写在前面:

评阅老师您好~很荣幸有机会接受知乎老师们的检阅。在经过这段时间知乎对我的几轮考核下来，我对知乎的感情愈加深切，真心希望可以加入组织的大家庭中。然而当我拿到此次大作业的题目之后顿觉惊呆，因为自己对 python 相关知识接触并不多，压力甚大。然而对于知乎的热爱显然已经占领了我大脑的高地@。@,于是我决定闭关修炼发愤图强充分利用知识迁移、查阅相关书籍以及发散思维,尽自己最大的可能在短时间内完成此次任务。项目中或许有好多地方在老师看来还比较初级，因为我确实对这个接触不多，但我相信一个正确的学习方法、一个优良的学习能力加上不断勤奋的学习，是攻克一切困难的秘钥！哈哈，我会一直不停地努力学习再进步的，真心希望老师能给我一个机会，让我为知乎奉献自己一丝绵薄的力量！最后，我爱知乎！我爱开发！嘎嘎！废话不多说，直接上正文：

题目背景调研：

1. 题目要求：使用 Python 编写一个实时的新闻墙，给定指定的话题获取新浪微博（或 twitter）上对应话题的最新内容，并做到实时更新。
2. 这里根据题目，做了简短的需求分析：
    1) 根据日常使用熟悉程度，选择新浪微博作为处理对象。
    2) 经过调研，发现可以两种方式来获取微博内容：
        a) 使用新浪微博对外提供的 API--->(需要申请权限，时间关系不允许)
        b) 使用网络爬虫对微博数据进行爬取--->(综合实际情况选这种)
    3) 要求使用 Python 语言，所以选择安装了 Python 2.7.10 并选取 Pycharm Community Edition 4.5.4 作为编译器
    4) 对爬取到的数据需要一个展示平台，所以我选择用简单的 html 网页来呈现。考虑使用 tornado 框架作为 web 展现工具因为它足够轻量级并且异步非阻塞；
    5) 最后对于爬取到的数据需要一个实时的更新。目前有现有的几种实时更新的方式有：
        a) 通过传统的客户端轮训发送请求的方式进行服务器与客户端的通信

b) 基于 HTTP 长连接的长轮询技术进行服务器与客户端的通信

c) 使用 Websocket 创建全双工的消息通道进行服务器与客户端的通信

这里我选择了第三种，因为 Websocket 建立了服务器与客户端的一个全双工消息通道，这样避免再用传统方式以及长轮询等方式，不在是客户端不停地向服务器轮训发送请求查看是否有新消息，而是在新消息到来时由服务器端主动推送消息到客户端。这样避免了资源的浪费提高了效率。

程序开发说明

1. 使用语言:

   Python 2.7.10

2. Web 开发工具:

   tornado 框架+websoket

3. 编辑器: Pycharm Community Edition 4.5.4

4. 用到的第三方包:requests-2.8.1、beautifulsoup4-4.4.1 和 tornado4.2.1

5. 程序环境构建工具：buildout

   由于这次程序中依赖了一些外部的安装包，再加上之前几轮面试的沟通中了解到知乎对于 Python 环境的构建以及版本的管理使用的是 buildout。因此我查询了一些关于 buildout 资料详见 http://blog.plotcup.com/a/136，这样就可以为这次的程序构造一个完全隔离虚拟的执行环境将程序需要的包连接起来。评阅老师只需执行如下三条命令即可直接运行程序(详见 readme)：

   a) python bootstrap-buildout.py

   b) bin/buildout

   c) bin/newswall

   并在浏览器中输入 http://localhost:8888 即可看到效果，建议使用 chrome 浏览器。

项目主要实现及流程说明

1. 程序文件组织结构

   如图 1 所示，项目 MicroBlogWall 主要有三个文件构成：

1) Main.py 中是整个项目主函数的入口，通过 main 函数启动 web 服务器,以及爬虫线程。
2) Crawler.py，该文件中实现了 Crawler 类。Crawler 类负责模拟登录微博，并根据指定话题获取相应话题下的微博数据。
3) Webserver.py，该文件中实现了 MainHandler 和 SocketHandler 类：这两个类为 web 服务器所调用，用于处理浏览器的请求，并将爬取到的最新内容推送到浏览器展示

