
SYSTEM 的权限(P)
允许
拒绝
6
完全控制
修改
读取和执行
列出文件夹内容
读取
确定
取消
7

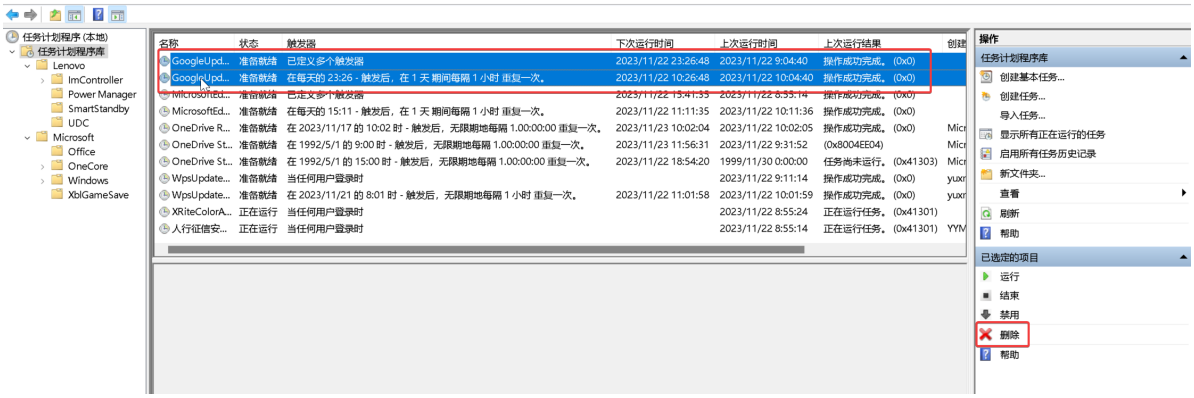
右击开始->打开计算机管理->服务和应用程序->服务，禁用所有和google有关的程序，如：



单击开始->所有应用->window工具系统->任务计划程序



删除所有和google相关的更新程序



采集IDEA环境

使用pycharm的community版本

下载地址

使用python 3.9.13 版本

下载地址

说明：为了确保使用的是系统解析器，建议先通过控制面版，卸载之前的所有python版本，包括虚拟环境。然后再安装 python3.9.13到d://app/python目录下。

安装依赖包

WIN键+R运行cmd进入命令行：

```
python -m pip install --upgrade pip -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install pymongo -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install pymysql -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install scrapy -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install pandas -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install sqlalchemy -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install bs4 -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install selenium -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install redis -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install bgutils-hddly -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com
pip install --upgrade bgutils-hddly -i http://mirrors.aliyun.com/pypi/simple ^
--trusted-host mirrors.aliyun.com

```

MR分析环境准备

需要开启hadoop集群

在master主机上运行：

```
start-all.sh
```

需要同步hadoop主机和从机的时间，以及windows时间

以下在master上和slave1,slave2上执行

```
systemctl stop ntpd
ntpdate cn.pool.ntp.org
service ntpd start & chkconfig ntpd on
```

需要设置目录的权限

以下在master或者在slave1,slave2上执行,效果都一样

```
hdfs dfs -chmod -R 777 /
```

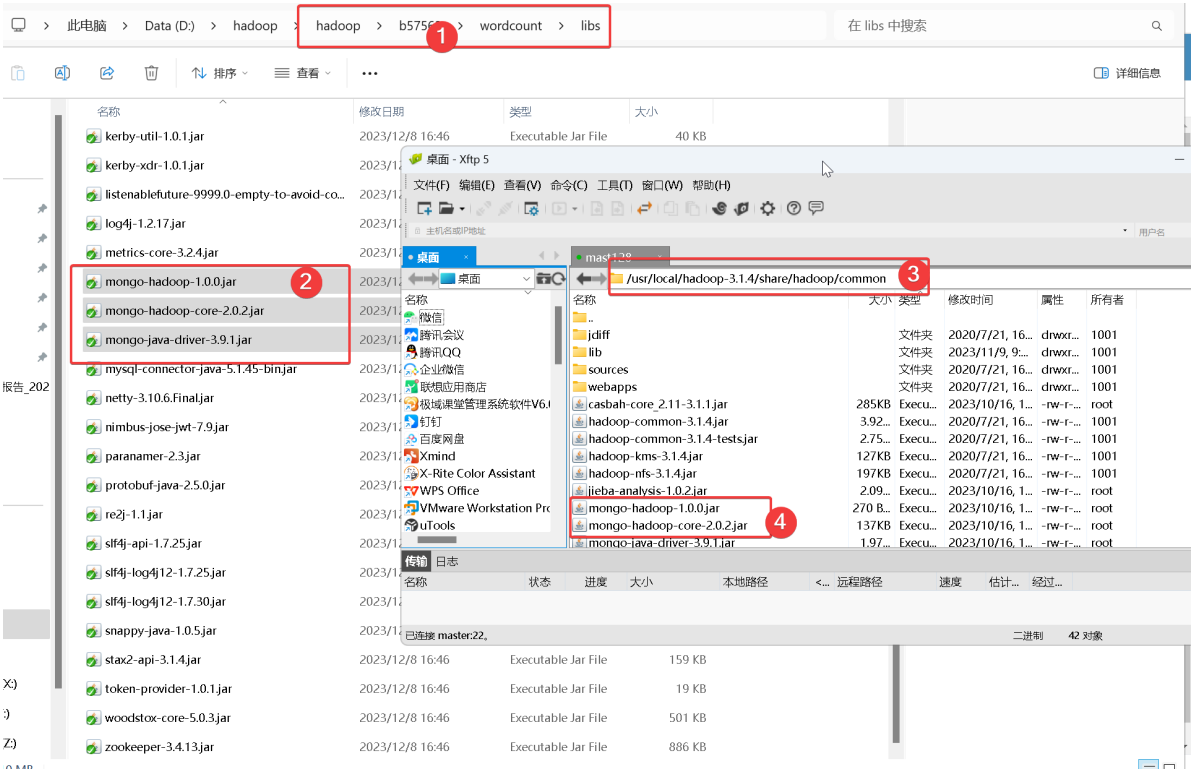
需要将mongodb相在的依赖包上传到集群

将程序打包上传到集群运行时，依赖mongodb包，因此需需要将相关的依赖包上传，方法如下：

源git clone：hadoop\b57562\wordcount\libs目录下mongo-hadoop-1.0.0.jar、mongo-hadoop-core-2.0.2.jar、mongo-java-driver-3.9.1.jar

集群master,slave1,slave2目标目录： /usr/local/hadoop-3.1.4/share/hadoop/common

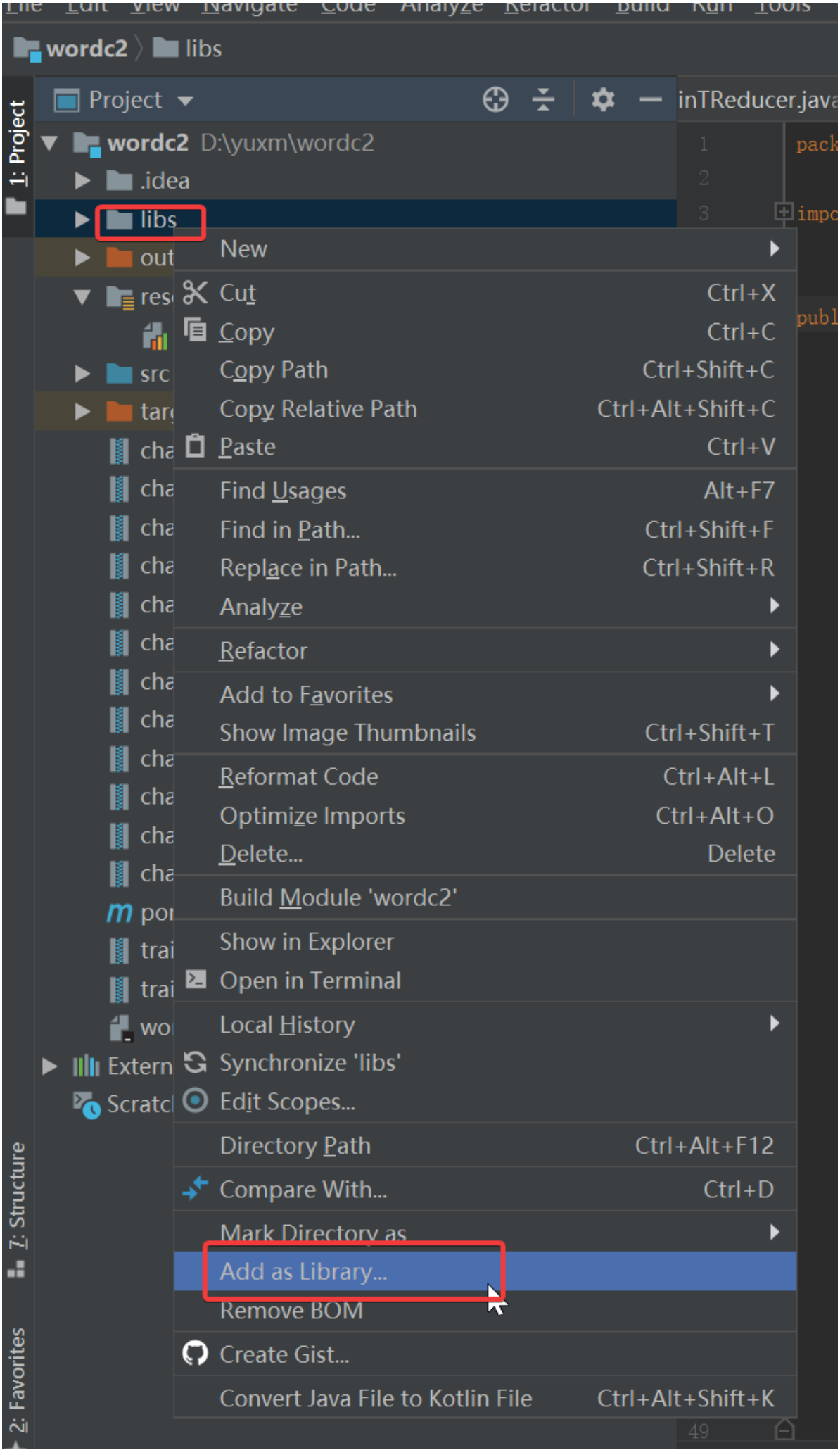
使用xftp工具分别连接到master,slave1,slave2，然后上传：



MR源码调试注意事项

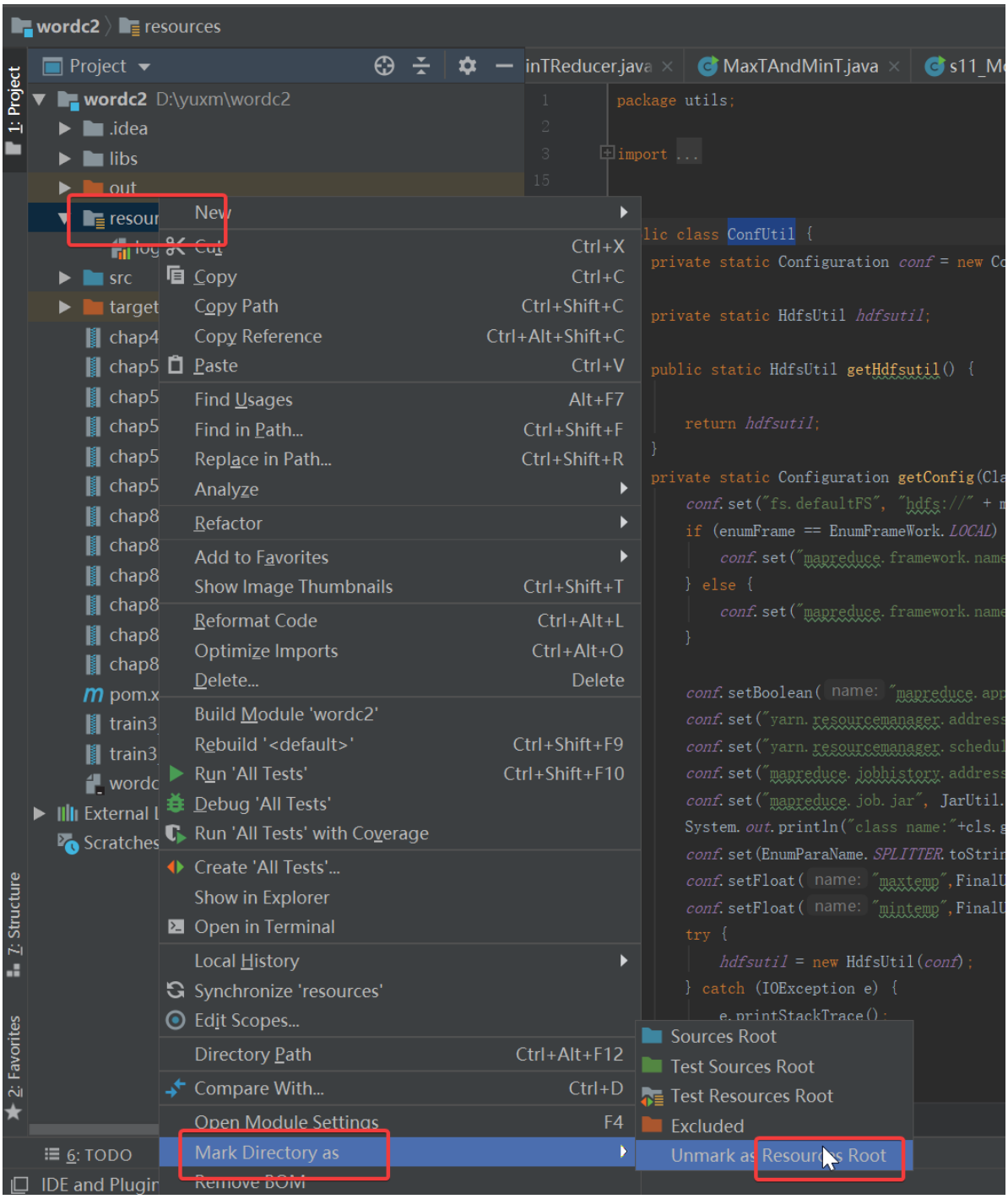
添加libs目录

从git clone下来的目录中，将\hadoop\b57562\wordcount\libs目录复制到本项目根目录下，注意右击libs目录从菜单中选择add as library:


wordc2
libs
Project
wordc2 D:\yuxm\wordc2
.idea
libs
out
src
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Relative Path Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Build Module 'wordc2'
Show in Explorer
Open in Terminal
Local History
Synchronize 'libs'
Edit Scopes...
Directory Path Ctrl+Alt+F12
Compare With... Ctrl+D
Mark Directory as
Add as Library...
Remove BOM
Create Gist...
Convert Java File to Kotlin File Ctrl+Alt+Shift+K
1: Project
7: Structure
2: Favorites

