文章编号：2096-1472(2021)-04-30-03
DOI:10.19644/j.cnki.issn2096-1472.2021.04.007

# 一种多源统一爬虫框架的设计与实现

潘洪涛

(保定电力职业技术学院，河北 保定 071051)
✉bddypht@126.com

**摘　要**：面向深层网数据的爬虫技术与反爬虫技术之间的对抗随着网站技术、大数据、异步传输等技术的发展而呈现此消彼长的趋势。综合对比当前主流的爬虫和反爬虫技术，针对高效开发、快速爬取的需求，MUCrawler(多源统一爬虫框架)被设计成一种可以面向多个网站数据源，以统一的接口形式提供爬虫开发的Python框架。测试结果显示，该框架不但能够突破不同的反爬虫技术获取网站数据，在开发效率、鲁棒性和爬取效率等方面也体现出较好的运行效果。

**关键词**：Python开发；网络爬虫；浏览器行为；HTTP请求
**中图分类号**：TP311.1　**文献标识码**：A

# Design and Implementation of a Multi-source Uniform-interface Crawler Framework

PAN Hongtao

( *Baoding Electric Power VOC. & TECH College*, *Baoding* 071051, *China*)
✉bddypht@126.com

**Abstract:** Confrontation between crawler technology for deep web data and anti-crawler technology has waxed and waned with development of website technology, big data, and asynchronous transmission technology. This paper proposes to develop a Multi-source Uniform-interface Crawler (MUCrawler) framework after comprehensively comparing current mainstream crawler and anti-crawler technologies and considering the needs of efficient development and fast crawling. MUCrawler framework can face multiple websites data sources and provide Python framework of crawler development in the form of a uniform interface. Test results show that the proposed framework can not only break through different anti-crawler technologies to obtain website data, but also show better operating results in terms of development efficiency, robustness, and crawling efficiency.

**Keywords:** Python program; web crawler; browser behavior; HTTP (High Text Transfer Protocol) request



## 1 引言(Introduction)

网络爬虫(crawler，也称spider、robot等)是面向互联网，能够通过URL(Uniform Resource Locator,统一资源定位器)自动获取Web页面数据的程序[1]。高性能的网络爬虫搜集互联网信息是搜索引擎(如Google、Baidu等)的基础。网络爬虫也是大数据和人工智能训练的一个重要数据来源，如社交网络情绪分类[2]、农业物资分析[3]或金融市场分析[4]等数据就可以采用网络爬虫从对应网站中采集。

无限制的网络爬虫可能对网站造成流量压力，因此，许多网站采用反爬虫技术对网页的自动化爬取进行限制，由此导致很多爬虫程序失效[5]。反爬虫技术的发展要求爬虫程序必须不断改进才能突破反爬虫限制获取网页内容。本文针对当前流行的爬虫和反爬虫技术进行对比分析，在综合各种爬虫技术的基础上提出了一种针对多数据源，提供统一接口的Python网络爬虫框架MUCrawler(Multi-source Uniform-interface Crawler，多源统一爬虫框架)，并针对招聘网站进行测试。

## 2 网络爬虫功能分析(Functional analysis of network crawler)

### 2.1 功能分类

依据数据存储展现方式的不同，Web网站可以分成表层网和深层网[6]，针对两种网络设计的爬虫程序也称为表层网爬虫和深层网爬虫。在当前的网络中，以静态页面为主要存储展现的网站(表层网)越来越少，更多的网站则是使用数据

库存取的动态页面、AJAX(Asynchronous JavaScript and XML，异步的JavaScript和XML)数据加载、JSON(JavaScript Object Notation，JavaScript对象简谱)数据传输的深层网，因此，深层网爬虫应用最广泛。

依据爬取Web数据的范围，网络爬虫可以分为通用型爬虫和主题型爬虫两类。通用型爬虫是对互联网所有Web信息进行遍历获取，这种爬虫主要作为搜索引擎的信息采集工具，具有全面性、高效率、高并发、海量存储等特点[7]。与通用型爬虫不同，主题型爬虫则是针对某一个或几个网站，进行特定主题信息的获取[8]。主题型网络爬虫在当前大数据分析领域应用较为广泛。

