

Larbin 体系结构的研究与优化

王璇^{1,2}, 霍义霞¹, 慈云飞¹, 史国振¹, 李莉^{1,2}

(1. 北京电子科技学院信息安全系, 北京 100070;

2. 西安电子科技大学计算机学院, 陕西 西安 710000)

摘要: 网络爬虫是搜索引擎的重要组成部分, 其性能直接影响搜索引擎的准确性和及时性。Larbin 是一个高效、简单、功能比较完善的开源爬虫框架, 基于此, 介绍了几种典型的开源爬虫框架, 并对其进行多维度比较; 对 Larbin 体系结构进行详细的介绍; 然后指出 Larbin 在程序结构和流程方面存在的不足, 提出对应的优化方案; 测试结果表明, 改进后的方案在速度和性能方面都有所提高。

关键词: 搜索引擎; 网络爬虫; Larbin; 开源; 优化

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2016.00076

Study and optimization on system architectures of Larbin

WANG Xuan^{1,2}, HUO Yi-xia¹, CI Yun-fei¹, SHI Guo-zhen¹, LI Li^{1,2}

(1. School of Information Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China;

2. School of Computer, Xidian University, Xi'an 710000, China)

Abstract: Web crawler is an important part of the search engine, its performance will directly affect the accuracy and timeliness of the search engine. Larbin is an efficient and simple open source crawler with relatively perfect in functions. Several typical open-source crawler were firstly introduced and a multi-dimensional comparison was made among them. Then, the system architecture and working mechanism of Larbin were given in detail. Its shortcomings in the program structure and process were pointed out, and improved programs were proposed. Experimental results show that improved program is better in speed and performance.

Key words: search engine, Web crawler, Larbin, open source, optimization

1 引言

随着信息技术和网络技术的高速发展, 互联网信息规模呈爆炸式增长, 内容呈现杂乱无章, 其中包括文本、视频、音频、图片等非结构化数据。为了使用户方便快捷地获取各种各样的目标信息, 搜

索引擎^[1]应运而生。数据收集, 即如何获取互联网数量庞大的信息是搜索引擎的技术基础。目前网络爬虫是公认的完成网络信息自动采集任务最有效的工具, 其本质是一种按照特定的搜索策略和算法自动抓取互联网信息的程序^[2]。网络爬虫的性能将直接影响搜索引擎查找信息的及时性和准确性^[3]。

收稿日期: 2016-05-12; 修回日期: 2016-08-02。通信作者: 李莉, lili103@besti.edu.cn

基金项目: 国家重点研发计划基金资助项目 (No.2016YFB0800304); 北京市自然科学基金资助项目 (No.4152048); 江苏省自然科学基金资助项目 (No.BK20150787); 北京电子科技学院 2016 年春蕾计划基金资助项目 (No.2016CL04)

Foundation Items: The National Key Research Program of China (No.2016YFB0800304), The Natural Science Foundation of Beijing (No.4152048), The Natural Science Foundation of Jiangsu Province (No.BK20150787), 2016 Spring Buds Project of Beijing Electronic Science & Technology Institute (No.2016CL04)

Larbin 是一个高效、简单、功能多样化的开源的网络爬虫框架,拥有完整的体系结构和功能设计,适合在此基础上进行二次开发。虽然 Larbin 的创始人在 2003 年就停止对其维护和升级。但是,目前仍有大量的人在研究和用它。例如,文献[4]为满足抓取微博的需求,在 Larbin 的基础上提出动态爬取、定时采集和代理采集的方案。文献[5]改进了 Larbin 中 URL 去重算法,使 URL 的冲突率由原来的 10%降低到 0.82%。为满足抓取企业内网的需求,文献[6]对 Larbin 进行功能扩展,实现了站点自动登录、保存网页、过滤无效 URL 以及网页消噪的功能。文献[7]对 Larbin 进行主题化改进,使其成为主题爬虫框架 TLarbin。

2 开源爬虫框架

本节首先介绍了几种社区活跃度较高的网络爬虫框架,并针对它们的特点进行多维对比。Larbin 是活跃度较高的开源爬虫框架之一,在研究 Larbin 的基础上,将对其体系结构进行详细的介绍。

2.1 开源爬虫框架比较

目前开源的网络爬虫框架比较成熟,在商业上取得较广泛的应用。其中,社区活跃程度较高的有 Larbin、Nutch、Heritrix、Scrapy、Pholcus、Snoopy 等。下面对 Larbin、Nutch、Heritrix、Scrapy 这 4 种常用的爬虫框架进行比较。

