文章编号：2096-1472(2023)-04-28-05
DOI:10.19644/j.cnki.issn2096-1472.2023.004.007

基于大数据的网络数据采集研究与实践

霍 英[1]，李小帆[1]，丘志敏[2]，李彦廷[1]

(1.韶关学院信息工程学院，广东 韶关 512005；
2.韶关学院智能工程学院，广东 韶关 512005)
✉huoying@sgu.edu.cn; 14929099@qq.com; 250437325@qq.com; kidi@qq.com

摘 要：在微博大数据环境下，文章以舆情数据采集、用户行为分析为应用背景，提出了一种爬虫数据采集系统的设计与实现方案。该方案主要采用的是聚焦爬虫和增量式爬虫相结合，同时基于内容评价的爬行策略，对用户给定的关键词进行搜索，并在其发生变化时对相关内容进行更新，从而实现数据采集的及时性和有效性。通过实际数据采集效果来看，本方案单机日数据采集量约为88万条，实际应用中用户可根据需求自定义爬取数据的速度，也可通过增加分布式爬虫数量提升爬取数据量与速度。

关键词：大数据；数据采集；网络爬虫
中图分类号：TP319 **文献标识码**：A

Research and Practice of Network Data Acquisition based on Big Data

HUO Ying[1], LI Xiaofan[1], QIU Zhimin[2], LI Yanting[1]

(1.*School of Information Engineering*, *Shaoguan University*, *Shaoguan* 512005, *China*;
2.*School of Intelligent Engineering*, *Shaoguan University*, *Shaoguan* 512005, *China*)
✉huoying@sgu.edu.cn; 14929099@qq.com; 250437325@qq.com; kidi@qq.com

Abstract: In the context of Weibo big data, this paper proposes to design and implement a crawler data acquisition system based on the application background of public opinion data collection and user behavior analysis. In this solution, the focused crawler is combined with the incremental crawler, and a content evaluation-based crawling strategy is used to search for the keywords given by the user and update the relevant content with the changes of the keywords, so as to achieve the timeliness and effectiveness of data acquisition. According to the actual data acquisition effect, the daily data acquisition volume of a single machine in this solution is about 1 million pieces. In practical application, users can customize the speed of crawling data according to their needs, and can also increase the amount and speed of crawling data by increasing the number of distributed crawlers.

Keywords: big data; data acquisition; network crawler

www.rjgcz.com

1 引言(Introduction)

数据对企业经营、政府决策、社会动态分析等起着极其重要的作用，如何大规模、快速地采集数据已成为有效提取数据价值的先决条件，数据采集的效率直接决定了数据的有效性和及时性。在大数据时代背景下，如何从大数据中采集有用的信息是大数据分析至关重要的一个环节，也是大数据分析的入口[1]。

对于大多数用户提出的与主题或领域相关的查询需求，传统的搜索引擎得到的结果往往不尽如人意，为了克服传统的搜索引擎的不足，提高抓取资源的质量，面向主题的爬虫即聚焦爬虫应运而生，并且迅速成为爬虫研究的热点之一。随着动态网页技术的发展，网络爬虫遇到越来越多的困难，很多动态网页根本无法搜索到[2]，例如聊天室系统等，还有许多页面必须登录才可以查看，因此产生了智能爬虫。但是，随着人工智能、大数据挖掘等行业的兴起，普通的单机版爬虫已不能满足行业发展的要求，因此又产生了分布式爬虫，如分

基金项目：广东省哲学社会科学规划学科共建项目(GD18XXW07)；广东省自然科学基金项目(2021A1515011803).

布式爬虫框架SeimiCrawler、WebCollector、Scrapy等[3]，相应也产生了大量爬虫企业和软件，如深圳数阔信息技术有限公司的八爪鱼数据采集器、杭州云微数据信息有限公司的云微大数据、北京宏博知微科技有限公司的知微事件、北京微指数科技有限公司的微指数等。

本文在微博大数据环境下，以舆情数据采集、用户行为分析为应用背景，提出了一种爬虫数据采集系统的设计与实现方案。该方案主要采用的是聚焦爬虫和增量式爬虫相结合，同时基于内容评价的爬行策略，对用户给定的关键词进行搜索，并在其发生变化时对相关的内容进行更新，从而实现数据采集的及时性和有效性。

