

第9章综合项目实训

第9章综合项目实训

CH09实训1: Hadoop综合实训项目

训练要点

需求说明

实现步骤

作业要求

环境准备

采集环境准备

chrome和chromedriver下载

chromedriver.exe部署

禁止chrome自动更新

采集IDEA环境

安装依赖包

MR分析环境准备

需要开启hadoop集群

需要同步hadoop主机和从机的时间, 以及windows时间

需要设置目录的权限

需要将mongodb相在的依赖包上传到集群

MR源码调试注意事项

添加libs目录

添加resources资源目录

ConfUtil文件修改

finalutil文件修改

关于源码根路径

报告模板

报告编写

1.1. 概述 (5分)

1.1.1. 训练要点(1分)

1.1.2. 需求说明(2分)

1.1.3. 实现步骤(2分)

1.1. 总体设计(30分)

1.1.1. 业务流程图(7分)

1.1.2. 数据流程图(8分)

1.1.3. 系统功能结构(5分)

1.1.4. 运行环境(10分)

1.2. 详细设计(60分)

1.2.1. 数据采集(10分)

1.2.2. 数据分析(MR)(40分)

1.2.3. 数据可视(10分)

1.3. 项目小结 (5分)

1.4. 附件

实现参考

数据采集

数据存储(mongodb->hdfs)

数据分析1(MR->Count)

数据分析2(MR->Sort)

数据可视

如何下载源码

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. 本实训允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：

- 1 百度网盘下载
- 2 链接：https://pan.baidu.com/s/1FF7HHQi9y2kVBXom_Omd_g
- 3 提取码：8ft4
- 4 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

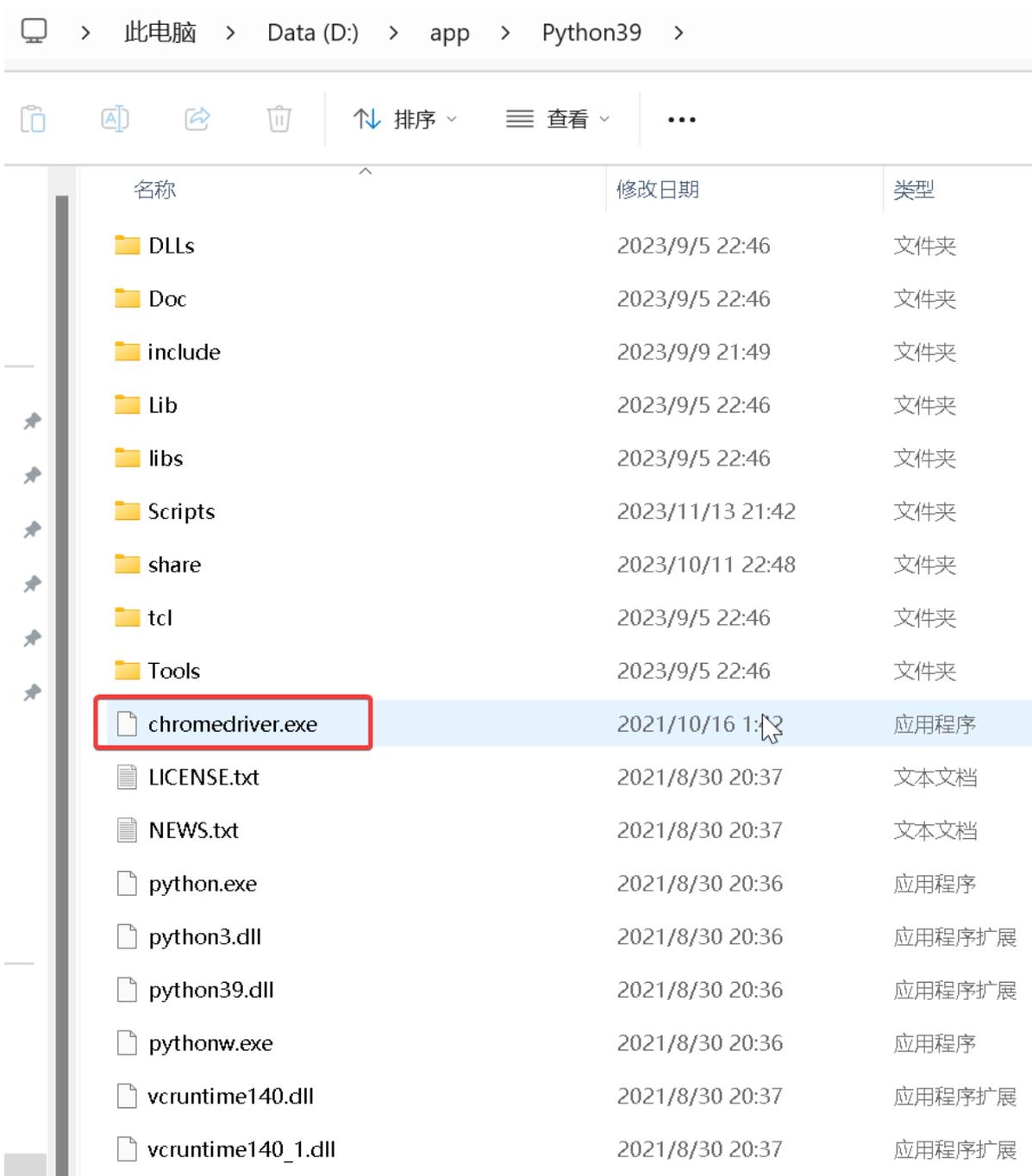
- 1 链接下载google 109版本
- 2 https://dl.google.com/release2/chrome/ad2uwza6rxngw4rvu7lrmj5rvtca_109.0.5414.75/109.0.5414.75_chrome_installer.exe
- 3 链接下载chromedriver版本109
- 4 https://registry.npmmirror.com/-/binary/chromedriver/109.0.5414.74/chromedriver_win32.zip

方式二，使用当前最新的chrome版本，如119,120,去下载最新的驱动

- 1 chromedriver最新版本下载：
- 2 <https://googlechromelabs.github.io/chrome-for-testing/>

chromedriver.exe部署

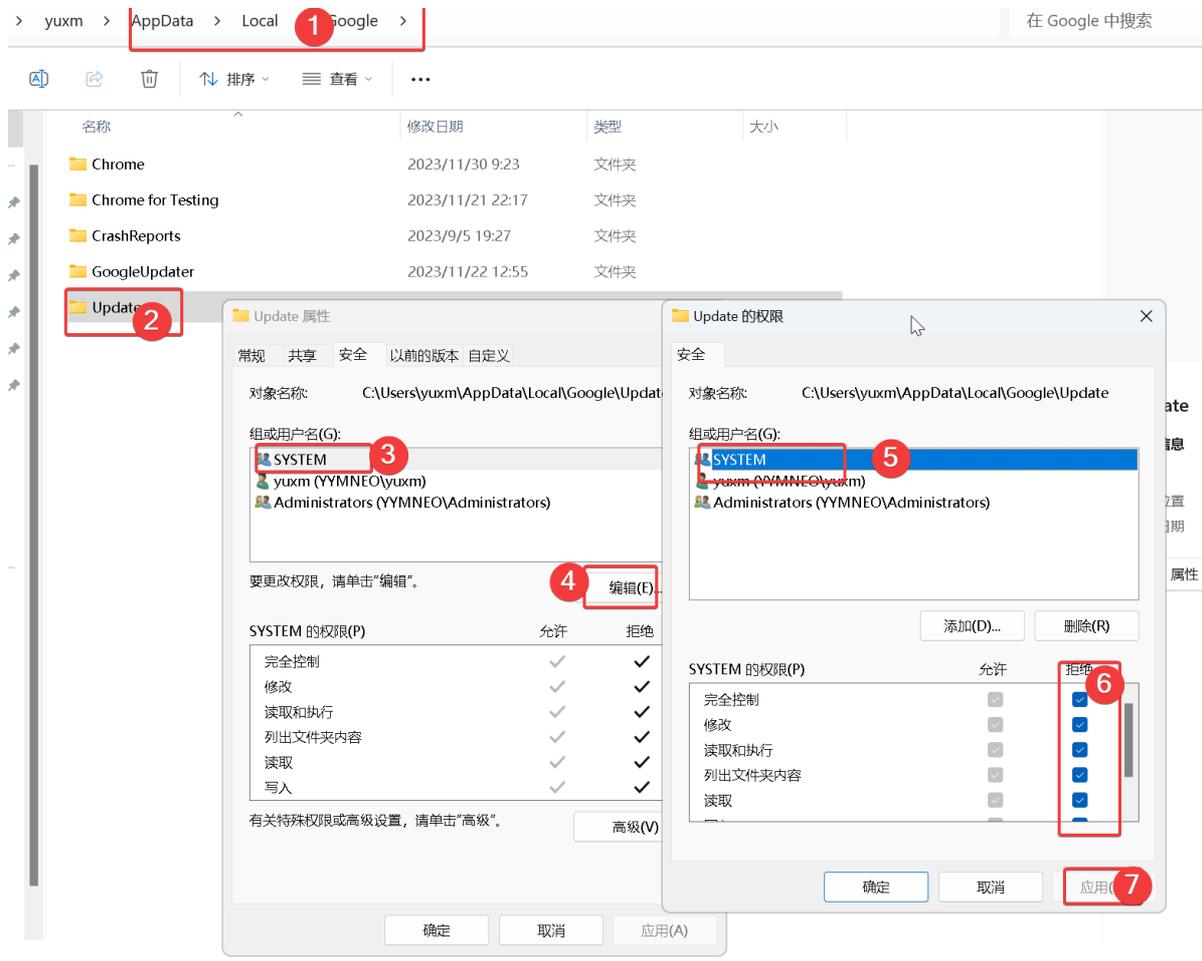
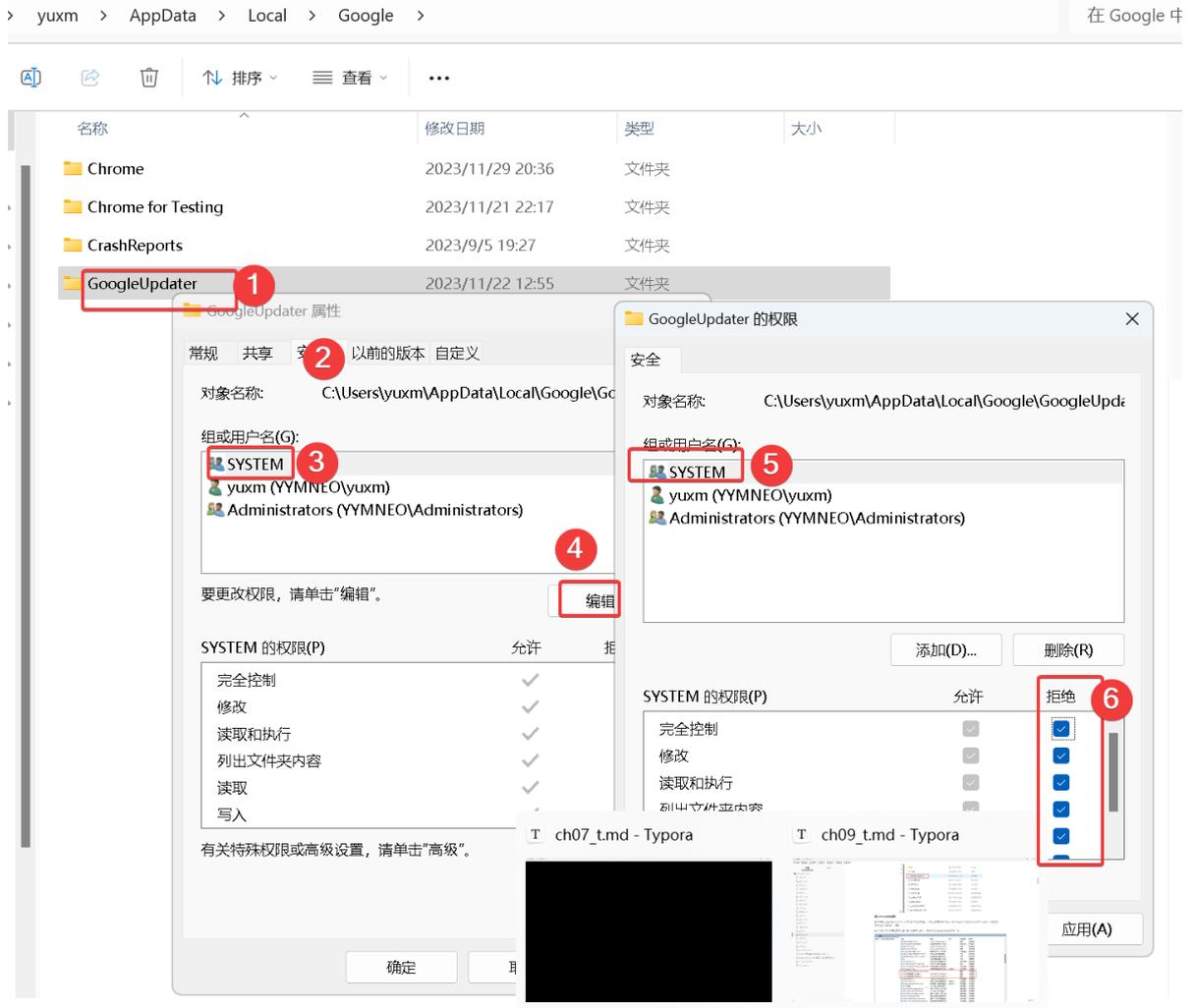
将下载下来的chromedriver_win32_95.zip解压出来的chromedriver.exe复制到python的路径下，如：



