华北电力大学

课程设计报告

(2019--2020 年度第二学期)

名　　称：面向对象程序(Python)课程设计
题　　目：爬取51Job官网里的职位信息
院　　系：控制与计算机工程学院
班　　级：软件1802
学　　号：120181080701
学生姓名：邵博深
指导教师：熊建国
设计周数：1 周

成　　绩：_____

日期： 2020 年 6 月 25 日

目录

1 课程设计的目的与要求

通过课程设计的实践训练，加强学生对 Python 语言基础知识、面向对象的概念、面向对象程序设计方法的理解和掌握，提高学生综合运用所学知识分析问题、解决问题的能力。

具体如下：

1. 巩固和加深学生对计算机网络基本知识的理解和掌握 ；
2. 根据题目要求进行需求分析，确定系统的功能需求、数据需求、性能需求等；
3. 根据分析的结果进行软件设计，包括类的设计、数据设计等；
4. 采用Python语言实现系统的主要功能；
5. 撰写课程设计报告，要求内容完整、格式规范。

2 设计正文

2.1 需求分析

2.1.1 功能需求

系统要实现的基本功能包括：

（1）在51job网站上，爬取2020年发布的“Python开发工程师”的职位的薪酬，并存储到数据库中；

（2）计算北京地区该职位的平均薪酬。

2.2 网络爬虫总体设计

2.2.1 总体文件结构.

```
|--.idea
|--51job                                  # 执行文件
|    |--.idea
|    |--__init__
|    |--Model                             # 数据库文件
|    |          |--__init__.py
|    |          |--model.py               # 连接数据库，插入和查询数据
|    |          |--results.sql            # 导出来的部分数据样本（备份）
|    |--51job调用关系分析图.png             # 基于Profile的代码性能分析
|    |--ava.py                            # 计算均值
|    |--config.py                         # 配置文件
|    |--spider.py                         # 数据爬取文件
|    |--Main.py                           # 程序入口
|--venv                                   # 本次所使用的虚拟环境，基于Python 3.7
|--requirtements.txt
|--Python课设报告.pdf
```

2.2.2 总体功能设计

在本爬虫程序中共有三个模块：

（1）爬虫调度端：启动爬虫。

（2）爬虫模块：包含三个小模块，URL管理器、网页下载器、网页解析器。

1）URL管理器：对需要爬取的URL进行管理，能够从URL管理器中取出一个待爬取的URL，传递给网页下载器。

2）网页下载器：网页下载器将URL指定的网页下载下来，存储成一个字符串，传递给网页解析器。

3）网页解析器：网页解析器解析传递的字符串，并且提取有效信息。

（3）数据输出模块：存储爬取到的有效数据到数据库。

2.3 网络爬虫详细设计

2.3.1 设计环境

IDE：pycharm
Python版本：python3.7

2.3.2 爬取目标分析

根据题目，可得：
（1）初始URL："https://search.51job.com/list/000000,000000,0000,00,9,99,python%25E5%25BC%2580%25E5%258F%2591%25E5%25B7%25A5%25E7%25A8%258B%25E5%25B8%2588,2,1.html "
（2）目标数据样式：

```
<div class="el">
        <p class="t1 ">
            <em class="check" name="delivery_em" onclick="checkboxClick(this)"></em>
            <input class="checkbox" type="checkbox" name="delivery_jobid"
value="123129644" jt="0" style="display:none" />
            <span>
                <a target="_blank" title="Python开发工程师"
href="https://jobs.51job.com/beijing/123129644.html?s=01&t=0"  onmousedown="">
                    Python开发工程师                </a>
            </span>
                                                                    </p>
        <span class="t2"><a target="_blank" title="扬州万方电子技术有限责任公司"
href="https://jobs.51job.com/all/co5022996.html">扬州万方电子技术有限责任公司
</a></span>
        <span class="t3">北京</span>
        <span class="t4">1-1.5万/月</span>
```

```
<span class="t5">06-25</span>
</div>
```

（3）页面编码：gbk

2.3.3 爬虫运行流程

（1）爬虫程序模块从给定的URL开始，从给定的初始URL开始，获取所有岗位信息的页面数，然后调用URL管理器的相关方法，将所有页码添加进去

（2）URL管理器将网址信息分发给网页下载器

（3）网页下载器将页面信息传递给网页解析器

2.3.4 具体实现

（1）数据库设计

```
create table python_51job
(
    id           int auto_increment
        primary key,
    job          varchar(255) null,
    company      varchar(255) null,
    place        varchar(255) null,
    salary       varchar(255) null,
    release_time varchar(255) null
);
```