### 2.2 基本功能

网络爬虫必须能够模拟浏览器行为，针对URL自动完成HTTP请求，并能够接收服务器传回的HTTP响应信息。Web服务器响应的信息一般为HTML(Hyper Text Markup Language，超文本标记语言)、XML(Extensible Markup Language，可扩展标记语言)或者JSON等格式的数据，也可能是图像或者视频格式文件。因此，网络爬虫需要将服务器相应的信息按照语法结构进行解析，从中过滤出有用的信息。

网络爬虫还要具有迭代查找或者构造URL的功能。当前多数网站是信息存放在数据库的动态网站，网络爬虫需要通过自动表单填写和提交，使用POST方法来获取新的URL以及分析网站URL结构，也可以通过字符串拼接方式构造GET方法的URL。如果网站响应信息为HTML响应信息，也可以从页面信息中过滤“href”“src”等标签或属性获取更深层次的URL。

能够实现网络爬虫基本功能的Python扩展库包括：Urllib、Requests实现HTTP请求和响应处理，Beautifulsoup、PyQuery、lxml等实现响应文档解析。

### 2.3 扩展功能

网络数据的大规模爬取、存储和处理，要求网络爬虫除了具备上述基本功能，还需要具备并行调度、数据去重和数据存储等功能[9]。短时间内获取海量网络数据需要提高采集效率，在Robots协议允许范围内，采取多线程并发的方式是主要途径[10]。网络信息中存在大量重复的URL或冗余数据，获取这些数据不但会浪费宝贵的计算资源、带宽资源和存储资源，还会给服务器造成不必要的压力，因此，去除这些重复URL和冗余数据对于提高网络爬虫效率至关重要。侯美静等基于DOM(Document Object Model，文档对象模型)结构计算页面相似度，实现智能URL去重提高爬取效率[11]。存储爬取数据的方式有多种，如数据库存储等结构化存储、JSON等半结构化存储及文本图像视频等非结构化存储，因此网络爬虫还应具备多种存储方式的接口。

### 2.4 反爬虫技术及对应策略

数据已经成为互联网宝贵的资源，多数大型网站对自己的数据都有防范措施，即采用反爬虫技术对网站爬取进行限制。常用的反爬虫机制和应对策略如表1所示[12-13]。

表1 反爬虫机制对策表

Tab.1 Strategy of anti-crawler

| 序号 | 反爬虫技术 | 应对策略 | 可利用工具 |
|---|---|---|---|
| 1 | 对客户端发送的Headers信息进行检查，拒绝非浏览器客户端访问 | 设置HTTP request headers相关选项，如user-agent等 | Urllib、Requests |
| 2 | 用户登录后才能访问网站信息 | 设置HTTP request cookies的值，模拟登录用户 | Requests |
| 3 | 通过AJAX等技术动态加载数据 | 模拟浏览器行为运行JS代码 | PhantomJ、Selenium、ghost.py |
| 4 | 通过IP地址等限制每个客户端并发连接次数和访问频度 | 设置并发连接或者访问频次 | 限制并发度和爬取速率 |

### 2.5 爬虫框架

随着搜索引擎，尤其是大数据技术的发展，网络爬虫技术的应用越来越广泛。采用Python基本功能库(如Requests)编码实现网络爬虫，可以灵活定制爬虫功能，但开发效率较低。因此，许多组织或个人开发了网络爬虫框架作为中间件来提高开发效率[14-15]，如Scrapy、Pyspider、Crawley等，其中应用最广泛的就是Scrapy，文献[16]就是基于Scrapy设计开发就业推荐系统。

Scrapy是一个高层次的、快速开源的网络爬虫框架，用于爬取网站并从页面中提取数据。Scrapy以Scrapy engine为中心，实现发起HTTP请求、接收响应、迭代提取URL等网络爬虫功能，并通过URL列表、数据列表输出的统一调度来控制并发，提高系统效率。但是，Scrapy针对使用AJAX等动态加载数据的反爬取技术的应对策略不足，无法突破高级反爬虫技术的屏蔽。

## 3 多源统一爬虫框架(Multi-source uniform-interface spider structure)

综上分析，各种网络爬虫技术均有自己的优势和不足，尤其是针对不同的反爬虫技术，有些爬虫技术受到限制而另外一些却能突破。大型网站的结构和内容在不断地变化，针对网络爬虫所采取的反爬虫措施也在不断强化。例如，2019年针对知名招聘网站A开发的网络爬虫技术，在2020年已经失效。因此，本文结合各种爬虫与反爬虫技术开发一种面向多数据源的统一爬虫框架。