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

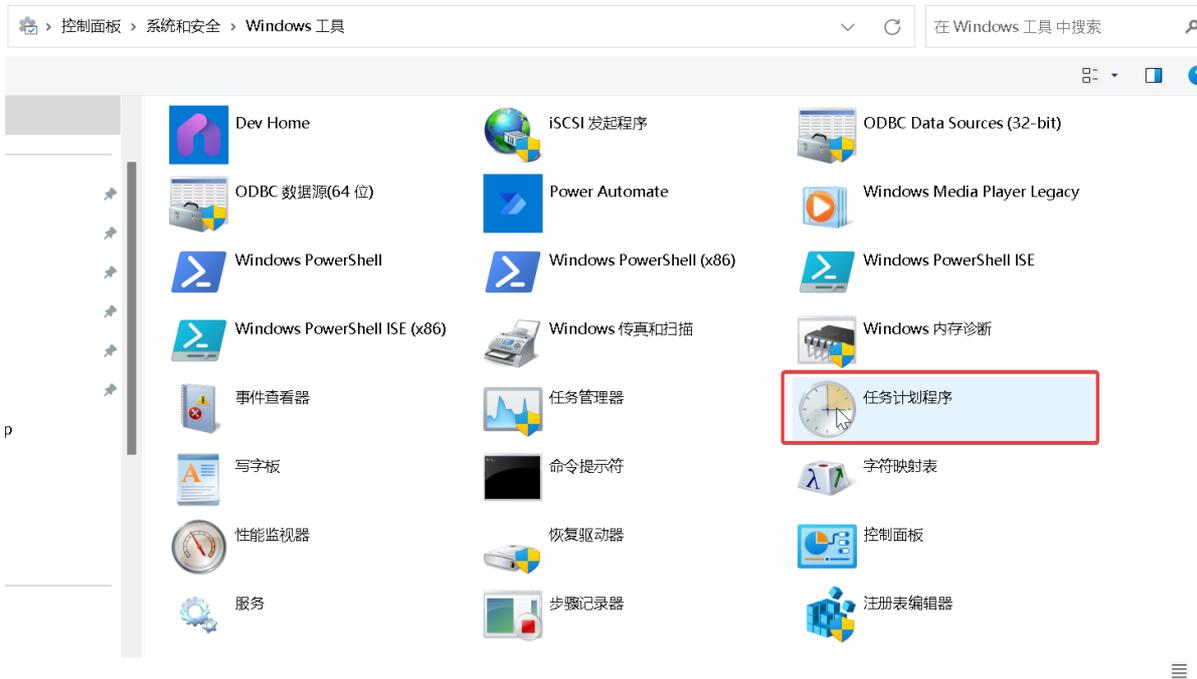
(如:C:\Users\yuxm\AppData\Local\Google) ->分别右击目录:GoogleUpdater 和目录:Update ->属性->安全->编辑->拒绝->应用,如下图:



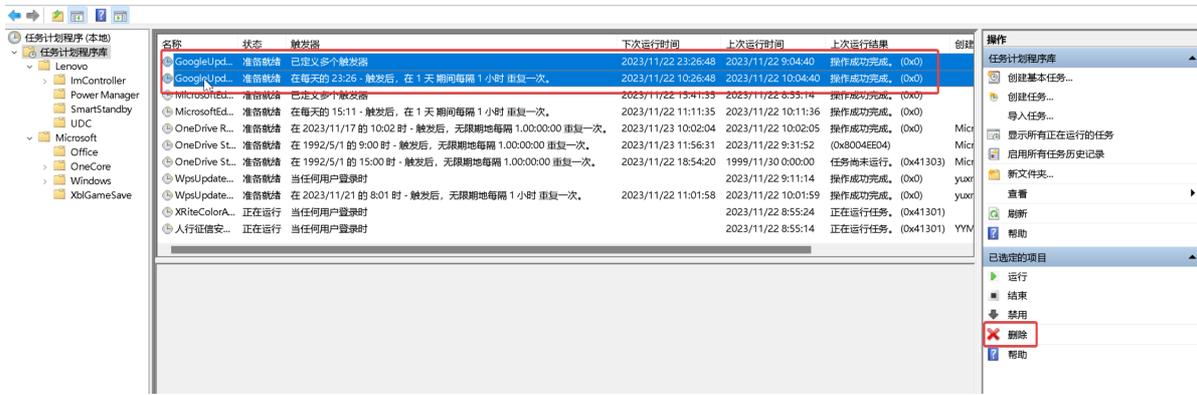
右击开始->打开计算机管理->服务和应用程序->服务，禁用所有和google有关的程序，如：



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，建议先通过控制面板，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：

```

1 python -m pip install --upgrade pip -i http://mirrors.aliyun.com/pypi/simple
  ^
2 --trusted-host mirrors.aliyun.com
3 pip install pymongo -i http://mirrors.aliyun.com/pypi/simple ^
4 --trusted-host mirrors.aliyun.com
5 pip install pymysql -i http://mirrors.aliyun.com/pypi/simple ^
6 --trusted-host mirrors.aliyun.com
7 pip install scrapy -i http://mirrors.aliyun.com/pypi/simple ^
8 --trusted-host mirrors.aliyun.com
9 pip install pandas -i http://mirrors.aliyun.com/pypi/simple ^
10 --trusted-host mirrors.aliyun.com
11 pip install sqlalchemy -i http://mirrors.aliyun.com/pypi/simple ^
12 --trusted-host mirrors.aliyun.com
13 pip install bs4 -i http://mirrors.aliyun.com/pypi/simple ^
14 --trusted-host mirrors.aliyun.com
15 pip install selenium -i http://mirrors.aliyun.com/pypi/simple ^
16 --trusted-host mirrors.aliyun.com
17 pip install redis -i http://mirrors.aliyun.com/pypi/simple ^
18 --trusted-host mirrors.aliyun.com
19 pip install bgutils-hddly -i http://mirrors.aliyun.com/pypi/simple ^
20 --trusted-host mirrors.aliyun.com
21 pip install --upgrade bgutils-hddly -i http://mirrors.aliyun.com/pypi/simple
  ^
22 --trusted-host mirrors.aliyun.com
23

```

MR分析环境准备

需要开启hadoop集群

在master主机上运行:

```
1 | 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

```
1 | systemctl stop ntpd  
2 | ntpdate cn.pool.ntp.org  
3 | 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