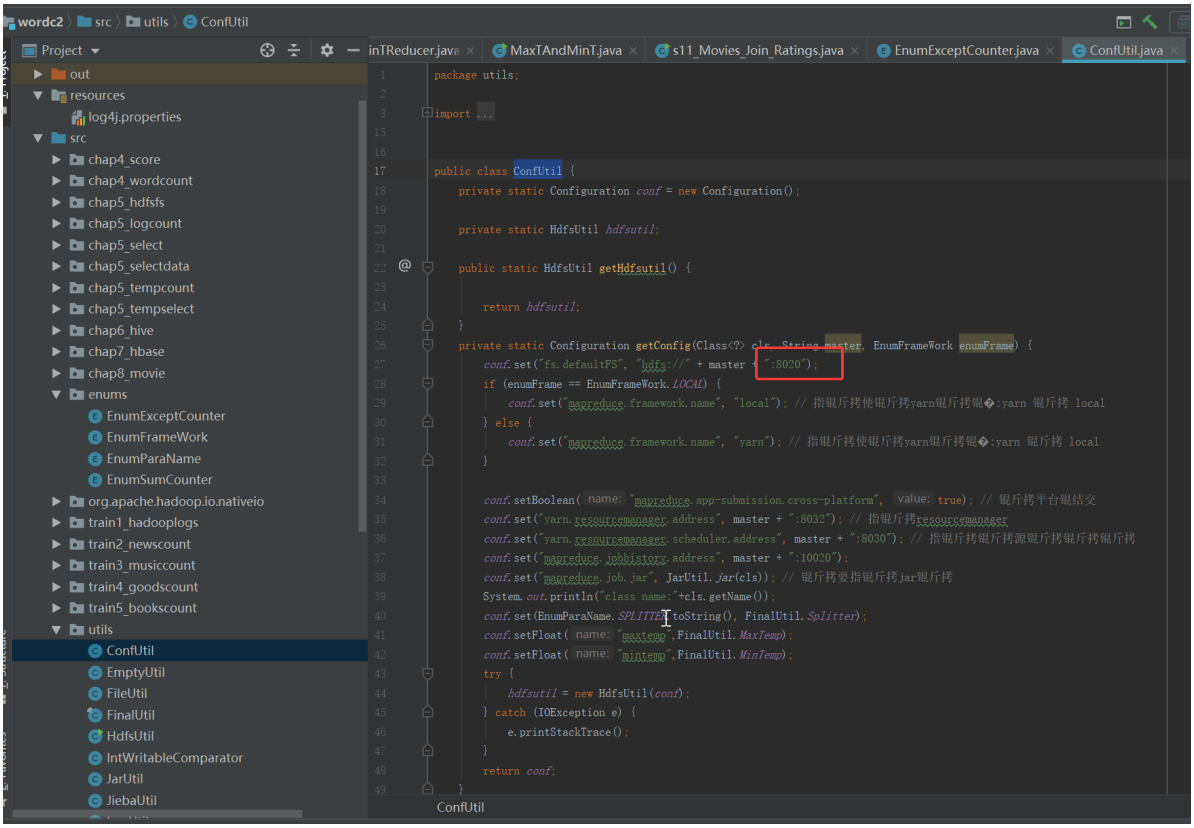
添加resources资源目录

从git clone下来的目录中，将hadoop\b57562\wordcount\src\resources目录复制到本项目根目录下，然后右击resources目录从菜单中选择make directory as -> resources root:



ConfUtil文件修改

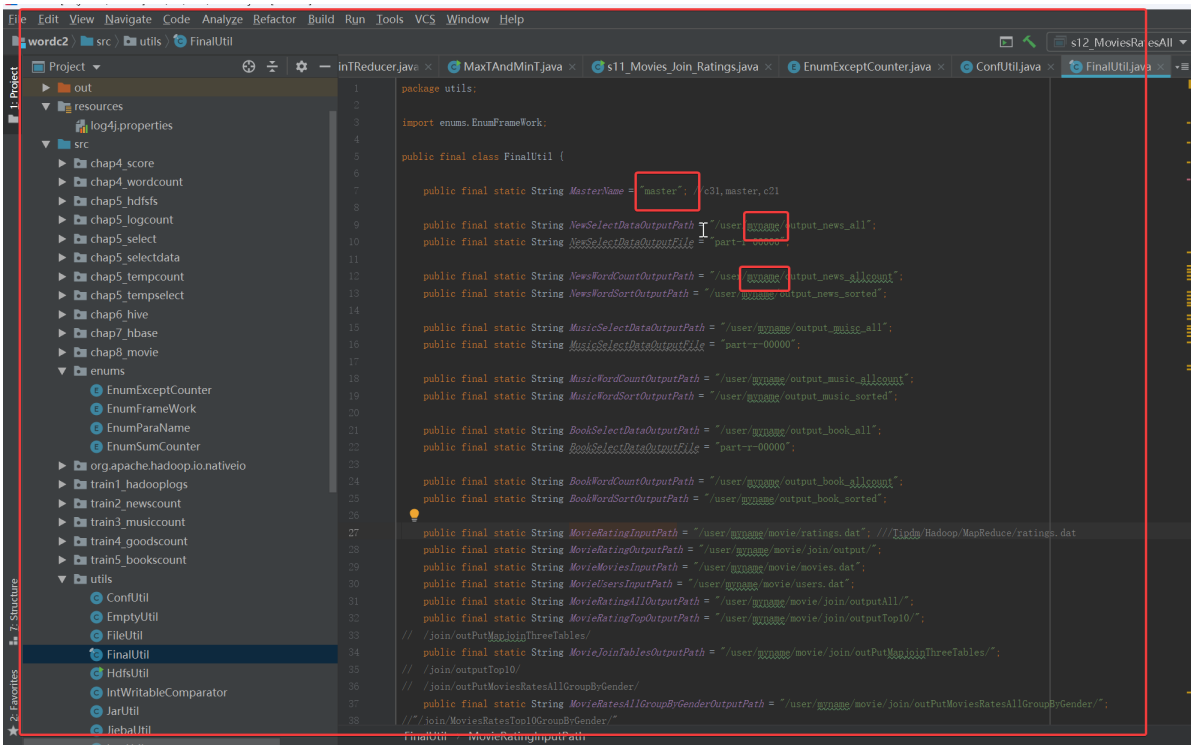
注意，如果集群上core-site.xml配置的端口是8020，注意这里要改成8080



finalutil文件修改

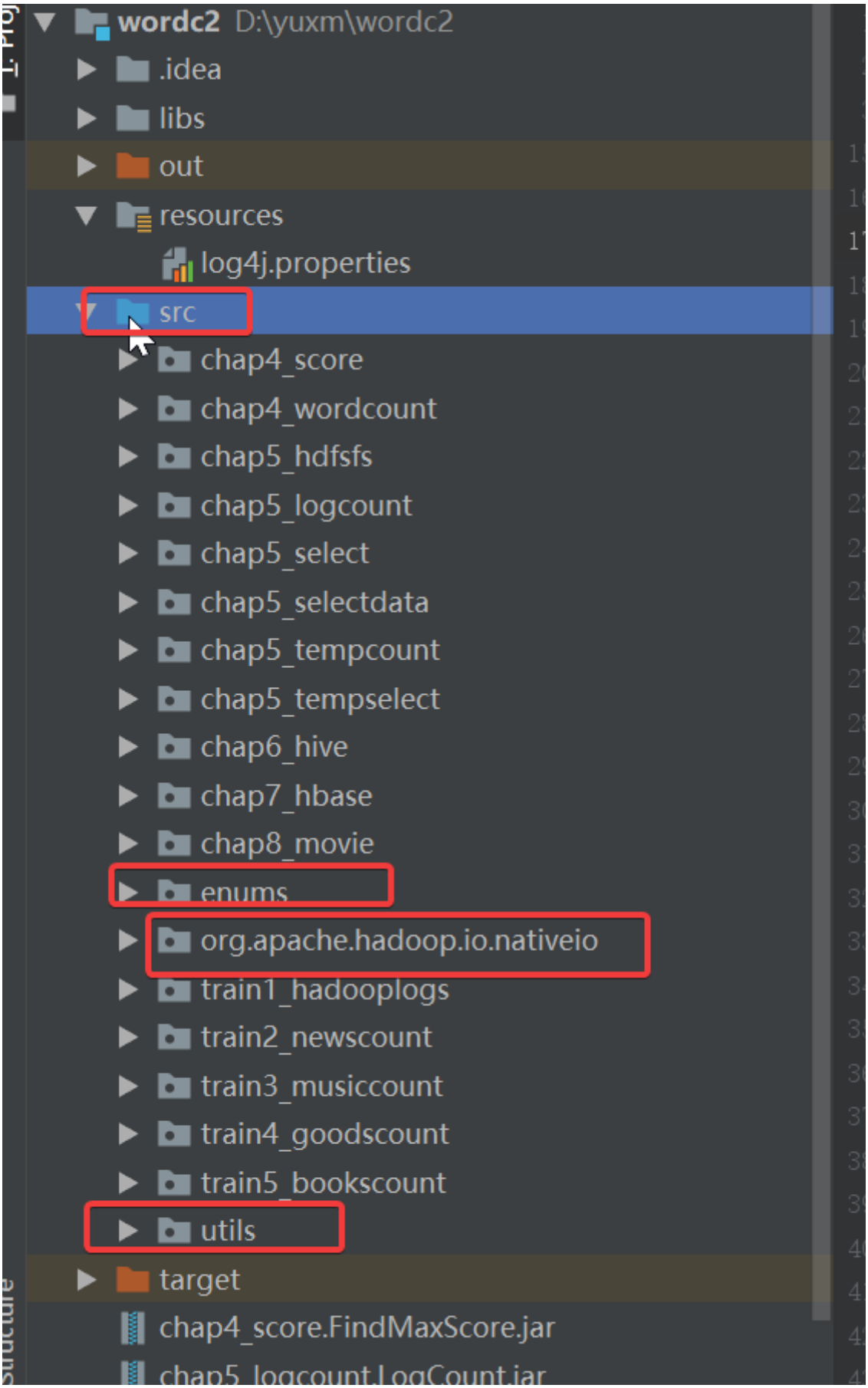
注意：

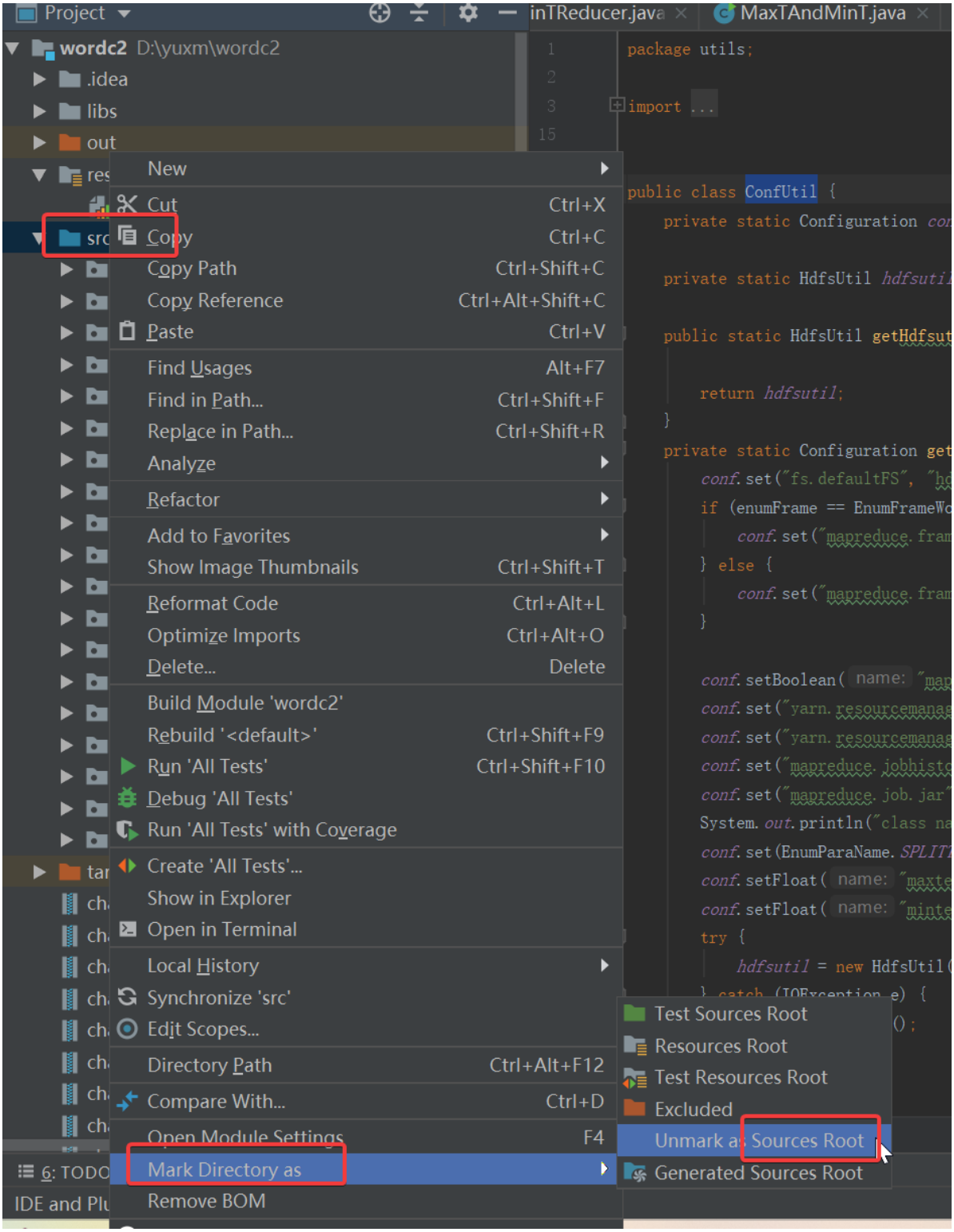
1. 修改主机名为"master"
2. 所有myname注意替换为本人姓名的拼音或者简拼
3. 发现MR运行失败可以尝试本地方式运行：public final static EnumFrameWork enumFrame = EnumFrameWork.LOCAL; // YARN,LOCAL