### 3.1 设计原则

周德懋等提出高性能网络爬虫应该具有可伸缩性、提高下载质量、避免下载垃圾问题的特点[9]，于成龙等还补充了礼貌爬行、并行性等特点[17]。这些特点都是本框架设计的原则，突出的主要有四点：

(1)多源通用：针对采用了各种不同反爬虫技术的网站，框架均具有适应性，且需要屏蔽采取爬虫技术底层细节，为用户提供统一的URL请求接口。

(2)提高性能：以客户端计算资源和带宽资源为基础，在框架中采用多线程网络爬虫实现并发数据采集。不同的网络爬虫技术采取不同的并发度，如Requests针对大型网站多主题数据爬取采用大量线程(线程数＞10)；针对中小型网站的多主题数据爬取采取少量线程并发(线程数≤10)；针对Selenium等模拟浏览器运行AJAX数据加载的网络爬虫则采取单线程，避

免出现错误。

(3)适度采集：网络爬虫爬取信息会挤占网站的计算资源和带宽资源，对同一网站无限制的并发大量请求连接会消耗其资源，影响正常用户的访问。因此，本框架采取“礼貌”爬取方式，限制单位时间内并发的请求连接数量以及两次请求之间的时间间隔。

(4)统一存储：用户对爬取的数据可能采取MySQL等数据库存储，也可能采取csv文件甚至文本文件存储。因此，本框架封装多种存储方式接口，调用方式统一，参数各异。

### 3.2 框架结构

本框架包括下载、分析、存储和调度四个模块，其结构如图1所示。

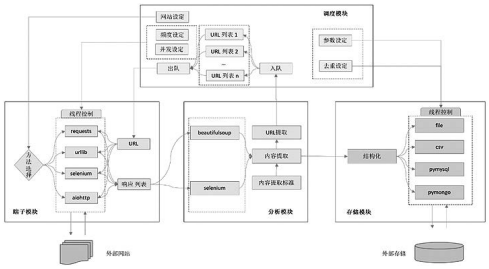

图1 MUCrawler框架结构图

Fig.1 Structure of MUCrawler framework

### 3.3 框架模块说明

(1)下载模块：下载模块是MUCrawler框架体现“多源统一”特点的模块。“多源”即多个数据源(Web网站)，这些Web网站可能采取不同的反爬虫策略；“统一”即针对不同数据源，MUCrawler封装了不同的爬取技术，如Requests、Selenium等，只需要设置网站域名(host)、对应的方法名和headers参数即可。

(2)分析模块：除了有用数据，页面包括大量的HTML标签、CSS样式、JavaScript代码等。分析模块的功能是从已经下载的页面中过滤出有用信息，主要采用Beautifulsoup、Selenium等技术，将特定标签或属性中的信息提取出来。在MUCrawler中，用户只需要设定标签或者属性与信息的对应关系，同时设置该条数据的名称，即可通过键值对的方式存入数组。

(3)调度模块：形成URL队列，通过对URL的入队和出队操作实现对URL请求进度的控制。当配置高并发的时候，会同时出队多个URL发送到下载模块来请求页面；当配置低并发或单线程时，每次只弹出少数几个甚至一个URL发送到下载模块来请求页面。通过设置两个URL请求间隔时间控制访问速度。

(4)存储模块：Web数据存储可以有多种形式，如文本文件存储、csv或excel文件存储、数据库存储等。针对这些形式，MUCrawler框架封装了文本文件存储接口，接口参数包括命名规则、文件大小限制等；csv或excel文件存储，接口参数包括命名规则、sheet设定、列名、文件大小限制等；数据库存储接口包括MySQL、NoSQL等多种，接口参数包括数据库连接参数、字段对应关系、重复数据判定字段等，封装了select、insert、delete等操作方法。

## 4 MUCrawler框架应用测试(MUCrawler application and test)

招聘类网站是典型的数据密集、更新快速的深层网网站。为了测试应用效果，MUCrawler框架针对招聘类网站进行了采集实践[15]。

### 4.1 测试环境

MUCrawler基于Python环境，软硬件配置如下：

服务器：DELL PowerEdge R210 II；CPU：Intel(R) Xeon(R)E3-1220,3.1 GHz；内存：8 GB；硬盘：1 TB；操作系统：Windows Server 2008 R2。