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

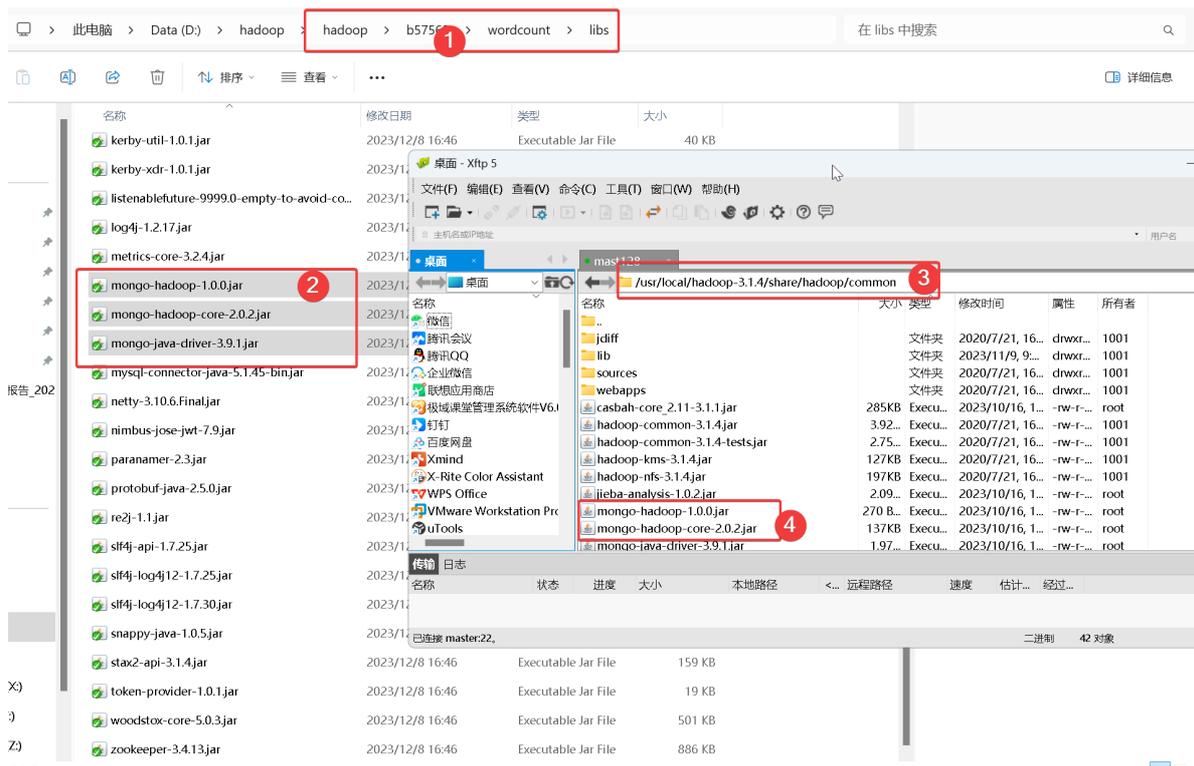
需要将mongodb相的依赖包上传到集群

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

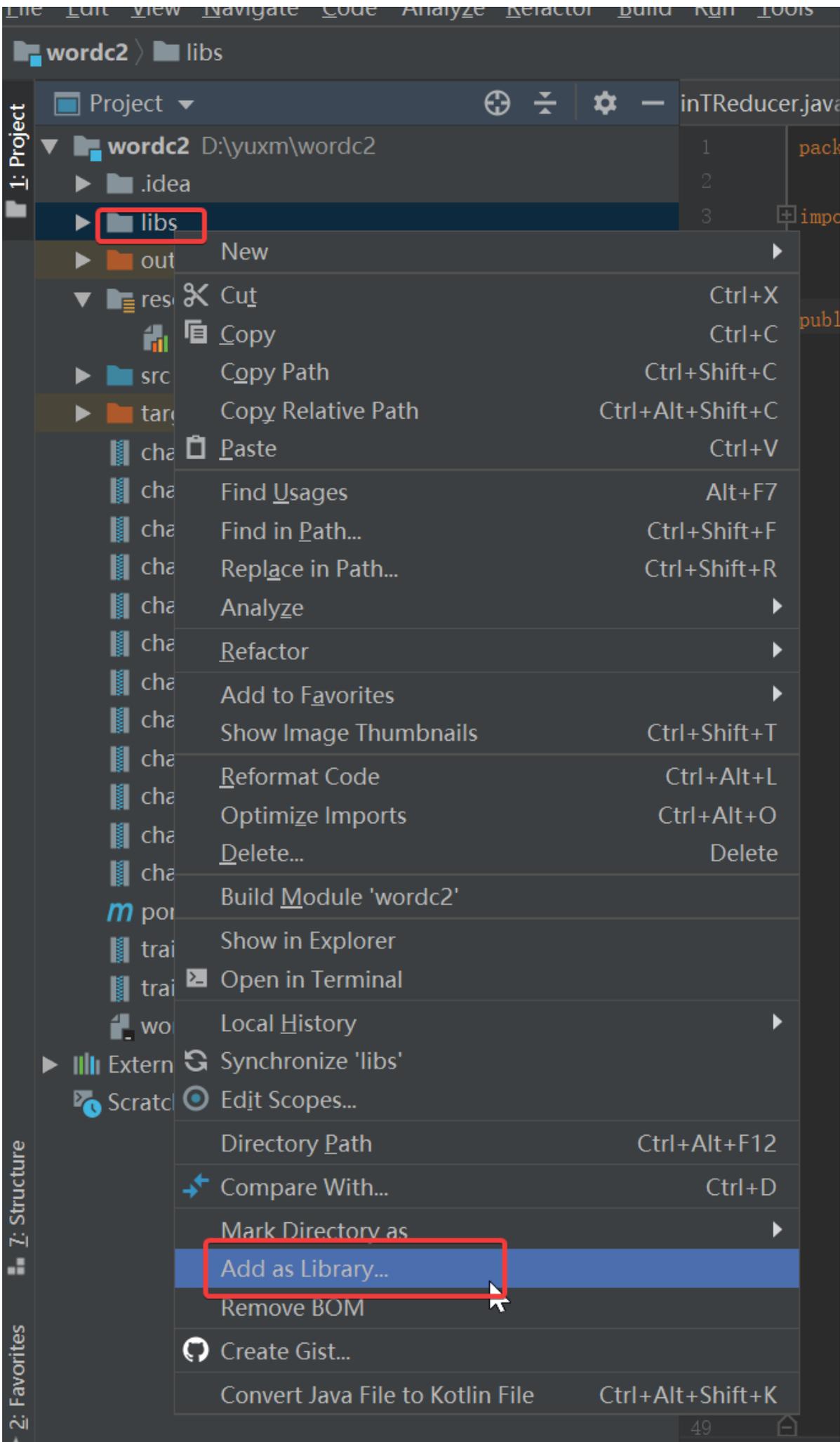
使用xftp工具分别连接到master,slave1,slave2，然后上传：



MR源码调试注意事项

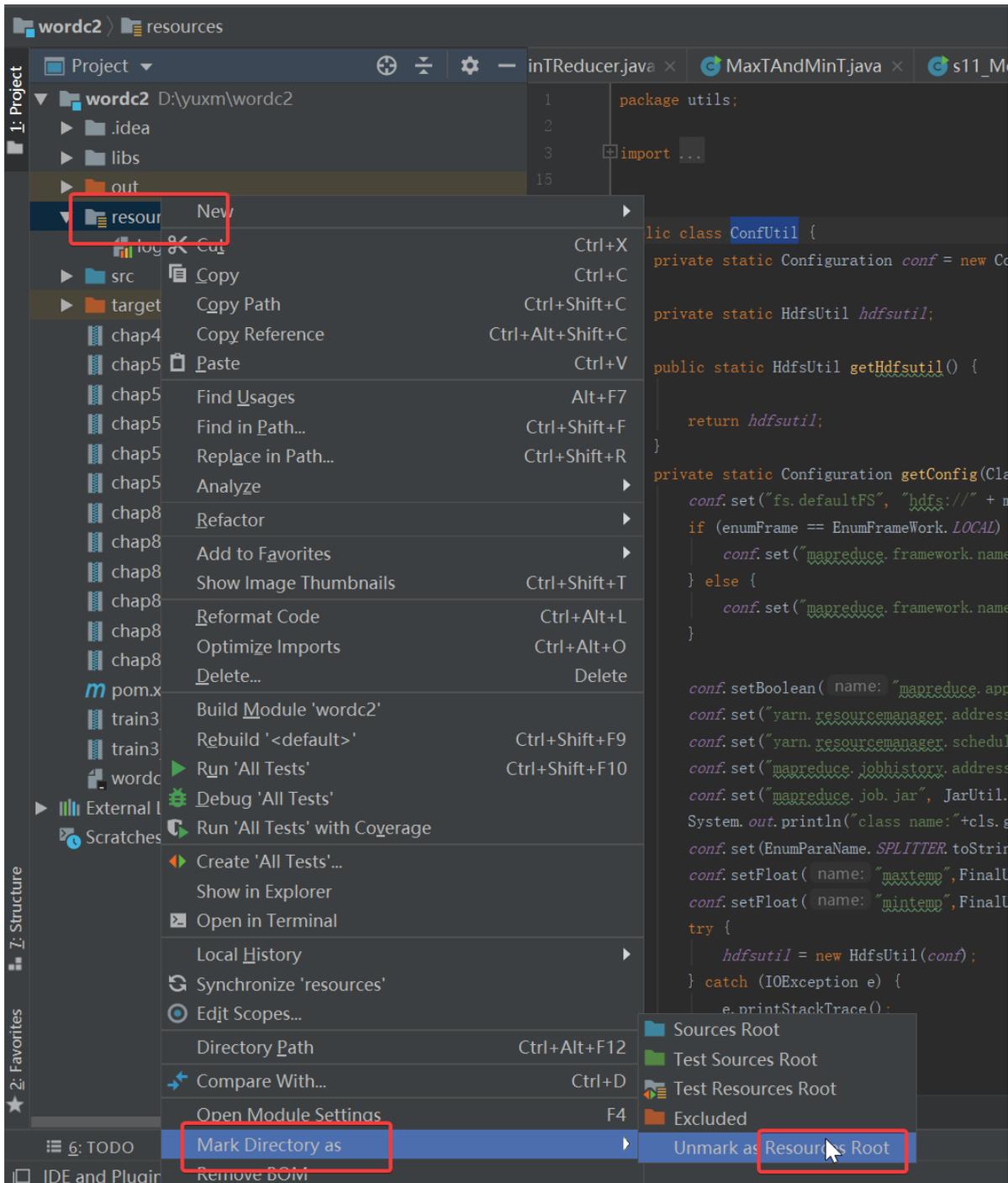
添加libs目录

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:



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

```

package utils;

import org.apache.hadoop.conf.Configuration;

public class ConfUtil {
    private static Configuration conf = new Configuration();

    private static HdfsUtil hdfsutil;

    @SuppressWarnings("unchecked")
    public static HdfsUtil getHdfsutil() {
        return hdfsutil;
    }

    private static Configuration getConfig(Class<?> cls, String master, EnumFrameWork enumFrame) {
        conf.set("fs.defaultFS", "hdfs://" + master + ":8020");
        if (enumFrame == EnumFrameWork.LOCAL) {
            conf.set("mapreduce.framework.name", "local"); // 指银行拷使银行拷yarn银行拷本地
        } else {
            conf.set("mapreduce.framework.name", "yarn"); // 指银行拷使银行拷yarn银行拷本地
        }

        conf.setBoolean("mapreduce.app-submission.cross-platform", true); // 银行拷平台解结
        conf.set("yarn.resourcemanager.address", master + ":8030"); // 指银行拷资源管理器
        conf.set("yarn.resourcemanager.scheduler.address", master + ":8030"); // 指银行拷资源管理器调度器
        conf.set("mapreduce.job.jar", JarUtil.Jar(cls)); // 银行拷要指银行拷jar银行拷
        System.out.println("class name:" + cls.getName());
        conf.set(EnumParaName.SPLITTER, FinalUtil.Splitter);
        conf.setFloat("name", FinalUtil.MaxTemp);
        conf.setFloat("name", FinalUtil.MinTemp);

        try {
            hdfsutil = new HdfsUtil(conf);
        } catch (IOException e) {
            e.printStackTrace();
        }

        return conf;
    }
}

