实验二：词频统计与可视化

Design by W.H Huang | Direct by Prof Feng

1 实验目的

通过本次实验，你应该：

- 熟悉 *hadoop+ Spark*下编程环境
- 掌握基于*Spark*的基本*MAP REDUCE* 操作
- 掌握基本大数据可视化工具
- 独立完成本次青年群体择偶观分析实验
- 【新增】远程开发相关知识

本次实验需小组内分工合作完成两个任务：

1. *WordCount* 词频统计

   你将会使用到 *jieba* 分词 & 基于 *pySpark* 的基本 *MAP REDUCE* 操作进行词频统计，在指定数据集上大数据分析青年群体择偶观倾向。

2. 大数据可视化

   你将使用 *echars & WordCloud* 两个可视化库来进行大数据可视化，小组独立完成核心代码编写、测试。

2 实验准备

2.0 成绩说明

本次实验，根据各位同学选择的**不同环境搭建方式**，不同的成绩分数说明如下。

【注】本次实验不要求使用分布式。

实验环境	最高分
云服务器	100
VM虚拟机	95

在本次实验我们给予学有余力的同学，在完成本次实验的基础上提出了扩展要求。

【注】加分后总分不超过100分。

扩展要求	加分	
1. 使用分布式完成本次实验	+5	可参考ex3~ex4
2. 扩充原有数据集（100M以上），或基于新的数据集进行实验	+5~+10	根据数据量、质量、难度给分
3. 新增更多可视化，如柱状图等	+5~+10	根据可视化工作量给分
4. 使用更好的算法分析数据，如应用深度学习模型Bert	+10	有对比实验最好

2.1 上传文件

在开始实验前， 首先要将代码及相关资源上传到服务器。该小节将介绍如何使用FTP软件将本地（Windows）文件上传到服务器(Linux)。

1. 下载软件

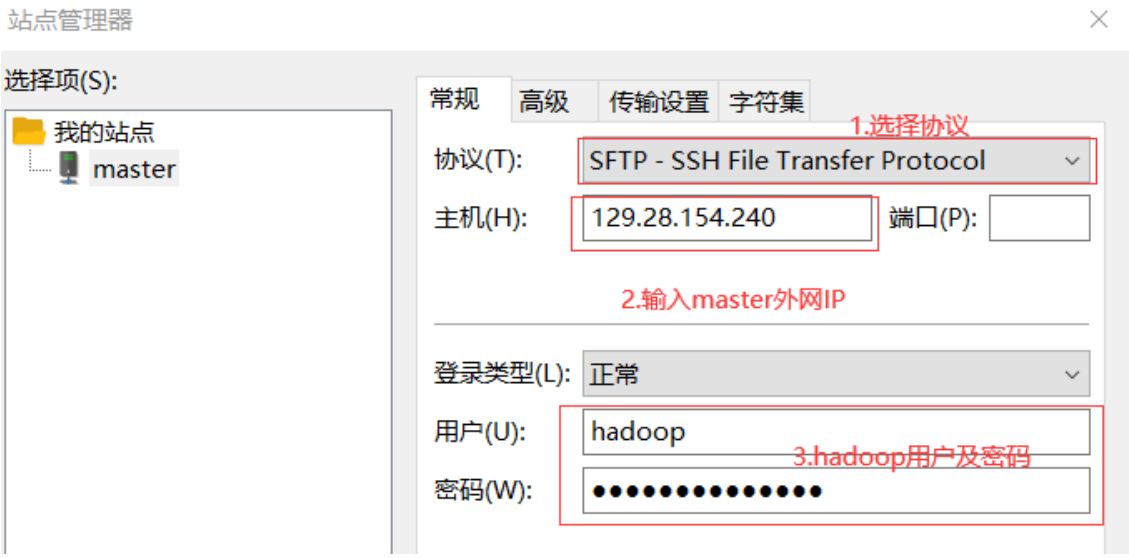
   FTP工具我们选择`Filezilla`，下载地址：Filezilla下载

   点击进行下载安装，安装过程较为简单不再赘述。

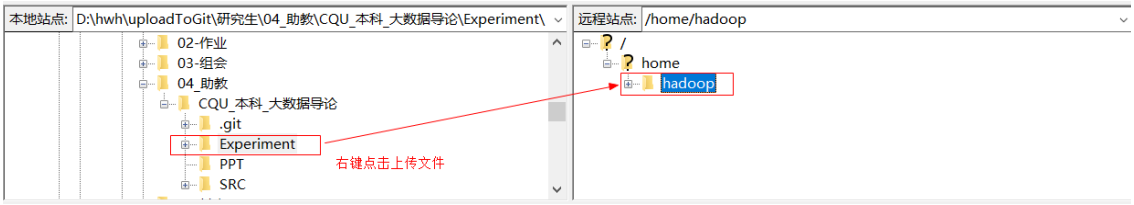
2. 连接服务器

   依次点击：文件 --> 站点管理器 --> 新站点

3. 上传文件

   如下图所示，左侧为本地文件，右侧为服务器文件目录（默认为`/home/hadoop`）

   上传完毕后，可在服务器上查看文件：

```
[hadoop@master Experiment]$ cd /home/hadoop/Experiment/
[hadoop@master Experiment]$ ll
total 20
drwxrwxr-x 2 hadoop hadoop 4096 Jan 26 18:54 Ex1_SettingUpEnvironment
drwxrwxr-x 5 hadoop hadoop 4096 Jan 27 00:34 Ex2_WordCount
```

2.2 【强烈建议】远程开发

> 教程支持：@white_windmills @Wanghui-Huang @ghc2000
>
> - 云环境：基于华为云，腾讯云/阿里云也可
> - 操作系统：基于Ubantu，CentOS下基本一致

在教程`ex0` & `2.1节`时，我们介绍了：

- `SSH`工具，如`Xshell`，可以很方便在**终端命令行操作**远程服务器；
- `FTP`工具，如`Filezilla`，可以很方便的**上传/下载**数据集和代码等；

但是因为**代码运行环境在远程服务器**上，我们只能这么进行代码编写、调试：

1. `Filezilla`上传数据集等资源；
2. `Xshell`连接服务器命令行操作，使用`vim`编写代码，关键地方`print`；
3. 编写好后命令行下执行`python code.py`，观察代码`print`输出 & 调试。

显然**这样做效率是很低下**的。或者你还想到：

1. VNC登陆云服务器，安装桌面、IDE等；
2. 在VNC中云桌面编写代码。

这种做法经过实际尝试也是不建议的：

- **资源问题**：安装桌面要**消耗服务器很大的资源**，对于本来就**资源不足**的丐版云服务器雪上加霜；
- **操作问题**：VNC桌面登陆，复制、粘贴等基本操作都很麻烦，不太方便；
- **网络问题**：实际操作下，网络延迟有时比较大，加上启动桌面很耗资源，经常会卡成PPT。

因此对于实际开发来说，我们还是**更希望能使用本地IDE去调试远程代码，而不是使用vim或者直接远程服务器安装IDE。**

那么，**如何在本地电脑使用IDE去调试远程机器上的代码**？

- 现在我们可以在VSCode/Pycharm 进行远程开发相关配置；
- VSCode/Pycharm 等IDE通常还集成了终端环境，方便我们操作服务器。

正式开始前，我们先**检查一下SSH安装情况。**

1. 查看是否安装了ssh-server服务

   非云系统默认只安装ssh-client服务。

   ```
   1 dpkg -l | grep ssh
   ```

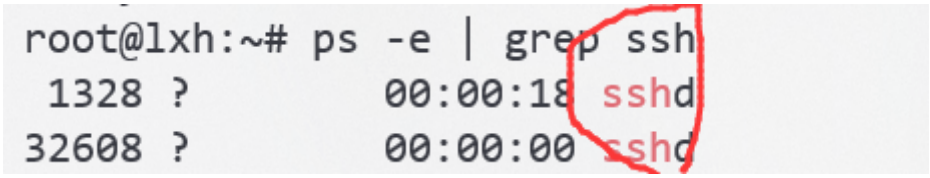
   ```
   Last login: Thu May 27 20:49:58 2021 from 218.70.255.143
   root@lxh:~# dpkg -l | grep ssh
   ii  openssh-client                      1:7.6p1-4ubuntu0.3                      amd64        secure shell (SSH) client, for secure access to remot
   e machines
   ii  openssh-server                      1:7.6p1-4ubuntu0.3                      amd64        secure shell (SSH) server, for secure access from rem
   ote machines
   ii  openssh-sftp-server                 1:7.6p1-4ubuntu0.3                      amd64        secure shell (SSH) sftp server module, for SFTP acces
   s from remote machines
   ii  ssh-import-id                       5.7-0ubuntu1.1                          all          securely retrieve an SSH public key and install it lo
   cally
   ```