Nutch 是基于 Lucene 的 Apache 的子项目^[8],使用 Java 语言开发,基于 Hadoop 的分布式处理,用户可以方便地根据自己的需求进行定制。Larbin 是一种 C++的开源爬虫,有其独特的亮点和作用,其高效性和高度可配置性为想用 C++研究爬虫的人们提供了很好的平台^[9]。Heritrix 是 SourceForge 上基于 Java 的开源爬虫,提供友好的 Web 界面、高度可扩展性,深受搜索引擎研究者的青睐^[10]。Scrapy 是用 Python 开发的一个开源

Web 爬虫框架,用于快速抓取 Web 站点并从页面中高效提取结构化的数据,但是其通用性差^[11]。开源爬虫框架对比如表 1 所示。

从表 1 可以得出:1) 同一时间打开多个连接,使 Larbin 相对其他爬虫框架在速度上具有明显的优势^[12],但是不支持分布式;2) 可以对网页进行镜像保存,作为搜索引擎的数据来源;3) 具有高度可配置性。总之, Larbin 是一个小型的爬虫框架,其针对小型爬虫具有良好的工作效率,不适合大型分布式抓取。

2.2 Larbin 的设计结构

Larbin 符合主流网络爬虫的设计思路,拥有完整的体系结构和功能设计。Larbin 采用模块化的程序设计,各功能模块划分清晰,流程如图 1 所示。根据图 1, Larbin 可分为以下几个模块:内存队列模块、DNS 解析模块、连接处理模块、解析 robots 文件、HTML 文件处理模块、URL 处理模块。下面根据一个 URL 抓取网页的过程介绍各模块间的衔接关系。

Step1 种子链接。从图 1 可以看出, Larbin 中种子链接有 3 个来源:配置文件、控制台、解析网页获得的子链接,作为种子 URL 在 Larbin 中的流向基本相同。根据优先级将其存储到硬盘或内存队列(URL 库),等待下一步处理。

Step2 从库中获取 URL。从 URL 库获取一个或者多个 URL,然后按照一定策略将其放入对应站点。

Step3 DNS 解析。在抓取之前需要进行 DNS 解析获取 IP 地址。Larbin 利用 ADNS 进行批量域名解析,以此提高效率,即对于一个站点只进行一次 DNS 解析。获取该站点下的 robots 文件,查看该站点下的抓取限制。

Step4 网页抓取。发起连接,检查 robots 文件,确保该站点可以抓取;对服务器发起请求报文,服务器返回应答报文。

Step5 解析 robots 文件和 HTML 文件。获取禁止抓取的 URL 列表;进行子链接提取。

表 1

开源爬虫框架对比

开源框架	二次开发	开发语言	支持分布式	镜像保存	优点
Larbin	易	C++	否	是	高效、高度定制化
Nutch	难	Java	是	否	支持分布式、提供抓取和索引、提供插件扩展
Heritrix	易	Java	否	是	高度可扩展、高度可控性
Scrapy	一般	Python	是	否	可用性高、功能完善