```

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`

```

package utils;

import enums.EnumFrameWork;

public final class FinalUtil {

    public final static String MasterName = "master"; // c31, master, c21

    public final static String NewSelectDataOutputPath = "/user/933886/output_news_all";
    public final static String NewSelectDataOutputFile = "part-r-000000";

    public final static String NewsWordCountOutputPath = "/user/933886/output_news_allcount";
    public final static String NewsWordSortOutputPath = "/user/933886/output_news_sorted";

    public final static String MusicSelectDataOutputPath = "/user/933886/output_music_all";
    public final static String MusicSelectDataOutputFile = "part-r-000000";

    public final static String MusicWordCountOutputPath = "/user/933886/output_music_allcount";
    public final static String MusicWordSortOutputPath = "/user/933886/output_music_sorted";

    public final static String BookSelectDataOutputPath = "/user/933886/output_book_all";
    public final static String BookSelectDataOutputFile = "part-r-000000";

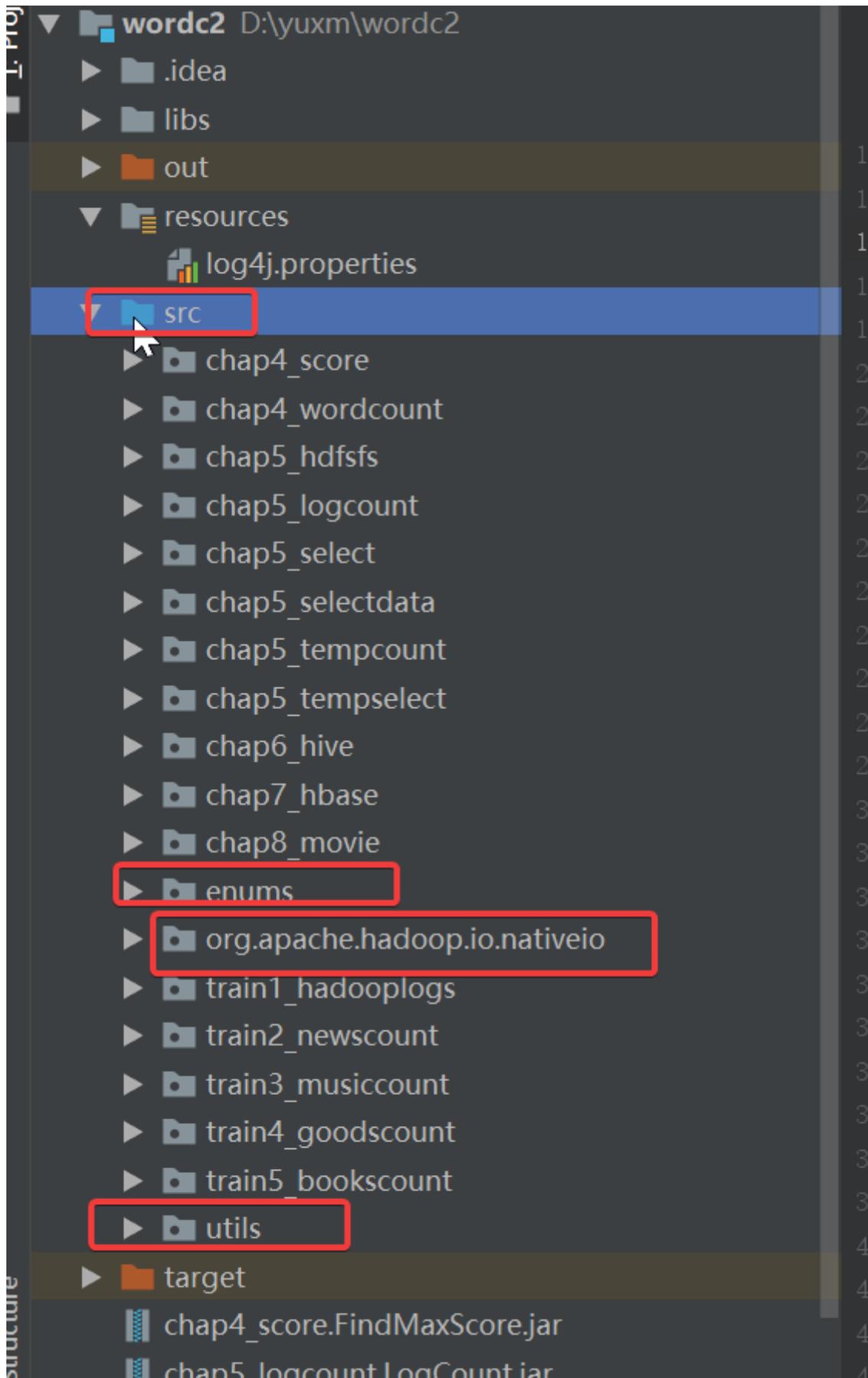
    public final static String BookWordCountOutputPath = "/user/933886/output_book_allcount";
    public final static String BookWordSortOutputPath = "/user/933886/output_book_sorted";

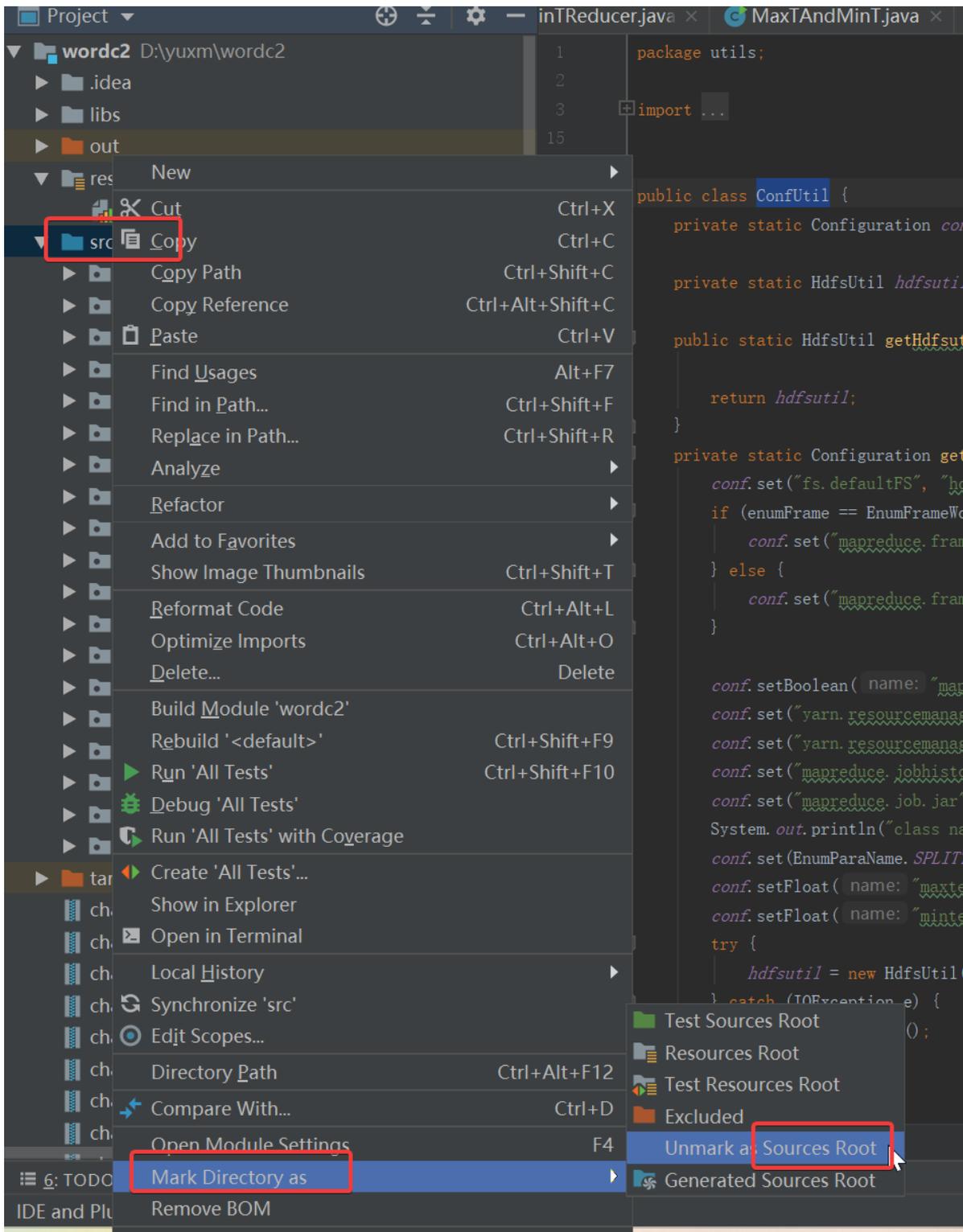
    public final static String MovieRatingInputPath = "/user/933886/movie/ratings.dat"; // /tmp/Hadoop/MapReduce/ratings.dat
    public final static String MovieRatingOutputPath = "/user/933886/movie/join/output/";
    public final static String MovieMoviesInputPath = "/user/933886/movie/movies.dat";
    public final static String MovieUsersInputPath = "/user/933886/movie/users.dat";
    public final static String MovieRatingAllOutputPath = "/user/933886/movie/join/outputAll/";
    public final static String MovieRatingTopOutputPath = "/user/933886/movie/join/outputTop10/";
    // /join/outputMapJoinThreeTables/
    public final static String MovieJoinTablesOutputPath = "/user/933886/movie/join/outputMapJoinThreeTables/";
    // /join/outputTop10/
    // /join/outputMoviesRatesAllGroupByGender/
    public final static String MovieRatesAllGroupByGenderOutputPath = "/user/933886/movie/join/outputMoviesRatesAllGroupByGender/";
    // /join/MoviesRatesTop10GroupByGender/
}

```

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. 本实训允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)