   华为云等云厂商一般还会默认安装ssh-server服务，如果没有：

   ```
   1 sudo apt-get install openssh-server
   ```

2. 确认ssh-server启动

   ```
   1 ps -e | grep ssh
   ```

   如果看到`sshd`，说明ssh-server已经成功启动。

   特别的，我们设置一下**允许密码登陆root用户**：

```
vim /etc/ssh/sshd_config
```

- 注释配置文件中的`PermitRootLogin without-password`，即前面加一个`#`号
- 增加`PermitRootLogin yes`一行

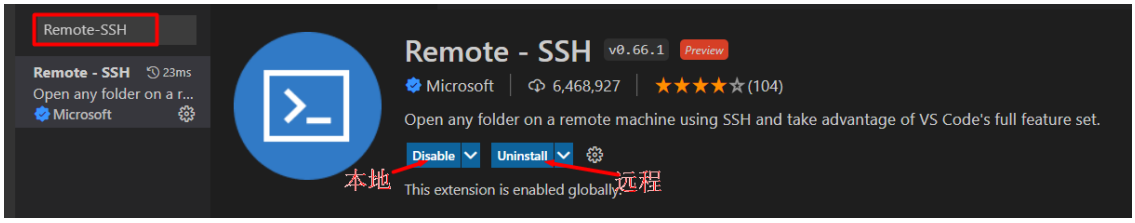
最后重启服务：

```
sudo /etc/init.d/ssh stop
sudo /etc/init.d/ssh start
```

2.2.1 VSCode远程开发实践

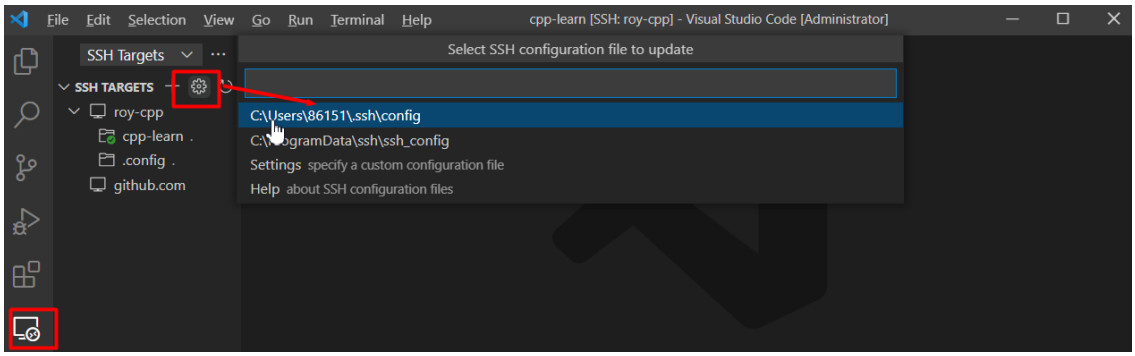
1. 安装Remote-SSH插件

   左侧Extension图标 ---> 输入`Remote-SSH` --->安装即可。注意，需要**远程和本地都进行安装。**



2. 配置Remote-SSH插件

   如图所示选择`.ssh/config`文件进行配置：



   打开文件后，需要设置以下字段：

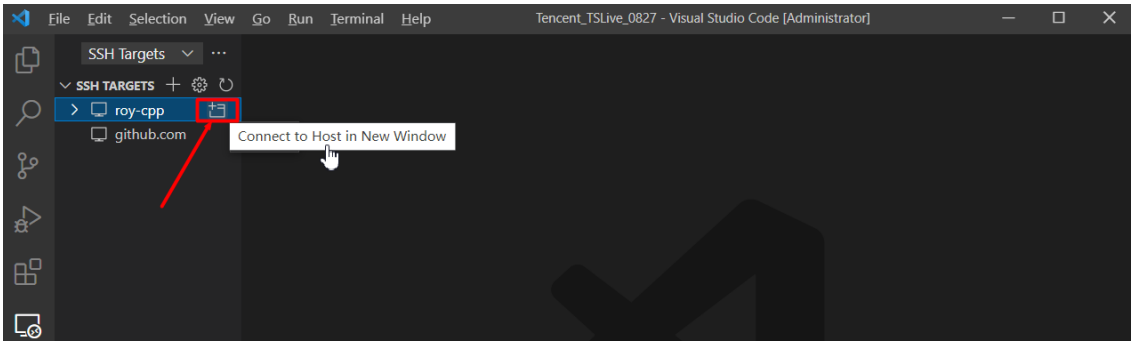
   - Host：自定义即可
   - HostName：云主机**公网IP**
   - User：登陆的用户，选择root
   - IdentityFile：免密登录私钥地址，如果没有配置则每次远程登录需要输入密码

   以下为示例：

```
Host roy-cpp
  HostName 121.36.59.23 # 云主机公网IP
  User  root
  IdentityFile D:/huawei_rsa/id_rsa # 这行可删除
```

3. 登陆测试

   点击下图按钮进行登陆：

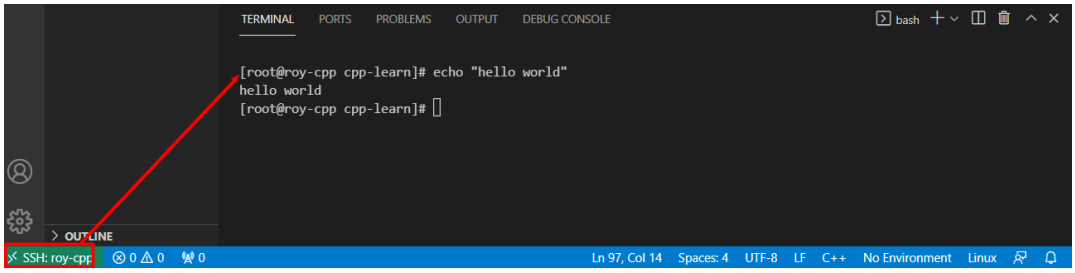


输入密码后，便可以看到远程服务器已经成功连接。

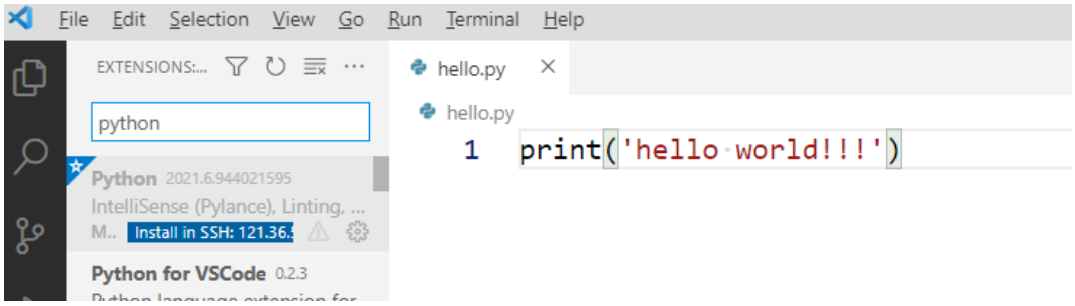
4. 远程开发实践

   连接上后就和我们本地使用VSCode没有什么区别：

   - **打开远程文件**：File--->Open Folder--->指定文件夹路径，类似打开本地文件；
   - **使用终端**：在VSCode下方`Terminal`一栏可进行命令行操作，可认为是一个ssh连接的界面，你可以在上面进行之前的所有操作；



