文章编号: 1003-5850(2012) 06-0065-03

多线程网络爬虫的设计与实现

张 超,闫宏印
(太原理工大学计算机科学与技术学院,太原 030024)

摘 要:针对互联网信息急剧增多,为了改善网络爬虫的爬行性能和提高爬虫程序的通用性,分析了网络爬虫的原理与架构,设计实现了一种高速多线程网络爬虫程序。该爬虫程序采用多个线程并行处理网页,采用宽度优先和深度优先结合的方式来控制网页爬取深度。实验证明该爬虫程序减少了网页下载过程中的平均等待时间,具有较好的性能。

关键词:搜索引擎,网络爬虫,多线程,URL队列,宽度优先
中图分类号: TP391 文献标识码: A

Design and Implementation of Multi-threads Web Crawler

ZHANG Chao, YAN Hong-yin
(*College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan* 030024, *China*)

Abstract For the increasingly information of the Internet, a new High-speed and Multi-threads Web Crawler(HMWC) was designed and implemented to improve the performance and enhance the versatility in this paper. The web crawler dealt with the web page by using multi-threads and controlled the depth of crawling by mixed method of breadth-first and depth-first. Experimental result show that as a web crawler, HMWC can reduce the average waiting time of web page downloaded and has better performance.

Key words search engine, web crawler, multi-threads, URL queue, breadth-first, depth-first

搜索引擎一般由搜集信息、整理信息、接受查询 3个部分组成。其中核心部分之一搜集信息的部分,就是通过网络爬虫来实现的,这个部分决定着整个搜索引擎系统的内容和信息更新。互联网网络爬虫程序是一个基于 HTTP协议的网络程序,爬虫从一些既定的初始页面开始,顺着从初始页面中分析出的链接进行访问。作为搜索引擎中非常核心的一部分,需要其尽可能多、尽可能快地为搜索引擎提供页面数据。一个优秀的搜索引擎要求其中的爬虫模块与所有模块能够高效地相互协作。世界上第一个网络爬虫是由麻省理工学院(MIT)的马休·格雷(Matthew Gray)在 1993年写成的。后来随着 Internet的发展,出现了商用级别的搜索引擎,比如 Google和 Bing。斯坦福大学设计了用于 Google的爬虫[1]。UbiCrawler[2]是一个具有良好通用性的分布式高性能爬虫。北大天网[3]是国内研究高性能网络爬虫的先行者,其开发的爬虫系统能胜任 10亿级别的网页数据。

1 网络爬虫的分析

1.1 网络爬虫的组成部分

网络爬虫主要分为 4个模块:① 页面获取模块:该模块从起始页面开始搜索,发送页面请求数据包,下载 URL页面。② 页面分析模块:对下载下来的页面进行分析,提取页面中的 URL。③ 链接过滤模块:对页面分析中提取到的链接进行操作,例如 URL去重、剔除错误 URL等。④ 链接队列模块:维护 URL队列。

1.2 网络爬虫的工作原理

虽然网络爬虫的种类多种多样,但是其核心原理是相同的。首先爬虫程序获得一个初始 URL,根据记录判断该 URL是否处理过,如果没有,则下载对应网页,并进行网页解析,获取引用的其他链接,并加入队列。然后,根据一定的遍历算法向主程序提交下一个 URL,以此类推,直到队列中没有可用的 URL为止。网络爬虫的工作过程如图 1所示。