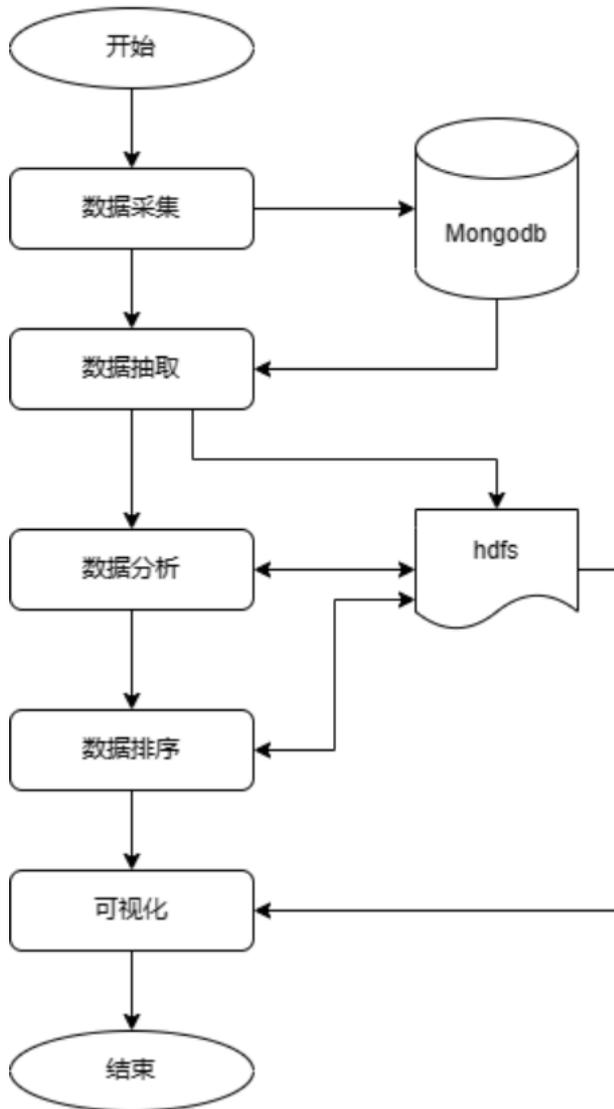
1. 数据采集: 使用scrapy框架实现某类题材网站的数据采集,存入mongo数据库。
2. 数据分析: 1) 使用java采集题材数据从mongo到hdfs; 2) 使用java对hdfs上的数据进行MR分析; 3) 使用java将分析结果进行排序。
3. 数据可视: 使用python将MR的排序后的结果进行可视化并上传到web服务器。

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

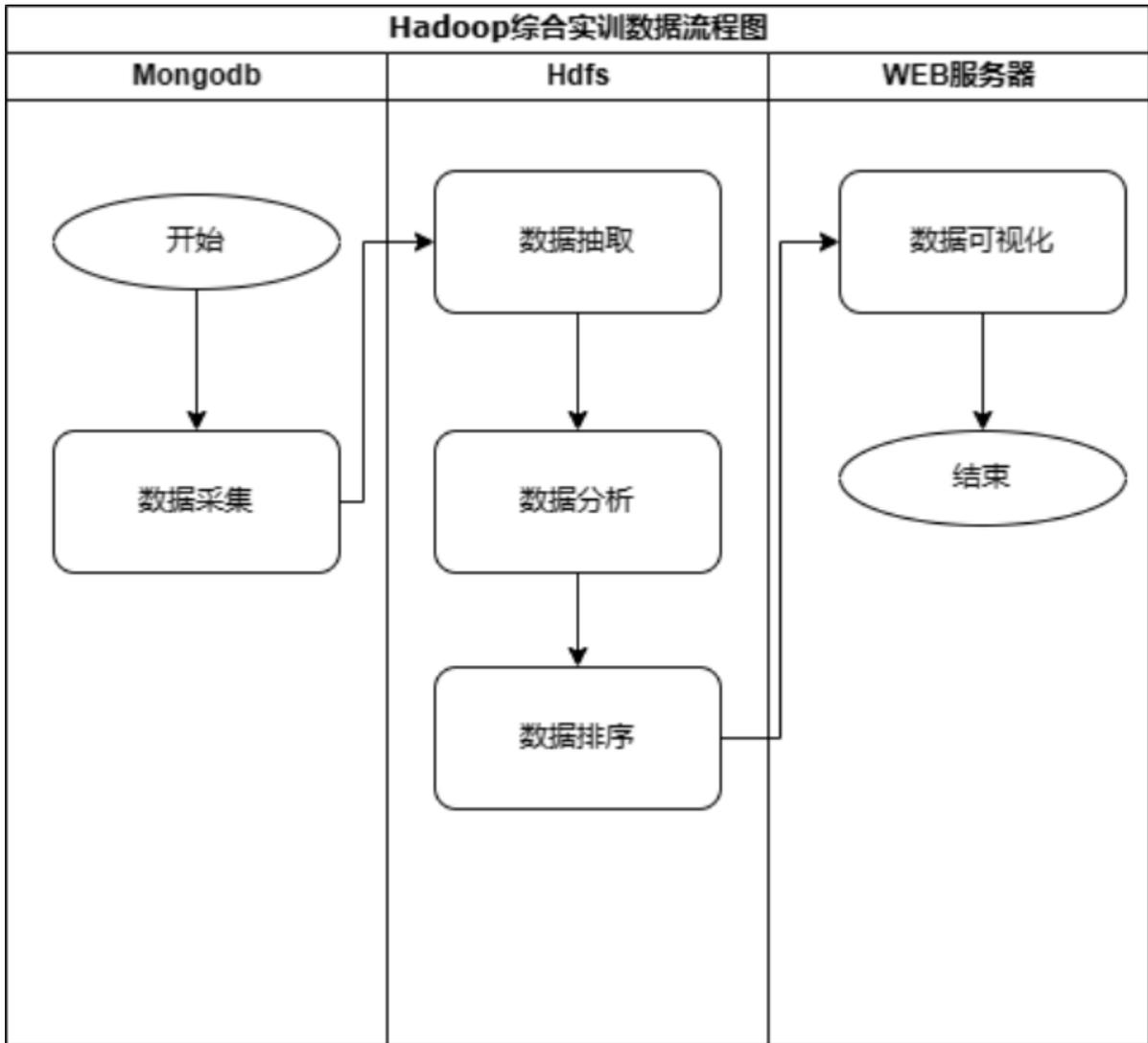


【业务流程说明】(2分)

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)

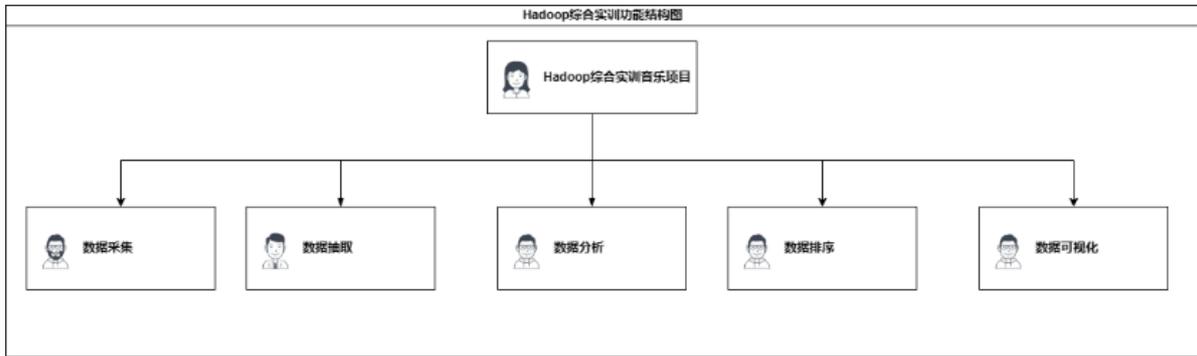


【数据流程说明】(3分)

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

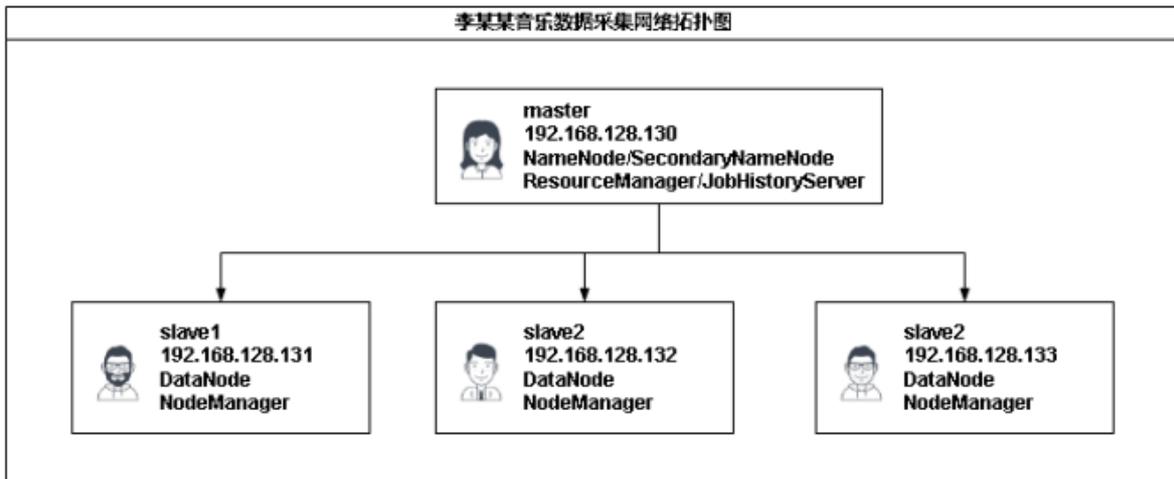


1.1.4. 运行环境(10分)