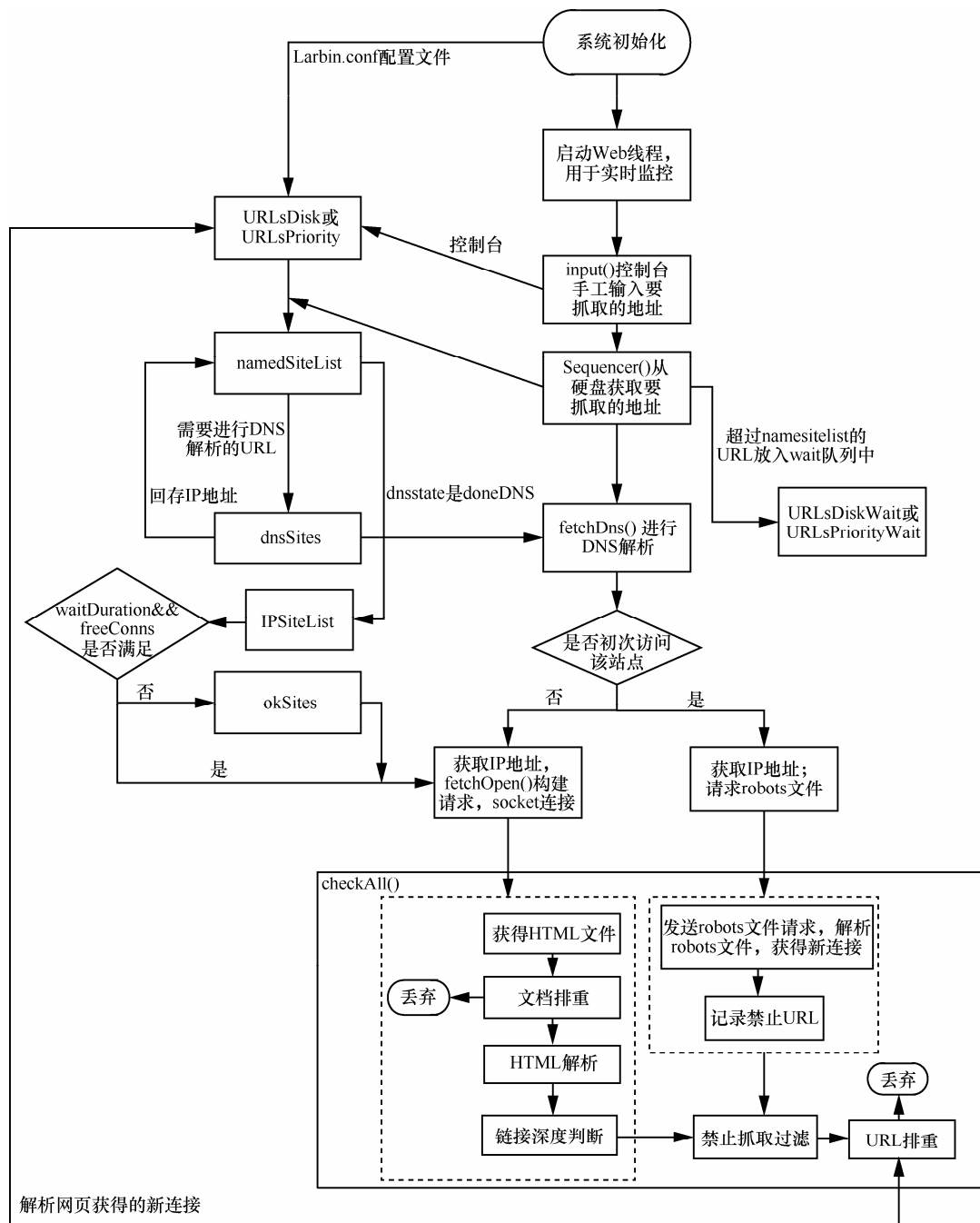


图1 Larbin 整体流程

Step6 对上一步获取的URL进行深度判断和排重处理。然后将合格的URL存入到URL库中。

至此, 一个URL的网页获取流程结束。

基于以上的介绍可以发现, Larbin 符合主流网络爬虫的设计思路, 采用单线程非阻塞模式, 对于单一站点的爬取在速度上具有明显的优势。但是, Larbin 在数据结构和程序流程方面也有其不足之处, 包括: 数据结构设计不合理, 造成 fifo

队列满以及爬取速率下降等问题; 程序流程设计不合理, 造成由于网络原因而导致的URL丢弃。另外, 目前Larbin只能抓取网页, 不提供网页解析、内容存储及索引服务, 在此提供一种思路: Larbin连接外部数据库并建立索引。

3 Larbin 结构和流程优化

本节针对Larbin的不足提出对应的改进方

案,其中包括:修复站点队列满的问题;优化速度下降问题;链接抓取失败后采用重试方案;提出实现网页解析、内容存储、索引服务的思路。

3.1 数据结构优化

Larbin的目标是对网页上的URL进行扩展性的抓取,为搜索引擎提供广泛的数据来源。但是随着系统的运行,出现网页获取速度下降的现象。这是因为从一个网页中抽取的子链接被批量放入内存或者硬盘队列。Larbin 批量读取这些子链接再次进行网页抓取时,它们之间的多样性越来越差,造成速度下降。因此,提出如下解决方案。