关于源码根路径

建议删除src目录下的java路径，将主要的包：enums,utils,org以及需要的包目录复制到src目录下，并将src设置为源码根路径：





报告模板

实训报告模板：

[Hadoop综合实训项目实验报告Ver2.0.1]

报告编写

1.1. 概述（5分）

1.1.1. 训练要点(1分)

- 掌握HDFS文件系统的操作
    1. 掌握hdfs目录创建,文件上传下载
    2. 掌握hdfs的API操作
- 掌握MapReduce的编程
    1. 掌握MapReduce方法和实现
    2. 掌握自定义数据类型
    3. 掌握自定义计数器
    4. 掌握MapReduce 参数的传递
    5. 掌握MapReduce通过使用Combiner,Partitioner来优化的方法
- 掌握MapReduce程序部署和测试
    1. 掌握MapReduce程序打包和运行
    2. 掌握MapReduce程序功能测试
- 应用跨学科知识
    1. 使用Linux的Shell编程
    2. 使用Python数据采集和数据可视化
    3. 应用软件工程项目管理方法

1.1.2. 需求说明(2分)

1. 本实训充许同学们采集各类题材数据,包括并不限于:商品、音乐、新闻、房产、书籍、招聘
2. 本实训要实现的功能是通过同学采集某类题材数据，采集题材数据到mongodb, 再从mongodb将所有同学采集的同题材数据采集hdfs，进行mapreduce分析，输出分析结果到hdfs，并将结果以可视化方式展现。

1.1.3. 实现步骤(2分)

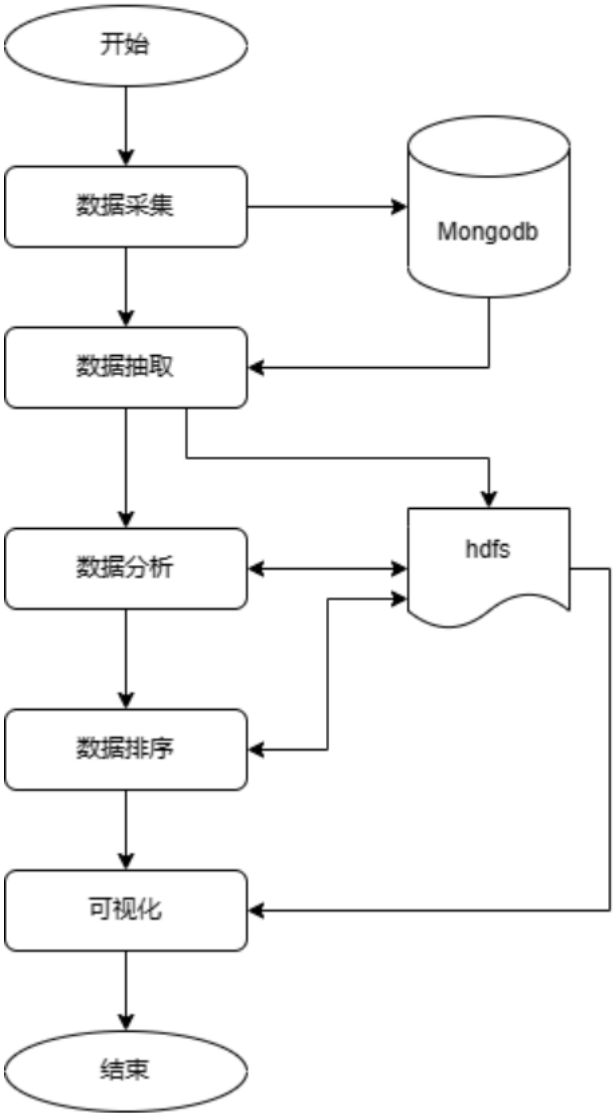
1. 数据采集：使用scrapy框架实现某类题材网站的数据采集，存入mongo数据库。
2. 数据分析：1）使用java采集题材数据从mongo到hdfs； 2）使用java对hdfs上的数据进行MR分析；3）使用java将分析结果进行排序。
3. 数据可视：使用python将MR的排序后的结果进行可视化并上传到web服务器。

1.1. 总体设计(30分)

1.1.1. 业务流程图(7分)

【业务流程图】(5分)

Hadoop综合实训流程

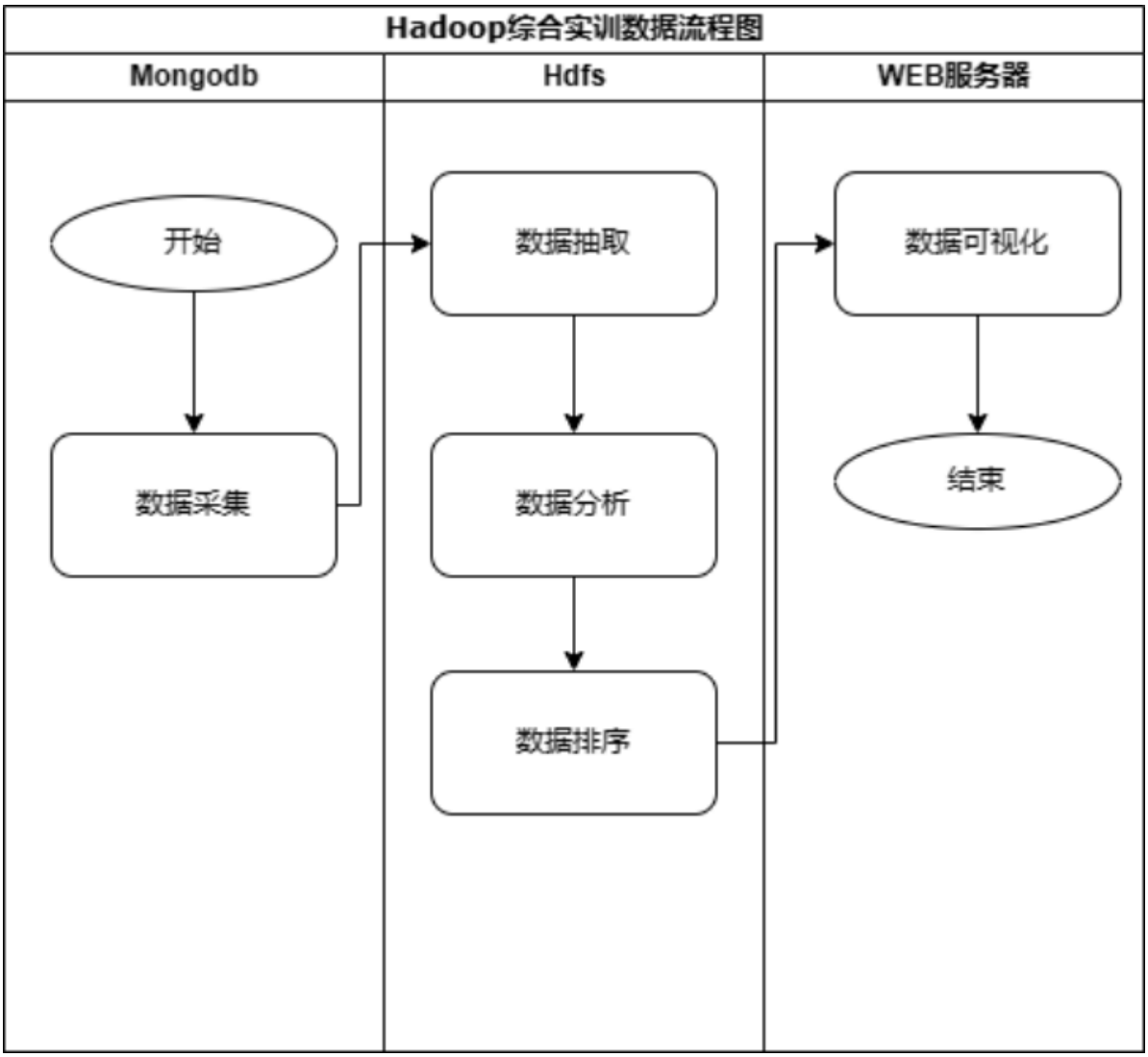


【业务流程说明】(2分)

1. 数据采集：从网站将音乐数据采集到mongodb库，使用scrapy框架
2. 数据抽取：从mongodb库中抽取hdfs文件系统
3. 数据分析：从hdfs取出文件，进行音乐词频分析，将分析后的结果存入hdfs
4. 数据排序：从hdfs取出数据分析的结果，进行按统计数据逆序排序
5. 数据可视化：将hdfs中数据排序的结果复制到可视化的脚本中，进行图形展示

1.1.2. 数据流程图(8分)

【数据流程图】(5分)



【数据流程说明】(3分)

1. 在数据采集阶段：从千千音乐网站将音乐数据采集到mongodb库，使用scrapy框架
2. 在数据存储阶段：

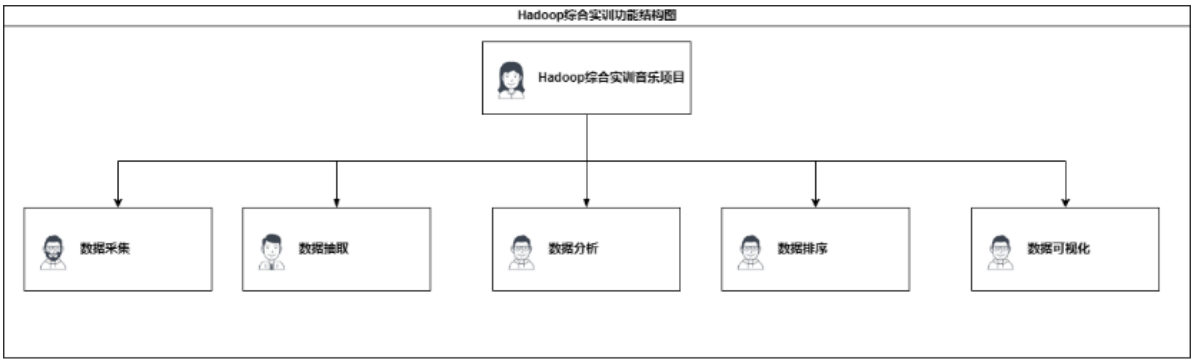
   1)数据抽取：从mongodb库中抽取hdfs文件系统

   2)数据分析：从hdfs取出文件，进行音乐词频分析，将分析后的结果存入hdfs

   3)数据排序：从hdfs取出数据分析的结果，进行按统计数据逆序排序
3. 在数据可视化阶段：将hdfs中数据排序的结果截取一部分复制到可视化的脚本中运行，展示音乐词频统计条形图

1.1.3. 系统功能结构(5分)

【模块组织结构图】(5分)

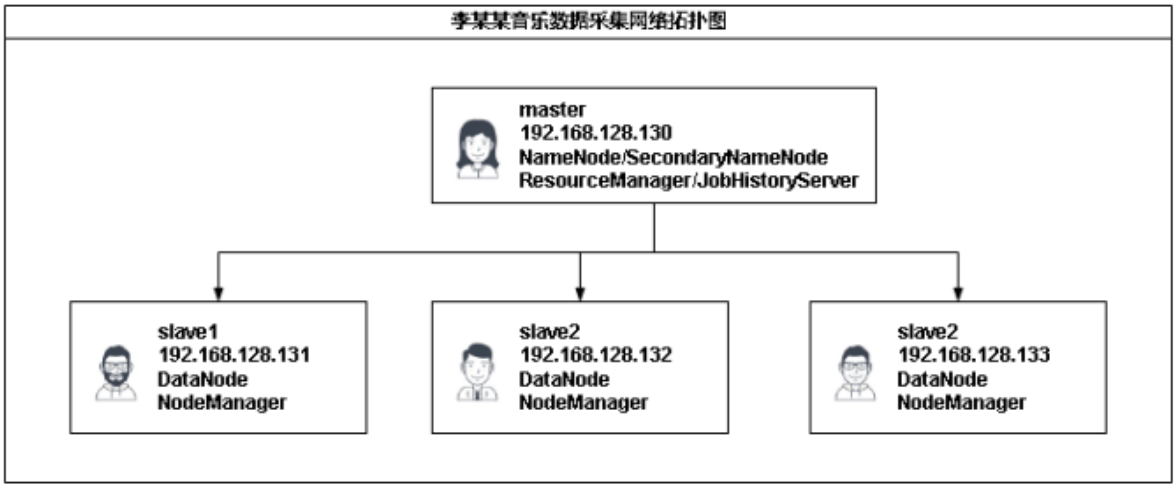


1.1.4. 运行环境(10分)

l 操作系统和软件依赖(5分)

子系统	操作系统	依赖软件	备注
数据分析	Linux,Window	Vmware,IDEA,Mongodb,Hadoop, jdk	
数据抓取	Window	PyCharm ,python,chrome,chromedriver	
数据可视化	Window	PyCharm,web, python	

l 网络拓朴图(5分)



1.2. 详细设计(60分)