* 收稿日期: 2012-01-03,修回日期: 2012-04-01

* * 张 超,男,1986年生,硕士研究生,研究方向:计算机系统结构,搜索引擎。

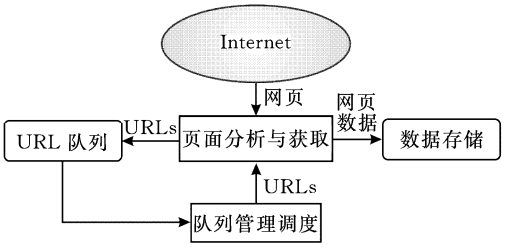

图 1 网络爬虫工作过程示意图

1.3 网络爬虫的爬取策略

网络爬虫所采用的主流的网页搜索策略主要有 3种,即: 深度优先、广度优先、最佳优先。深度优先搜索[4]是在网络爬虫开发使用中比较多的方法。深度优先搜索沿着网页文件上的超级链接走到底为止,形成一条完整的访问链,然后返回到开始网页文件,再继续选择此文件中的其他超级链接。当不再有其他超级链接可选择时,说明搜索已经结束。深度优先搜索方式的优点是能遍历完全 Web站点深层嵌套的文档集合。缺点是假如 Web结构相当深,有可能造成爬虫程序深陷在某一 Web站点。广度优先搜索策略的基本思想是: 与初始 URL在一定距离内的网页重要性较高,因此可以从起始网页开始,先搜索完一个 Web页面中所有的超级链接,然后再继续搜索下一层,直到底层为止[5]。这样做的优点是保证了对浅层 Web页面的优先处理。但是缺点是如果要遍历一个指定的站点的页面集,用宽度优先搜索策略则需要花费较长时间。最佳优先搜索策略按照一定的网页估值算法,预测候选 URL与目标网页的相似度权值,选取其中权值最高的一个或几个 URL进行抓取[6]。

2 网络爬虫的设计

2.1 网络爬虫的设计目标

目前衡量搜索引擎的主要标准是搜索时的查全率和查准率。查准率(Precision)即检出的目标文献与检出的全部文献的百分比。普遍表示为: 查准率= (检索出的相关信息量 /检索出的信息总量) 100%。查全率(Recall)即检出的相关文献与全部相关文献的百分比。查全率= (检索出的相关信息量 /系统中的相关信息总量)× 100%[7]。本文所介绍的爬虫程序在设计的时候主要考虑解决以下几个问题:① 网络爬虫可以解析抓到的网页里的链接,能够提取出网页中的摘要和标题信息进行分析存储;② 爬虫爬取网页深度、网页个数、重要性关系如表 1所示,要求网络爬虫程序能对所搜索的网页进行搜索深度的限制;③ 爬虫要有可配置性,方便以后继续增加开发模块或者集成到其他项目里进行应用;④ 要求能够采用多线程的方式,提高搜索效率。

表 1 网页深度、网页个数和重要性关系表

网页深度	网页个数	重要性
0	1	10
1	20	8
2	600	5
3	2 000	2
4	> = 6 000	很小

2.2 网络爬虫程序的结构

网络爬虫程序的构造方法有两种: 递归方式和非递归方式。递归方式的主要思想是当爬虫分析完成一个页面以后,使用分析出来的 URL再次调用页面下载程序本身。递归的方式使程序的结构清晰,但是当爬行页面数目比较大的时候,会占用大量的资源来构建堆栈,严重降低爬虫程序的运行效率,并且由于共用堆栈的原因不能使用多线程。所以在本文实现的爬虫程序中使用了非递归的方式。非递归方式使用构建 URL队列的方法来实现对程序流程的控制。这里一共用到了 4种队列来描述 URL的不同状态,即放置等待被处理和新发现的 URL的等待队列,处理和判重 URL的处理队列,放置错误 URL的错误队列以及放置已经完成的 URL的完成队列。

2.3 页面的获取与分析

URL是统一资源定位符(Uniform /Universal Resource Locator)的缩写,是因特网上标准的资源的地址,一般的格式为:

协议类型: //服务器地址 (必要时需加上端口号) /路径 /文件名

网络爬虫程序的两个主要任务就是从给定的 URL下载网页数据和从下载的网页中提取新的 URL。HTML超文本标记语言(Hypertext Markup Language),是用于描述网页文档的一种标记语言。HTML中有 5种数据类型: 文本、注释、简单标签、起始标签以及结束标签。网络爬虫最为关注的是超链接标签。

本文的网络爬虫程序使用了 HttpClinet提供的类库来下载 Internet上面的网页,使用其中 GetMethod类实现网页下载的算法如下:① 生成一个 HttpClinet对象 MyHttpClient并设置的参数;② 生成一个 GetMethod对象 CGetMethod并设置的参数;③ 用 MyHttpClient来执行 Get方法;④ 处理 Http响应状态码;⑤ 若响应状态正常,处理响应内容;⑥ 完成后,释放连接。

完成下载以后,使用开源的 JAVA库类 HTMLParser来处理下载的网页,提取其中的文本信息和 URL,随后将其中的文本信息放入用来存储的数

据库供以后进一步的分词检索使用,而提取出来的 URL则放入等待队列,等待处理。HTMLParser将页面按照二进制数据流的方式读取,进行词法分析操作,生成属性层次结构的节点集合。然后使用 NodeClassFilter指定要分析的节点类型,过滤出 <frame>、<a>标签,使用 Parser实例的 Parser方法获得返回的节点数组,找出页面中的所有 URL。