2　网络数据采集技术(Network data collection technology)

网络数据采集是指通过网络爬虫或网站公开API(Application Programming Interface，应用程序编程接口)等方式从网站上获取数据信息，该方法可以将非结构化数据从网页中抽取出来，将其存储为统一的本地数据文件，并以结构化的方式存储。

2.1　网络爬虫技术简介

网络爬虫[4]又称为网页蜘蛛、网络机器人，在FOAF(Friend-of-a-Friend,朋友的朋友)社区中被称为网页追逐者，是一种按照某种规则，自动抓取互联网上信息的程序或脚本；它是搜索引擎中最重要的组成模块，是舆情采集系统中最核心的部分，也是数据分析工程师必须掌握的一种工具。爬虫主要解决的问题有两个：一是如何有效获取网页内容。爬虫在运行过程中会遇到各种反爬策略，如限制访问频率、header头部信息校验、JavaScript动态生成页面、IP限制、验证码限制、登录限制等一系列技术。在解决上述问题的情况下，还必须思考如何在短时间内获得更多、更准确的信息。除此之外，在数据获取的过程中还有一系列需要遵守的协议，如robots.txt协议。二是如何对获取到的网页进行数据的提取。网络爬虫首先根据搜索的目的建立待爬取URL队列，并对这些URL所对应的网页进行访问，然后把从互联网上抓取下来的资源进行校验，从提取的新URL中获取数据，直到URL队列中的所有URL全部爬取完毕或满足一定要求为止。

2.2　聚焦网络爬虫

聚焦网络爬虫将爬取目标定位在与主题相关的页面中，主要应用在对特定信息的爬取，并为某一类特定的人群提供服务[5]。聚焦网络爬虫的爬行策略分为基于内容评价的爬行策略、基于链接评价的爬行策略、基于增强学习的爬行策略、基于语境图的爬行策略。

2.3　增量式网络爬虫

增量式更新指的是在更新的时候只更新改变的地方，未改变的地方则不更新，只爬取内容发生变化的网页或者新产生的网页，能在一定程度上保证所爬取的网页尽可能是新网页。

本文给出的爬虫数据采集系统主要采用的是聚焦网络爬虫和增量式网络爬虫相结合，同时基于内容评价的爬行策略。

3　数据采集系统设计(Design of data acquisition system)

本文以舆情数据采集、用户行为分析为应用背景，因此系统主要是通过采集用户指定话题的微博，并对数据进行初步加工后，通过业务系统展示出来，能清晰直观地体现某热点话题的传播速度，以及公众对该事件的态度。系统处理流程如图1所示。

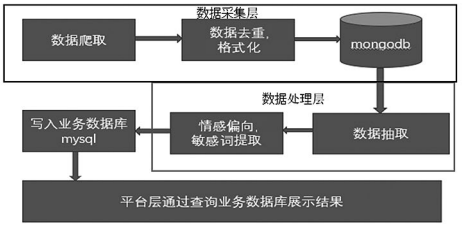

图1 系统处理流程

Fig.1 System processing flow

3.1　系统处理流程

从总体角度来看，微博舆情采集系统的主要工作流程如下：首先从微博采集数据，在获取数据后，对其进行简单的处理后[如数据结构化，HTML(超文本标记语言)标签剔除、时间格式化、去重等]存入原始数据库，然后定时从原始数据库中抽取数据并生成结论化数据，写入业务数据库并通过Web页面展示，让用户更加直观地掌握微博上的舆情动态。

由于微博平台有数以亿计的用户，每天发布的消息不计其数，想要全部获取不太可能，也没有必要[6]。因此，数据采集过程是根据用户输入的关键字进行采集，并且只把转发评论达到一定数量(评论、点赞、转发任意一项达到阈值以上)的博文存入业务数据库即可。

3.2　功能设计

数据采集系统主要是为后期业务数据分析与展示提供数据支持，由于需要经常性修改，因此不需要太过复杂的交互功能，重点是提供基本的定时任务设置、日志分析、爬虫监控功能及爬虫管理功能即可，其功能模块如图2所示。