   - **远程debug**：使用`python`插件，注意远端&本地都要安装。

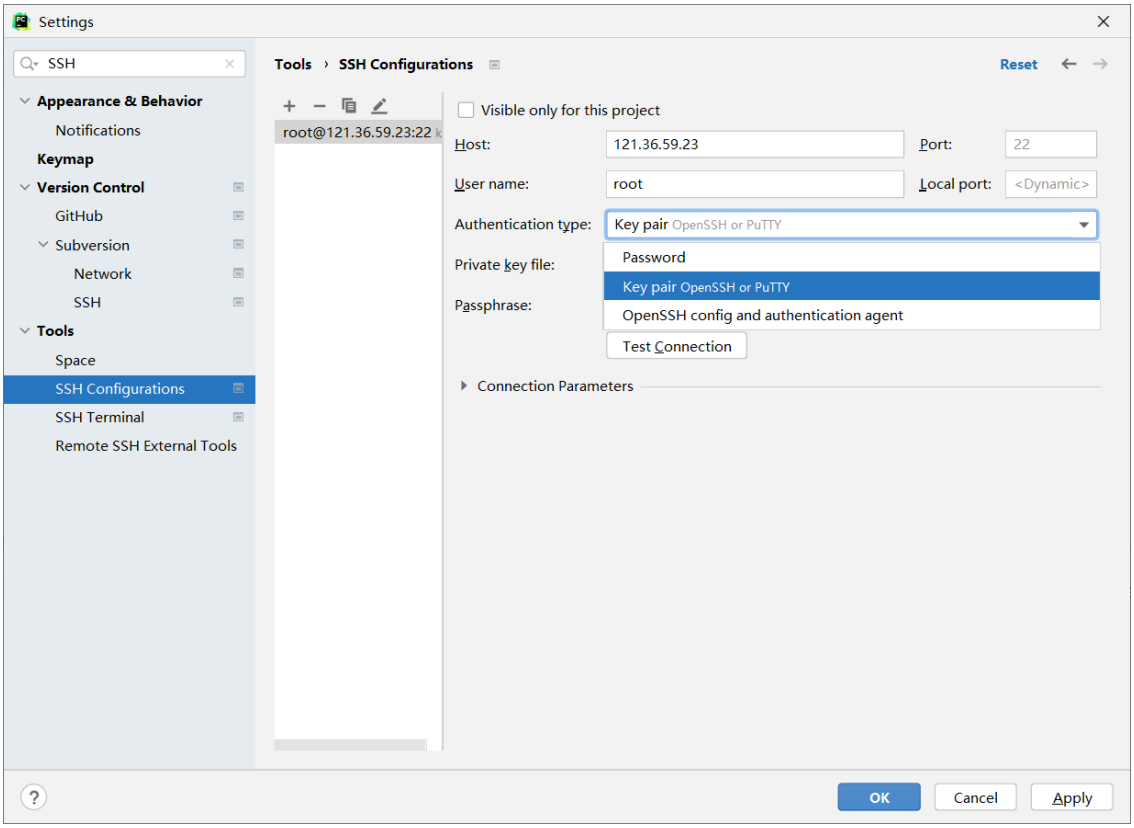


2.2.2 Pycharm远程开发实践

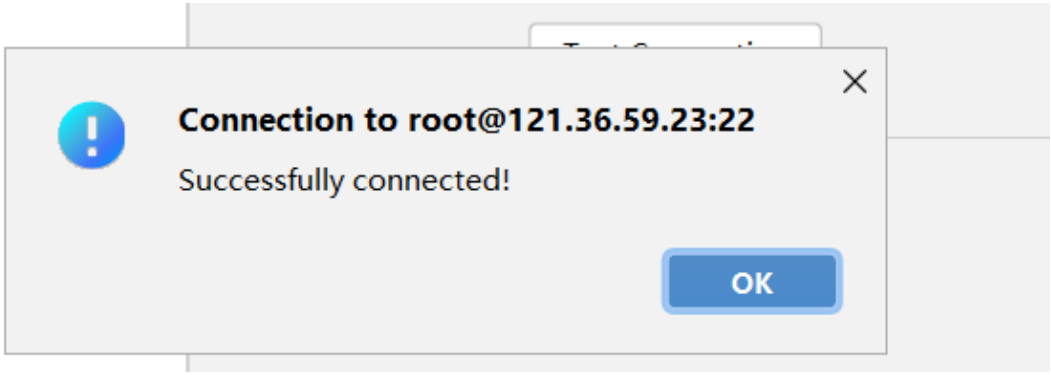
> 在Pycharm中此功能需要用**专业版**，正版的话直接去官网学生申请即可（我们重大的邮箱没法申请，需要拍学生证申请）。

1. SSH配置

   打开File->Settings->SSH Configurations，进行如下配置：

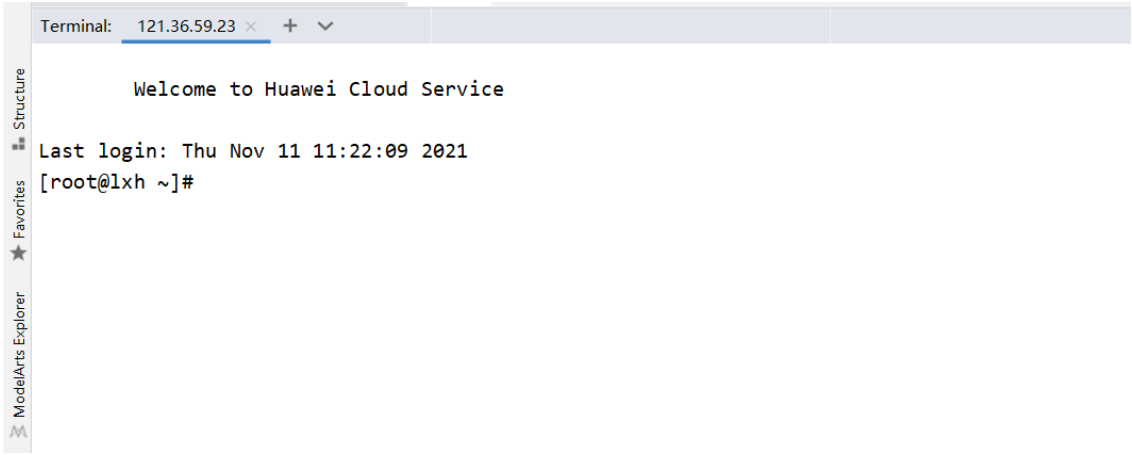


输入完成后，点击Test Connection，测试一下连接状态



2. 新建SSH Session

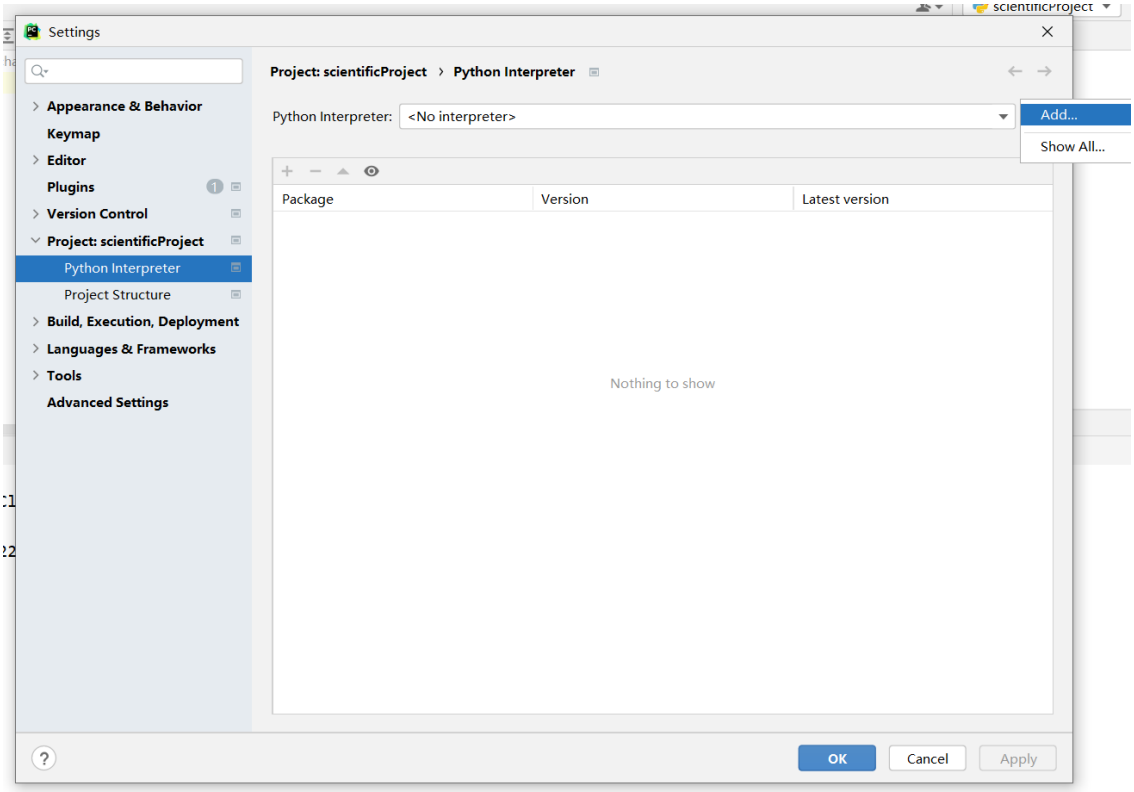
点击：Tools--->Start SSH Session...--->Edit credentials，现在可以进行对服务器的命令行操作（可以视作之前的ssh窗口）：