2.4 URL链接过滤模块设计

在 URL去重的模块中,使用了 Bloom filter算法来消去重复的 URL链接。Bloom filter是一种效率很高的数据结构,一般用于在大量数据集中判断元素是否存在。Bloom filter的实现需要一个位数组来表示一个集合,还需要性能好的 Hash函数,将某个 URL链接映射到位数组的某一位,来判定是否为重复 URL。Bloom filter存在一个缺点是有可能会把不重复的 URL判定为重复的 URL,这种情况被称为 Bloom filter的误判。在应用 Bloom filter时,若给定允许的错误率 p,及需要判定的元素个数 n,可以根据公式[8,9]:

$$m \geqslant -\frac{n^{\circ}\ \ln p}{(\ln 2)^2}$$

来得出合适的位数组大小。若 Hash函数的个数为 k,则可根据公式:

$$k = \ln 2^{\circ}\ \frac{m}{n} \approx 0.7\frac{m}{n}$$

来得出应该使用的 Hash函数的个数。

Bloom filter算法的流程如下:

① 对位数组进行初始化;② 查看 URL等待队列,若非空,则取其中一个 URL;③ 对待判定的 URL通过 k 个 Hash函数生成 k 个 $\vdash m$ 的随机数;④ 在位数组中查找相应的 k个位。若 k 个位均为 1,则表示待判定 URL重复了,将其放入错误队列,转②,否则继续下一步;⑤ 对 URL对应网页进行下载、数据采集等工作,将新采集的 URL加入等待队列,转②;

根据本文处理 URL的数量级,选定 Hash函数的个数 k为 10,位数组大小 m 为待判定 URL队列元素个数 n的 20倍时,误判的概率为 $p \leqslant 8.89 \times 10^{-5}$,可基本满足本文网络爬虫的需求。

2.5 多线程

网络爬虫运行的瓶颈在于程序在和服务器交互信息以后等待服务器响应,而多线程可以将平均等待时间降低,提高程序的效率。文中实现的网络爬虫程序使用了 JAVA语言对多线程技术的支持来提高其效率。JAVA语言内置有对多线程的天然支持,根据多线程应用程序继承或者实现对象的不同可以采用两种方式:一种是并发运行的对象直接继承 JAVA的线程类 Thread;另外一种方式是定义并发执行对象实现 Runnable接口。在程序的具体实现时采取了第一种方式,实现了爬虫程序线程类 CrawlerThread。CrawlerThread类基于对多线程控制的 ThreadController类。ThreadController在等待队列中存在等待的 URL或者当爬取的 URL层数还没有达到规定的层次数的时候,创建一个新的线程,并且通过参数限制了爬取页面的层数和最大线程数。当没有需要爬取 URL时,CrawlerThread自行终止,通过消息系统通知 ThreadController,由 ThreadController进行队列的转换工作。

3 网络爬虫程序的测试

3.1 程序实现

根据以上所述设计方案,采用 JAVA编程语言,Eclipse平台实现了网络爬虫。当把爬虫程序的线程数设为 5,爬取深度设为 1,且初始 URL设为作者单位官网: http //www.tyut.edu.cn时运行界面如图 2所示。

图 2 爬虫工作界面示意图

3.2 程序测试

实验的环境如表 2所示。

表 2 网络爬虫的实验环境

软件环境	硬件环境
Ubuntu10.04	Intel Core2 Duo T5750× 2 2.00 GHz 内存: 2 GB 硬盘: 250 GB 网络: 校园网 100 M

将网络爬虫的初始 URL设为: http //www.tyut.edu.cn,当将爬取层数统一设为 3时,线程数分别为 1, 10, 50时,所得实验结果如下:

表 3 网络爬虫测试结果

线程数	所用时间	访问 URL数	下载页面数	平均时间
1	2 420	398	372	6.08
10	1 180	1 490	1 278	0.79
50	830	5 980	5 670	0.13

(下转第 70页)

源的或免费的系统,即大名鼎鼎的 Kidmet和 Snort 目前网上有关于这两个系统的大量教程,您不妨试试。当然,如果您愿意花钱,还可以考虑 AirMagnet AirDefence AirTight等国外公司的产品。