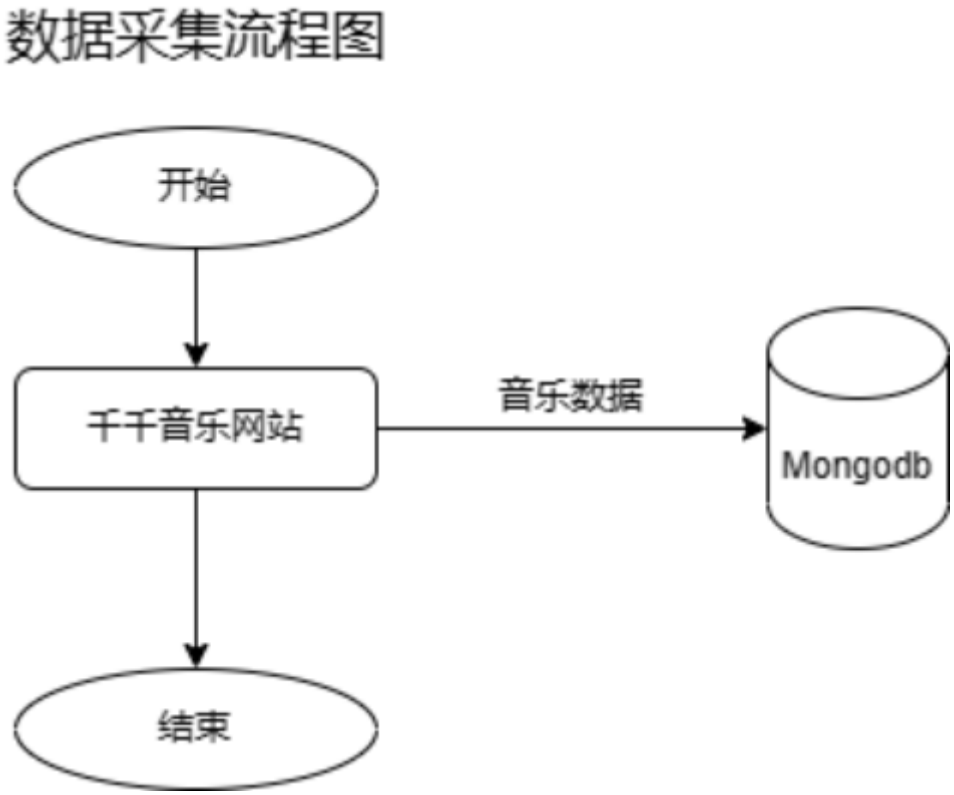
1.2.1. 数据采集(10分)

【功能说明】(1分)：

采集内容：音乐数据，包括歌曲名称，歌曲url，歌手等数据
采集过程：利用Pycharm从千千音乐网站采集音乐数据到mongodb库中
框架：使用scrapy框架

【功能设计】(2分)：

将千千音乐网站的音乐数据，包括歌曲名称，歌曲url，歌手等信息采集到mongodb远程数据库中



【源码实现】(4分）：

scrapy脚本:

```python
import json
import time

import scrapy
from scrapy import Selector
from selenium import webdriver

from musics.items import MusicsItem


class QingqingSpider(scrapy.Spider):
    name = 'qingqing'
    allowed_domains = ['music.taihe.com']
    # start_urls = ['http://music.taihe.com/']
    start_urls = ['https://music.taihe.com/artist']
    # 按歌手来查
    cookie_str = 'Hm_lvt_d0ad46e4afeacf34cd12de4c9b553aa6=1638943302;
token_type=access_token;
access_token=MGQzMDNjZWZlNjlhODgwN2Q5MWM3Y2I1OTQ4YmY0NWU=;
refresh_token=MGQzMDNjZWZlNjlhODgwN2Q5MWM3Y2I1OTQ4YmY0NWU=;
userid=61611712; Hm_lpvt_d0ad46e4afeacf34cd12de4c9b553aa6=1638946179'
#1.请补充

    def __init__(self):
        self.browser = webdriver.Chrome()
        self.browser.set_page_load_timeout(30)
```

```
    # 将cookie字符串转化为dict类型
    def get_cookie_list(self):
        cookie_str=self.cookie_str
        cookies = {}
        for line in cookie_str.split(';'):
            key, value = line.split('=', 1)
            cookies[key] = value
        return cookies
    # 获取歌手的总页数
    def get_singer_pages_by_selector(self,response):
        selector = Selector(text=response.text)
        pages =
selector.xpath('//*[@id="__layout"]/div/div[2]/div[4]/ul/li[last()-1]
/text()').extract() #2.请补充
        # //*[@id="__layout"]/div/div[2]/div[4]/ul/li[8]
        if len(pages)>0:
            print('pages:',pages[0])
            return int(pages[0])
        else:
            print('no found pages')
            return 0
    #     //*[@id="__layout"]/div/div[2]/div[4]/ul/li[8]
```

```
    # 使用scrapy的Selector解析歌手的信息
    def get_singer_url_by_selector(self, response):
        selector = Selector(text=response.text)
        scripts = selector.xpath("/html/body/script")
        if len(scripts) >= 1:
            scripts_data = scripts[0].extract()
            Codes=scripts_data.split('"')
            artistCodes=[]
            for code in Codes:
                if code.startswith('A') and len(code)==9:

artistCodes.append('https://music.taihe.com/artist/'+code)
            #       https://music.taihe.com/artist/A10047635
            return artistCodes
        else:
            print('error:scripts!')
            return None


    # 重写start_requests()方法，把所有URL地址都交给调度器,也可以不重写，这
里需要传入cookie,需要重写
    def start_requests(self):
```

```
        cookies=self.get_cookie_list()
        yield scrapy.Request(url=self.start_urls[0],
callback=self.parse, cookies=cookies, dont_filter=True)

    # 获取总页数，然后回调每一页
    def parse(self, response):
        print('parse.response.url:', response.url)
        # print('parse.response.text:',response.text)

        cookies = self.get_cookie_list()
        # ipages=self.get_singer_pages_by_selector(response)
        ipages=3
        for i in range(2,ipages):

page_url='https://music.taihe.com/artist?pageNo=%d&artistFristLetter=
&artistRegion=&artistGender'%i #4.请补充
            #
```

```
            yield scrapy.Request(url=page_url,meta={'itotalPage':
ipages,'icurrPage': i},
                                 callback=self.parse_onepage,
cookies=cookies, dont_filter=True)

        singerurls = self.get_singer_url_by_selector(response)
        itotalPage = len(singerurls)
        icurrSinger = 0
        for singleurl in singerurls:
            icurrSinger += 1
            yield scrapy.Request(url=singleurl, meta={'itotalPage':
itotalPage, 'icurrPage': icurrSinger},
                                 callback=self.parse_onesinger,
cookies=cookies, dont_filter=True)
        return

    def parse_onepage(self, response):
        print('parse_onepage.response.url:', response.url)
        # print('parse_onepage.response.text:', response.text)
```

2、pipline脚本

```
import datetime

from itemadapter import ItemAdapter

from bigdata.utils.MongoBase import MongoBase


class MusicsPipeline:
    def process_item(self, item, spider):
        item['collector'] = '郑丹阳'  # 6.请改为自己的名字
        item['coll_time'] = datetime.datetime.now()
        db = MongoBase('music_data')
        db.process_item(item)
        print('collector:', item['collector'],
db.get_db_count(item['collector']))
        return item
```

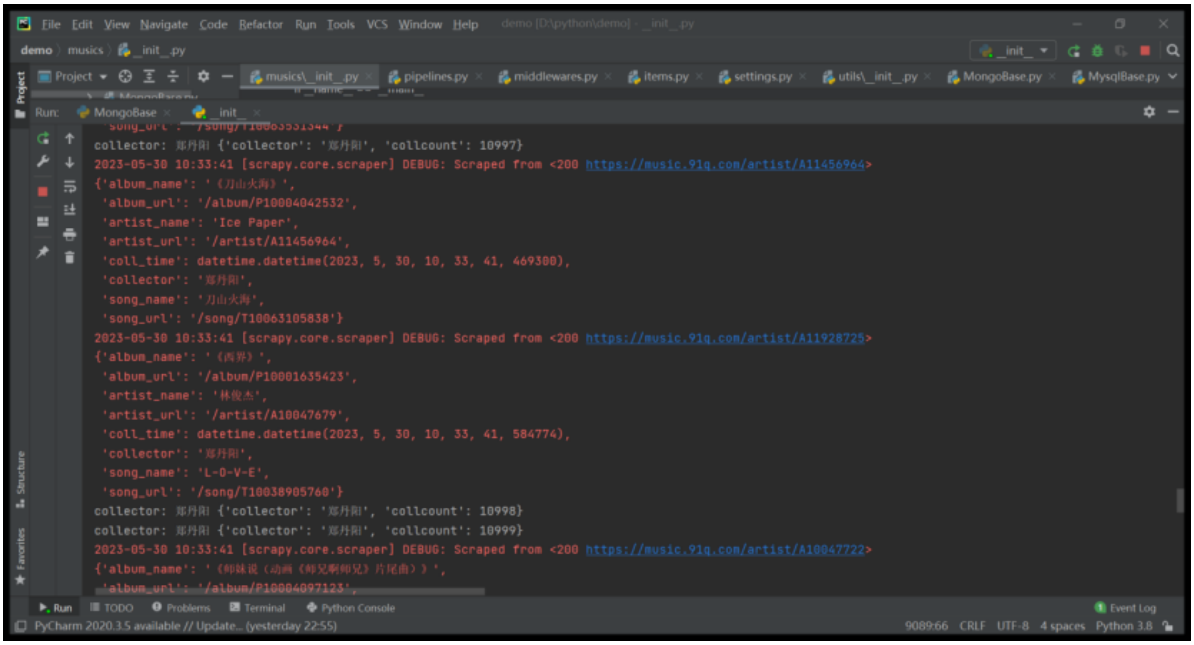
3、 midlleware脚本

```
import time

from scrapy import signals

# useful for handling different item types with a single interface
from itemadapter import is_item, ItemAdapter
from scrapy.http import HtmlResponse
from selenium.common.exceptions import TimeoutException


class MusicsSpiderMiddleware:
    # Not all methods need to be defined. If a method is not defined,
    # scrapy acts as if the spider middleware does not modify the
    # passed objects.

    @classmethod
    def from_crawler(cls, crawler):
        # This method is used by Scrapy to create your spiders.
        s = cls()
        crawler.signals.connect(s.spider_opened,
signal=signals.spider_opened)
```

4、item脚本

```python
import scrapy


class MusicsItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    id = scrapy.Field()
    song_url = scrapy.Field()
    song_name = scrapy.Field()
    artist_url = scrapy.Field()
    artist_name = scrapy.Field()
    album_url = scrapy.Field()
    album_name = scrapy.Field()
    collector = scrapy.Field()
    coll_time = scrapy.Field()
```

【运行截图】(3分)：



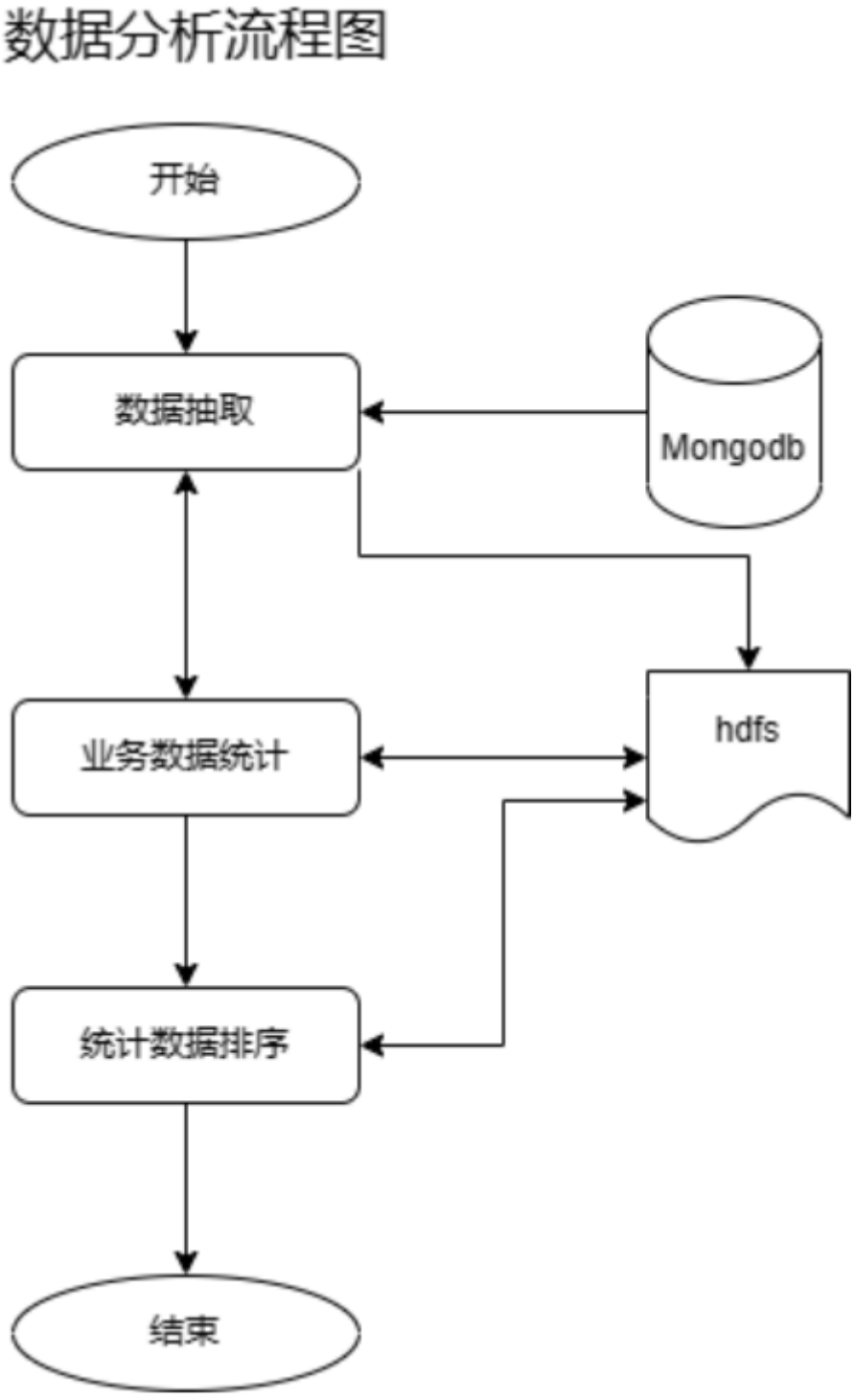
1.2.2. 数据分析(MR)(40分)