I 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	

I 网络拓扑图(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

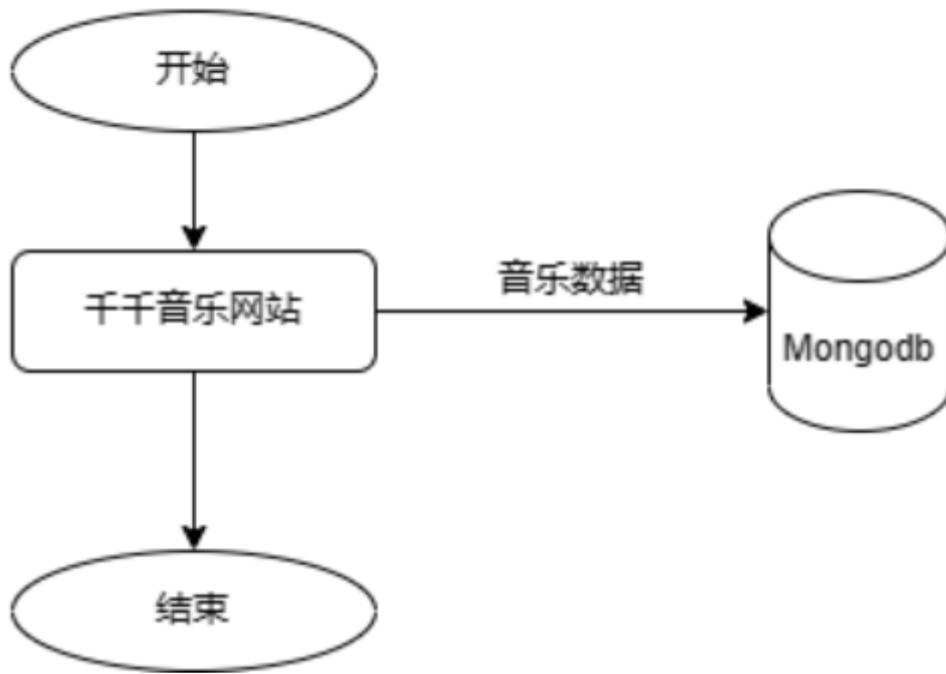
采集过程: 利用Pycharm从千千音乐网站采集音乐数据到mongodb库中

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

数据采集流程图



【源码实现】(4分)：

scrapy脚本：

```
import json
import time

import scrapy
from scrapy import Selector
from selenium import webdriver

from music.items import MusicItem

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 I

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

```

2、pipeline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、middleware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)

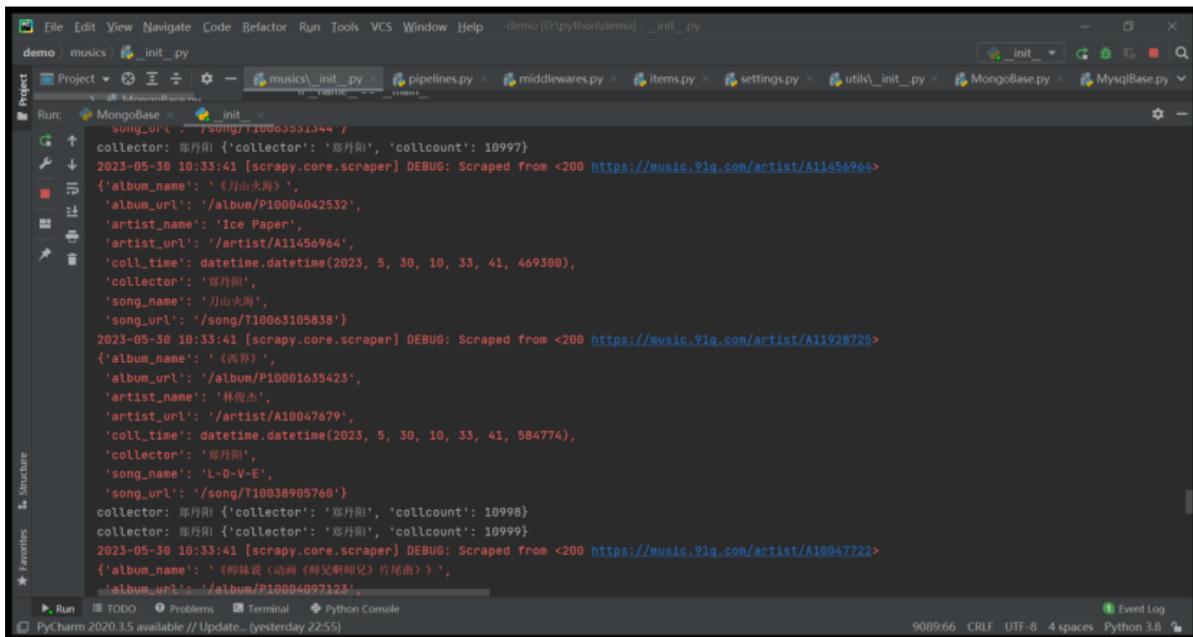
```

4、item脚本

```
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

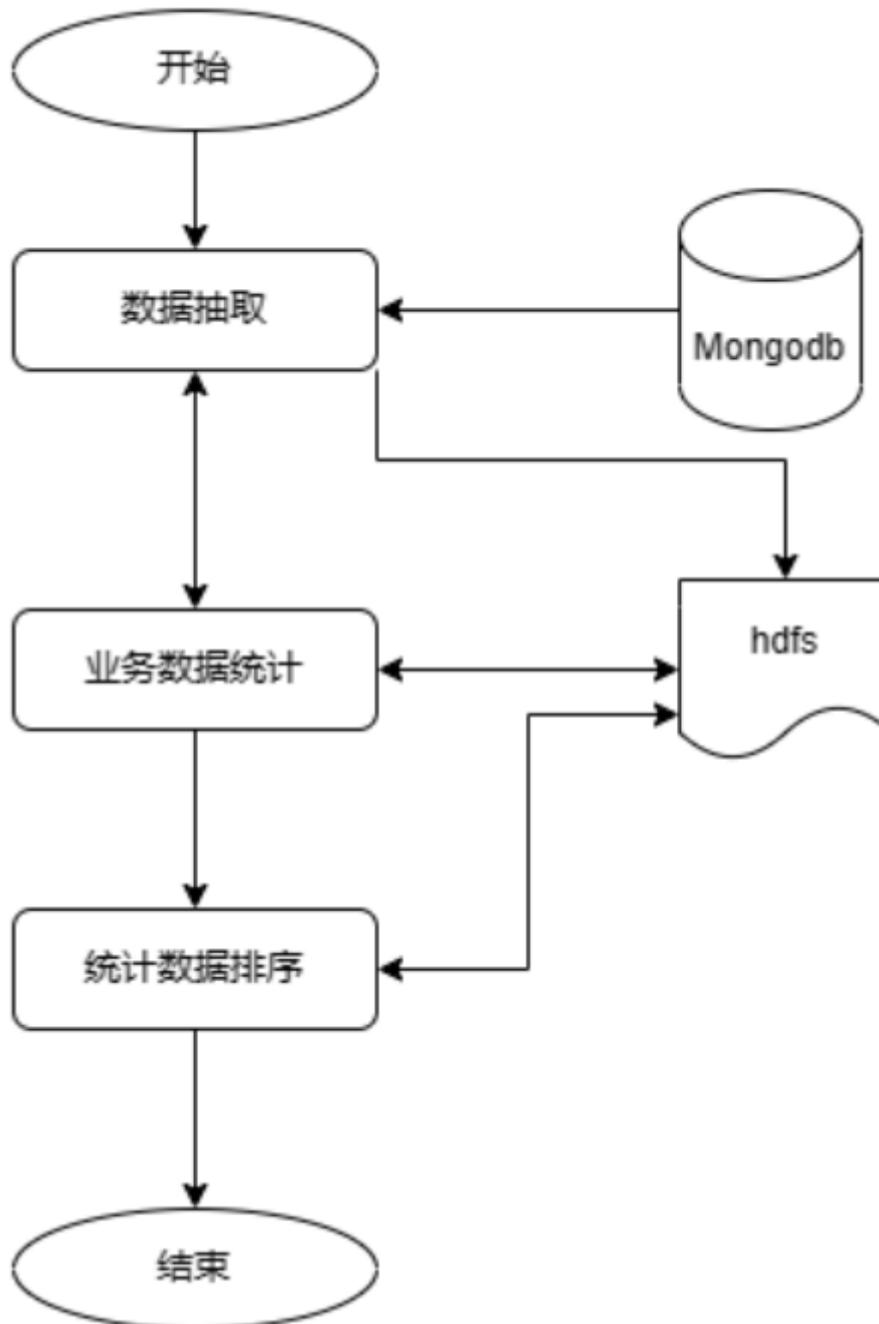
使用方法：

- 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收集到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