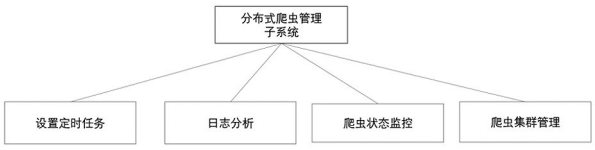

图2 系统功能图

Fig.2 System function diagram

设置定时任务：用户可自定义爬虫任务计划，系统默认24 h不间断地采集当天数据。

日志分析：可查看爬虫爬取的数据条目、爬取速度、页面请求错误、开始运行时间及最后一次请求页面的时间等信息。

爬虫状态监控：可查看各个节点的爬虫运行状态。

爬虫集群管理：可开始和终止任意一个节点的爬虫。

3.3　存储设计

由于关系型数据库插入性能较差且对存储的数据格式有要求，故爬虫爬取的数据将使用非关系型数据库Mongodb存储。Mongodb的格式灵活，插入性能高，数据之间没有耦合性，容易水平扩展，而且它不需要设计表结构，只需要指定

www.rjgcz.com

相应的集合即可，非常适合大数据爬取数据的存储。

3.4 搜索关键字功能设计

通过定时扫描业务数据库中的关键字表，获得关键字后，检索用户自定义的关键字映射文件，即存储该关键字同义词的文件。该文件需用户自行维护，并以“关键字.txt”的格式存储在爬虫所在的目录下，若未定义则默认使用该关键字。例如，“新冠.txt”文件中存储了“新冠，冠状病毒，新冠肺炎，肺炎，新型冠状病毒肺炎，Covid 19”，则使用这六个词进行搜索，若未建立该文件，则将使用“新冠”进行搜索，并根据其搜索返回的结果，进行数据的采集和分类。

4 数据采集系统实现(Implementation of data acquisition system)

4.1 技术方案

Scrapy是一个基于Python开发的Web抓取框架，通过它可以方便地从Web页面上提取结构化的数据，常常被用于数据挖掘、监测和自动化测试[7]，它良好的扩展性得到了大部分人的青睐，因为它是一个框架，所以任何人都可以根据需求修改它。

Scrapy提供了一个可通过简单的JsonAPI(应用程序接口)快速部署或控制爬虫项目的软件Scrapyd，然而其功能过于简陋，并不能满足大数据环境下实际应用的需求，因此系统实现上借用了Github上的一个基于Scrapyd的开源项目ScrapydWeb，其良好的用户交互界面，可轻松地实现管理本系统应用的爬虫项目，再在其基础上安装相应的插件、实现相关代码即可。

4.2 对目标网站信息进行分析

要对网站数据进行采集，需要先对相应的页面进行分析，由于本系统是针对微博的数据进行采集，为保证数据的完整性，本研究对微博的三个不同版本的页面[8][桌面端、移动端(触屏版)、移动端(weibo.cn)]分别进行分析，并分别采集后再进行数据合并。

4.2.1 桌面端(s.weibo.com)页面分析

Chrome等浏览器可以方便地查看页面的信息以及请求/响应信息，桌面端是微博三个版本中页面最复杂、最难爬取的页面，但是数据量是三个版本中最多的，高级查询功能可以小时为单位查询数据。

分析请求的URL(Uniform Resoure Locator，统一资源定位器)地址比对不同时间段不同页面的URL地址如图3所示，得出URL的生成规则为“https：//s.weibo.com/weibo?q=搜索关键字&typeall=1&suball=1&timescope=custom:xxxx-xx-xx-x:xxxx-xx-xx-x&Refer=g&page=页数”。

图3 微博URL分析

Fig.3 Analysis of Weibo URL

通过上文分析得出了URL的生成规则和响应页面格式，接下来可通过Xpath(XML Path Language，XML路径语言)提取数据。分析页面的HTML文件(图4) 中的内容，包含微博内容、点赞、转发、收藏、评论、原始微博(转发的微博才有)及微博ID等数据。

图4 微博搜索页HTML

Fig.4 Weibo search page HTML

主要的Xpath表达式如下。

1. region=response.selector.xpath('//span[@class="ctips"][2]/text()').extract()[0]

2. find_result=response.selector.xpath("//div[@class='card card-no-result s-pt20b40']").extract()

3. cur_page=response.selector.xpath('//div[@class="m-page"]//li[@class="cur"]/a/@href').extract()[0][-6:]

4. page_urls=response.selector.xpath('//div[@class="m-page"]//li/a/@href').extract()

5. item['weibo_id']=card.xpath(".//@mid").extract()[0]

