基于 Linux 的 python 多线程爬虫程序设计*

李俊丽
（晋中学院信息技术与工程学院　晋中　030619）

摘　要　微博作为国内最受欢迎的社交平台，海量的微博数据必然包含丰富的知识资源。如何获取这些非结构化的数据，是进行微博数据挖掘的基础。根据微博网页的特点，提出了一种基于 Linux 的 python 多线程爬虫程序设计方法，通过模拟登录新浪微博，自动获取网页内容，再从网页内容中抽取微博和用户数据，以结构化的 CSV 数据格式存储或存入 MySQL 数据库，从而获取微博海量数据和用户信息。通过和基于开放 API 的爬虫程序进行比较，结果表明，从较长时间考虑，基于 Linux 的 python 多线程爬虫程序拥有更加优异的性能。

关键词　微博网页；网络爬虫；模拟登录
中图分类号　TP301　　**DOI**：10.3969/j.issn1672-9722.2015.05.026

Python Multithreading Web Crawler Program Based on Linux

LI Junli
(School of Information Technology and Engineering, Jinzhong College, Jinzhong　030619)

Abstract　Microblog is one of the country's most popular social networking platform. Vast amounts of microblog data must contain rich intellectual resources. How to get these unstructured data, it is the basis of microblog data mining. According to the characteristics of the microblogging site, this paper proposes a python multithreading crawlers method based on Linux, by simulating the login of microblog, obtaining web content automatically, and then extracting microblog and user data from the web page content. Thus CSV data in a structured format is stored or deposited in the MySQL database and then huge amounts of microblog data and user information is acquired. Through comparing with crawlers based on open API, the results show that from a long time consideration, python multithreading crawlers based on Linux have more excellent performance.

Key Words　microblog page, web crawler, simulating login
Class Number　TP301

1　引言

互联网时代的信息爆炸，“大数据”时代的到来，大量的网络信息中包含丰富的知识宝藏。社交网络[1]已经成为网络信息平台的代表，其用户数据极其有价值。而新浪微博[2]作为国内最受欢迎的社交网络平台，具有很好的及时性、发布信息快速，信息传播的速度快等优势，大量的微博数据[3]需要数据挖据[4~6]以得到有用的信息。文中主要研究如何设计一个高并发、高性能、功能齐全，健壮和可靠的多线程微博网络爬虫，并在基于 Linux 操作系统上使用 Python 编程语言[7~8]来实现。

2　相关概念

2.1　网络爬虫

网络爬虫[9]（Web Crawler），是一个非常强大的自动提取 Web 网页的应用程序，它为搜索引擎从互联网上下载 Web 页面，是搜索引擎的重要组成部分。爬虫从一个或多个初始页面的 URL，通过分析源文件的 URL，提取新的 web 链接，通过这些链接，然后继续寻找新的链接，如此不断循环，直到抓取和分析所有页面。当然这是理想情况下的执行情况，但事实上是不可能抓取互联网上所有的网页。根据现在公布的数据，最好的搜索引擎也只

* 收稿日期：2014 年 11 月 17 日，修回日期：2014 年 12 月 29 日
作者简介：李俊丽，硕士，讲师，研究方向：数据库与软件工程技术。

爬取整个互联网 40%的网页。

2.2 模拟登录

不同于传统 web 网站,以前的网站不需要登录,现在的社交网站进入个人主页需要登录,不登录访问将会跳转到登录页面。所以需要设计一种社交网站爬虫程序。该程序支持登录,而且可以获取大量用户的信息。我们采用 Session 机制来解决。Session 通过 Cookie 和 URL 重写来实现用户登录。通过 cookie 可以实现 Session 会话,但如果客户禁用了 Cookie,那么可以使用 URL 重写。

2.3 python 多线程

Python 使用线程主要有两种方式:用函数或使用类来包装线程对象。使用函数实现多线程:线程模块的 start_new_thread()函数来创建一个新的线程。用类包装线程对象:通过创建线程子类来包装一个线程对象。

3 基于 Linux 的 python 多线程爬虫程序设计

设计主要包括两大模块,分别是模拟登录模块和抓取微博数据模块。模拟登录模块是基础,不成功登录微博,没有权限查看用户主页,个人信息和关注页面,只有成功登录,才能爬取信息。

