**系统需求分析**

--24组

--姓名：葛萌萌（组长）

--学号：2017210461

--班级：2017211102

--小组其他成员：罗平

--时间：2019.06.10

目录

[1、任务概述 1](#_Toc11361607)

[2、需求规定 2](#_Toc11361608)

[3、运行环境规定 5](#_Toc11361609)

[4、尚需解决的问题 + 项目未来发展方向 6](#_Toc11361610)

# 1、任务概述

**1.1、目标**

背景：互联网时代为我们提供了海量的数据资源，而这些资源并非直接呈现，需要采用合法合理高效的手段进行提取，所以爬虫应运而生，而克服多种反扒机制以及提升爬虫性能和简易化实现爬虫项目分发部署在实际中尤为重要。用户信息也是现在的热门话题，而各大社交平台提供了海量的公开用户信息，通过挖掘分析这些信息可以发现很多有趣的事情。

因此便有了我们**PowerFul crawler**项目组的“**用户信息爬虫**”项目。

项目基本情况：专注于爬虫方面开发，爬取知乎、微博、即刻APP上的用户信息，并且进行性能优化，项目部署管理，以及简要数据分析。

**1.2、系统（或用户）的特点**

[1]系统特点：

本产品专注于对有关平台用户信息的爬取，能够爬取到大量的用户的基本信息，并着重提升爬虫的性能，之后将这些信息存储并呈现出来，最后部署成为便于使用的软件包如docker镜像，对于用户界面不会做过多着重。

[2]用户特点：

用户为对有关待爬取平台的使用人群的基本信息比较感兴趣或者在做有关平台的调查问卷的人，大量的用户信息中蕴含着很多潜在的信息可用于分析。

**1.3、假定和约束**

[1]功能约束：

本产品更关注于爬虫信息的采取，爬虫对抗反扒以及提升爬虫性能等方面，对数据的分析以及可视化着重程度会较小

。

[2]时间约束：

基本的爬取知乎、微博以及即刻app的爬取，基本的数据可视化呈现，项目的部署发布包制作等会在课程时间范围内完成，有待完善的验证码、更精细数据可视化、facebook,twitter等国外网站的用户信息爬取等拓展模块将在课程后续中去加以实现。

# 2、需求规定

**2.1、软件功能说明**

[1]爬虫功能：借助分布式，多线程技术等技术升级爬虫效率；添加cookies池，proxy池支持，并验证码破解等技巧，加强反”反爬虫”能力，从而实现知乎微博两大反扒机制较强但是有着丰富的开放合法用户信息的社交网站

[1.1]APP爬虫实现即刻APP用户信息的爬取

[2]数据处理方面：将存储到nosql中的爬取到的数据通过mongodb-charts进行可视化处理，并对于用户的关系进行存储

[3]爬取提速：使用scrapy-redis 分布式爬虫，multi-processing多进程和AIO异步io网络请求对爬虫爬取速度进行提速。

[4]项目管理：用共享云笔记协同工作、GIT多分布版本控制项目管理，然后将项目上传至docker-hub以及阿里云的docker registry

[5]项目部署：采用scrapyd+scrapyd-client进行分布式部署并且通过docker容器技术，制作docker-image实现项目发布，便捷部署。

**2.2、 用户界面**

由于对界面并没有做过多工作，此处忽略

。

**2.3、对性能的一般性规定**

2.3.1、精度

输出以及处理传输过程中的精度：

爬取到的数据应该保证有效性，可用性，以及实时性

分布式环境下保证数据无失存储到数据库中

保存到数据库中的数据应该保证其精确不易失性

2.3.2、时间特性要求

[1]爬虫时间需求，要求爬虫在有网络IO限制情况下仍然实现高速运行，基于分布式，多线程的爬虫从而实现更高的爬取速度，基于代理池，cookies池保证爬取过程阻力较小，基于超时限制及异步爬取保证网络IO延时对于性能影响尽可能小。