6. user_id=card.xpath(".//a[@class='name']/@href").extract()[0]

7. item['user_id']=re.split("\?",re.split("/",user_id)[-1])[0]

8. r.lpush("weibo_spider:start_urls","https://weibo.cn/{}/info".format(item['user_id']))

9. origin_weibo_user_id=header+card.xpath(".//a[@class='name']/@href").extract()[1]

10. item['origin_weibo_user_id']=re.split("/",origin_weibo_user_id)[-1]

11. names=card.xpath(".//a[@class='name']/@nick-name").extract()

12. urls=card.xpath(".//p[@class='from']/a[@target='_blank']/@href").extract()

13. item['create_at']=card.xpath(".//p[@class='from']/a[@target='_blank'][1]/text()").extract()[0]

14. item['tool']=card.xpath(".//p[@class='from']/a[@rel='nofollow'][1]/text()").extract()[0]

15. item['create_at']=card.xpath(".//p[@class='from']/a[@target='_blank']/text()").extract()[0]

16. item['tool']=card.xpath(".//p[@class='from']/a[@rel='nofollow']/text()").extract()[0]

17. item['content']=card.xpath(".//p[@class='txt']").extract()[1]

18. like_num=card.xpath(".//div[@class='card-act']//a[@action-type='feed_list_like']/em/text()").

www.rjgcz.com

extract()[0]

19. item['like_num']=int(re.findall(r"\d+\.?\d*",like_num)[0])

20. repost_num=card.xpath(".//div[@class='card-act']//a[@action-type='feed_list_forward']/text()").extract()[0]

21. item['repost_num']=int(re.findall(r"\d+\.?\d*",repost_num)[0])

22. comment_num=card.xpath(".//div[@class='card-act']//a[@action-type='feed_list_comment']/text()").extract()[0]

提取到的数据格式如图5所示。


```
{
    "_id" : ObjectId("5e8b6441a4c13a88bcd4da64"),
    "province_info" : "湖南",
    "county_info" : "长沙",
    "weibo_id" : "4489701325271371",
    "user_id" : "5373332592",
    "origin_weibo_user_id" : "2803301701",
    "user_name" : "LuckyC1122",
    "origin_weibo_user" : "人民日报",
    "weibo_url" : "https://weibo.com/2803301701/IBCY9qrRD?refer_flag=1001030103_",
    "origin_weibo_url" : "https://weibo.com/5373332592/IBDxym7k7?refer_flag=1001030103_
    "create_at" : "2020-04-03 17:34:00",
    "tool" : "微博视频",
    "content" : "                                                【#新冠肺炎如何攻陷全世界#？】在最
    "like_num" : 0,
    "repost_num" : 0,
    "comment_num" : 0,
    "crawl_time" : "2020-04-07 01:17:53"
},
```


图5 提取的数据格式

Fig.5 Extracted data format

字段描述如图6所示。

```
class TweetsItem(Item):
    """ 微博信息 """
    weibo_id = Field()  # 微博id
    user_name = Field()  # 用户名
    user_id = Field()     # 用户ID
    weibo_url = Field()  # 微博URL
    create_at = Field()  # 微博发表时间
    like_num = Field()  # 点赞数
    repost_num = Field()  # 转发数
    comment_num = Field()  # 评论数
    content = Field()  # 微博内容
    tool = Field()  # 发布微博的工具
    province_info = Field()     # 省份
    county_info = Field()   # 市
    crawl_time = Field()  # 抓取时间戳
    origin_weibo_url = Field()  # 原始微博URL
    origin_weibo_user_id = Field()  # 原作者ID
    origin_weibo_user = Field()  # 原作者名称
```