3.1 模拟登录模块

为了增加用户的数量和功能需求,在访问新浪微博用户主页时都需要用户登录验证,只有登录用户才可以查看页面内容,或者跳转到登录页面要求用户必须登录。而微博为保护自己的数据,避免不必要的资源消耗,设置了复杂的登录方法故意采取措施防止爬虫访问。通过抓包分析和查看新浪通行证 url(http://login.sina.com.cn/signup/signin.php)的源代码,你可以找到新浪微博登录 js 加密文件(http://login.sina.com.cn/js/sso/ssologin.js),登录过程包括访问干扰如获取随机参数和多个加密方法,新的登录过程中需要重复请求服务器以完成登录过程,在这个过程中也对密码进行了多次 RSA 加密[10]和 BASE64 加密[11]。这使得模拟登录[12]很困难。模拟登录步骤如下:

第一步,从配置文件得到已经配置好的用户名和密码。

第二步,添加自己的 username,提取希望的 servertime nonce,pubkey 和 rsakv 值。当然可以将 pubkey 和 rsakv 值写死在代码中,这是一个固定值。

第三步,BASE64 加密用户名,而密码需要首先创建一个 RSA 公钥,公钥的两个参数,微博给了固定的值,但都是十六进制字符串,首先是第一步登录的 pubkey,第二个是 js 加密中的'10001',这两个值需要从十六进制转换为十进制,但也可以写死在代码中。再次使用 RSA 公钥对拼接了 servertime 和 nonce 值的字符串进行 RSA 加密来获取一个中间密码,最后中间密码转为十六进制,获得真正的密码字符串。

第四步,前三步得到的 servertime nonce,rsakv、su、sp 值填充请求参数,其余的请求参数值作为常量,例如:"entry":"weibo","gateway":"1","savestate:"7"、"userticket':'1','ssosimplelogin':'1','VSNF":"1"、"service":"miniblog"、"pwencode':'rsa2"、"编码":"utf-8"、"prelt":"115",然后请求头文件分配 headers={"User-Agent":"Mozilla/5.0(8.0 X11,Linux i686:rv:8.0)Gecko/20100101 Firefox 8.0 Chrome//20.0.11 Safari/536.11"},最后通过 http://login.sina.com.cn/sso/login.php?Client=ssologin 以 POST 方法。Js(v1.4.4)发送请求,从返回的数据中提取 location.replace(*)中的*指代的 URL。

第五步,上一步得到的 URL 使用 GET 方法发送请求给服务器,保存请求的 Cookie 信息,这就是需要的登录 Cookie。在这一点上,完全可以模拟正常登录的过程,保存登录成功的 Cookie。

3.2 微博数据爬取模块

通过模拟登录成功登录微博,通过线程调度,获得微博博主 ID,访问主页(例如:http://weibo.com/1197161814/profile),可以分页查看用户所有的微博,通过修改请求参数,反过来,下载包含所有页的微博用户数据。微博数据包括消息 ID、消息内容、消息@看到的用户数量、转发消息、消息评论的数量、新闻发布时间等。微博数据爬取步骤如下:

第一步,当前线程调度微博博主 ID,首先创建一个存储新浪微博数据的数据结构,本系统采用 Python 字典数据结构的基本数据类型,它包含多个键值对的数据结构,在创建后,初始化每个键的初始值,部分使用默认值,如消息的转发数量,新闻评论的数量的初始值为 0。

第二步,读取配置文件的配置信息,配置信息包括抓取数据的 URL(用户主页添加一些请求参数),提取结果是否格式化 needExtract,下载网页最大的数量,数据存储格式和存储路径(CSV 文件路径和 MySQL 连接参数),需要提取的名称字段,

字段值在一个 web 页面位置信息和正则表达式的正则提取规则。位置信息包括前标签 bIdent、后标签 eIdent、前忽略长度值 bOffset、后忽略长度值 eOffset 等。读取后的配置文件数据以含有配置信息属性的类的数组进行存储。

第三步，下载第二步获取的待爬取数据的 URL 页面数据，需要修改参数以下载所有分页。在下载的过程中，允许多次尝试，最大数量的尝试在第二步中通过配置文件访问已经获取、下载后的页面数据存储以字符串格式存储。