[2]Cookies proxy pool 提供可用proxy cookies的高效性，这关系到其内部getter,tester,interface以及scheduler的高效性。

[3]数据库访问存储高效性

2.3.3、灵活性

[1]针对不同网站使用了相同的逻辑进行用户信息采集----用户->粉丝关注列表->用户信息的迭代过程

[2]在不同平台实现灵活的部署

2.3.4、输入输出要求

爬虫

输入：对配置文件进行一定的修改/直接使用已经打包好的dockerimage作为服务启动

输出：将爬取到的数据存储到中心/本地数据库mongodb中，然后通过mongodb-charts进行可视化处理。

Cookies/proxy pool：

输入：可用的账号（通过网络可以购买获得）

输出：经过模拟登陆并且验证其验证获得的可用的cookies

输入2：制定的免费代理网站ip提取代码

输出2：经过验证器验证并进行打分排序从而返回的高分(高可用性)的代理IP

最终通过flask网络接口给爬虫提供获取cookies以及代理IP的高效接口

**2.4、数据管理能力要求**

说明需要管理的文卷和记录的个数、表和文卷的大小规模，要按可预见的增长对数据及其分量的存储作出估算。

[1]使用redis维护代理池，cookies池，以及分布式爬取指纹以及请求队列，所以规模不会很大，关键是通过高效的数据结构进行保存，从而实现快速获取数据

[2]使用mongodb保存爬取到的数据，随着爬取内容增多，将会有大量的数据，需要有结构化的存储方便进一步的分析读取利用，关键的是存入速度以及存储的结构化易读性可编程性，对于读取速度要求并不高

但对于上述二者都需要做到去重，所以在数据库接口函数的调用上需要使用update，并且在数据库存储所用的数据结构通常为set等无重复结构

**2.5、故障处理要求**

**可能出现的故障**：

[1]反扒措施导致的报错

[2]Ip被封

[3]Cookies被封（账号被封）

[4]Pool网络接口崩溃

[5]数据库IO错误

[6]自动化操作延时错误

[7]项目部署过程中出现的环境依赖不足问题

[8]APP的https无法有效抓取分析

[9]爬虫过程中可能出现的IOError UrlError HttpError等等错误

**补救措施：**

[1]基础的HEADERS伪装 包括useragent auth cookies proxy的添加

[2][3]购买大量微博账号，手机网上免费代理，构建自动检测并且动态更新获取提供可用proxy及cookies的获取接口的cookies-pool,proxy-pool，并通过检测reponse响应状态码进行proxycookies切换保证爬取顺利进行

[4]Pool网络借口由于分布式爬虫中的大量访问可能出现响应不及时甚至崩溃现象，在项目后续大规模扩展中可以更换flask为支持异步请求的tarnado实现更快的接口，并且通过nginx反向代理，实现多个API接口服务器的负载均衡技术。

[5]数据库IO错误，检查链接信息，重启数据库，查看数据库日志分析等等

[6]自动化操作中延时导致的问题，通过selenium的Waite部分实现元素加载等待机制，知道元素出现在进行操作，并且手动通过time模块设置延时实现翻页等操作中的认为等待操作。

[7]项目部署的环境依赖问题确实比较严重，但是随着虚拟化技术的发展，容器的诞生极大地简化了这个过程， 通过docker可以实现方便快捷的项目部署，妈妈再也不用担心我的环境安装了，docker在手，开箱即用。

[8]首先在抓包软件端进行设置，其次在安卓模拟器端安装CA证书，由于安卓7.0以上不支持私人安装的CA证书，所以需要使用安卓6.0以下版本，本项目考虑到APP支持的安卓版本兼容问题，使用安卓6.0版本进行

[9]用trycatch语句包装，logging以及ctx模块日志记录错误信息提取，在符合程序主逻辑下的忽略请求，重新请求处理，通过检测异常类型使用pool等方式对异常进行有效的检测处理保证程序的持续运行。