VMware虚拟机：CPU：1颗2核；内存：2 GB；硬盘：20 GB。

操作系统：Windows 7 professional 64 bit。

软件环境：Python 3.7.7 64 bit。

网络环境：100 MB以太网物理网络，NAT虚拟网络设置。

### 4.2 测试方案

MUCrawler框架的测试目标主要是功能测试、开发效率测试、鲁棒性测试和爬取效率测试等。为了对比框架运行效果，选择其他主流方法、库或框架进行对比。选择了三家知名的招聘网站作为测试对象，分别以L、W、Z表示。按照不同的关键字进行搜索，每个网站得到100个URL链接，将这300个链接作为测试URL库。框架测试对比项目如表2所示。

表2 测试对比项目说明

Tab.2 Instruction of test item comparison

| 测试项目 | 说明 | 标准/单位 |
|---|---|---|
| 功能测试 | 能够爬取、提取并存储对应的Web信息 | Y：能够爬取<br>N：不能爬取 |
| 开发效率测试 | 实现相同功能所编写的代码数量 | 行 |
| 鲁棒性测试 | 爬取相同数量的数据所产生的异常数量 | 次/100条记录 |
| 爬取效率测试 | 爬取相同URL数据所需要的时间(因为礼貌爬取原因，只作小数据量测试) | 秒/100条记录 |

### 4.3 测试结果

按照测试方案，功能测试针对L、W和Z网站的URL分别进行爬取，爬取功能实现结果如表3所示，“Y”代表成功爬取，“N”代表爬取失败。

表3 爬取功能实现结果

Tab.3 Result of crawler function test

| 爬取对象 | Requests | Selenium | Scrapy | MUCrawler |
|---|---|---|---|---|
| L网站 | N | Y | N | Y |
| W网站 | Y | Y | Y | Y |
| Z网站 | N | Y | N | Y |

分别采用Requests、Selenium、Scrapy和MUCrawler四种技术对三个目标网站的URL链接库进行开发效率测试、鲁棒性测试和爬取效率测试。为了便于对比，本文将测试结果进行归一化，如公式(1)所示，其中$n_i$为第$i$个网站的测试值。

$$r=\sum_{i=1}^{3}n_i\Big/\max\left(\sum_{i=1}^{3}n_i\right) \tag{1}$$

四种技术的开发效率、鲁棒性和爬取效率的测试结果经过归一化处理，均转换为0—1的小数，数值越小性能越优，结果如图2所示。

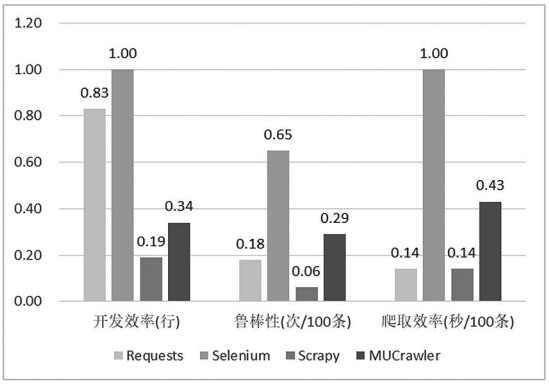


图2 开发效率、鲁棒性、爬取效率的对比测试结果

Fig.2 Test result of comparing development, robustness and crawling efficiency

### 4.4 结果分析

图2所展示的测试结果表明，在性能上Scrapy的开发效率、鲁棒性和爬取效率均为最优，其次是Requests，MUCrawler第三，而Selenium则在这些方面均处于劣势。但结合表1的功能测试，针对部分网站的反爬措施，只有Selenium、MUCrawler能够实现三个网站的Web信息爬取，且MUCrawler针对三个测试网站的爬取效率高于Selenium。

## 5 结论(Conclusion)

MUCrawler网络爬虫框架综合各种Python爬虫技术的优势，能够突破常用爬虫技术的限制实现信息的爬取。然而，MUCrawler并非Python原生开发的类库，只是基于Requests、Selenium等技术进行的二次开发，因此在爬取性能上还不能做到最优。

## 参考文献(References)

[1] BRIN S, PAGE L. The anatomy of a large-scale hypertextual web search engine[J]. Computer Networks, 1998, 30(1):107-117.