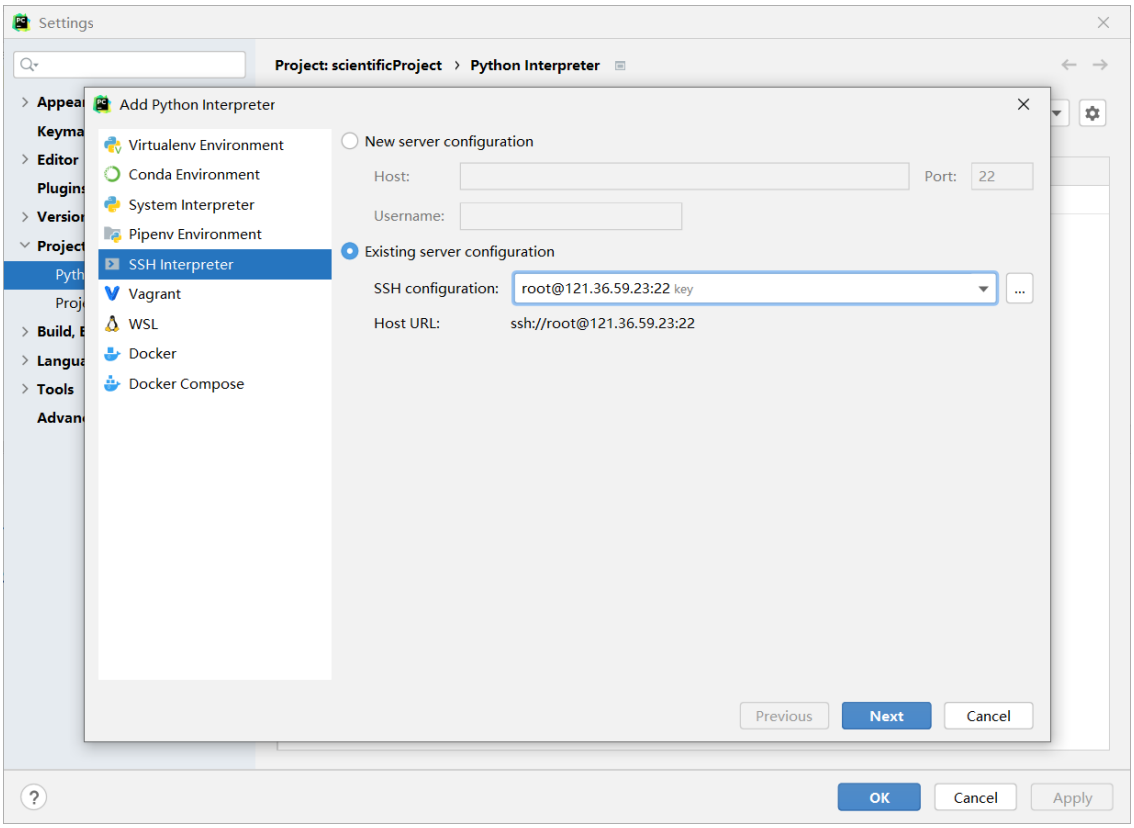
仅仅使用命令行是肯定不够的，pycharm提供了很多更便捷的工具。

3. 配置远端python解释器

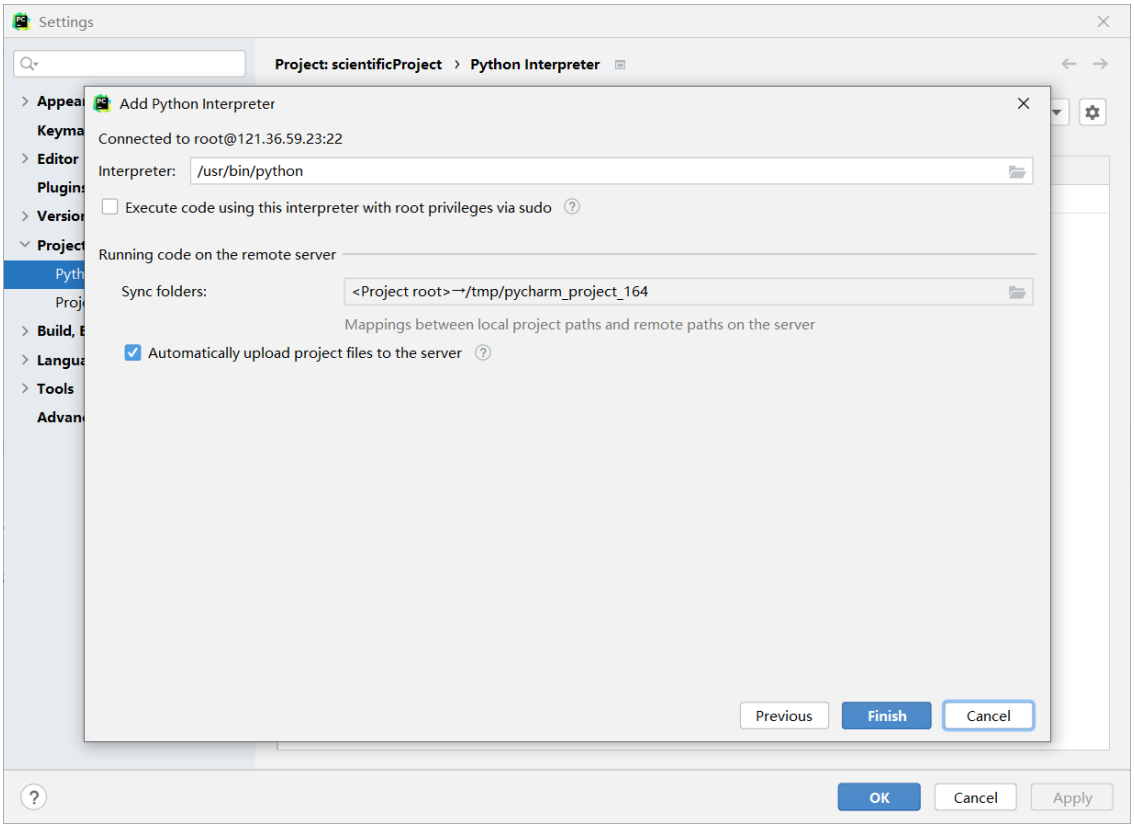
点击：File--->Settings--->Project:...-->Project Interpreter，添加python解释器：