Step1 解决速度下降问题最直接的方法是打乱URL库中链接的排序,在这里修改对硬盘和内存队列的调度时间间隔,达到间接打乱顺序的需求。

Step2 根据系统运行内存大小,适当地增加maxUrlsBySite的值,即每个站点中可以存储的URL最大数,本文修改为10 000。

Step3 waitDuration是同一个网站送出2次请求之间等待的时间,当对多个网站进行抓取时,适当减小其值,可以提高抓取速度,本文设置为30。

3.2 程序流程优化

流程优化1 优化站点队列的操作流程

Larbin可能提示队列满的错误,提示如下。

Larbin: site.cc:97: void NamedSite::putInFifo(url*): Assertion`inFifo!=outFifo' failed。

出现这种错误是因为Larbin对限定值设定不当:Larbin源码中limit设定为1,maxUrlsBySite设置为40。假设当前站点中链接数量nburls=39,由于nburls > maxUrlsBySite-limit不成立,当前URL放入fifo队列,从而导致fifo队列满。

设置一个全局变量控制是否从库中获取URL。当站点中的链接数量达到限定值后,控制队列暂时不获取URL,避免取出后再进行回存,对系统性能产生影响。详细方案如下。

Step1 首先改进限定值设定:limit设置为大于1的数字。

Step2 设置一个全局变量isGet_que,初始值默认设为true,表示所有站点URL数量均未达到限定值,可以继续从库中获取一个URL。

Step3 当某个站点队列中的URL数量达到限定值时,isGet_que置为false,表示不再从URL库中取URL。

Step4 再次取URL时,首先判断isGet_que的状态,决定是否从库中获取URL。

优化后伪代码如下。

/*初始化 isGet_que 队列*/

初始化为true,表示可以从URL库中取URL;

/*判断 isGet_que 队列的状态*/

static bool canGetUrl () {

判断该站点下URL是否达到限定值

达到限定值,提示暂时不获得URL;

未达到限定值,从URL库中获取URL并加入到namedSiteList;

}

/*将URL放入fifo队列,更新isGet_que状态*/

void NamedSite::putGenericUrl(url *u) {

nburls++;

判断该站点下URL是否达到限定值

如果达到限定值,isGet_que置为false;

如果没有达到,isGet_que置为true;

...

}

/*从fifo中取出一个URL,更新isGet_que*/

inline url *IPSite::getUrl () {

从fifo中获取URL;

Nburls--;

判断该站点下URL是否达到限定值

如果达到限定值,isGet_que置为false;

如果没有达到,isGet_que置为true;

...

}

优化前后流程对比如图2所示。

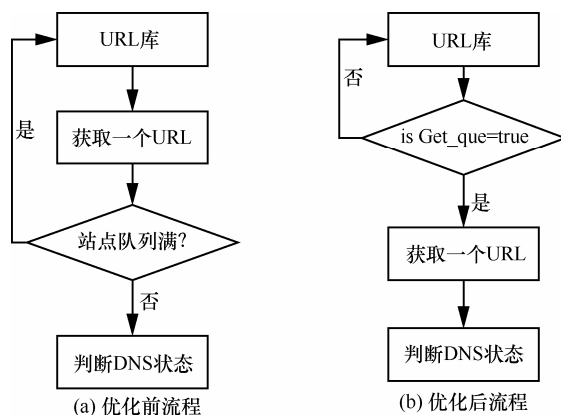


图2 站点队列操作流程对比

流程优化2 链接抓取失败后采用重试方案

在 Larbin 中, 当 DNS 解析失败、无连接以及 robots 文件禁止时, 则调用 forgetUrl() 进行 URL 忽略操作, 即从队列中直接删除。但在正常的抓取过程中, 可能由于网络不稳定、带宽有限、网站临时关闭等原因造成 socket 连接不成功、DNS 解析中断或失败 (robots 文件禁止抓取的链接在此不做优化)。当出现此问题时, 不会直接将这条 URL 删除, 而是将其重新放回站点队尾, 等待下次抓取, 改进方案设计如下。

Step1 当一个 URL 的 DNS 状态为解析失败或者无连接状态时, 将当前 URL 重新放入 namedSiteList 队尾, 并设置为等待 DNS 解析状态, 默认最多重试 3 次。

Step2 设置一个全局变量 reconn 为 3, 每次

抓取失败重试后, 更新 reconn 的值。

优化后伪代码大致如下。

```
int reconn = 3; // 每个 URL 增加一个属性参数, 表示可以重试的次数

void failure (url *u, FetchError reason) {
    if(u->reconn > 0){
        DNS 置为等待解析状态
        将 u 重新加入到 namedSiteList;
        // 重试次数减 1
    }
}
```