[2] 彭纪奔,吴林,陈贤,等.基于爬虫技术的网络负面情绪挖掘系统设计与实现[J].计算机应用与软件,2016,33(10):9-13;71.

[3] TIAN F, TAN H, CHENG Z, et al. Research and construction of the online pesticide information center and discovery platform based on web crawler[J]. Procedia Computer Science, 2020, 166:9-14.

[4] LIU P, XIA X, LI A. Tweeting the financial market: Media effect in the era of big data[J]. Pacific-Basin Finance Journal, 2018, 51(7):267-290.

[5] 张晔,孙光光,徐洪云,等.国外科技网站反爬虫研究及数据获取对策研究[J].竞争情报,2020,16(01):24-28.

[6] 曾伟辉,李淼.深层网络爬虫研究综述[J].计算机系统应用,2008(05):122-126.

[7] ARASU A, CHO J. Searching the web[J]. ACM Transactions on Internet Technology, 2001, 1(1):2-43.

[8] 林椹勘,袁柱,李小平.一种主题自适应聚焦爬虫方法[J].计算机应用与软件,2019,36(5):316-321.

[9] 周德懋,李舟军.高性能网络爬虫:研究综述[J].计算机科学,2009,36(08):26-29;53.

[10] BEDI P, THUKRAL A, BANATI H, et al. A multi-threaded semantic focused crawler[J]. Journal of Computer Science and Technology, 2012, 27(6):1233-1242.

[11] 侯美静,崔艳鹏,胡建伟.基于爬虫的智能爬行算法研究[J].计算机应用与软件,2018,35(11):215-219;277.

[12] 胡立.Python反爬虫设计[J].计算机与网络,2020,46(11):48-49.

[13] 余本国.基于Python网络爬虫的浏览器伪装技术探讨[J].太原学院学报(自然科学版),2020,38(1):47-50.

[14] LI J T, MA X. Research on hot news discovery model based on user interest and topic discovery[J]. Cluster Comput, 2019, 22(7):8483-8491.

[15] PENG T, HE F, ZUO W L. A new framework for focused web crawling[J]. Wuhan University Journal of Natural Sciences, 2006, 11(9):1394-1397.

[16] 陈荣征,陈景涛,林泽铭.基于网络爬虫和智能推荐的大学生精准就业服务系统研究[J].电脑与电信,2019(Z1):39-43.

[17] 于成龙,于洪波.网络爬虫技术研究[J].东莞理工学院学报,2011,18(03):25-29.

## 作者简介：

潘洪涛(1979-)，男，硕士，副教授.研究领域：网络安全，软件开发和计算机职业教育.

(上接第42页)

核算要求。本次实现中考虑到数据计算部分各航运公司的需求不尽一致，且数据计算部分是整个POC核算平台最为核心的部分，因此本设计充分从前瞻性和灵活性的角度出发，利用结构化的设计，对不同对象赋予不同算法变量。实际运行结果表明，POC核算平台能够很好地满足新会计准则的核算要求，其灵活的拓展性亦能满足新局势的发展，具备一定的推广价值。

## 参考文献(References)

[1] 朱乐明.谈完工百分比法核算在航运企业的应用[J].交通财会,2009(10):64-66.

[2] 中国财政部.企业会计准则2006[S].北京:人民出版社,2006.

[3] 张爱琴.完工百分比法应用分析及建议[J].财会月刊,2014(23):38-40.

[4] SAP. Why SAP[EB]. https://www.sap.com/why-sap.html, 2020.

[5] 龙海.SAP系统在国内成功实施的关键因素分析[J].华北电力大学学报(社会科学版),2016(05):94-98.

[6] GONSALVES, ANTONE. SAP business objects offer joint data migration services[J]. Intelligent Enterprise, 2008(11):12-15.

[7] Ahmed. ABAP development for SAP HANA[M]. Bonn: Rheinwerk Publishing, 2015:213-220.

[8] SUSHIL M, KAUSHIK R. Selection-screens[J]. Sap Abap, 2014(12):447-512.

## 作者简介：

黄　震(1977-)，女，硕士，高级讲师/高级信息系统项目管理师.研究领域：项目管理，财务信息化.