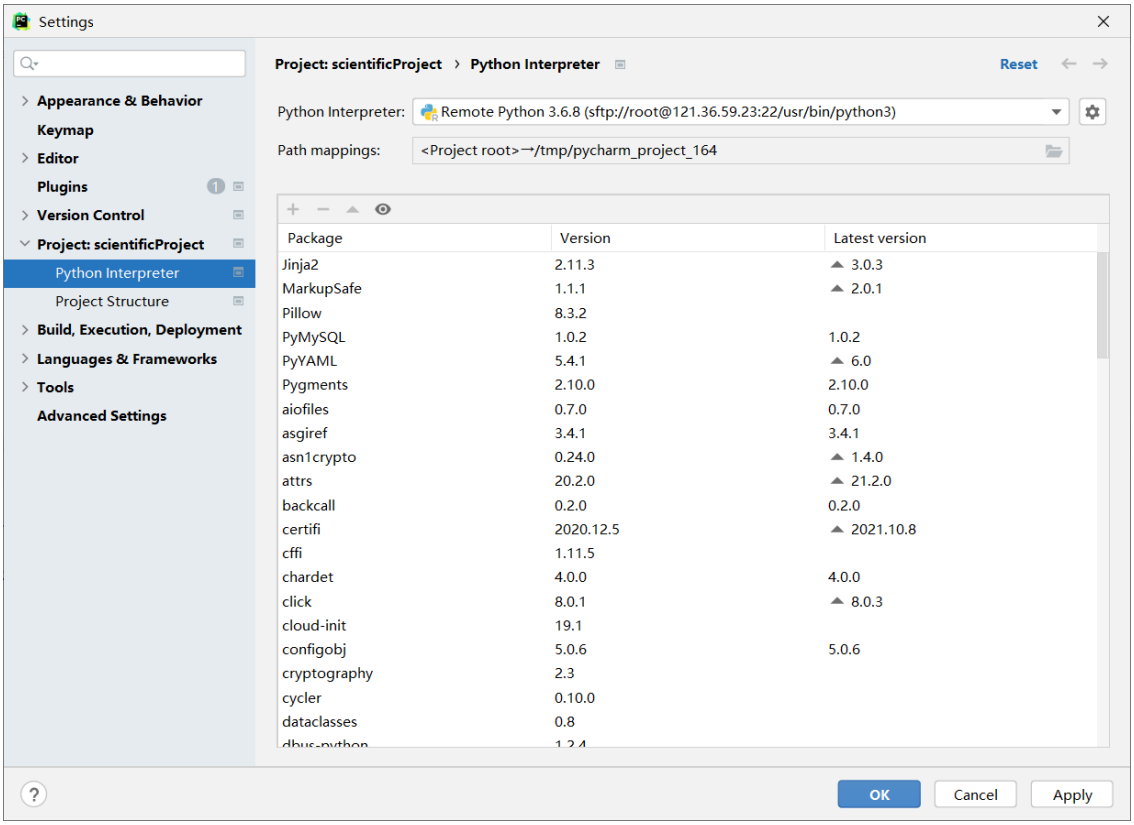
选择SSH Interpreter，选择刚刚配置好的服务器：



选择服务器中你需要的解释器，以及项目文件的映射路径，勾选自动同步等配置：



配置完成后就和本地解释器的使用差不多了：

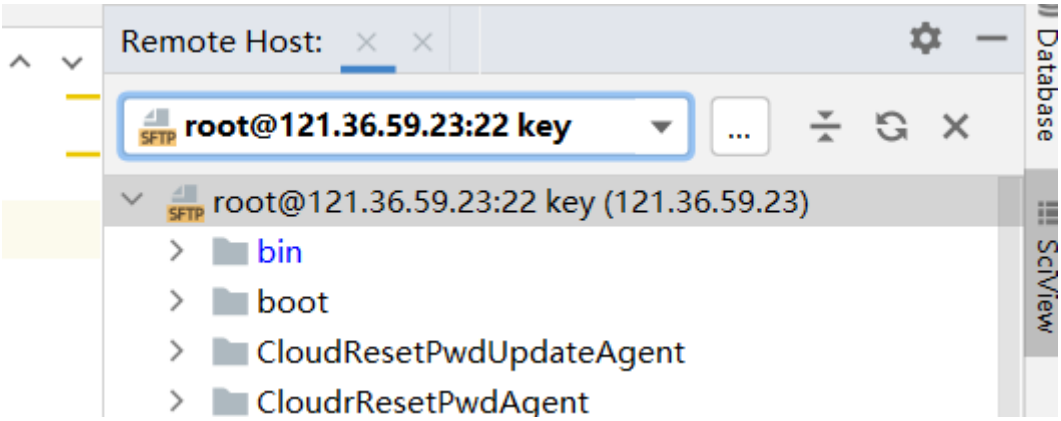


特别的如果需要修改映射路径可以在：tools->Deployment->Configurations，下进行修改。

4. 打开远端目录

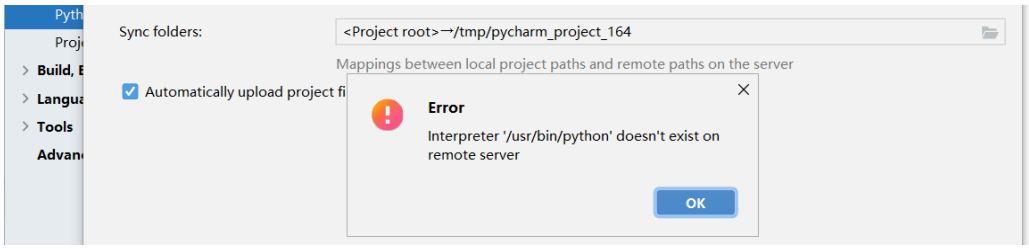
通过：tools-->Deployment-->Configurations-->Browse Remote Host，打开远端目录文件树。

之后就可以操作远程目录的文件：

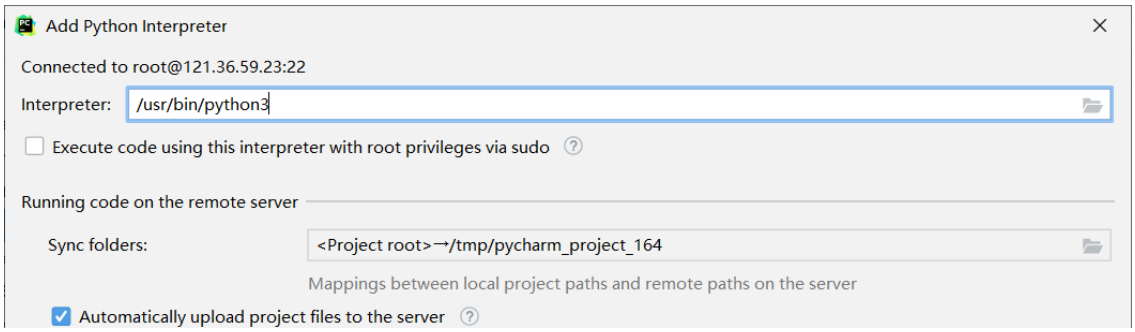


5. 问题解决

> [**ERROR#1**] 远端解释器连接失败。



一般是因为解释器路径写错了，很多系统默认装的是python3，修改解释器路径即可：



> [**ERROR#2**] 华为云的openEuler系统ssh连不上。

可能是服务器设置问题，需要修改/etc/ssh/sshd_config文件：

首先，运行如下命令

```
vi /etc/ssh/sshd_config
```

接着，我们要做如下修改

```
AllowAgentForwarding yes
AllowTcpForwarding yes
GatewayPorts yes
```

2.2.3 扩展阅读：C++远程开发实践

> 最近在学C++，想配置C++远程开发环境？最好还能调试代码？

请参考：C++从零开始（一）：环境搭建（上）之VSCode远程开发

2.3 安装相关库

> ⚠ 注意，**本实验需要你安装桌面环境**：请参考`ex0:2.1.2 可视化界面`。

先更新`pip`，否则可能会出现安装失败。

```
sudo pip3 install --upgrade pip
```

如果以下调用yum命令出错，请将以下两个文件的第一行改为`python2/ 2.7`。相关讨论可见：[issue#23](#)。

1. 修改urlgrabber-ext-down

```
sudo vim /usr/libexec/urlgrabber-ext-down
```

2. 修改yum

下面开始正式安装本次实验所需的库文件。

- 安装字体文件

  本次实验，需要wqy-microhei.ttc文件，但在`/usr/share/fonts`可能并不存在该文件。需要自行安装：

```
sudo yum install wqy-microhei-fonts
```