优化前、后流程对比如图 3 所示。

流程优化3 网页解析、内容存储、索引服务的基本思路

Larbin 只负责抓取网页, 从网页中提取出子链接, 加入到 URL 库进行下一步的扩展性抓取。

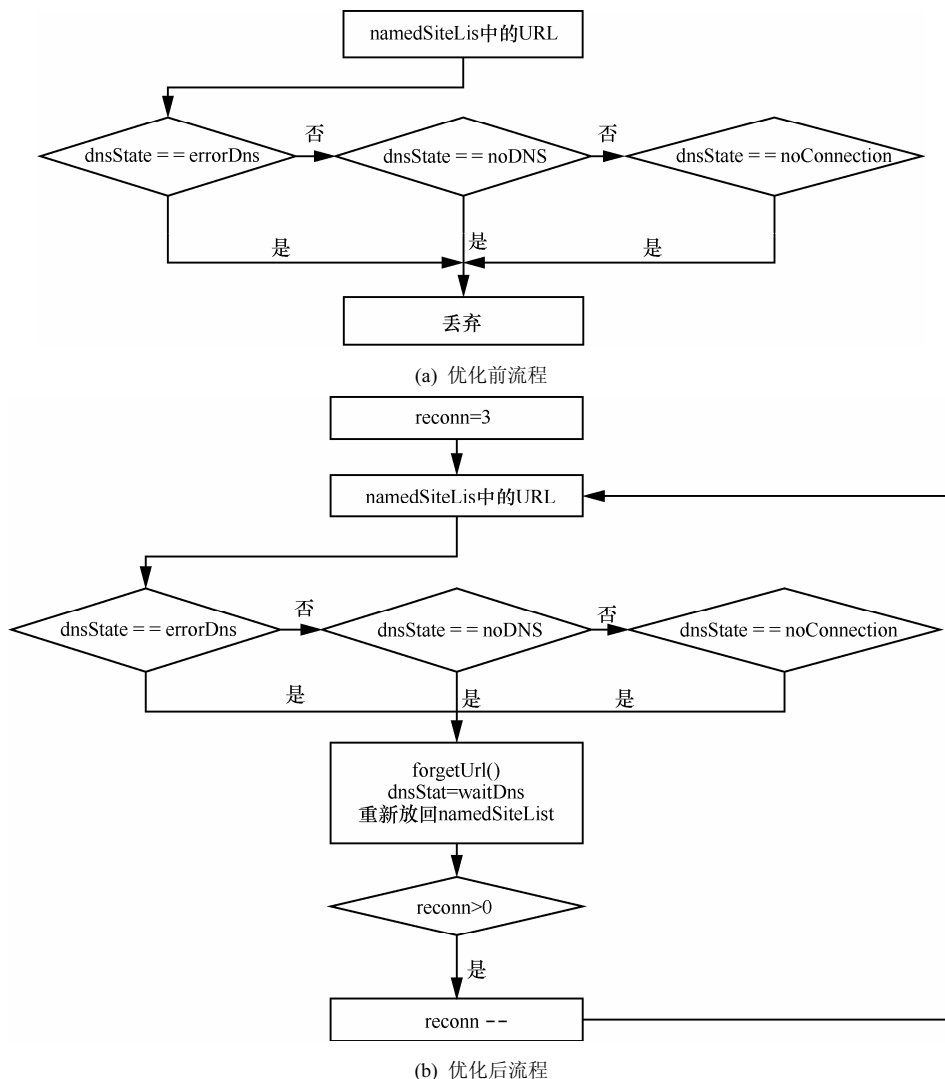


图3 优化前后流程

但是不负责网页内容的解析、提取内容的存储以及建立索引等操作^[13]。在此提出实现以上功能的基本思路,设计结构如图4所示。

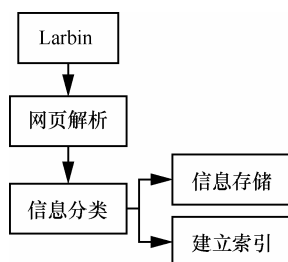


图4 功能实现基本流程

1) 网页解析。Larbin 将抓取的网页保存到本地,可以对本地的网页文件进行页面内容解析,按相关规则提取符合要求的目标数据。这样既保证网页解析的速度,同时也不会影响 Larbin 继续抓取网页的速度。目前主流的网页解析工具如表2所示。