2.5 加强安全意识

加强学校全体员工的网络安全意识,对于专业人员和普通用户应该有不同的要求。网络管理人员需要不断学习,了解最新的网络安全技术,必要时可以参加一些培训班。如果学校有能力,可以对网络管理人员做专业培训,尤其是上岗前,必须经过专业培训才能上岗。网络管理员是校园无线网络安全工作中重要的一环,必须保证每个管理员有丰富的网络安全知识和一定的工作经验。

对校园无线网络用户,不仅要教会他们如何使用无线网络,还要定期进行网络安全知识宣传和普及,加强师生的无线网络安全意识。一定要培养师生养成良好的网络使用习惯,例如:安装防火墙和杀毒软件、注意系统更新、定期对自己的电脑进行杀毒和优化、下载文件要先杀毒再使用、不要使用简单密码等。

3 总 结

由于无线网络的便捷,不仅大学校园慢慢开始普及无线网络,机场、酒店、大型公司等单位或者家庭用户也开始使用无线网络。随着无线网络规模的扩大,随之而来的新的网络安全问题也越来越严重。高校需要的是安全的校园无线网络,如果安全设置不到位,无线网络很可能造成资源丢失、数据被盗窃等严重后果。所以,校园无线网络要求所有人一定要有高度的安全意识,专业人员一定要有过硬的专业素质,设备也要根据情况及时更新,打造一个安全的校园无线网络。

参考文献:

[1] 程海英. 校园无线局域网的安全策略探讨 [J].软件导刊, 2010(3): 128-129.

[2] 王 亮,翟乃强. 浅析校园网中无线网络的应用 [J]. 信息系统工程, 2011(4): 79-80.

[3] 李 园,王燕鸿.无线网络安全性威胁及应对措施 [J]. 现代电子技术, 2007(5): 91-94.

[4] 田进华. 高校校园网无线局域网通信安全策略探析 [J].信息系统工程, 2011(3): 27-45.

[5] 任小琦,王丽霞,王艳丽. 基于校园网教学系统的研究 [J].电子科技, 2010(6): 114-116.

[6] 伍永锋.无线局域网在高校信息化建设中的应用探讨 [J].福建电脑, 2004(5): 88-89.

[7] 邵 丹. 无线局域网安全标准的比较和分析 [J].长春大学学报, 2009(12): 61-64.

(上接第 67页)

从测试结果中可以分析到,当爬虫程序使用的处理网页的线程数逐渐增多时,爬虫处理网页的效率不断提高,性能不断增强。但是不能无限度地增加网络爬虫的线程数,因为每个线程都要占有一定的系统资源,当线程数增加的时候,爬虫程序的效率不会无限提高。

4 结束语

分析了网络爬虫程序的原理,结合具体的开发环境,给出了一种网络爬虫的设计,解决了网页分析与下载,URL队列与 URL去重操作及多线程并行爬行页面等问题。所给的解决方案有一定的通用性和扩展性,效率比较高。下一步的工作主要集中在以下几个方面:① 页面下载的时候,爬虫加入 DNS缓存,减少每次 URL解析的时间;② 加入对网站中爬虫限制协议 robots.txt的处理,防止被对方封 IP地址;③ 加入页面判重,防止相同的页面多次下载。

参考文献:

[1] Brin S, Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine [J]. Computer Networks, 1998(30): 107-117.

[2] Boldi P, Codenotti B, Santini M. UbiCrawler: A Scalable Fully Distributed Web Crawler [J]. Software Practice and Experience, 2004, 34(8): 711-726.

[3] 雷 鸣, 王建勇, 赵江华, 等. 第三代搜索引擎与天网二期 [J]. 北京大学学报(自然科学版), 2001, 37(9): 735-740.

[4] 刘世涛. 简析搜索引擎中网络爬虫的搜索策略 [J].阜阳师范学院学报, 2006(9): 60-61.

[5] Cho J, Garcia-Molina, Page L. Efficient Crawling Through URL Ordering [C]//Proceedings of the seventh international conference on World Wide Web 7, 1998.

[6] 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.

[7] 焦玉英. 信息检索进展 [M]. 北京: 科学出版社, 2003 165.

[8] 谢 鲲, 文吉刚, 张大方, 等.布鲁姆过滤器查询算法 [J]. 软件学报, 2009, 20(1): 96-108.

[9] 丁振国, 吴宝贵, 辛友强. 基于 Bloom Filter的超大规模网页去重策略研究 [J]. 现代图书情报技术, 2008, 24(3): 45-50.

[10] 斯格尔特. Java编程艺术 [M].北京: 清华大学出版社, 2004.