- 安装`jieba`

```
sudo pip3 install jieba -i https://pypi.tuna.tsinghua.edu.cn/simple
```

- 安装`wordcloud`

```
sudo pip3 install wordcloud -i https://pypi.tuna.tsinghua.edu.cn/simple
```

- 安装`pyecharts`

  > 需要指定安装1.7.0版本，否则下面代码API调用接口不正确。

```
sudo pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple
pyecharts==1.7.0
sudo pip3 install snapshot-selenium
```

- 安装驱动（易错）

  > 具体讨论可参考：@[issue#30](#) @[issue#21](#)

  `pyechars`模块保存图片需要安装相应**chromedriver和google-chrome**。

- 安装google-chrome

```
wget -O /etc/yum.repos.d/CentOS-Base.repo
http://mirrors.aliyun.com/repo/Centos-7.repo

curl https://intoli.com/install-google-chrome.sh | bash
ldd /opt/google/chrome/chrome | grep "not found"

google-chrome --no-sandbox --headless --disable-gpu --screenshot
https://www.baidu.com/
```

- 安装chromedrive

  chromedrive版本要和google-chrome对应，所以我们先查看google-chrome版本号：

```
google-chrome-stable --version
```

  记录版本号后，去https://npm.taobao.org/mirrors/chromedriver/下载对应的驱动。下载方式：

  - 推荐：`wget 下载链接` 形式，例如下载95版本的chromedriver

```
sudo rm -f chromedriver_linux64.zip*
wget
https://npm.taobao.org/mirrors/chromedriver/95.0.4638.54/chromed
river_linux64.zip
```

  - 笨一点：本地机器下载后，使用FTP工具上传到服务器

  下载后，在下载的**压缩文件所在的路径**：

```
# 1. 先删除之前的chromedriver
sudo rm -f /usr/bin/chromedriver
sudo rm -f /usr/local/bin/chromedriver
sudo rm -f /usr/local/share/chromedriver

# 2.unzip 解压
# 解压出文件chromedriver
unzip chromedriver_linux64.zip

# 3.赋予777权限
chmod 777 chromedriver

# 4.mv 移动到/usr/bin/路径
mv  chromedriver /usr/bin/
```

2.4 设置日志级别

> 由于`spark`在运行时会打印非常多的日志，为了便于调试观察，我们设置日志级别为`WARN`。

以下为全局设置日志级别方式，你也可在代码中临时设置`sc.setLogLevel("WARN")`（详见`ex3.pdf`）。

1. 切换到`conf`目录

```
cd /usr/local/spark/conf
```

2. 设置配置文件

```
cp log4j.properties.template log4j.properties
vim log4j.properties
```

修改 `log4j.rootCategory=WARN,console`

```
# Set everything to be logged to the console
log4j.rootCategory=WARN, console
log4j.appender.console=org.apache.log4j.ConsoleAppender
log4j.appender.console.target=System.err
```

3 实验流程

（已弃用）3.0 命令行下完成实验

> ⚠ **该小节编写、调试代码方式已被弃用，采用远程开发模式** ⚠
>
> `2021-12-20`：建议大家使用**远程开发**，进行代码编写，实验调试等。远程开发教程请参考：2.2节【强烈建议-远程开发】。

在实验开始之前，我们建议你按照以下流程完成实验：

1. **命令行** 下完成代码 **单元测试**
2. 单元测试无误，将代码填充在相应给出的`py`文件函数中
3. `spark-submit`方式提交代码

> ☺ 如何在命令行下完成单元测试?

1. 启动`pyspark`

```
cd /usr/local/spark
bin/pyspark
```

> ⚠ 后续实验都是在集群环境下（本次实验不需要），启动`pyspark`应该按以下方式：
>
> - 启动集群
>
> ```
> # 启动hadoop集群
> cd /usr/local/hadoop
> sbin/start-all.sh
> # 启动spark集群
> cd /usr/local/spark
> sbin/start-master.sh
> sbin/start-slaves.sh
> ```
>
> - 启动`pyspark`
>
> ```
> bin/pyspark --master spark://master:7077
> ```

2. 命令行下单元测试

例如，本次实验要求你完成`jiebaCut`函数编写：

> 完成下列指定位置编码，使得`str`为所有答案拼接而成的字符串

```
def jiebaCut(answers_filePath):
 """
```

```
    结巴分词
    :param answers_filePath: answers.txt路径
    :return:
    """
    # 读取answers.txt
    answersRdd = sc.textFile(answers_filePath) # answersRdd每一个元素对应answers.txt每一行

    # 利用SpardRDD reduce()函数,合并所有回答
    # 【现在你应该完成下面函数编码】
    str = answersRdd.reduce(lambda )

    # jieba分词
    words_list = jieba.lcut(str)
    return words_list
```

⚠ 命令行模式下，不用设置 `SparkContext、SparkSession` 实例：

```
conf = SparkConf().setAppName("ex2").setMaster("local")
sc = SparkContext(conf=conf)
```

会自动生成实例 `sc`，可直接使用！

首先定义 `answers_filePath` ，查询此前代码指定按照如下方式进行拼接：

```
# 结巴分词
words_list = jiebaCut("file://" + SRCPATH + "answers.txt")
```

```
>>> answers_filePath =
'file:///home/hadoop/Experiment/Ex2_WordCount/src/answers.txt'
```

按照流程读入文件：

```
>>> answersRdd = sc.textFile(answers_filePath)
>>> answersRdd.take(10)  # 展示前10行数据验证
```

现在你可以尝试在命令行下 **实时交互** 完成 `str = answersRdd.reduce(lambda )` 这行代码完整编写。

例如，你可以如此进行测试：

```
>>> answersRdd.reduce(lambda a,b: a+b)
'★★★更新于4个月以后★7月14日晚★★★写这个答案时，刚刚过完春节，在家被催婚心烦意乱，随手刷到，一时兴起...'
```

会实时显示交互结果，验证是否编码正确。

3. 提交代码

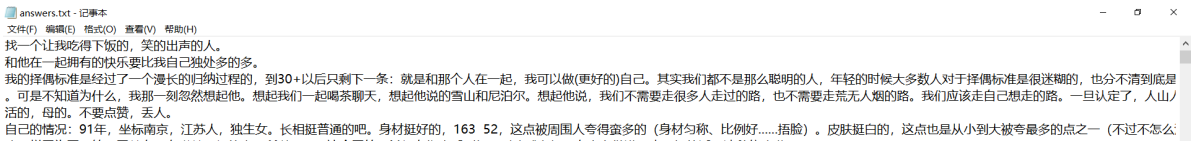
在命令行下单元测试后，便可以填写在相应 `py` 文件中。

最后通过 `spark-submit` 方式提交代码。相应如何提交，在实验后有详细介绍，这里不再赘述。

3.1 数据集介绍

> 本次实验数据集来源于2019级研究生@***W.H. Huang***，数据集包含了知乎全站 **择偶观** 相关问题下所有 *答案&作者&年龄&地区等信息* 。原**完整**数据存储于`mysql`数据库，出于简化实验数据部署等目的，本次实验**仅使用其中部分数据**以`txt`文本形式展示。