数据分析流程图



【源码实现】(16分)：

S1_MusicSelectData代码：

```
Map/Reduce - MemberCount1/src/train3_musiccount/S1_MusicSelectDatajava - Eclipse
文件(F) 编辑(E) 源码(S) 重构(T) 浏览(N) 搜索(a) 项目(P) 运行(R) 窗口(W) 帮助(H)

MaxTAndMin... S1_MusicSele... S2_MusicWord... S3_MusicWord... FinalUtilJava UtilTestJava ConfUtilJava
1 package train3_musiccount;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24 public class S1_MusicSelectData {
25
26     public static class SelectDataMapper extends Mapper<Object, BSONObject, Text, IntWritable> {
27         private final static IntWritable one = new IntWritable(1);
28
29         @Override
30         protected void map(Object key, BSONObject value, Context context) throws IOException, InterruptedException {
31
32             try {
33                 if (value.get("collector")!=null && value.get("song_name")!=null)
34                     {
35                         Text txt = new Text(value.get("collector").toString() + "::" + value.get("song_name").toString());
36                         context.write(txt, one);
37                     }
38                 else
39                     {
40                         context.getCounter(EnumExceptionCounter.MusicSelectNull).increment(1);
41                     }
42             } catch (IOException e) {
43                 e.printStackTrace();
44             } finally {
45             }
46         }
47     }
48
49     public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
50         private IntWritable result = new IntWritable();
51
52         public void reduce(Text key, Iterable<IntWritable> values, Context context)
53             throws IOException, InterruptedException {
54             int sum = 0;
55             for (IntWritable val : values) {
56                 sum += val.get();
57             }
58             result.set(sum);
59             context.write(key, result);
60         }
61     }
62 }
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
Map/Reduce - MemberCount1/src/train3_musiccount/S1_MusicSelectDatajava - Eclipse
文件(F) 编辑(E) 源码(S) 重构(T) 浏览(N) 搜索(a) 项目(P) 运行(R) 窗口(W) 帮助(H)

MaxTAndMin... S1_MusicSele... S2_MusicWord... S3_MusicWord... FinalUtilJava UtilTestJava ConfUtilJava
50
51
52
53 public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
54     private IntWritable result = new IntWritable();
55
56     public void reduce(Text key, Iterable<IntWritable> values, Context context)
57         throws IOException, InterruptedException {
58         int sum = 0;
59         for (IntWritable val : values) {
60             sum += val.get();
61         }
62         result.set(sum);
63         context.write(key, result);
64     }
65 }
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

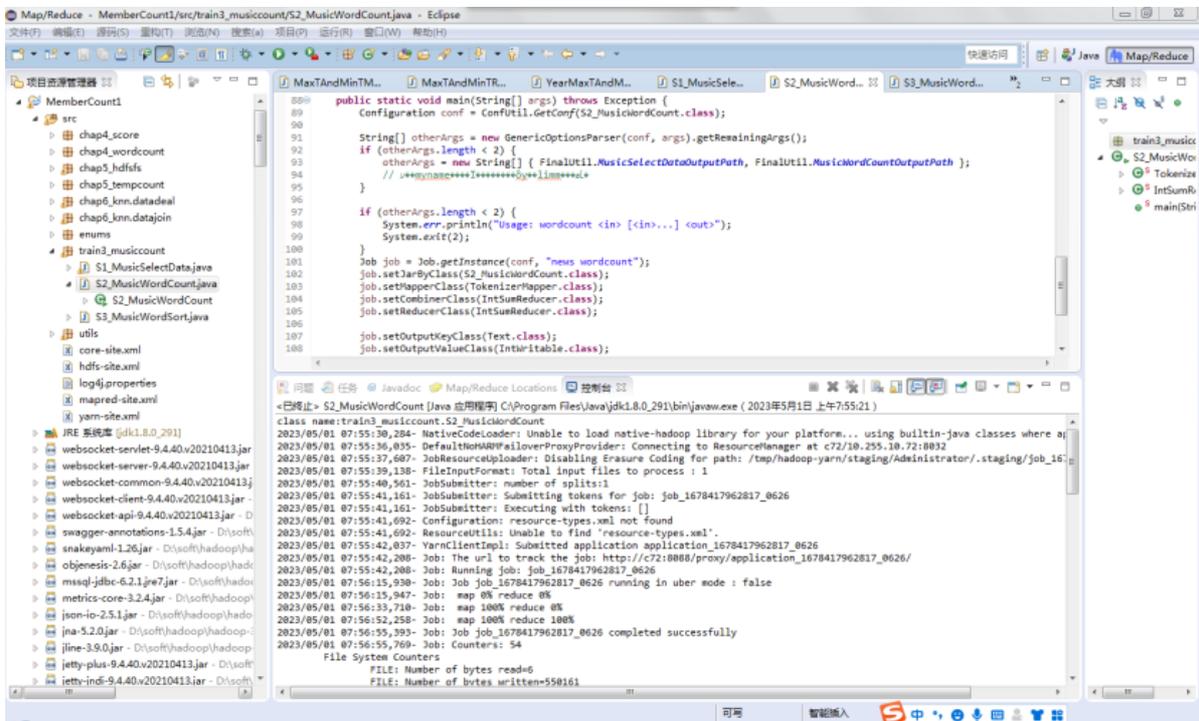
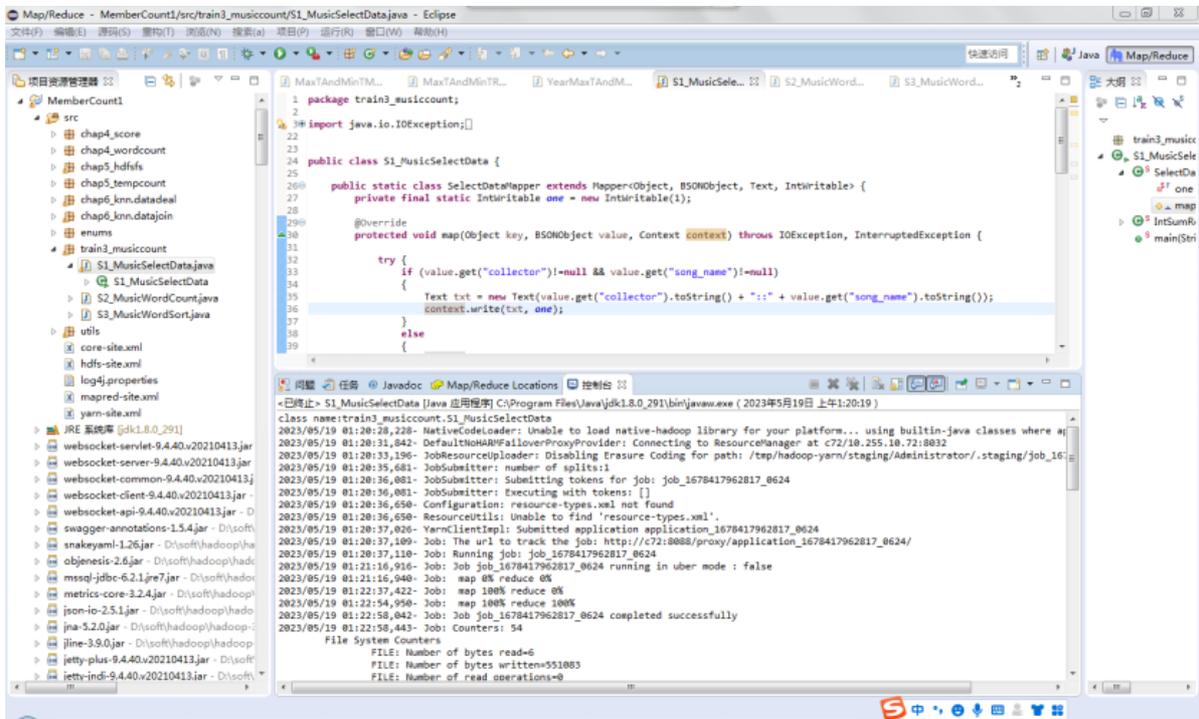
S2_MusicWordCount 代码

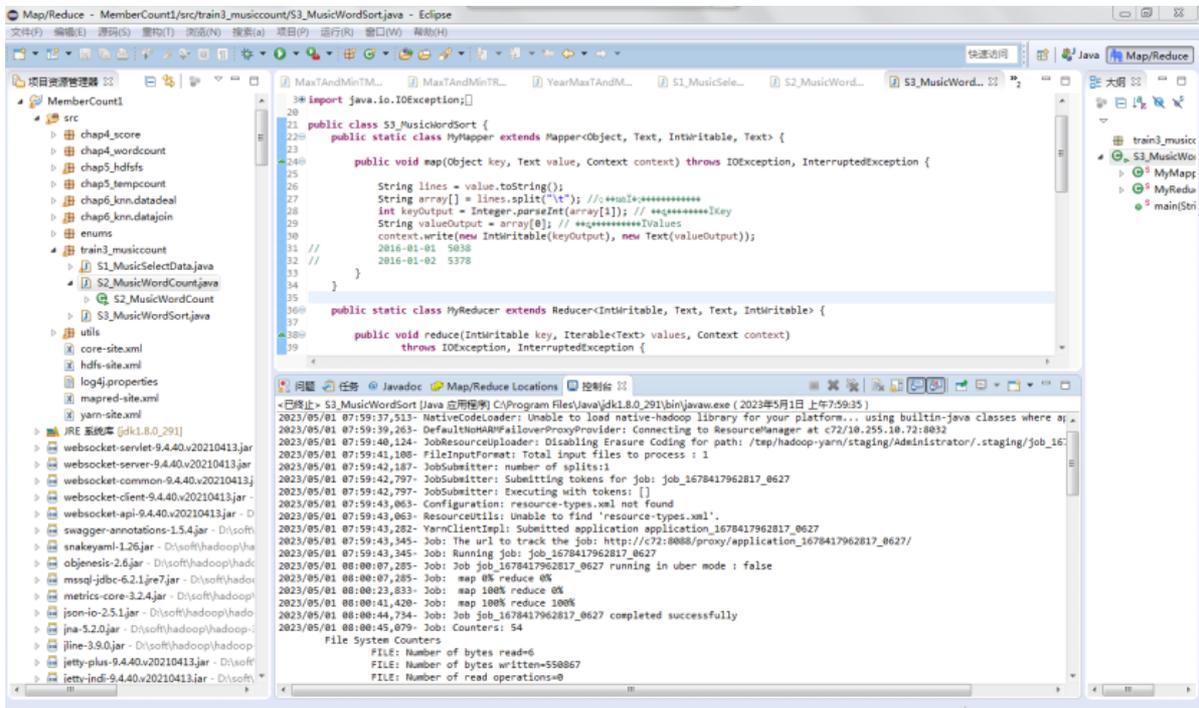

```
1 package train3_musiccount;
2
3 import java.io.IOException;
4
5 public class S3_MusicWordSort {
6     public static class MyMapper extends Mapper<Object, Text, IntWritable, Text> {
7
8         public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
9
10             String lines = value.toString();
11             String array[] = lines.split("\t"); //::+::+::+::+::+::+::+
12             int keyOutput = Integer.parseInt(array[1]); // +::+::+::+::+::+::+::+::+::+::+::+::+
13             String valueOutput = array[0]; // +::+::+::+::+::+::+::+::+::+::+::+::+
14             context.write(new IntWritable(keyOutput), new Text(valueOutput));
15             //
16             // 2016-01-01 5038
17             // 2016-01-02 5378
18         }
19     }
20
21     public static class MyReducer extends Reducer<IntWritable, Text, Text, IntWritable> {
22
23         public void reduce(IntWritable key, Iterable<Text> values, Context context)
24             throws IOException, InterruptedException {
25             for (Text value : values) {
26                 context.write(value, key);
27             }
28             //
29             // 5038 ,2016-01-01
30             // 5378 , 2016-01-02
31         }
32     }
33
34     public static void main(String[] args) throws Exception {
35         Configuration conf = ConfUtil.GetConf(S3_MusicWordSort.class);
36         String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
37         if (otherArgs.length < 2) {
38             otherArgs = new String[] { FinalUtil.MusicWordCountOutputPath, FinalUtil.MusicWordSortOutputPath };
39             // myname::+::+::+::+::+::+
40             if (otherArgs.length < 2) {
41                 System.err.println("Usage: wordcount <in> [<in>...] <out>");
42                 System.exit(2);
43             }
44             Job job = Job.getInstance(conf, "newsWordSort");
45         }
46     }
47 }
48
```