图6 字段描述

Fig.6 Field description

4.2.2 移动端(触屏版)页面分析

触屏版页面抓包如图7所示，页面的数据是通过接口生成的，因此获取数据更方便，其返回的是json格式的数据，但是它只能返回1,000 条数据。由于提取数据方便，因此可以协助研究人员从桌面端获取数据。桌面端提取数据困难，但是在获得微博ID和用户ID的情况下，研究人员可切换为触屏版通过接口提取数据。通过分析接口可以得到用户详细信息的数据接口为“http://m.weibo/cn/profile/info?uid=用户ID”，微博详情页的数据接口为https://m.weibo.cn/comments/hotflow?id=微博ID&微博ID+&max_id=max_id&max_id_type=0，只需解析json数据便可提取数据。


```
{ok: 1, data: {,…}}
▼data: {,…}
   banners: null
  ▶cardlistInfo: {v_p: "42", containerid: "100103type=1&q=肺炎", t
  ▼cards: [{card_type: 11,…}, {card_type: 4,…},…]
    ▶0: {card_type: 11,…}
    ▶1: {card_type: 4,…}
    ▶2: {card_type: 9, is_hotweibo: 1, display_socialtitle: 1, dis
    ▶3: {card_type: 9, is_hotweibo: 1, display_socialtitle: 1, dis
    ▶4: {card_type: 11,…}
    ▶5: {card_type: 11,…}
    ▶6: {card_type: 11,…}
    ▶7: {card_type: 11, title: "实时微博", card_group: [{card_type:
    ▶8: {card_type: 9, mblog: {visible: {type: 0, list_id: 0}, cre
    ▶9: {card_type: 9,…}
```


图7 触屏版抓包信息

Fig.7 Packet capture information of touch screen version

提取的数据格式存储后如图8所示。

_id	comment_user_id	weibo_id	created_at	like_num	crawl_time	content
4487720510422228	6478740110	4487127239063129	2020-03-29 07:50:10	0	2020-03-29 07:50:49	
4487719357440170	7390061465	4487599433566429	2020-03-29 07:45:35	0	2020-03-29 07:46:18	全球趋势。
4487719025937113	2759852105	4487709114851314	2020-03-29 07:44:16	0	2020-03-29 07:47:21	就说检力的n95怎么没有了
4487718169826853	5172033980	4487709114851314	2020-03-29 07:40:52	1	2020-03-29 07:47:20	希望他们不戴口罩只是因为没有，i
4487718010568380	2491450071	4487709114851314	2020-03-29 07:40:14	1	2020-03-29 07:47:20	【#法国累计确诊新冠肺炎升至375
4487717985158145	7391366526	4487603132402807	2020-03-29 07:40:08	0	2020-03-29 07:48:38	登上美国和日本的报刊周末报道中
4487717863449304	1793836921	4487568819731365	2020-03-29 07:39:39	0	2020-03-29 07:47:20	早日调空，关门大吉，但同时更要
4487717737795773	5125039328	4487353269984050	2020-03-29 07:39:08	0	2020-03-29 07:42:13	海运直停
4487717679737997	5599620719	4487709114851314	2020-03-29 07:38:55	0	2020-03-29 07:47:21	现在开始定口罩了，肯定蒸逼的起
4487717633538597	6447255457	4487709114851314	2020-03-29 07:38:43	1	2020-03-29 07:47:21	救援是否能引起重视？
4487717595535882	5512907056	4487709114851314	2020-03-29 07:38:34	4	2020-03-29 07:47:21	希望那些经视中国的国家，端正态
4487717179902791	3088929821	4487626239208444	2020-03-29 07:36:56	0	2020-03-29 07:37:27	科比说过@艾曼佳的little brother
4487717142317064	2106293693	4487545637231417	2020-03-29 07:36:47	0	2020-03-29 07:37:27	
4487717058924217	5593273852	4487551144500919	2020-03-29 07:36:27	0	2020-03-29 07:40:25	~_~//@校晨云无心 隔离、维持性
4487717012654273	3791317392	4487545637231417	2020-03-29 07:36:16	0	2020-03-29 07:37:27	l7
4487716878366641	3306863127	4487568819731365	2020-03-29 07:35:43	1	2020-03-29 07:47:20	👏👏👏👏
4487716848794248	3012889857	4487626239208444	2020-03-29 07:35:36	0	2020-03-29 07:37:27	现在说这些大早了 还是要好好看看
4487716659863042	5326665126	4487622904581869	2020-03-29 07:34:52	0	2020-03-29 07:37:28	小碧那这么高龄
4487716278359024	6616376912	4487709114851314	2020-03-29 07:33:21	4	2020-03-29 07:47:20	10亿只嘲哄啊，为毫百姓再订购10
4487715905176882	3180363132	4487119500280522	2020-03-29 07:31:52	0	2020-03-29 07:37:19	科比泡沫有需求咋办啊？