结构如图1。

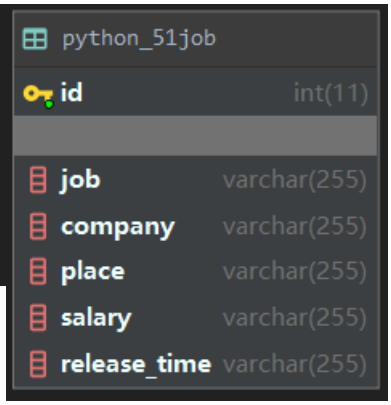

图1

（2）URL管理器

首先是从官网上获取到总页面数length，然后将数值按从1到length，存入队列。

```
def Get_Pages():
    """
    1.提取页码
    2.构造队列
    :return:
    """
    url = "https://search.51job.com/list/000000,000000,0000,00,9,99," \
          "python%25E5%25BC%2580%25E5%258F%2591%25E5%25B7%25A5%25E7%25A8%258B%25E5%25B8%2588,2,1.html "

    try:
        html = requests.get(url, headers=header, timeout=20)
        html.encoding = 'gbk'

        # 提取页码
    soup = BeautifulSoup(html.content, 'html.parser')
        td = soup.find_all('span', class_="td")
        length = re.search(r'\d+', td[0].text)

        length = int(length.group())  # 得到页码长度

    # 为下一步的多线程做准备
    # 将页号视作被竞争资源，依次存入队列中
    pages = queue.Queue(length)
        for i in range(1, length + 1):
            pages.put(i)

        return pages

    except Exception as e:
        print(type(e), ':', e)
```

（3）网页下载器

根据分析发现，51job里，URL具有如下特征，如图2：

图2

所以，只需要定向地替换html起倒数第一个数即可确定一个新的url作为下载资源的源的地址：

```
url = 'https://search.51job.com/list/00000,000000,0000,00,9,99,' \
      'python%25E5%25BC%2580%25E5%258F%2591%25E5%25B7%25A5%25E7%25A8%258B%25E5%25B8%2588,2,{}.html'.format(
    pages.get())  # 临界资源出队列

# print(url)  # 输出拼接后的url
html = requests.get(url, headers=header, timeout=20)
html.encoding = 'gbk'
```

（4）网页解析器

根据上文2.3.2对爬取目标的分析：

```
soup = BeautifulSoup(html.content, 'html.parser')
resultList = soup.find_all(name="div", attrs={'class': 'dw_table', 'id': 'resultList'})
els = resultList[0].find_all(name="div", attrs={'class': 'el'})
positions = []
for el in els[1:]:
    position = []
    t1 = el.find_all('p', class_='t1')
    t11 = t1[0].find_all('span')
    position.append(t11[0].text.strip())
    t2 = el.find_all('span', class_='t2')
    position.append(t2[0].text)
    t3 = el.find_all('span', class_='t3')
    position.append(t3[0].text)
    t4 = el.find_all('span', class_='t4')
    position.append(t4[0].text)
    t5 = el.find_all('span', class_='t5')
    position.append(t5[0].text)
    positions.append(position)
# 数据存入数据库
model.Jobs.append(positions)
```