【功能说明】(4分)：

分析内容：分析Pychram采集到的音乐数据
分析过程：
1、从mongodb库中抽取hdfs文件系统
2、从hdfs取出文件，进行音乐词频分析，将分析后的结果存入hdfs
3、从hdfs取出数据分析的结果，进行按统计数据逆序排序
使用方法：
1、业务分析过程要包含基本的MapReduce类，如main方法，map类，reduce类
2、业务分析过程中要用到自定义数据类型、自定义计数器、Combiner优化方法
3、S1_SelectData_MusicSelectData,实现数据从mongodb到hdfs的采集
4、S2_CountData_MusicWordCount,实现业务数据统计
5、S3_SortData_MusicWordSort,实现统计数据的排序

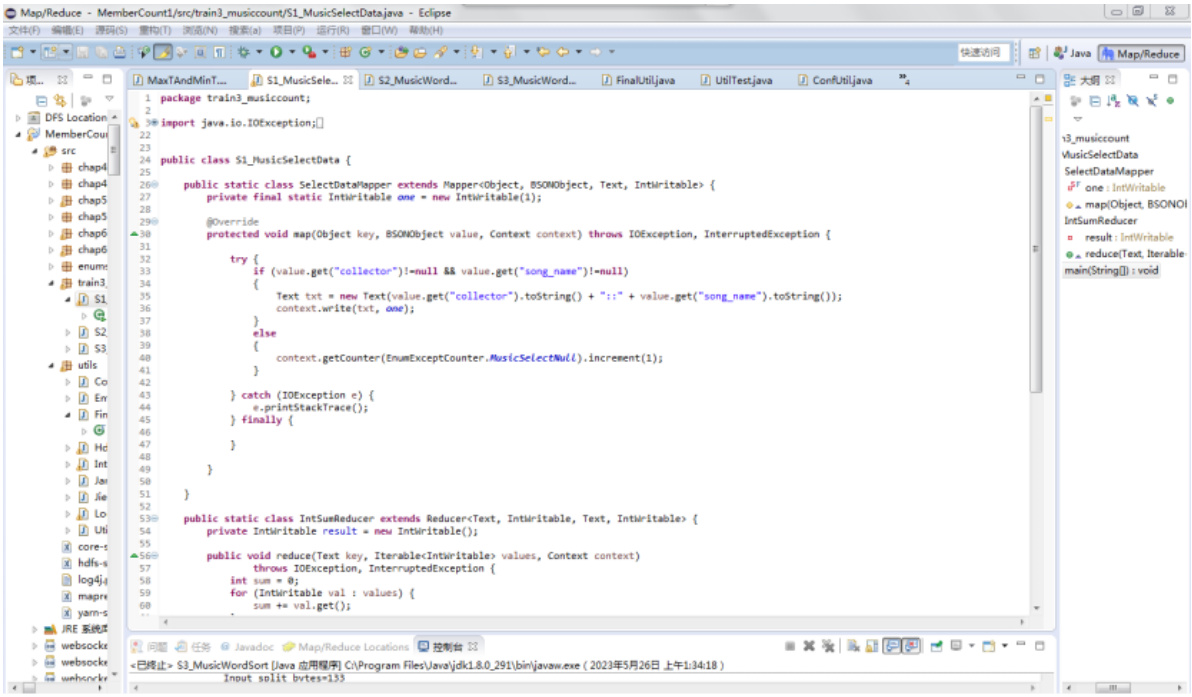
【功能设计】(8分)：

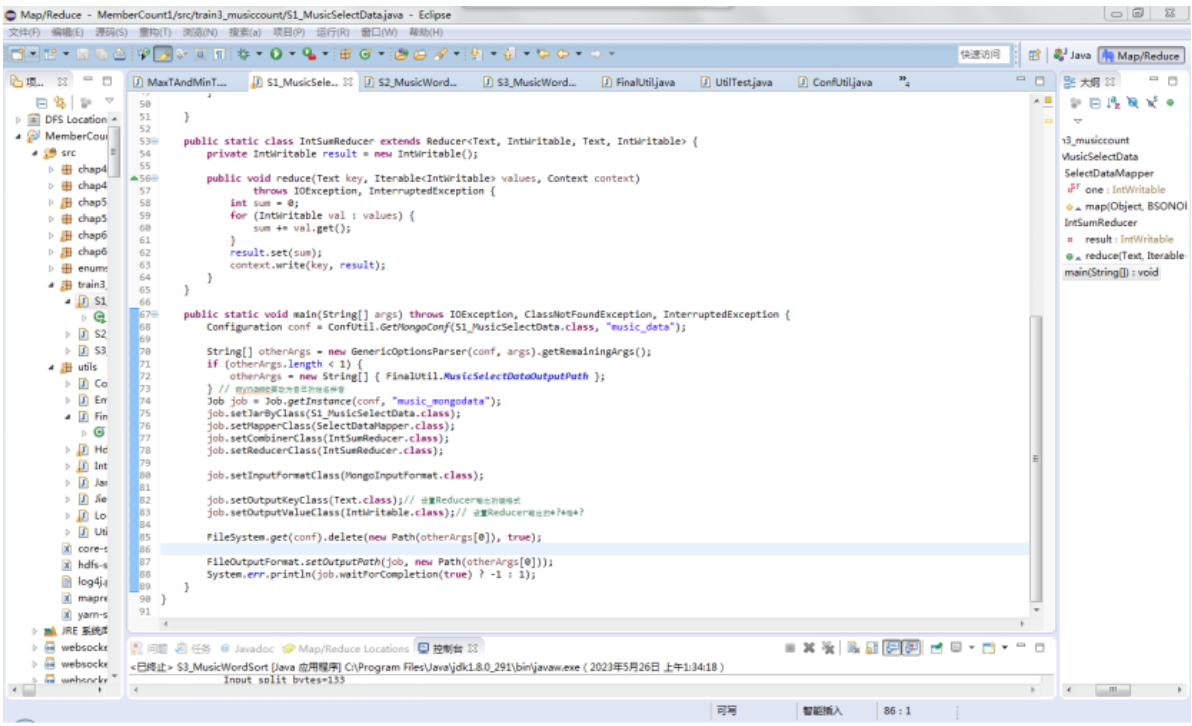
1、将mongodb中采集到的音乐数据采集到hdfs中
2、将hdfs中的音乐数据取出，实现音乐数据的统计分析，分析结果存入hdfs
3、从hdfs中取出音乐数据分析的结果，进行按统计数据逆序排序