图8 提取的评论数据

Fig.8 Extracted review data

4.2.3 移动端(weibo.cn)页面分析

移动端页面较为简单，但其用户的个人描述比触屏版更全面，因此可通过上面提取到的用户ID在移动端提取个人信息，其URL生成规则为https://weibo.cn/用户ID/info。分析HTML页面，如图9所示， 的标签里的内容即为微博个人信息内容。通过Xpath表达式"//div[@class="c"]"提取所有包含个人信息的div块，再通过正则切割字符串[9]提取其详细内容，并保存到数据原始数据库中。


```
▼<div class="c">
 "昵称:新浪广东"
 <br>
 "认证:新浪广东官方微博，新浪看点广东站官方微博"
 <br>
 "性别:"
 <br>
 "地区:广东"
 <br>
 "认证信息: 新浪广东官方微博，新浪看点广东站官方微博"
 <br>
 "简介:新浪广东频道包括广东新闻、广东美食、消费投诉、广东旅游等多个栏目，立足广东本土，有“资”有“味”！新浪广东新闻部
 &gt;&gt; (020) 8558 3938"
 <br>
 "标签:"
 <a href="/search/?keyword=%E5%B9%BF%E4%B8%9C%E8%8E%E4%BA%A&stag=1">广东达人</a>
 " "
 <a href="/search/?keyword=%E5%B9%BF%E4%B8%9C%E6%97%85%E6%B8%B8&stag=1">广东旅游</a>
 " "
 <a href="/search/?keyword=%E5%85%AB%E5%8D%A6&stag=1">八卦</a>
```


图9 移动端个人信息提取页面

Fig.9 Mobile personal information extraction page

提取到的数据格式如图10所示。


```
{
    "_id" : "7427448962",
    "province" : "北京",
    "tweets_num" : 54,
    "vip_level" : "未开通",
    "follows_num" : 124,
    "nick_name" : "微笑的小门蝴蝶",
    "fans_num" : 1,
    "birthday" : "1976-04-13",
    "crawl_time" : 1586188633,
    "gender" : "男"
},
```


图10 提取的个人信息数据格式

Fig.10 Extracted personal information data format

4.3 网络爬虫模块

网络爬虫模块主要包括以下子模块(各子模块调度流程如图11所示)。

消息队列模块：主要用来存放初始的URL链接，以及后续提取到的URL链接，供爬虫使用。

调度器模块：负责分发Request请求，并且对提取到的URL进行去重。

下载器模块：接收调度器下发的请求，从微博下载数据，即获取微博的HTML页面。

爬虫模块：接收下载器下载到的HTML页面，通过Xpath表达式和正则表达式提取网页中的数据。

中间件模块：在调度器把Request请求分发给下载器时添

www.rigczz.com

加一些信息，如UA(User Agent，用户代理)，Cookie(储存在用户本地终端上的数据)，meta等信息。

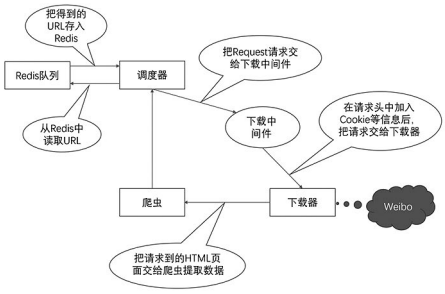

图11 爬虫各子模块调度流程

Fig.11 Scheduling process of each sub-module of the crawler

4.3.1 消息队列模块

使用Redis(Remote Dictionary Server，远程字典服务)存储需要爬取的URL，可以多个爬虫实例共享同一个消息队列，从而实现分布式爬虫，多个爬虫可以依次从队列中提取URL并记录URL是否已请求过，使调度器模块在调度时候能过滤掉重复的Requests请求，即实现不会多个爬虫爬取同一个链接。

4.3.2 调度器模块

接收从Scrapy引擎发来的请求，压入队列，当引擎再次请求的时候返回给引擎，决定下次爬取的URL链接，Scrapy原生的队列并不支持多个爬虫实例共享一个队列，当使用Scrapy-redis爬虫后，运行方式如图12所示。