第四步，第三步获得的使用正则表达式匹配的字符串或使用字符串查找与截取的方法来提取数据，正则表达式规则或字符串查找和截取规则已经在第二步中通过配置文件访问，通过遍历配置文件数据，依次读取每个需要爬取的字段名称和提取规则。提取微博新闻内容，比如屏幕姓名在配置文件中得到常规值的正则表达式提取规则是空的，也就是说，这个字段并不是通过正则表达式匹配提取，而是通过字符串搜索和截取方法提取，屏幕前的配置文件名称前标签 bIdent 值为 node－type＝“feed_list_content”〉，后标签 eIdent 值为〈 \ / div 〉，忽略 bOffset 前长度值为 0，忽略 eOffset 后长度值 0，提取结果是否格式化 needExtract 值为 0，所以在页面中找到顺次相邻的 bIdent 值和后标签 eIdent 值，截取二者之间的字符串，结果不进行格式化处理，然后将结果保存到第一步初始化的字典数据结构中。

第五步，在第四步中，顺次抽取配置文件中配置的字段数据，初始化的第一步字典数据结构的每个字段都被赋值，现在需要把这些数据存储，首先以正则表达式校验获得的微博数据数据是否符合数据格式要求，然后根据第二步通过配置文件数据存储格式和存储路径（CSV 文件路径和 MySQL 连接参数）将微博数据以结构化的 CSV 格式存储或存入 MySQL 数据库。

第六步，完成第五步，微博消息数据提取完成，这个时候需要确定是否还有微博页面数据，如果有的话，返回到第四步提取下一条微博数据，若无则需要判断当前页是不是该用户的最后一页微博数据，若不是，则返回第三步，下载下一页微博数据页面，若是，则需要通过主线程的调度，获得下一个微博博主的 ID，若成功获取，则返回第一步，若主线程已经将所有微博博主的 ID 分配完，当前线程结束。

4　性能对比

4.1　基于开放 API 的爬虫程序

基于开放 API 爬虫，理论上是具有最优性能的程序，它直接调用 API 底层直接连接新浪微博的数据库，获得数据效率很高。但是存在一个致命的弱点，新浪微博开放的 API 对开发人员访问频率有限，为第三方开发者普通授权针对一个服务器 IP 的请求次数限制是 1000 次/小时。这样一个低数量使基于开放的 API 部分程序只能在间隔的片段时间内处于数据爬取状态，其余的时间处于闲置状态。与活跃的时刻基于开放 API 爬虫程序为基准，衡量基于 Linux 多线程的 python 爬虫的性能，可以非常直观的评价其性能。

4.2　性能评测

基于 Linux 的 python 多线程爬虫程序和基于开放 API 爬虫分别在同一环境中抓取微博数据，都开启 12 个线程进行爬取，给定相同的微博博主 ID 列表，且数据存储在相同的 MySQL 数据表中。统计数据库中爬取时间字段，获得二者爬取微博数随时间变化如图 1 所示。

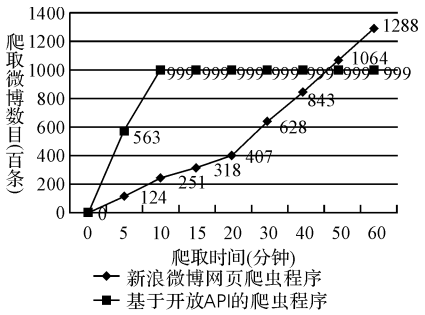

图 1　微博数据爬取性能比较

由于新浪微博的限制，基于开放 API 的爬虫程序每次请求最多获取 100 条微博，且请求次数限制是 1000 次/小时，所以在基于开放 API 的爬虫程序运行的前一小时最多获取 100000 条微博数据。从图 1 可以看出，基于开放 API 的爬虫程序在运行 10～15 分钟的某个时间点后达到请求次数限制，从那个时间点开始，其爬取的微博数保持在 99900 多次不再增加。之所以未达到 100000 次是因为 1000 次的请求大概有 1 次未收到服务器的响应。而基于 Linux 的 python 多线程爬虫程序在运行 40～50min 的某个时间点后，爬取的微博数量超过 100000 条，且之后仍不断增加。

5　结语

通过两种爬虫程序的性能评测，基于开放 API

（下转第 876 页）

[4] 董方鹏.织女星网格中的资源发现机制研究[D].沈阳:中国科学院计算技术研究所,2006.
DONG Fangpeng. Research on the Resource Discovery Mechanism in VEGA Grid[D]. Shenyang: Institute of Computing Technology, Chinese Academy of Science,2006.

[5] P. Mockapetris. Domain names-concepts and facilities. RFC 1034,1987.

[6] R. Raman, M. Livny, M. Solomon. Matchmaking: distributed resource management for high throughput computing[C]//Proceedings of the 7th IEEE Symposium on High Performance Distributed Computing(HPDC-7),1998.