表2 网页解析工具

分类	工具
Python	Beautiful Soup、html5lib
Java	Jsoup、htmlparser、JTidy
C/C++	Gumbo、Htmlcxx、Streaming HTML parser、libhtml

2) 信息存储。将解析分类后的目标数据存入本地数据库,目标数据包括标题、正文、时间、正文图片、点击率、评论数、文章摘要等内容。Larbin 对接的外部数据库可以是 mysql 传统数据库,也可以是 redis 等非关系型数据库。

3) 索引服务。将获得的目标数据传送到非结构化信息搜索引擎,通过建立硬盘索引,实现对数据信息的快速索引。

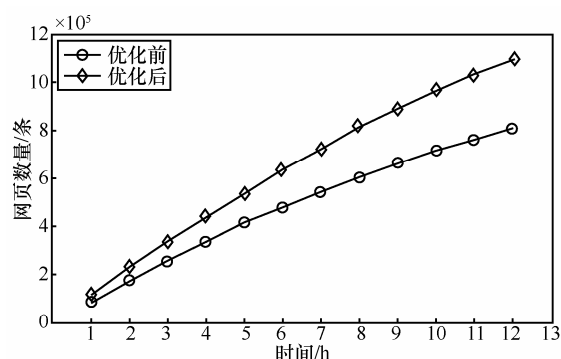
4 性能测试与分析

通过搭建环境,在单机环境下对优化后的爬虫系统进行测试,通过与原系统的抓取能力进行对比,验证本文优化方案设计的高效性和合理性。本文基于 CentOS6.5 平台对优化前和优化后的 Larbin 进行测试,限定到20个网站,运行12 h。

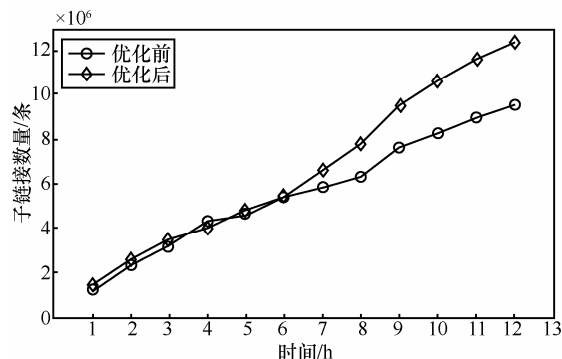
实验结果1 工作效率对比

工作效率的对比是指优化代码前后,抓取网

页数量和子链接数量的对比,由于 Larbin 自身并没有提取内容的功能,所以在此不涉及抓取内容量对比。由图5可知,优化后的系统在网页抓取和子链接获取方面更高效。12 h 之内,优化前后抓取的网页数量大约分别是80万条和110万条,效率平均提高约35%;从网页中解析的子链接数量大约分别是950万条和1200万条,效率平均提高约20%;另外,6 h 以内的子链接数量优化前后基本相等,6 h 后,本文方法的子链接数量相对优化前有了较大提升,从而速度下降问题得到缓解。



(a) 网页数量对比



(b) 子链接数量对比

图5 网页和子链接获取速度对比

实验结果2 性能对比

进行网页抓取的 URL 有3个状态:成功、robots 禁止、DNS 解析失败。通过对比优化前后 URL 状态分布情况,验证抓取失败进行重试操作对成功率的影响。由于网络状态将对网络爬虫产生一定的影响,测试当前网络的分组丢失率达到1%,优化前通过查看 Webserver 的统计结果,超时、DNS 解析失败等原因导致的子链接获取失败约占2.3%,如图6(a)所示。采取失败重试后,子链接失败率降低到1.0%,如图6(b)所示。

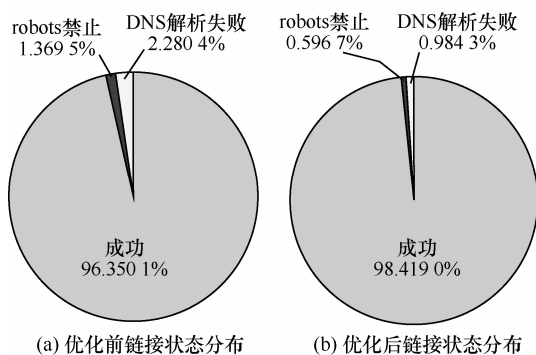


图6 子链接状态分布情况对比图

5 结束语

Larbin 作为一个开源的网络爬虫框架, 给搜索引擎提供了丰富的数据来源。该框架适合进行二次开发, 使用者可以根据自己的需求制定自己的 Larbin 爬虫。本文对 Larbin 的体系结构进行了介绍, 并针对其在数据结构和流程方面的不足提出了优化方案。实验测试证明, 该优化方案可行有效, 使 Larbin 的程序逻辑更加合理并能够更加高效地抓取网络数据。