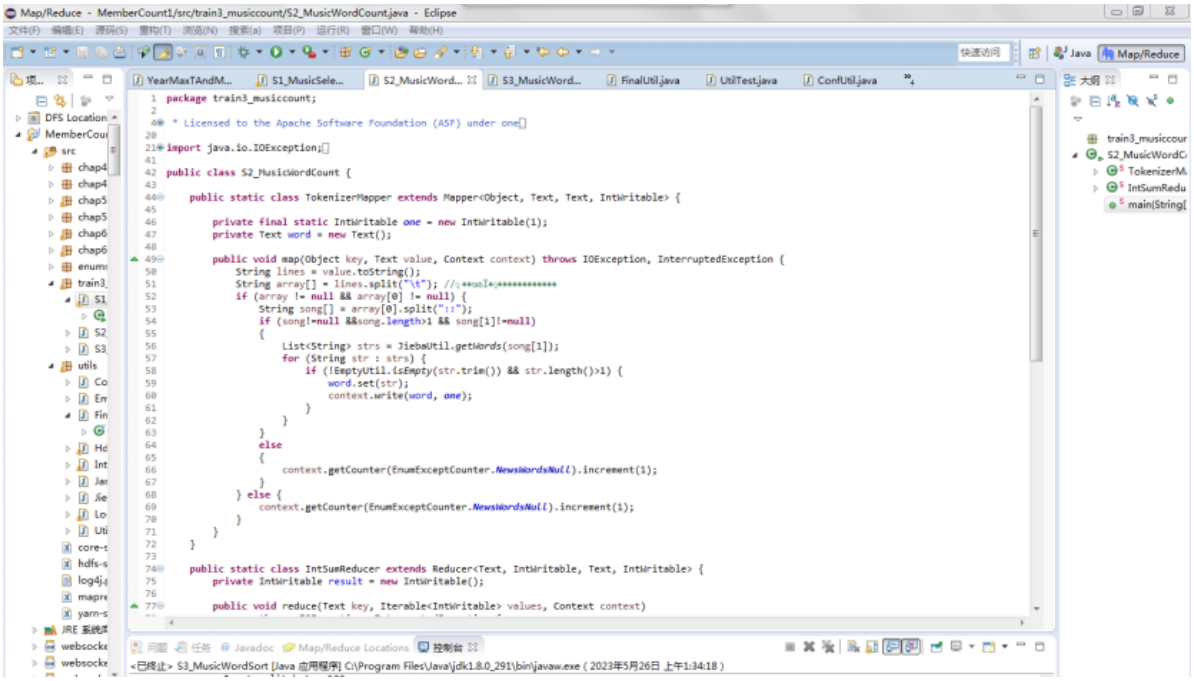
【源码实现】(16分)：

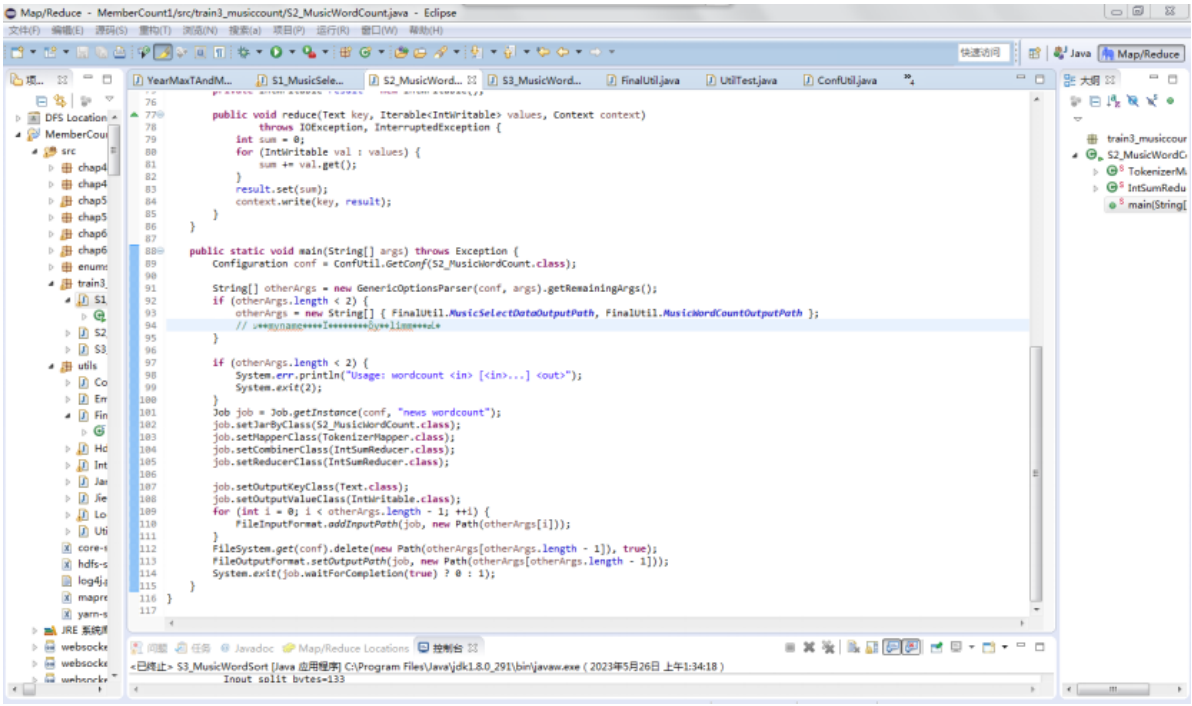
S1_ MusicSelectData代码：



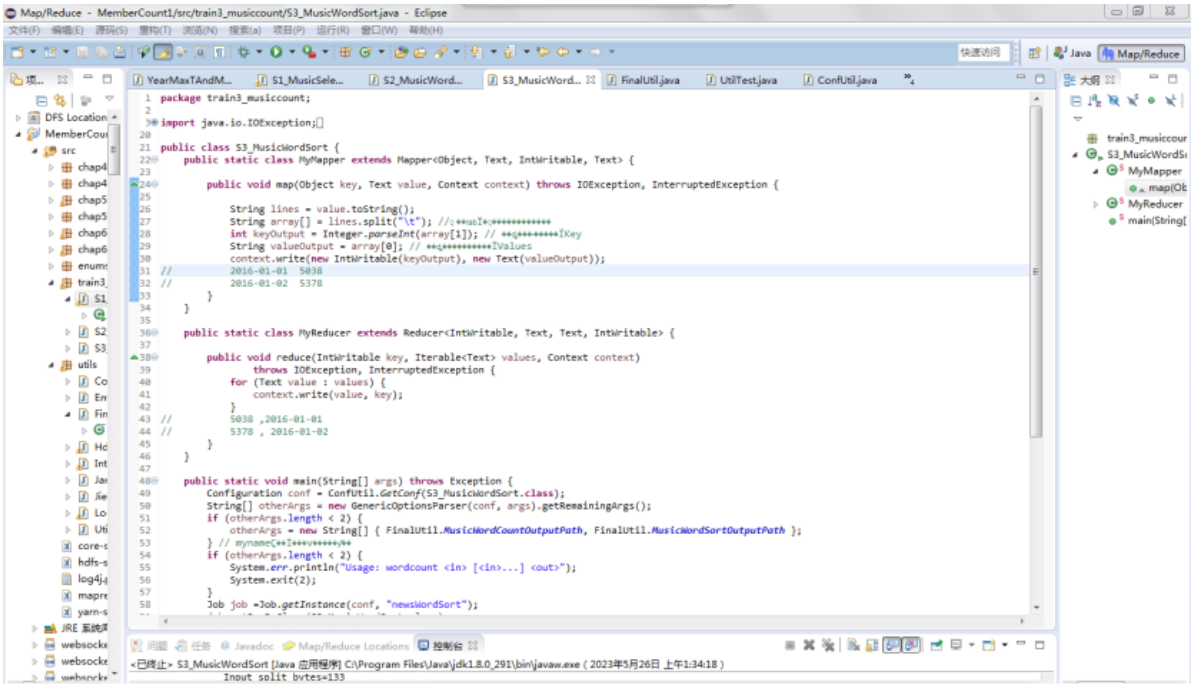


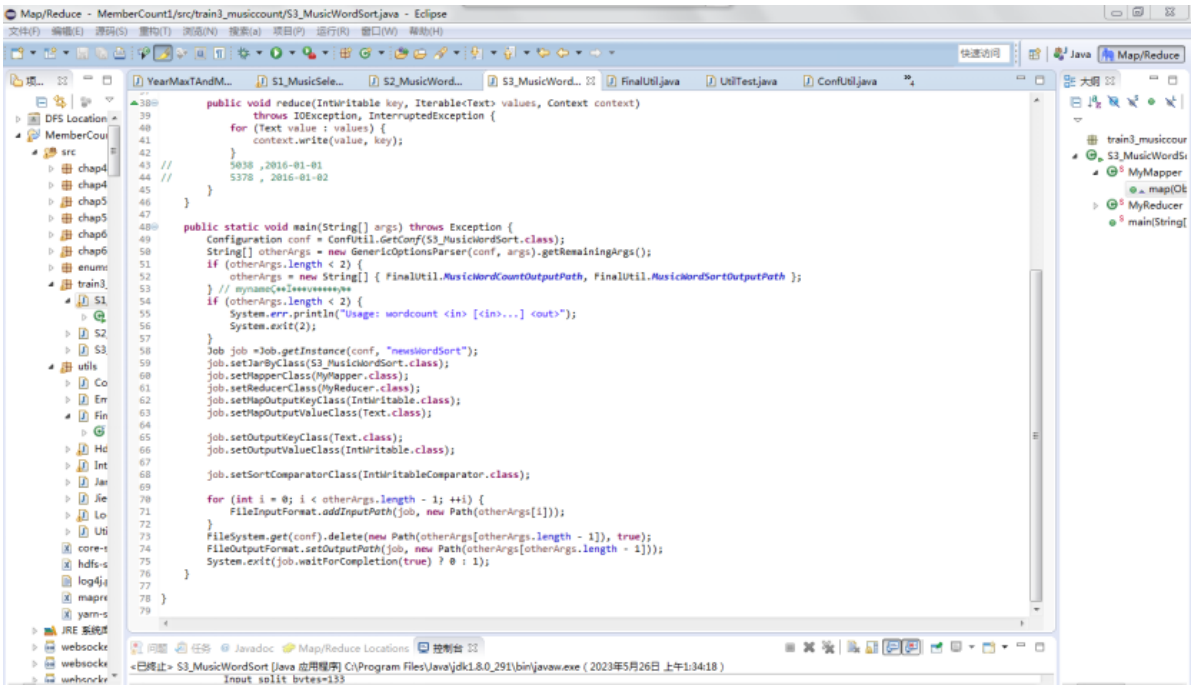
S2 _MusicWordCount 代码





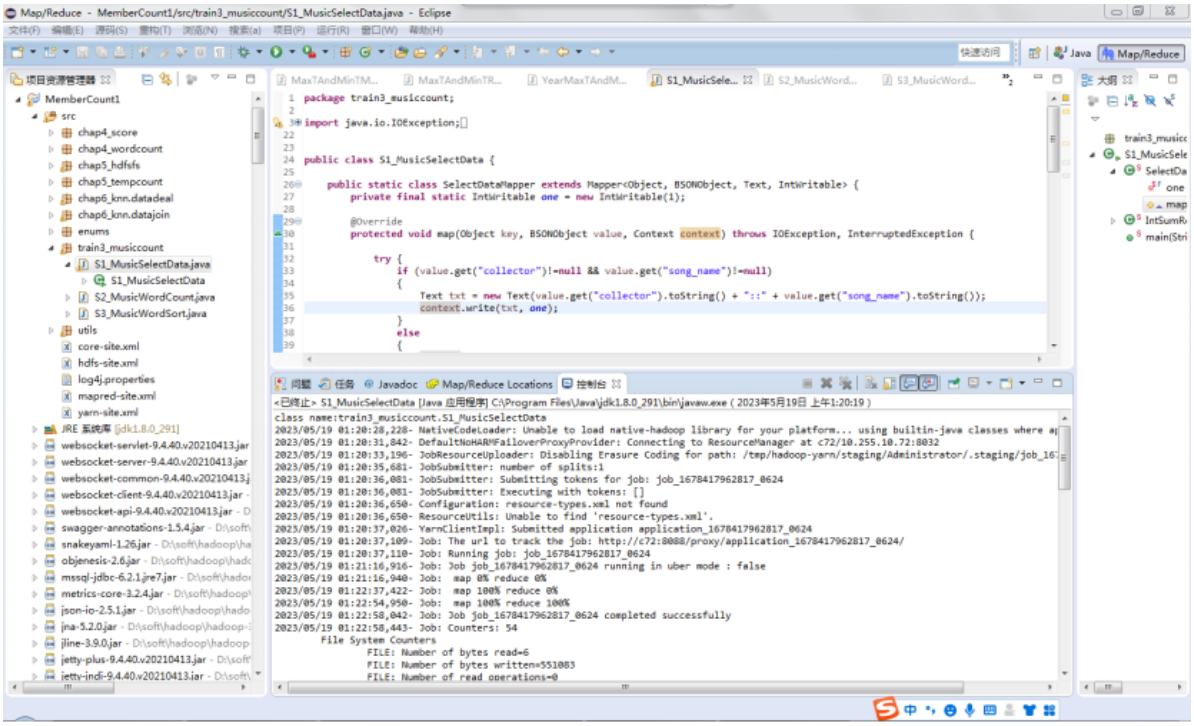
S3 _MusicWordSort

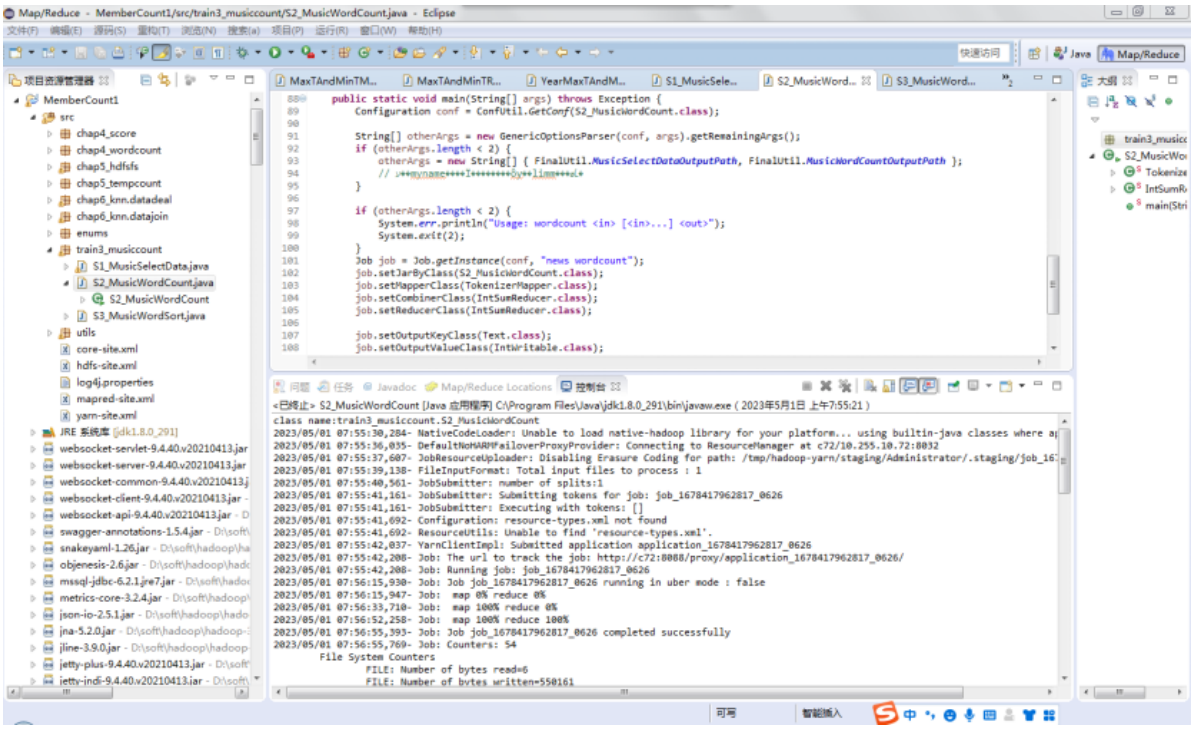


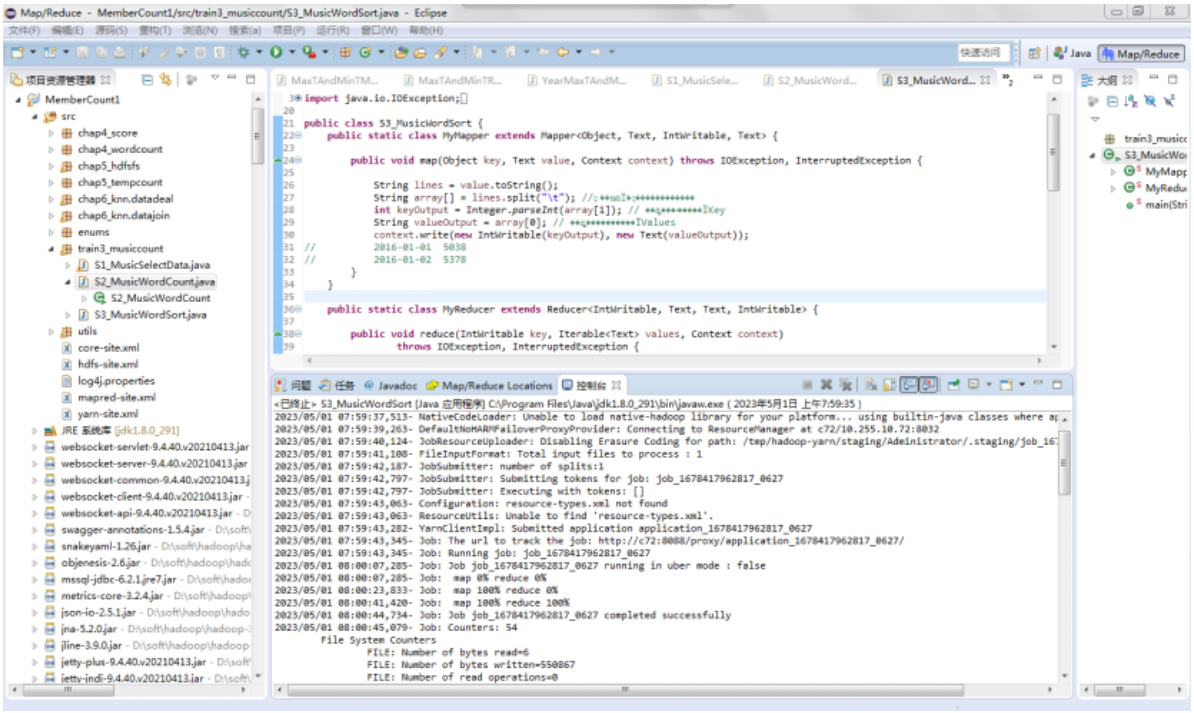


【运行截图】(12分)：

音乐数据采集代码运行成功截图

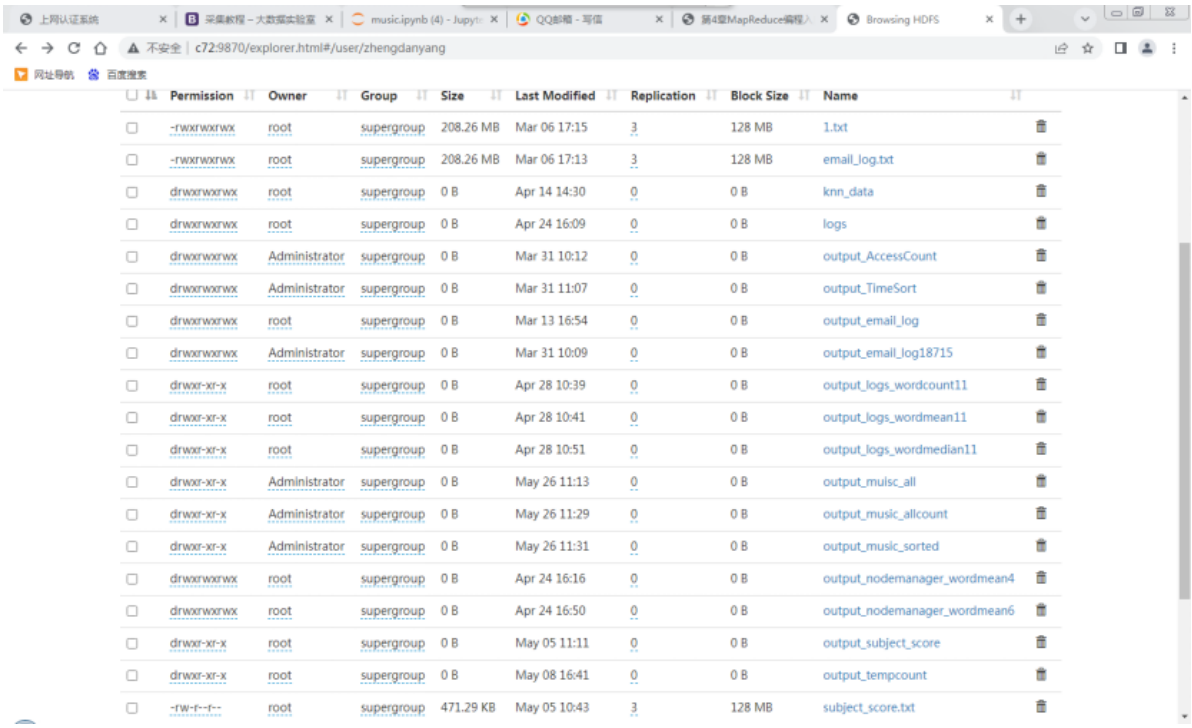

Map/Reduce - MemberCount1/src/train3_musiccount/S1_MusicSelectData.java - Eclipse
package train3_musiccount;
import java.io.IOException;
public class S1_MusicSelectData {
public static class SelectDataMapper extends Mapper<Object, BSONObject, Text, IntWritable> {
private final static IntWritable one = new IntWritable(1);
@Override
protected void map(Object key, BSONObject value, Context context) throws IOException, InterruptedException {
try {
if (value.get("collector")!=null && value.get("song_name")!=null)
{
Text txt = new Text(value.get("collector").toString() + "::" + value.get("song_name").toString());
context.write(txt, one);
}
else
{
class name:train3_musiccount.S1_MusicSelectData
2023/05/19 01:20:28,228- NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where ap
2023/05/19 01:20:31,842- DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at c72/10.255.10.72:8032
2023/05/19 01:20:33,196- JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Administrator/.staging/job_167
2023/05/19 01:20:35,681- JobSubmitter: number of splits:1
2023/05/19 01:20:36,081- JobSubmitter: Submitting tokens for job: job_1678417962817_0624
2023/05/19 01:20:36,081- JobSubmitter: Executing with tokens: []
2023/05/19 01:20:36,650- Configuration: resource-types.xml not found
2023/05/19 01:20:36,650- ResourceUtils: Unable to find 'resource-types.xml'.
2023/05/19 01:20:37,026- YarnClientImpl: Submitted application application_1678417962817_0624
2023/05/19 01:20:37,109- Job: The url to track the job: http://c72:8088/proxy/application_1678417962817_0624/
2023/05/19 01:20:37,110- Job: Running job: job_1678417962817_0624
2023/05/19 01:21:16,916- Job: Job job_1678417962817_0624 running in uber mode : false
2023/05/19 01:21:16,940- Job: map 0% reduce 0%
2023/05/19 01:22:37,422- Job: map 100% reduce 0%
2023/05/19 01:22:54,950- Job: map 100% reduce 100%
2023/05/19 01:22:58,042- Job: Job job_1678417962817_0624 completed successfully
2023/05/19 01:22:58,443- Job: Counters: 54
File System Counters
FILE: Number of bytes read=6
FILE: Number of bytes written=551083
FILE: Number of read operations=0


Map/Reduce - MemberCount1/src/train3_musiccount/S2_MusicWordCount.java - Eclipse
public static void main(String[] args) throws Exception {
Configuration conf = ConfUtil.GetConf(S2_MusicWordCount.class);
String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
if (otherArgs.length < 2) {
otherArgs = new String[] { FinalUtil.MusicSelectDataOutputPath, FinalUtil.MusicWordCountOutputPath };
}
if (otherArgs.length < 2) {
System.err.println("Usage: wordcount <in> [<in>...] <out>");
System.exit(2);
}
Job job = Job.getInstance(conf, "news wordcount");
job.setJarByClass(S2_MusicWordCount.class);
job.setMapperClass(TokenizerMapper.class);