（5）北京地区，平均工资计算

从数据库里提取北京地区的薪资(不为空白的)数据，经过单位统一换算后，直接进行计算。

2.4 性能分析

借助于pycharm的profile功能，对整个程序进行了性能分析，效果如图3所示：

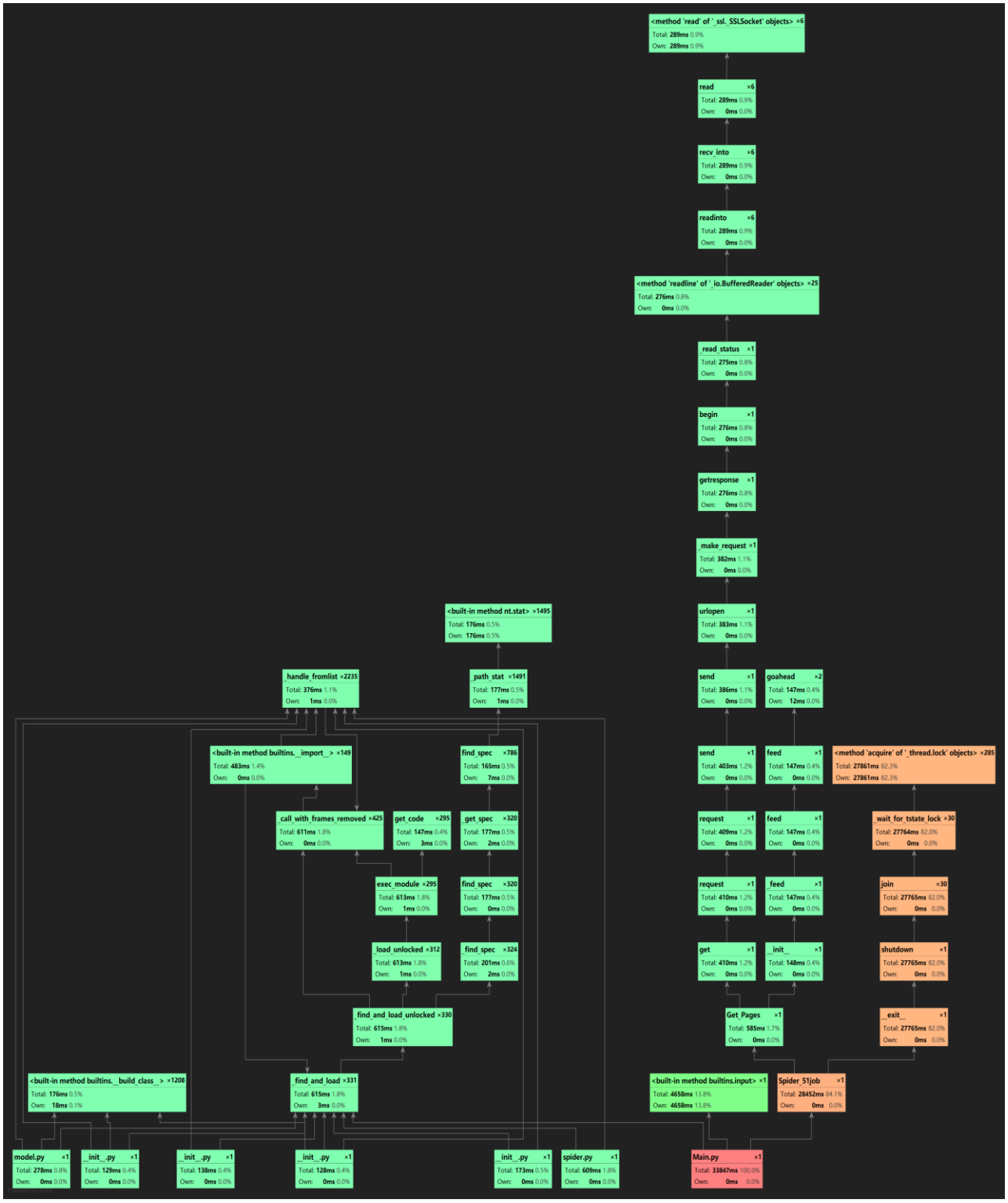

图3

}

3 课程设计总结

3.1 亮点

（1）爬取和数据库插入均采用多线程技术，从总体上缩短了时间成本，提高了CPU利用率；

（2）文件框架设计上，向商用看齐，对不同类型和功能的文件进行了分包和重构，便于后续维护和扩展；

（3）数据库设计上，采用ORM技术；

（4）在研究51job官网的过程中发现，该官网逻辑较为平易，甚至没有反爬机制。因此，本代码具有如果仍针对51job进行其他的信息爬取，仍具有很强的移植性。

3.2 不足

（1）由于是基于requests开发的，程序本身爬取速度略微逊色。

（2）从代码组织的逻辑体系上看，离商业化标准仍有一段距离。

（3）受网站本身无反爬机制的局限，未展示ip代理池、Selenium等技术。

3.3 新的认识

通过这次Python爬虫课程设计：

（1）对网页架构和Python都有了更深刻的理解；

（2）在不断调整文件组织结构的过程中，也渐渐体会到一个清晰明了的代码文件组织结构对于维护和扩展的重要性；

（3）在不断提高程序效率和性能的过程中，意识到了软件架构对于性能的决定性作用。

参考文献

[1] 钱程,阳小兰,朱福喜等.基于Python的网络爬虫技术[J].黑龙江科技信息,(36):273.

[2] 戚利娜,刘建东.基 于 Python的简单网络爬虫的实现[J].电脑编程技巧与维护,(8):72-73.

```
}
```