参考文献:

- [1] BRIN S, PAGE L. Reprint of: the anatomy of a large-scale hyper-textual Web search engine[J]. Computer Networks, 2012, 56(18): 3825-3833.
- [2] 孙骏雄. 基于网络爬虫的网站信息采集技术研究[D]. 大连: 大连海事大学, 2014.
SUN J X. The study on technology of website information collection based on Web crawler[D]. Dalian: Dalian Maritime University, 2014.
- [3] 吕阳. 分布式网络爬虫系统的设计与实现[D]. 成都: 电子科技大学, 2013.
LV Y. Distributed Web crawler system design and implementation [D]. Chengdu: University of Electronic Science and Technology of China, 2013.
- [4] 单月光. 基于微博的网络舆情关键技术的研究与实现[D]. 成都: 电子科技大学, 2013.
SHAN Y G. The research and implementation of network public opinion's key techniques based on microblog[D]. Chengdu: University of Electronic Science and Technology of China, 2013.
- [5] 李跃健, 朱程荣. 基于 Larbin 的网络爬虫体系结构的研究与改进[J]. 计算机技术与发展, 2012, 22(7): 147-150.
LI Y J, ZHU C R. Study and improvement on system architectures of Larbin Web crawler [J]. Computer Technology and Development, 2012, 22(7): 147-150.
- [6] 罗浩. 基于 CLucene 和 Larbin 的企业搜索引擎的研究与实现[D]. 成都: 电子科技大学, 2010.
LUO H. The research and implementation of enterprise search engine based on CLucene and Larbin [D]. Chengdu: University of Electronic Science and Technology of China, 2010.
- [7] 杜一平. 主题搜索网络爬虫的设计与研究[D]. 合肥: 中国科学

技术大学, 2009.

- DU Y P. Design and implementation of topical search engine Web crawler [D]. Hefei: University of Electronic Science and Technology of China, 2009.
- [8] CAFARELLA M, CUTTING D. Building nutch: open source search.[J]. Queue, 2004, 2(2): 54-61.
- [9] 敖东阳, 刘好杰. Larbin 分析与 Windows 平台下移植[J]. 智能计算机与应用, 2009(4): 23-24.
AO D Y, LIU H J. Analysis of Larbin and transplanted to Windows system [J]. Computer Study, 2009(4): 23-24.
- [10] 张敏, 孙敏. 基于 Heritrix 限定爬虫的设计与实现[J]. 计算机应用与软件, 2013, 30(4): 33-35.
ZHANG M, SUN M. Design and implementation of qualified spider based on Heritrix [J]. Computer Applications and Software, 2013, 30(4): 33-35.
- [11] 赵本本, 殷旭东, 王伟. 基于 Scrapy 的 GitHub 数据爬虫[J]. 电子技术与软件工程, 2016(6):199-202.
ZHAO B B, YIN X D, WANG W. GitHub Web crawler based on Scrapy[J]. Journal of Electronic Technology And Software Engineering, 2016(6): 199-202.
- [12] KARKI R, GENNERT M A. Fresh analysis of streaming media stored on the Web [J]. 2011.
- [13] TONG W, XIE X Y. A research on a defending policy against the WebCrawler's attack[C]//International Conference on Anti-Counterfeiting, Security, and Identification in Communication. c2009: 363-366.

作者简介:



王璇 (1991-), 女, 山东菏泽人, 西安电子科技大学硕士生, 主要研究方向为多核调度。



霍义霞 (1991-), 女, 河北廊坊人, 北京电子科技学院硕士生, 主要研究方向为网络安全。

慈云飞 (1989-), 男, 安徽池州人, 北京电子科技学院硕士生, 主要研究方向为访问控制和信息安全。

史国振 (1974-), 男, 河南济源人, 博士, 北京电子科技学院副教授、硕士生导师, 主要研究方向为网络与系统安全、嵌入式安全。

李莉 (1974-), 女, 山东青岛人, 西安电子科技大学博士生, 北京电子科技学院副教授、硕士生导师, 主要研究方向为网络与系统安全、嵌入式系统安全应用。