job.setCombinerClass(IntSumReducer.class);
job.setReducerClass(IntSumReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
class name:train3_musiccount.S2_MusicWordCount
2023/05/01 07:55:30,284- NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where ap
2023/05/01 07:55:36,035- DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at c72/10.255.10.72:8032
2023/05/01 07:55:37,607- JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Administrator/.staging/job_167
2023/05/01 07:55:39,138- FileInputFormat: Total input files to process : 1
2023/05/01 07:55:40,561- JobSubmitter: number of splits:1
2023/05/01 07:55:41,161- JobSubmitter: Submitting tokens for job: job_1678417962817_0626
2023/05/01 07:55:41,161- JobSubmitter: Executing with tokens: []
2023/05/01 07:55:41,692- Configuration: resource-types.xml not found
2023/05/01 07:55:41,692- ResourceUtils: Unable to find 'resource-types.xml'.
2023/05/01 07:55:42,037- YarnClientImpl: Submitted application application_1678417962817_0626
2023/05/01 07:55:42,208- Job: The url to track the job: http://c72:8088/proxy/application_1678417962817_0626/
2023/05/01 07:55:42,208- Job: Running job: job_1678417962817_0626
2023/05/01 07:56:15,930- Job: Job job_1678417962817_0626 running in uber mode : false
2023/05/01 07:56:15,947- Job: map 0% reduce 0%
2023/05/01 07:56:33,710- Job: map 100% reduce 0%
2023/05/01 07:56:52,258- Job: map 100% reduce 100%
2023/05/01 07:56:55,393- Job: Job job_1678417962817_0626 completed successfully
2023/05/01 07:56:55,769- Job: Counters: 54
File System Counters
FILE: Number of bytes read=6
FILE: Number of bytes written=550161



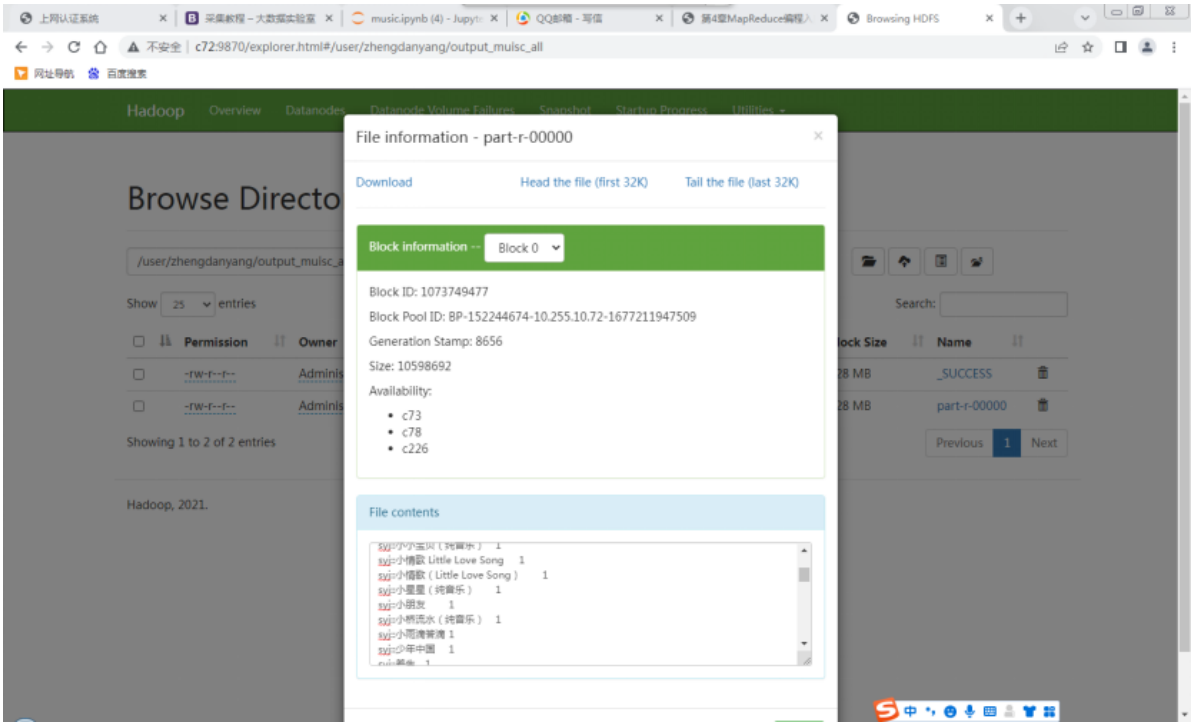
在http://c72:9870的文件系统中，运行输出结果截图

1、zhengdanyang目录下的文件

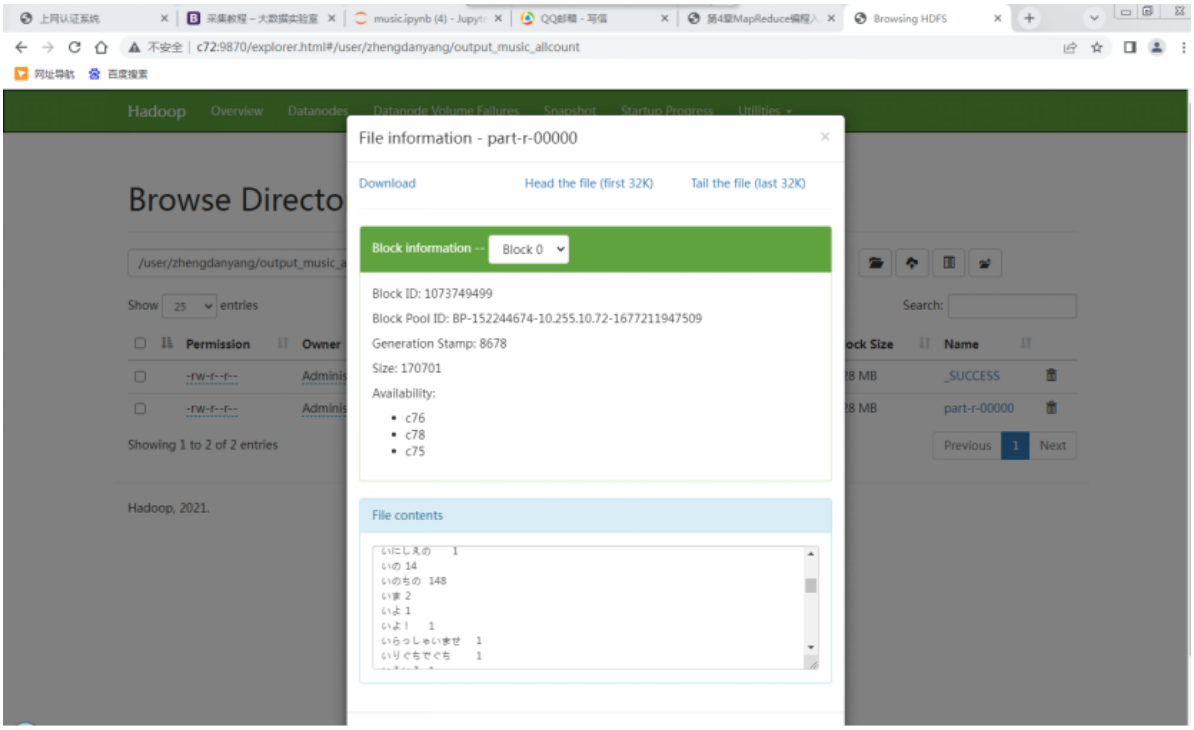


Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxrwxrwx	root	supergroup	208.26 MB	Mar 06 17:15	3	128 MB	1.txt
-rwxrwxrwx	root	supergroup	208.26 MB	Mar 06 17:13	3	128 MB	email_log.txt
drwxrwxrwx	root	supergroup	0 B	Apr 14 14:30	0	0 B	knn_data
drwxrwxrwx	root	supergroup	0 B	Apr 24 16:09	0	0 B	logs
drwxrwxrwx	Administrator	supergroup	0 B	Mar 31 10:12	0	0 B	output_AccessCount
drwxrwxrwx	Administrator	supergroup	0 B	Mar 31 11:07	0	0 B	output_TimeSort
drwxrwxrwx	root	supergroup	0 B	Mar 13 16:54	0	0 B	output_email_log
drwxrwxrwx	Administrator	supergroup	0 B	Mar 31 10:09	0	0 B	output_email_log18715
drwxr-xr-x	root	supergroup	0 B	Apr 28 10:39	0	0 B	output_logs_wordcount11
drwxr-xr-x	root	supergroup	0 B	Apr 28 10:41	0	0 B	output_logs_wordmean11
drwxr-xr-x	root	supergroup	0 B	Apr 28 10:51	0	0 B	output_logs_wordmedian11
drwxr-xr-x	Administrator	supergroup	0 B	May 26 11:13	0	0 B	output_muisc_all
drwxr-xr-x	Administrator	supergroup	0 B	May 26 11:29	0	0 B	output_music_allcount
drwxr-xr-x	Administrator	supergroup	0 B	May 26 11:31	0	0 B	output_music_sorted
drwxrwxrwx	root	supergroup	0 B	Apr 24 16:16	0	0 B	output_nodemanager_wordmean4
drwxrwxrwx	root	supergroup	0 B	Apr 24 16:50	0	0 B	output_nodemanager_wordmean6
drwxr-xr-x	root	supergroup	0 B	May 05 11:11	0	0 B	output_subject_score
drwxr-xr-x	root	supergroup	0 B	May 08 16:41	0	0 B	output_tempcount
-rw-r--r--	root	supergroup	471.29 KB	May 05 10:43	3	128 MB	subject_score.txt