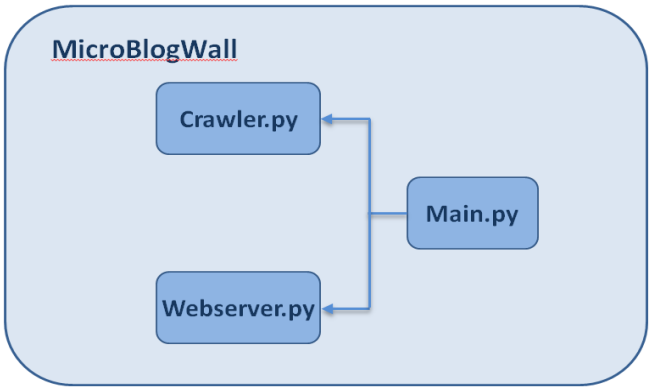

图 1 程序文件组织结构

2. 程序主要逻辑流程

图 2 展示了程序主要的逻辑流程。Main 程序作为整个程序的入口。以 main 函数为主线程调用 crawler.set_topic(topic)模拟话题切换。此外 main 函数创建了两个线程，其中线程 1 调用调用 start_web_server 启动 webserver 来对 MainHandler 和 SocketHandler 两个类进行处理，完成服务器端的逻辑。线程 2 调用 crawler.start()启动爬虫进行数据爬取的逻辑：首先进行模拟登录，其次调用 update_content 启动爬取内容，通过调用 get_topic_content 根据当前 topic 获取相应数据，并进行随机数时间的循环更新。 最后，针对每一次的更新内容，在 get_topic_content 中通过调用 SocketHandler 中的 send_news 函数进行消息推送，通过 websocket 建立的服务器与客户端的消息通道，将更新内容显示在 index.html 页面中。

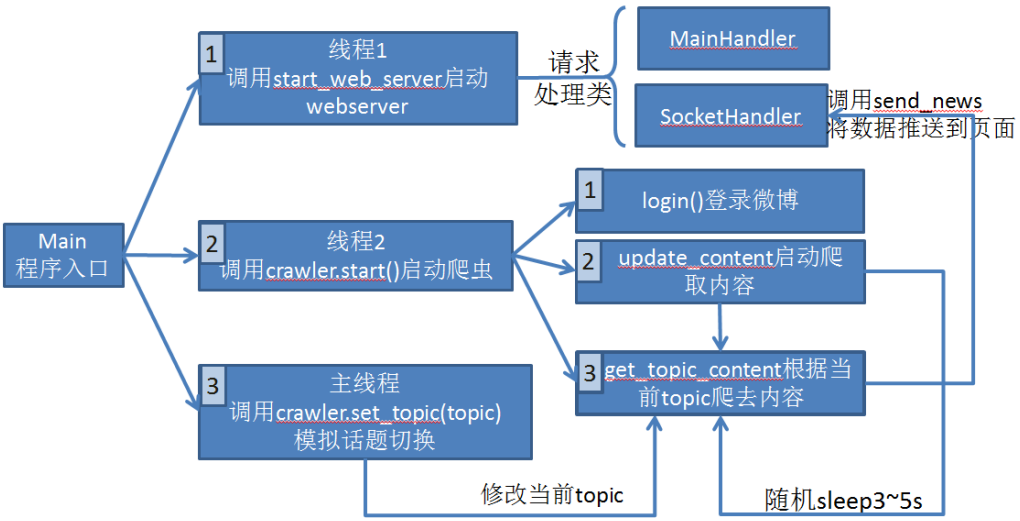

图 2 程序逻辑流程

3. 程序运行效果：

图 3 及图 4 所示是程序运行起来的截图，页面展示的是对于当前给定话题的最新的 10 条微博以及最热的 1 条微博。对于每一条微博都能够展示出：微博用户、当前话题、微博正文内容、相关链接、图片（如果有）、点赞数、转发数、评论、微博发送时间以及来源。

同时停留并保持观察页面一段时间可以发现，页面上的各类数据均能够实时根据微博上的更新进行相应的更新。

马屁入姐u:#最美背影女孩#朱自清的背影么... [组图共9张]

原图 赞[8401] 转发[1271] 评论[3213] 收藏 11月01日 19:17 来自iPhone 6s Plus

吉露琳Lu:#最美背影女孩#

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自OPPO智能手机

any要天去追边u:#最美背影女孩#可惜只是背影杀手

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自Android

那那那耳男Y:#最美背影女孩#好久没发微博了，凑个热闹~ [组图共2张]

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自三星 GALAXY S5

丹 Sara:#最美背影女孩#是一把[偷笑][偷笑][偷笑][羞嗒嗒]

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自iPhone 6s Plus

我是天蝎女王:#最美背影女孩# [组图共2张]

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自三星 GALAXY S6

茄子奋是我的小太阳u:#最美背影女孩#两年前腐的时候[拜拜][拜拜]

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自天才的iPhone 5s

Dianeer~oo:#最美背影女孩#不是所有的鱼都生活在同一片海里。 [组图共9张]