数据集 `answers.txt` 每一行代表一个完整回答，一共有3W条回答，如下图：

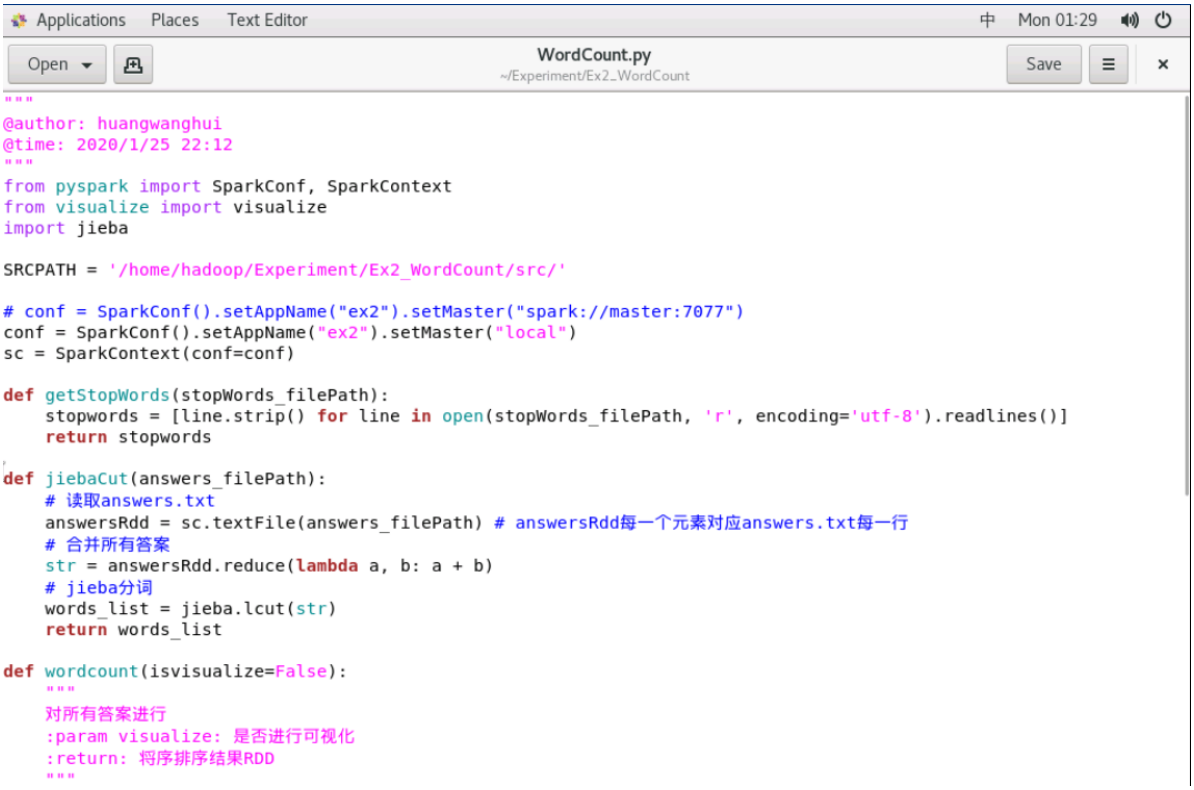


每一条回答均已去除图片、视频URL、HTML标签等。

3.2 WordCount.py

3.2.1 完成编码

你现在应该独立完成`WordCount.py`编码，可在服务器上查看：



`WordCount.py` 有`3`个函数，它们的作用如下：

- `getStopWords`： 读取`stop_words.txt`所有停用词，返回一个`python List`
- `jiebaCut`：将所有答案合并，并进行分词，返回一个`python List`
- `wordcout`：核心函数，利用`SparkRdd`完成词频统计

`jiebaCut`

完成下列指定位置编码，使得`str`为所有答案拼接而成的字符串

```
1  def jiebaCut(answers_filePath):
2      """
3      结巴分词
4      :param answers_filePath: answers.txt路径
5      :return:
```

```
    """
    # 读取answers.txt
    answersRdd = sc.textFile(answers_filePath) # answersRdd每一个元素对应answers.txt每一行

    # 利用SpardRDD reduce()函数,合并所有回答
    # 【现在你应该完成下面函数编码】
     str = answersRdd.reduce(lambda  )

    # jieba分词
    words_list = jieba.lcut(str)
    return words_list
```

`wordcout`

完成下面指定位置编码，使得`resRdd`包含所有词频统计结果，且降序排列。

打印`resRdd`前十个元素应该为如下结果，`resRdd.take(10)`：

```
[('身高', 2627), ('家庭', 2018), ('父母', 2002), ('性格', 1882), ('男生', 1640),
 ('朋友', 1618), ('条件', 1568), ('学历', 1445), ('女生', 1380), ('感情', 1301)]
```

```
def wordcount(isvisualize=False):
    """
    对所有答案进行
    :param visualize: 是否进行可视化
    :return: 将序排序结果RDD
    """
    # 读取停用词表
    stopwords = getStopWords(SRCPATH + 'stop_words.txt')

    # 结巴分词
    words_list = jiebaCut("file://" + SRCPATH + "answers.txt")

    # 词频统计
    wordsRdd = sc.parallelize(words_list)

    # wordcount：去除停用词等同时对最后结果按词频进行排序
    # 完成SparkRDD操作进行词频统计
    # 提示：你应该依次使用
    #       1.filter函数进行停用词过滤&去除长度=1的词汇
    #       2.map进行映射，如['a','b','a'] --> [('a',1),('b',1),('a',1)]
    #       3.reduceByKey相同key进行合并 [('a',2),('b',1)]
    #       4.sortBy进行排序，注意应该是降序排序
    #【现在你应该完成下面函数编码】
    resRdd = wordsRdd.filter(lambda word: ) \
                     .filter(lambda word: ) \
                     .map(lambda word: ) \
                     .reduceByKey(lambda a, b: ) \
                     .sortBy(ascending=False, numPartitions=None,keyfunc=lambda x: x[1])\

    # 可视化展示
    if isvisualize:
        v = visualize()
        # 饼状图可视化
        pieDic = v.rdd2dic(resRdd,10)
        v.drawPie(pieDic)
```

```
        # 词云可视化
        wwDic = v.rdd2dic(resRdd,50)
        v.drawWorcCloud(wwDic)
    return  resRdd
```

3.2.2 提交代码

此时主函数代码，设置可视化为`False`：

```
if __name__ == '__main__':

    # 进行词频统计，不进行可视化
    resRdd = wordcount(isvisualize=False)
    print(resRdd.take(10))  # 查看前10个
```

1. 切换到`spark`目录

```
cd /usr/local/spark
```

2. 提交代码（单机模式）

> ⚠ 如果启动了集群需要先关闭，因为**本次实验并非在集群环境下运行**：

```
cd /usr/local/hadoop
sbin/stop-all.sh
```

```
cd /usr/local/spark
sbin/stop-all.sh
```

```
bin/spark-submit /home/hadoop/Experiment/Ex2_WordCount/WordCount.py
```

3. 查看结果

你应该得到如下结果：

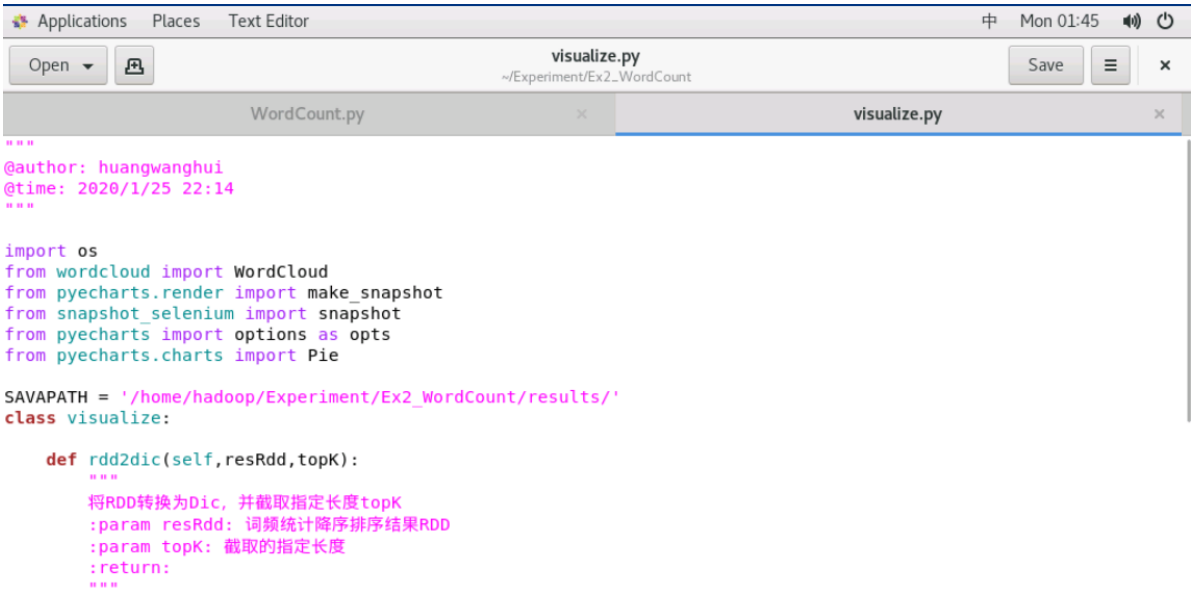
```
[hadoop@master spark]$ bin/spark-submit /home/hadoop/Experiment/Ex2_WordCount/WordCount.py
20/01/27 01:34:41 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin
-java classes where applicable
Building prefix dict from the default dictionary ...
Loading model from cache /tmp/jieba.cache
Loading model cost 1.790 seconds.
Prefix dict has been built successfully.
20/01/27 01:35:31 WARN TaskSetManager: Stage 1 contains a task of very large size (18076 KB). The maximum recomm
ended task size is 100 KB.
[('身高', 2627), ('家庭', 2018), ('父母', 2002), ('性格', 1882), ('男生', 1640), ('朋友', 1618), ('条件', 1568),
 ('学历', 1445), ('女生', 1380), ('感情', 1301)]
```