[7] 谭树斐.网格环境资源发现机制的研究[D].长沙:中南大学,2005.
TAN Shufei. Research on the Resource Discovery Mechanism in Grids[D]. Changsha: Central South University,2005.

[8] 龚奕利.分布式环境中的资源发现研究[D].沈阳:中国科学院计算技术研究所,2006.
GONG Yili. Research on the Resource Discovery in Distributed System[D]. Shengyang: Institute of Computing Technology, Chinese Academy of Science,2006.

[9] 陈敏.OPNET 网络仿真[M].北京:清华大学出版社,2004.
CHEN Min. OPNET Network Emulation[M]. Beijing: Tsinghua University Press,2004.

[10] 王文博,张金文.OPNET Modeler 与网络仿真[M].北京:人民邮电出版社,2003.
WANG Wenbo, ZHANG Jinwen. OPNET Modeler& Network Emulation[M]. Beijing: Posts & Telecom Press,2003.

(上接第 863 页)

的爬虫程序有请求次数限制,达到请求次数限制后,其爬取的微博数就不再变化,而基于 Linux 的 python 多线程爬虫程序无此限制,可以 24 小时不间断地采集数据。因此,从较长时间考虑,基于 Linux 的 python 多线程爬虫程序拥有更出色的表现。

参 考 文 献

[1] 郭涛,黄铭钧.社区网络爬虫的设计与实现[J].智能计算机与应用,2012,2(4):65-67.
GUO Tao, HUANG Mingjun. Design and Implementation of the Social Network Crawler[J]. Intelligent computer and Applications,2012,2(4):65-67.

[2] 韩英.浅析大数据时代的数据挖掘与精细管理[J].成都航空职业技术学院学报,2013(4):63-64.
HAN Ying. Analysis of Data Mining and Fine Management in Age of Big Data[J]. Journal of Chengdu Aeronautic Polytechnic,2013(4):63-64.

[3] Pieter N, Michiel H. Mining Twitter in the cloud: A casestudy[C]//Proceedin gs of the 2010 IEEE 3rd Internat ionalConference on Cloud Computing, CLOUD 2010. Miami, USA: IEEE Computer Society, 2010: 107-114.

[4] 廉捷,周欣,曹伟,等.新浪微博数据挖掘方案[J].清华大学学报:自然科学版,2011,51(10):1300-1305.
LIAN Jie, ZHOU Xin, CAO Wei, et al. SINA microblog data retrieval[J]. Journal of Tsinghua University, 2011,51(10):1300-1305.

[5] Boldi P, Codenotti B, Santini M. UbiCrawler: A scalable fully distributed web crawler[J]. Software: Practice & Experience,2004,34:711-726.

[6] 柴化磊.分布式环境下基于文本的海量数据挖掘[D].上海:上海交通大学,2013.
CHAI Hualei. Document-Oriented Massive Data Mining under Distributed Environment[D]. Shanghai: Shanghai Jiaotong University,2013.

[7] Alex Martelli, Anna Ravenscroft, David Ascher. Python Cookbook[M]. USA: O'Reilly Media, Inc,2005.

[8] Mark Lutz. Learning Python[M].北京:机械工业出版总社,2009.

[9] 周立柱,林玲.聚焦爬虫技术研究综述[J].计算机应用,2005,25(9):1965-1969.
ZHOU Lizhu, LIN Ling. Survey on the research of focused craw ling technique[J]. Computer Application, 2005,25(9):1965-1969.

[10] 谢会娟,韩昌豪,吴明珠.RSA 加密算法的有效实现及在云计算中的应用[J].电脑知识与科技,2014,10(14):3263-3265.
XIE Huijuan, HAN Changhao, WU Mingzhu. Analysis to effective realization of RSA Encryption algorithm and its application in the cloud computing[J]. Computer Knowledge and Technology,2014,10(14): 3263-3265.

[11] 罗江华.基于 MD5 与 Base64 的混合加密算法[J].计算机应用,2012,32(S1):47-49.
LUO Jianghua. MD5-Base64 based hybrid encryption algorithm[J]. Journal of Computer Applications, 2012,32(S1):47-49.

[12] 孙青云,王俊峰,赵宗渠,等.一种基于模拟登录的微博数据采集方案[J].计算机技术与发展,2014,24(3):6-10.
SUN Qingyun, WANG Junfeng, ZHAO Zongqu, et al. A Microblog Data Collection Method Based on Simulated Login Technology[J]. Computer Technology and evelopment,2014,24(3):6-10.