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自iPhone 5s

SPC李小鹏u:#最美背影女孩#我想静静，，，

原图 赞[1] 转发[0] 评论[0] 收藏 1分钟前 来自美图手机2

爱杰伦的茉儿u:#最美背影女孩#[doge][doge][doge]就是我。 [位置]北京市 显示地图 [组图共6张]

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自iPhone 6

我是小太阳贾贾哒:蹭个热搜[嘻嘻]#最美背影女孩#

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自iPhone 6

图 3 程序运行效果概览

i-李阿嬌:#最美背影女孩#我也来凑个热闹吧[哈哈][哈哈][哈哈] [位置]武汉市 显示地图 [组图共2张]

原图 赞[0] 转发[0] 评论[0] 收藏 1分钟前 来自iPhone 6

图 4 单条微博效果展示图

其次，程序还对输入的话题内容进行了参数检验，如果输入话题为空，或是输入一系列微博上并不存在相应内容的话题，或是乱码如”*&#”等，页面上会对用户进行相应提示：”您搜索的话题没有找到相应内容噢”，如图 5：

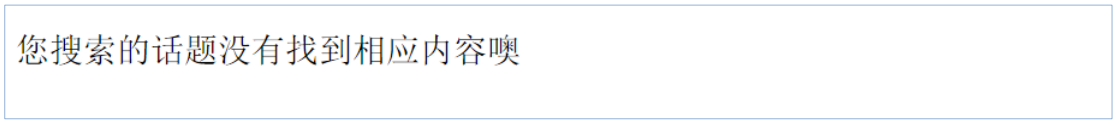

图 5 查找话题不存在相应内容示意图

4. 备注：
   1) 由于微博不同于其他普通网页，它是需要先登录才能够爬取到详细内容，所以这里先进行模拟登录，且只需一次即可；
   2) 考虑到微博更新时间的差异性，这里选择创建了一个随机数去循环获取，既可以比较高效且省资源，又避免被封号=。=

待完成的部分及可优化处理

由于时间的原因，以上是我目前完成的部分。但是综合考虑整个题目，我认为还可以从以下几个方面进行进一步完善以及优化：

1. 功能方面：
   当前只展示出相关话题最新的 10 条微博，后期考虑将增加”更多微博”的功能，如果用户感兴趣，可以爬取更多条微博并进行实时展示。
2. 交互性方面：
   1) 话题的输入可以放到页面中，让用户可以在页面中输入自己所感兴趣的话题并点击确认进行搜索。
   2) 模拟登录模块的用户名及密码的输入也可以放到页面中，变成用户自主选择任意用户名及密码的方式来登陆。
3. 测试方面：
   对于我给出的测试用例，应该进一步编写自动化测试脚本来执行这些测试用例，以提高代码的质量以及整个项目推进的效率。

小结与思考

由于完成这次题目基本是从零基础开始的，所以程序中还有很多不完善的地方，还请评阅老师多多指点。在这次完成项目的过程中，我收获了许多。

首先，在处理一个比较复杂并且没有头绪的问题的时候，我学会了将大问题切分成小问题，复杂问题简单化，寻找一个合适的切入点，进行逐轮迭代式的解

决方式。例如这次的题目，我就是按照先完成爬取任务从而拿到数据，再数据进行实时更新，最后进行网页展示的顺序完成的。在处理爬取数据的时候，我又发现直接爬取是不可行的，所以先完成了模拟登录的过程，然后在处理后面的逻辑。其他模块也都是按照这个顺序从小到大一次完成的，这样不但可以将完成的任务时间向前推进，而且可以稳步前进，减少问题出现的可能性，提高定位并解决问题的效率。

其次，我在选择完成一个问题的时候，我学会了不但要做到将功能完成，而且还要将性能进行保证。对于一个问题有多种实现方式，那么我们一定是选择占用资源最少，性能最高的方式。例如在对数据进行实时更新展示的时候，我通过查阅资料发现有很多种方式进行客户端和服务端的通信，有传统的客户端不停发送请求的方式、有基于 HTTP 长连接的长轮询技术、还有采用 Websocket 的方式。在调研的过程中我发现 Websocket 能够建立一个服务器与客户端的全双工消息通道，在新消息到来时由服务器端主动推送消息到客户端而不是客户端无休止的轮训，这样避免了资源的浪费提高了效率。由于这样的优势，成为了我选择使用它的理由。

最后，虽然这次项目对于我而言比较陌生，但正是由于这样的开始，让我能够从一个自己不太熟悉的领域慢慢入手去解决一个问题，拥有了一段珍贵的经历。在这个过程中我学会了如何去拆解题目，如何去定位需求，如何去了解一些新的技术，如何将这些知识快速定位自己所需部分并成功运用到自己的需求当中。

感谢知乎的老师，给了我这样一段难忘的经历。感谢国家，感谢知乎，感谢各位大虾~