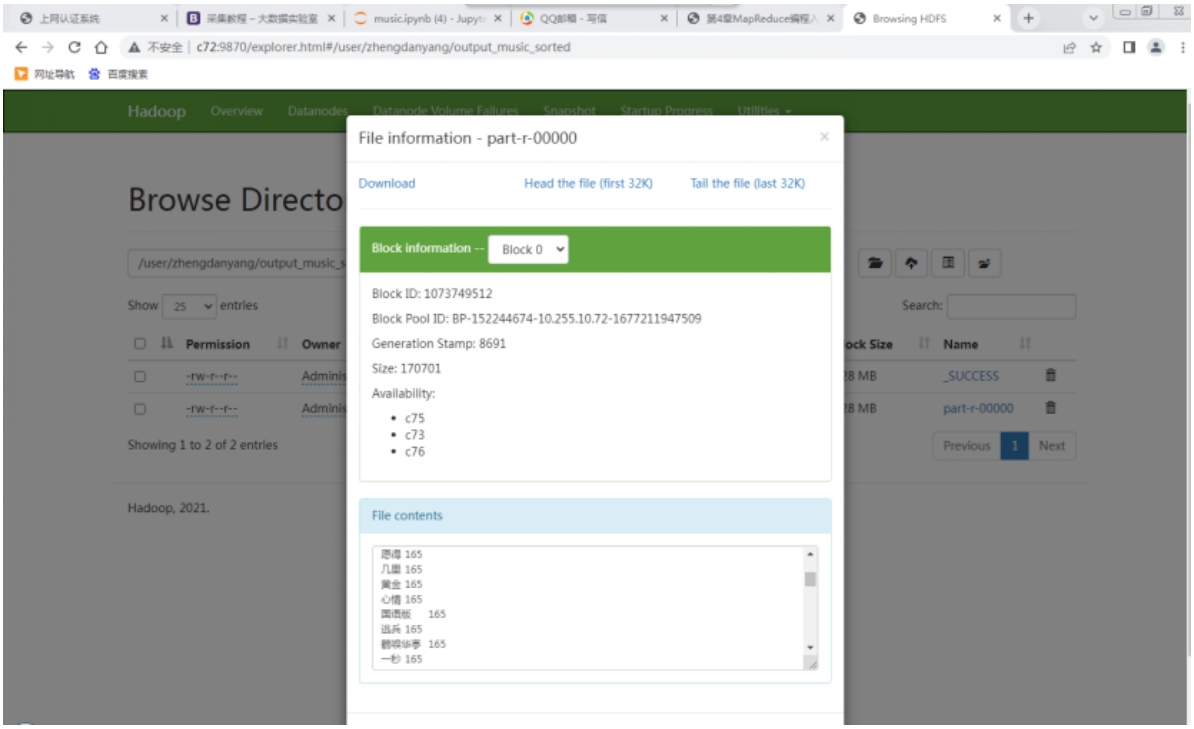
2、output_music_all



3、output_music_allcount



3、output_music_sorted



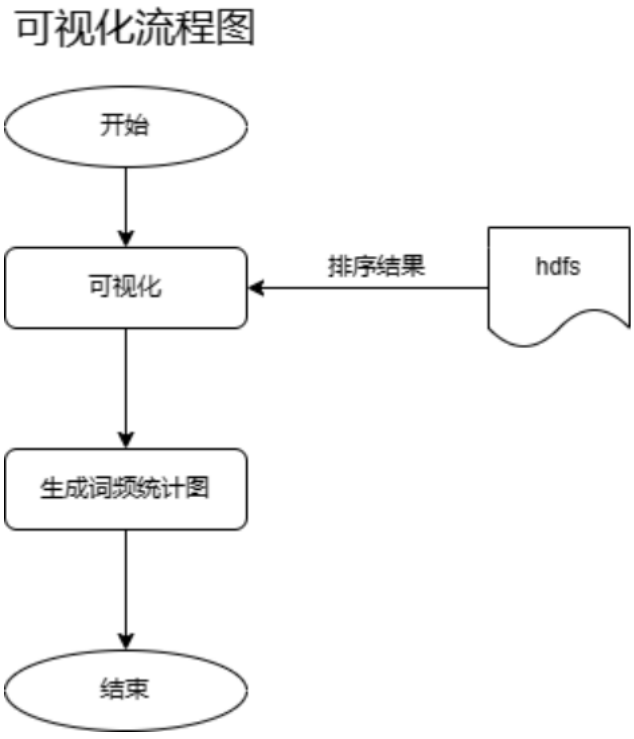
1.2.3. 数据可视(10分)

【功能说明】(1分）：

利用Pycharm将hdfs的部分数据排序结果进行可视化，得到音乐词频统计条形图

【功能设计】(2分）：

将hdfs中的部分音乐词频统计数据在Pycharm中进行可视化，生成音乐词频统计图



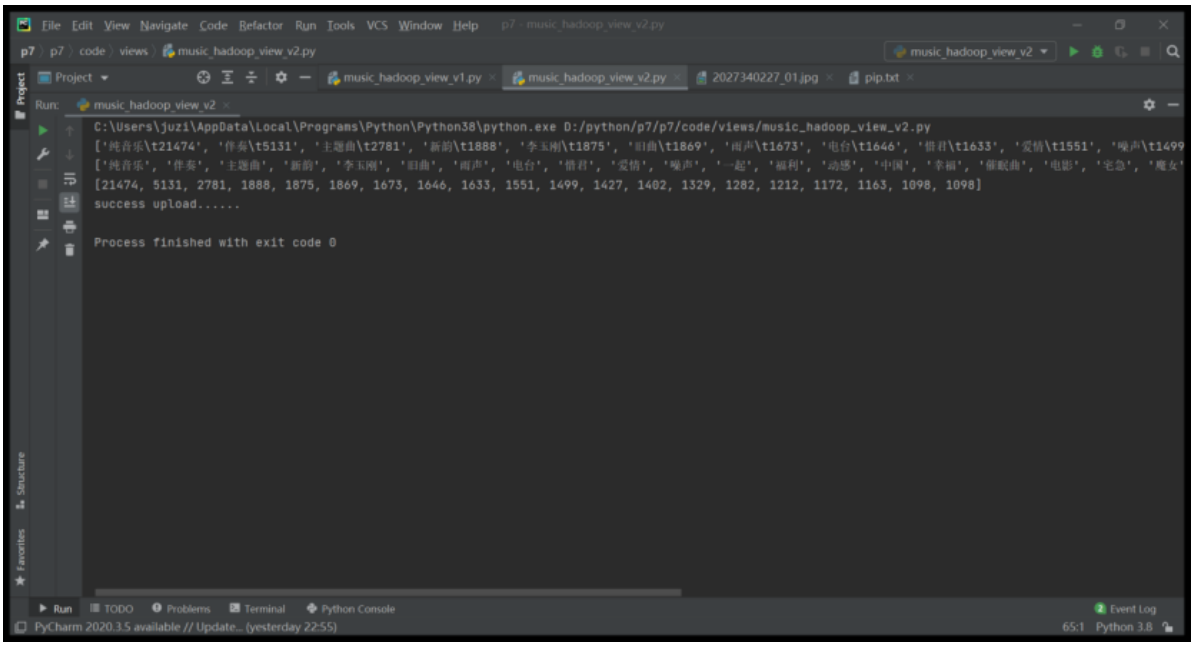
【源码实现】(4分）：

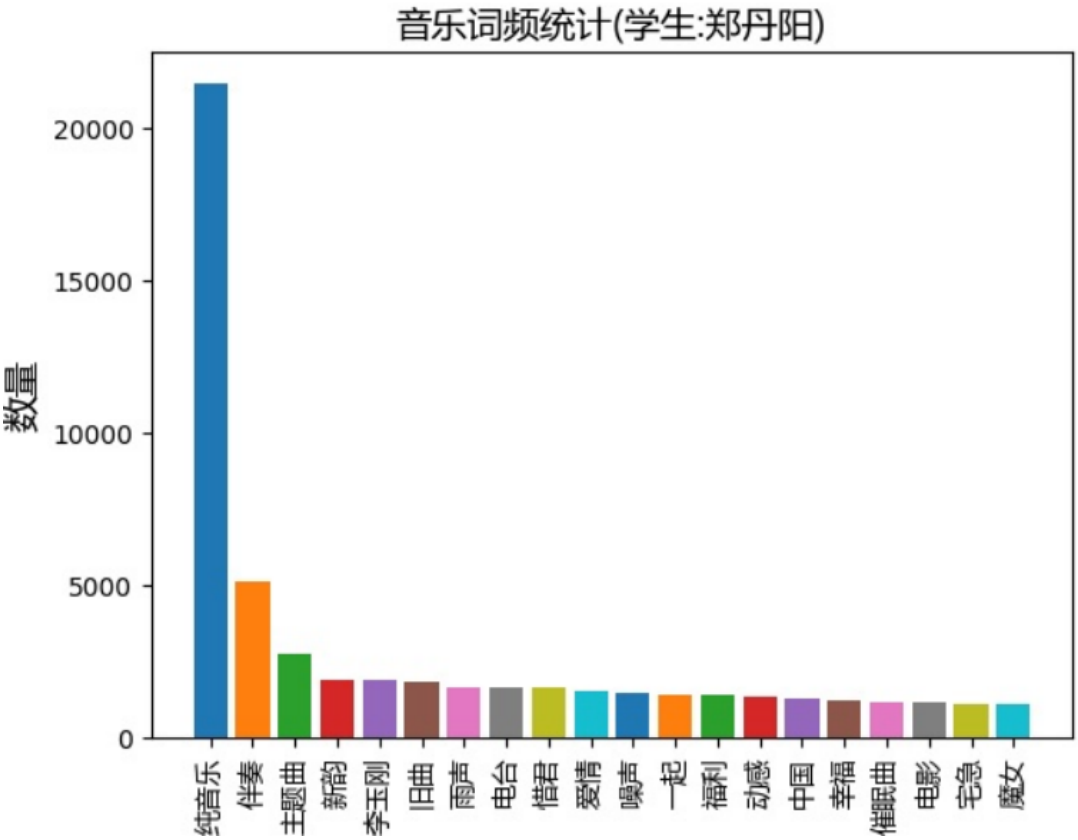
```
from bgutils.ftpUtil import ftpUtil

# git url:
https://jihulab.com/biglab-share/scrapy/-/tree/main/b50506/p7/code/vi
ews
class drawpic:
    def draw(self,myname,mycode):

        ftputil=ftpUtil()
        indata = '''纯音乐    21474
伴奏 5131
主题曲    2781
新韵 1888
李玉刚    1875
旧曲 1869
雨声 1673
电台 1646
惜君 1633
爱情 1551
噪声 1499
一起 1427
福利 1402
动感 1329
中国 1282
幸福 1212
催眠曲    1172
电影 1163
宅急 1098
魔女 1098'''

        s_data = indata.split('\n')
        print(s_data)

        import matplotlib.pyplot as plt
        from matplotlib.font_manager import FontProperties

        font = FontProperties(fname=r"msyh.ttc", size=14)

        #
plt.xticks(range(len(s_data)),s_data,fontproperties=font,rotation=90,
size=10)
```

```
        x_datas = []
        y_datas = []
        for i in range(len(s_data)):
            o_data = s_data[i]
            sp_data = o_data.split('\t')
            x_data = sp_data[0]
            x_datas.append(x_data)
            y_data = int(sp_data[1])
            y_datas.append(y_data)
            # print(x_data)
            # print(y_data)
            plt.bar(x_data, y_data)

        print(x_datas)
        print(y_datas)
        plt.xticks(range(len(x_datas)), x_datas, fontproperties=font,
rotation=90, size=10)
        plt.title("音乐词频统计(学生:"+myname+")", fontproperties=font)
        plt.xlabel("关键词", fontproperties=font)
        plt.ylabel("数量", fontproperties=font)

        # plt.show()
        idxproject= "23301"  #23301 是项目编号，长度 5 位
        idxpic="" #04 是图片在该项目中的顺序号，长度 2 位
        filepath=mycode+"_"+idxpic +".jpg" #生成的文件以学号+序号+后缀组成
        plt.savefig(filepath)
```

```
        ftputil.putfile_stud(filepath, idxproject,mycode,idxpic);



if __name__ == "__main__":
    draw=drawpic();
    myname="郑丹阳" #请将引号中的张三替换为本人姓名,如：张三
    mycode="2027340227" #请将引号中的 2019001001 替换为本人学号,
如:2019001001
    draw.draw(myname,mycode);
```

【运行截图】(3分)：





1.3. 项目小结（5分）

我通过此次Hadoop综合实训音乐项目，重新巩固了大二及大三期间学习的网络信息抓取技术和数据可视化的相关内容，在爬取上万条音乐数据并对音乐词频进行分析的过程中，我对这部分内容的理解比过去更加深刻，掌握程度也有所提升，希望可以将学习到的内容灵活运用到之后的暑期实习当中去。
作为本学期实训课程的小组组长，在进行Hadoop集群搭建的过程中，我也学习到了很多东西，不仅学会了搭建集群的基本流程，还在解决报错情况中掌握了检索错误和解决错误的方法。通过多次完成数据分析的实训项目，掌握了数据分析的基本方法，并在此次音乐项目中进一步巩固。

1.4. 附件

整个过程中实现的源码，包括采集的源码，mr分析源码，可视化的源码，全部打包成rar文件另行提交。

实现参考

数据采集

源码参考：

https://jihulab.com/biglab-share/scrapy/-/tree/main/b57562/chap9/music_scrapy?ref_type=heads

数据存储(mongodb->hdfs)

源码参考：

https://jihulab.com/biglab-share/hadoop/-/blob/main/b57562/wordcount/src/train3_musiccount/S1_MusicSelectData.java?ref_type=heads

数据分析1(MR->Count)

源码参考

https://jihulab.com/biglab-share/hadoop/-/blob/main/b57562/wordcount/src/train3_musiccount/S2_MusicWordCount.java?ref_type=heads

数据分析2(MR->Sort)

源码参考

https://jihulab.com/biglab-share/hadoop/-/blob/main/b57562/wordcount/src/train3_musiccount/S3_MusicWordSort.java?ref_type=heads

数据可视

源码参考(pycharm)：

https://jihulab.com/biglab-share/scrapy/-/tree/main/b57562/chap9/music_view?ref_type=heads

如何下载源码

以上源码包含在两个git项目：

```
https://jihulab.com/biglab-share/scrapy.git
https://jihulab.com/biglab-share/hadoop.git
```

由于 jihulab.com进入收费，我们即将切换到自建git服务器上，切换方法如：

使用资源管理器进入scrapy目录，在地址栏中输入cmd进入命令行，运行：

```
git remote set-url origin http://home.hddly.cn:8093/biglab-share/scrapy.git
```

同样，使用资源管理器进入hadoop目录，在地址栏中输入cmd进入命令行，运行：

```
git remote set-url origin http://home.hddly.cn:8093/biglab-share/hadoop.git
```

GIT安装和下载源码视频

学习视频