```
39     public void reduce(IntWritable key, Iterable<Text> values, Context context)
40         throws IOException, InterruptedException {
41         for (Text value : values) {
42             context.write(value, key);
43         }
44         //
45         // 5038 ,2016-01-01
46         // 5378 , 2016-01-02
47     }
48
49     public static void main(String[] args) throws Exception {
50         Configuration conf = ConfUtil.GetConf(S3_MusicWordSort.class);
51         String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
52         if (otherArgs.length < 2) {
53             otherArgs = new String[] { FinalUtil.MusicWordCountOutputPath, FinalUtil.MusicWordSortOutputPath };
54             // myname::+::+::+::+::+::+
55             if (otherArgs.length < 2) {
56                 System.err.println("Usage: wordcount <in> [<in>...] <out>");
57                 System.exit(2);
58             }
59             Job job = Job.getInstance(conf, "newsWordSort");
60             job.setJarByClass(S3_MusicWordSort.class);
61             job.setMapperClass(MyMapper.class);
62             job.setReducerClass(MyReducer.class);
63             job.setMapOutputKeyClass(IntWritable.class);
64             job.setMapOutputValueClass(Text.class);
65             job.setOutputKeyClass(Text.class);
66             job.setOutputValueClass(IntWritable.class);
67             job.setSortComparatorClass(IntWritableComparator.class);
68
69             for (int i = 0; i < otherArgs.length - 1; ++i) {
70                 FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
71             }
72             FileSystem.get(conf).delete(new Path(otherArgs[otherArgs.length - 1]), true);
73             FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
74             System.exit(job.waitForCompletion(true) ? 0 : 1);
75         }
76     }
77 }
78
```

【运行截图】(12分)：

音乐数据采集代码运行成功截图



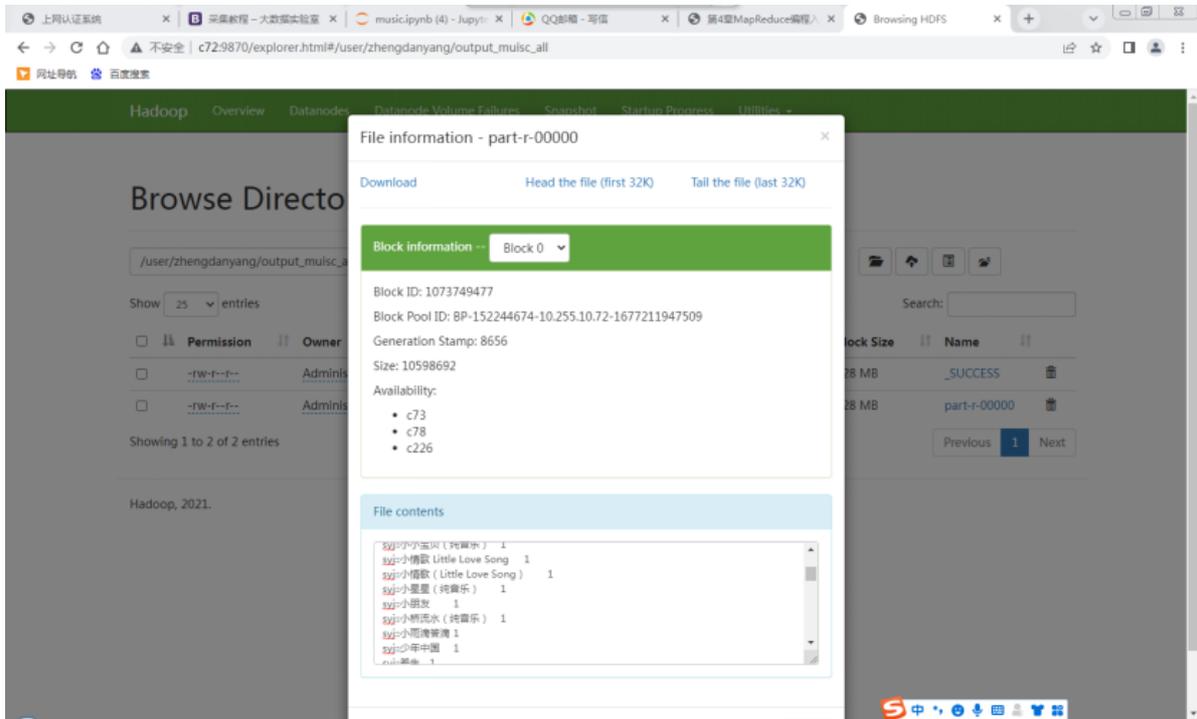


在<http://c72:9870>的文件系统中，运行输出结果截图

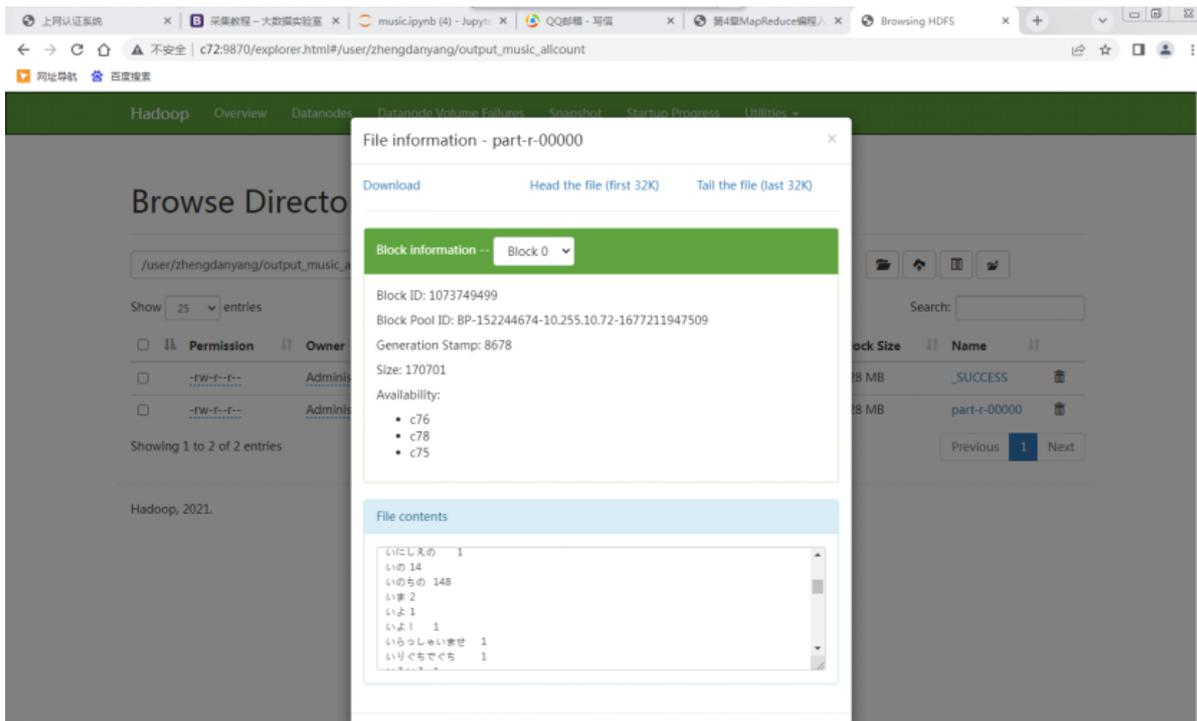
1、zhengdanyang目录下的文件

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_musc_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
-rwxr-xr-x	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


```

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);

```

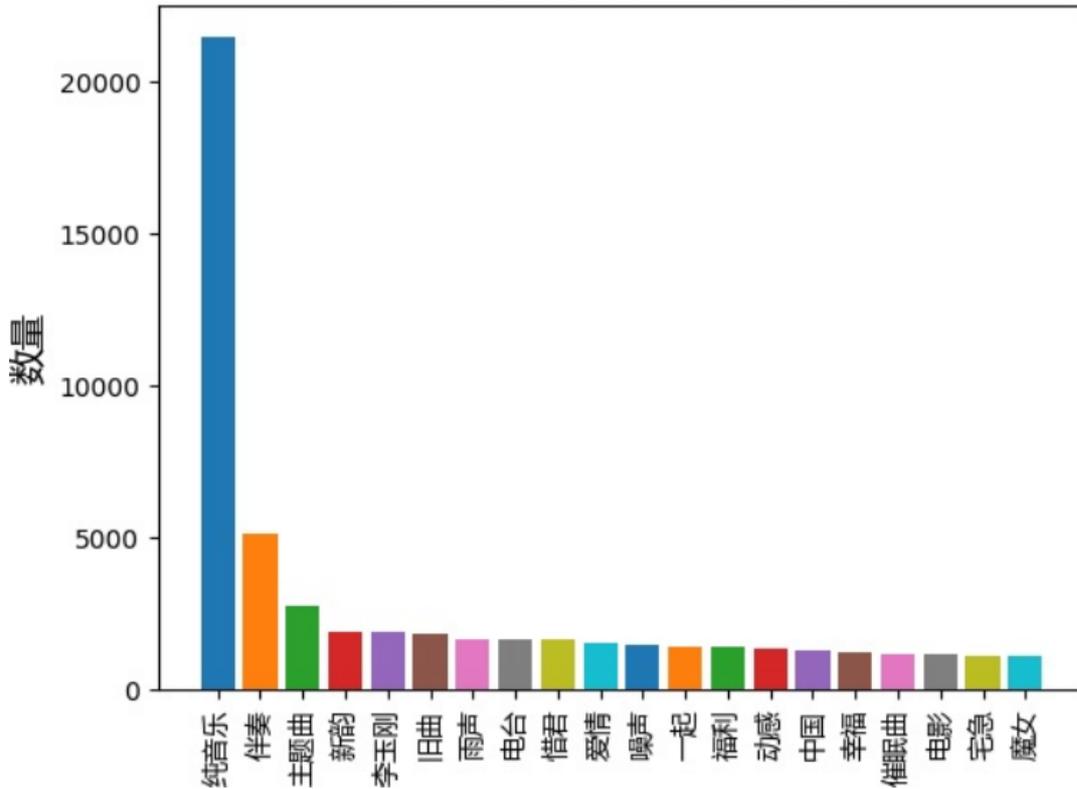
【运行截图】(3分) :

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help p7 - music_hadoop_view_v2.py
p7 p7 code views music_hadoop_view_v2.py music_hadoop_view_v2.py 2027340227_01.jpg pip.txt
Run music_hadoop_view_v2
C:\Users\juzi\AppData\Local\Programs\Python\Python38\python.exe D:/python/p7/p7/code/views/music_hadoop_view_v2.py
['纯音乐\t21474', '伴奏\t5131', '主题曲\t2781', '新韵\t1888', '李玉刚\t1875', '旧曲\t1869', '雨声\t1673', '电台\t1646', '借君\t1633', '爱情\t1551', '噪声\t1499']
['纯音乐', '伴奏', '主题曲', '新韵', '李玉刚', '旧曲', '雨声', '电台', '借君', '爱情', '噪声', '一起', '福利', '动感', '中国', '幸福', '催眠曲', '电影', '宅急', '魔女']
[21474, 5131, 2781, 1888, 1875, 1869, 1673, 1646, 1633, 1551, 1499, 1427, 1402, 1329, 1282, 1212, 1172, 1163, 1098, 1098]
success upload.....
Process finished with exit code 0
PyCharm 2020.15 available // Update... (yesterday 22:55)

```

音乐词频统计(学生:郑丹阳)



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：

```
1 | https://jihulab.com/biglab-share/scrapy.git
2 | 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：

```
1 | 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：

```
1 | 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](#)