3.3 visualize.py

在本节你应该完成对`visualize.py`核心代码编写。

3.3.1 完成编码

可在服务器上查看`visualize.py`文件如下：



`visualize.py` 一共有 `3` 个函数，它们的作用如下：

- `rdd2dic`： 将 `resRdd` 转换为 `python Dic`，并截取指定长度 `topK`
- `drawWorcCloud`：进行词云可视化，同时保存结果
- `drawPie`： 进行饼图可视化，同时保存结果

`rdd2dic`

完成下面指定位置编码，将 `resRDD` 转换为 `python Dic`，并截取指定长度

```
def rdd2dic(self,resRdd,topK):
    """
    将RDD转换为Dic，并截取指定长度topK
    :param resRdd: 词频统计降序排序结果RDD
    :param topK: 截取的指定长度
    :return:
    """
    # 提示：SparkRdd有函数可直接转换
    #【现在你应该完成下面函数编码】

    resDic =
    # 截取字典前K个
    K = 0
    wordDicK = {}
    for key, value in resDic.items():
        # 完成循环截取字典


    return wordDicK
```

3.3.2 提交代码

此时主函数代码，设置可视化为 `True`：

```
if __name__ == '__main__':

    # 进行词频统计，并可视化
    resRdd = wordcount(isvisualize=True)
```

1. 切换到`spark`目录

```
cd /usr/local/spark
```

2. 提交代码

> 依旧强调，非特殊说明不要在root用户执行代码，可能会出现以下错误：
>
> “ERROR:zygote_host_impl_linux.cc (90)] Running as root without --no-sandbox is not supported.“
>
> 本次操作是在hadoop用户下。

```
bin/spark-submit /home/hadoop/Experiment/Ex2_WordCount/WordCount.py
```

3. 问题解决

- **[ERROR#1] selenium. common. exception. webdriver exception: message: chrome not reachable**

  通常是因为chromedriver程序占用了端口，控制台登陆重启服务器即可。相关讨论可见 issue#5 @lympassion

- **[ERROR#2] Operation not permitted**

  > 感谢a-fly-fly-bird 同学提供的解决方案。

  1. **jieba.cache操作不足**： PermissionError: [Errno 1] Operation not permitted: '/tmp/tmpg255ml7f' -> '/tmp/jieba.cache'

     这是因为 jieba 默认情况下在 /tmp 下存储缓存文件，然而不是 root 用户，权限不够。解决办法是修改默认缓存文件的目录，把缓存文件放在用户的目录下面。

     ```
     vim /usr/local/lib/python3.6/site-packages/jieba/__init__.py
     ```

     将self.tmp_dir 赋值为用户目录下的任意目录，如下图：

     ```
     53 class Tokenizer(object):
     54
     55     def __init__(self, dictionary=DEFAULT_DICT):
     56         self.lock = threading.RLock()
     57         if dictionary == DEFAULT_DICT:
     58             self.dictionary = dictionary
     59         else:
     60             self.dictionary = _get_abs_path(dictionary)
     61         self.FREQ = {}
     62         self.total = 0
     63         self.user_word_tag_tab = {}
     64         self.initialized = False
     65         self.tmp_dir = "/home/hadoop/cache"
     66         self.cache_file = None
     ```

  2. **render.html权限不够**： Permission denied: 'render.html'

     修改render.html权限：

     ```
     cd /usr/local/spark
     sudo chmod 777 render.html
     ```

- **[ERROR#3] 运行chromedriver生成echarts图片报错**

```
echarts_pic1
C:\ProgramData\Anaconda3\python.exe E:/phpstudy/PHPTutorial/WWW/dzyf/public/echarts_pic1.py
Traceback (most recent call last):
  File "E:/phpstudy/PHPTutorial/WWW/dzyf/public/echarts_pic1.py", line 44, in <module>
    make_snapshot(snapshot, bar_chart().render(), path1)
  File "C:\ProgramData\Anaconda3\lib\site-packages\pyecharts-1.3.1-py3.7.egg\pyecharts\render\snapshot.py", line 32, in make_snapshot
    content = engine.make_snapshot(file_name, file_type, delay, pixel_ratio, **kwargs)
  File "C:\ProgramData\Anaconda3\lib\site-packages\snapshot_selenium\snapshot.py", line 52, in make_snapshot
    return driver.execute_script(snapshot_js)
  File "C:\ProgramData\Anaconda3\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 636, in execute_script
    'args': converted_args})['value']
  File "C:\ProgramData\Anaconda3\lib\site-packages\selenium\webdriver\remote\webdriver.py", line 321, in execute
    self.error_handler.check_response(response)
  File "C:\ProgramData\Anaconda3\lib\site-packages\selenium\webdriver\remote\errorhandler.py", line 242, in check_response
    raise exception_class(message, screen, stacktrace)
selenium.common.exceptions.JavascriptException: Message: javascript error: echarts1 is not defined
  (Session info: headless chrome=75.0.3770.90)


Process finished with exit code 1
```

重新安装snapshot-selenium即可解决：

```
pip3 uninstall snapshot-selenium
pip3 install snapshot-selenium
```

4. 查看结果

   查看目录 `/home/hadoop/Experiment/Ex2_WordCount/results` ：

   可以看出：`身高、家庭、性格、父母、学历` 等是青年群体择偶最在意的几个特质。

4 扩展实验

根据第2节：在本次实验我们给予学有余力的同学，在完成本次实验的基础上提出了扩展要求。

【注】加分后总分不超过100分。

扩展要求	加分	
1. 使用分布式完成本次实验	+5	可参考ex3~ex4
2. 扩充原有数据集（100M以上），或基于新的数据集进行实验	+5~+10	根据数据量、质量、难度给分
3. 新增更多可视化，如柱状图等	+5~+10	根据可视化工作量给分
4.使用更好的算法分析数据，如应用深度学习模型Bert	+10	有对比实验更好

当然，**如果你有更好的idea**来完善更新本次实验，请联系老师或助教，我们还会考虑为你申请本年度的优秀课设（每一年都有同学通过该方式获得优秀课设）。

详情你可参考：CQU_bigdata-开源贡献。

5 实验小结

本次实验中你使用Hadoop&Spark编写了自己第一个项目，相信你一定有所收获。当然，你还体验了下远程开发的快感。

接下来的实验，你将会进一步学习在分布式集群下进行大数据分析，如：在集群执行任务、`hdfs`的使用等。同时你也将开始了解基本机器学习算法在大数据分析的应用，这将包括`Kmeans`、`SVM`等经典机器学习算法。

我们将尽量设计有趣、生动的实例来帮助你理解。最后，恭喜你完成第一个在基于`Spark`平台的大数据分析小项目，希望你从中获得了不少乐趣 :) 。