**2.7、其他专门要求**

用户需要有docker环境以及一定的docker使用基础

用户需要有分布式主机环境，通过购买云主机可以实现

用户需要有基本的链接远程主机操作的能力

总而言之：

如果用户想要使用本系统的核心—爬虫模块，或是从属模块—cookies/proxy pool，用户需要是有一定开发经验的开发人员

如果用户仅仅想要对与系统的爬取结果进行观察/分析利用，只需要连接到公开的远程数据库/使用我们最终提供的公开数据源/直接使用我们分析可视化的结果。

# 3、运行环境规定

**3.1、设备**

安装docker的主机+云主机+redis.mongodb本地存储/云存储

**3.2、支撑软件**

工程开发工具：

pycharm vscode git docker redis mongodb 数据库连接可视化工具appium charles mitmproxy Andriod\_SDK xshell Genymotion VMware VirtualBox 、

开发语言：

python3 Dockerfile + 基本的docker/git命令

工具软件的名称、版本：

Pycharm vscode 集成开发环境作为基本的编码调试运行工具

Redis mongodb 以及一些链接客户端作为数据库以及数据库操作可视化工具

Appium app 自动化操作工具

Charles mitmproxy App抓包工具

Xshell 远程ssh连接工具

Git 项目管理工具

Docker 项目发布部署工具

Genymotion PC端的安卓模拟器

Andriod\_SDK 主要借助其中adb工具进行移动端控制

VMware VirtualBox PC端的虚拟机

**3.3、接口以及协议**

协议：http+https协议

外部接口：用户只需要用安装有docker的主机即可通过pull docker-image的方式获取项目运行环境，并通过架设多台主机以及一台装有redis的主机实现分布式部署，但是过程中需要对得到的docker镜像中的项目代码中的配置文件进行简要修改，从而适应于分布式以及远程连接环境。

内部接口：

(1)数据库接口

(2)flask提供的cookies/proxy获取接口

**3.4、控制**

[1]运行方法：将scrapy部署好的项目包直接发布到运行在安装docker并且运行scrapyd的container的指定服务器上(自己的电脑)，就可以实现分布式爬虫的部署运行监控调度

[2]控制信号：使用scrapyd所提供的网络请求形式的接口/scrapyd-API所提供的函数调度接口对项目运行情况进行监控调度

# 4、尚需解决的问题 + 项目未来发展方向

[1]尚需解决优化的问题：

**数据分析可视化**

爬取更多以足够分析的数据，使用pandas,matplotlb,pandas,charts进行更加灵活全面的数据分析可视化并以web页面方式展示

**Cookies Pool完善**

补充验证码破解模块，可以通过调用远程验证码破解接口破解困难的验证码，从而完善cookies池自动获取cookies功能.

**Pool接口优化**

通过将flask网络借口改变成tarnado等异步式式web框架提高获取速度

使用nginx反向代理应对大批量分布式访问

**部署方面**

通过熟悉docker的高级使用，进而将cookies,proxy pool部署成为docker镜像， 进而整合到知乎微博的爬虫docker镜像中，并尝试实现APP爬虫的Docker部署

**分布式部署完善**

通过熟悉docker的高级使用，进而将cookies,proxy pool部署成为docker镜像， 进而整合到知乎微博的爬虫docker镜像中，并尝试实现APP爬虫的Docker部署

[2]项目未来发展方向：

**爬取方向扩展**

通过facebook+twitter提供的接口实现国外社交网站更全面用户信息的爬取

**ADSL Proxy pool**

通过拨号主机ADSL拨号申请动态IP地址，并且搭建动态DNS解析服务器，实现代理池V2.0,从而解决付费IP成本，以及免费IP不可用性等问题

**验证码破解**

当前仅通过tensorflow实现了对于字母数字这种传统验证码的CNN识别，需要拓展其他破解方法，e.g.训练RNN/CNN模型，算法破解，而不只是上传人工服务平台

**自动化爬取**

向着不局限于站点，而是通用性，自动化提取数据信息的智能化爬虫方向发展