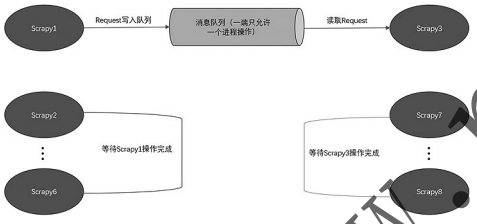

图12 Scrapy-redis获取URL链接

Fig.12 Scrapy-redis gets the URL link

4.3.3 中间件模块

位于Scrapy引擎和其他模块之间的模块，可分为爬虫中间件、调度中间件、下载器中间件。负责引擎到各个模块之间的请求和响应。这里主要实现Cookie中间件、IP(Internet Protocol，网际协议)代理中间件、UA中间件、重定向中间件(Redirect Middleware)等。

Cookie中间件：从准备好的账号池中随机选取一个账号，并把其传入Request的meta域[包含了所有本次HTTP请求的Header信息，比如用户IP地址和用户Agent(代理)]中。

IP代理中间件：从准备好的IP池中，随机选取一个IP，并把其传入Request的meta域中。

UA中间件：从准备好的UA池中，随机选取一个UA，并把其传入Request的meta域中。

重定向中间件：当Request请求响应错误代码(418，302，403)时，重新发送Request请求。

4.3.4 下载器模块

下载网页的内容(响应的HTML页面)，并将网页内容返回给Spider(爬虫)实例，其建立在名叫Twisted的高效异步模型上，这里表现为当其发出一个Request请求后，不必等待其返回的结果，而是继续发起其他请求，当请求数据被返回时，通过调用回调函数处理响应的数据，整个过程中不会出现等待返回结果的情况。

4.3.5 爬虫模块

从下载器返回的内容(请求得到的HTML页面)中提取微博评论、微博详细信息、用户个人信息和下一次请求的URL链接等数据，即Item(Items类中包含的字段)，必须在Spider目录下编写，继承于RedisSpider类，当从下载器中获得返回的内容后，通过Xpath表达式和正则表达式提取数据。

5 结论(Conclusion)

通过本数据采集系统在网络中的实际采集效果来看，微博的一个页面最多有20 条微博信息，爬虫每个页面大概能爬取19 条数据，一分钟可采集612 条数据，单机日采集量约为88万条。实际应用中，用户还可根据需求自定义爬取数据的速度，也可以通过增加分布式爬虫数量提升爬取数据量与速度。通过爬取的数据生成的结论化数据来看，也与预期结果一致。当采集的数据量进大时，在提升爬取数据量与爬取速度时，对于服务器的性能也有较高的要求，后期将在降数据规模算法方面进行进一步的优化。

参考文献(References)

[1] 崔金栋,李晨雨,李菲菲.大数据背景下主流融媒体热点发现机制研究[J].情报科学,2021,39(12):72-79.

[2] LAN Y X，LIAN Z X，ZENG R X, et al. A statistical model of the impact of online rumors on the information quantity of online public opinion[J]. Physica A: Statistical Mechanics and its Applications, 2020, 541:123623.

[3] 米硕,孙瑞彬,李欣,等.Scrapy分布式爬虫原理分析与概述[J].中国新通信,2018,20(04):234.

[4] 舒万畅.爬虫技术在大数据领域中的应用分析[J].科学技术创新,2018,36:91-92.

[5] 赵建华,蒋劲松.基于聚焦爬虫的搜索引擎的设计与实现[J].系统仿真技术,2018,14(03):221-226.

[6] 赖凯声,付宏,晏齐宏,等.地理舆情:大数据时代舆情研究的新路径[J].情报理论与实践:2020,43(8):64-69.

[7] 罗咪.基于Python的新浪微博用户数据获取技术[J].电子世界,2018,5:138-139.

[8] 雷刚,赖宇,余梦雨.基于公有云的微博网络爬虫及舆情可视化分析实验平台设计[J].实验室研究与探索,2021,40(10):273-276,294.

[9] 王晓艳,王珍珍.基于查询日志分析的中文网页关键词抽取方法[J].广西师范大学学报(自然科学版),2015,33(02):42-48.

作者简介：

霍 英(1975-)，女，博士，教授.研究领域：大数据与物联网，社会网络.

李小帆(1996-)，男，本科，工程师.研究领域：舆情处理.

丘志敏(1973-)，男，硕士，副教授.研究领域：社会网络.

李彦廷(1975-)，男，博士，副教授.研究领域：舆情处理.

www.rjgcz.com