Web站点拓扑结构获取方法研究

潘庆和[1],徐耀群[1],赵星驰[2]

(1. 哈尔滨商业大学 计算机与信息工程学院,哈尔滨 150028;
2. 哈尔滨科学技术职业学院 外语系,哈尔滨 150300)

摘 要:提出了一种使用深度优先遍历方式实现的Web站点拓扑结构获取策略,使用Python语言实现,并可扩展成用于数据采集的爬虫.利用这种方式可以对目标网站进行拓扑探测,了解其内部组织结构,为进一步的研究提供基础.

关键词:站点拓扑结构;深度优先遍历;Python;爬虫
中图分类号:TP393 **文献标识码**:A **文章编号**:1672-0946(2015)05-0573-05

Research on acquisition method of Web site topology

PAN Qing-he[1], XU Yao-qun[1], ZHAO Xing-chi[2]

(1. School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China;
2. Department of Foreign Languages, Harbin Vocational College of Science and Technology, Harbin 150300, China)

Abstract: In this paper, the strategy on the obtainment of website topology structure was put forward based on depth first traversal. The Python language was used to implement this strategy and the program can be extended to construct a web crawler. By obtaining web site topology structure it could detect and understand the internal organizational structure of the web-site and would provide a basis for further research.

Key words: website topology structure; depth first traversal; Python; crawler

1 Web站点拓扑获取的意义

当前,互联网已经成为人们工作生活不可或缺的一部分.截至2014年9月,全球互联网网站数量已超过10.6亿[1],我国在2014年就开通新网站95.2万个,平均每月7.9万余个[2],至2014年底,我国网站数量已经达到364.7多万个.这些网站所提供的信息不尽相同,类型各异,比如新闻类,电商类,论坛类,咨询类等等,而每种类别又可以进一步地细化分类,通过这种内容的分类,可以对我国互联网发展得到更加客观和深入的认识,无论对于网络政策的制定,还是未来网络发展的规划,都具有十分重要的意义.本研究也是对网络中的各类网站进行分析和研究,但研究对象不是针对互联网各网站的内容,而是对各网站的拓扑结构进行研究分析,并提出了一种切实可行的技术来获得各网站的拓扑结构.网站拓扑结构本质上就是网站的内容层次结构,除去了具体内容的含义,从抽象的角度研究网站的层次结构.这种研究具有十分重要的意义,比如说通过对一个网站拓扑结构的获取,可以有效地了解网站在运行过程中关键的信息节点,这些节点往往是性能瓶颈存在的位置;可以进一步对关键节点进行评测,从而设计合理的安全 防护策略.另外,在对网站拓扑探测的同时,如果同时保留了所探测的页面信息,可实现对网站整体信息的抓取.本研究设计的拓扑获取策略即可达到这种效果.

收稿日期:2015-01-27.
基金项目:黑龙江省自然科学基金资助(F201035)
作者简介:潘庆和(1981-),男,博士,讲师,研究方向:大数据分析,数据抓取,数据挖掘和机器学习,数据可视化.

2　站点拓扑结构获取方法

本文设计了一种基于深度优先遍历的网站拓扑结构获取方法. 基本思想如下:从网站的根路径对应的首页面开始,依次逐个访问该页面中所包含的链接,这些链接因为都在首页中,因此都处于相同的层次,可认为是根路径的下一个层次. 在每一个链接对应的页面中,再次重复这个步骤,如果一个页面中所包含的链接全部都被访问过,那么返回上一层次,在上一层次中找到没有被访问过的链接,继续依照这种模式进行访问. 在这个过程中,未访问链接的数量会越来越少,当网站所有的链接都被访问后,就认为任务结束,根据访问过程中保留下来的线索信息即可生成网络的拓扑. 该思想的示意图如图1所示.

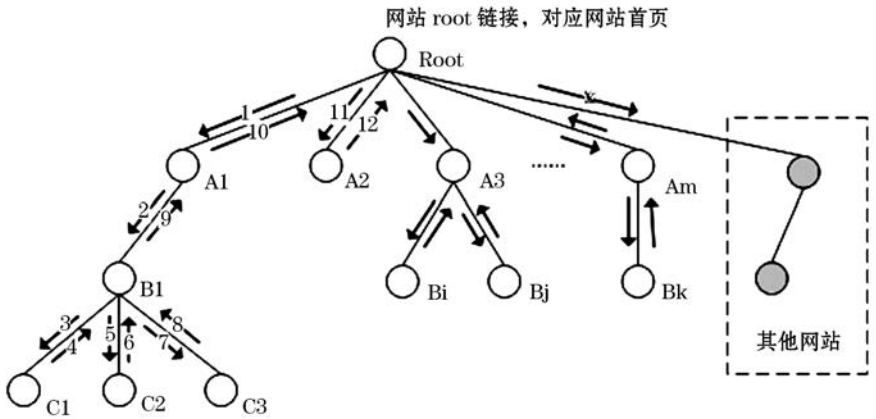

图1　网络拓扑深度遍历过程示意图

图1给出了前面设计思想的示意. 图中使用树来描述获取网站拓扑的过程. 树的根节点为 Root,对应待探测网站的入口链接,一般该链接指向网站的首页. 树中的每个节点均代表链接,比如 Root 下的 A1,A2,A3,...,Am 表示点击 Root 链接所到达的页面中包含的未访问链接. B1 的子节点 C1,C2,C3 表示点击链接 B1 后到达的页面中所包含的未访问链接. C1 没有子节点,表示点击链接 C1 后,到的页面中已无未访问链接. 箭头描述了拓扑发现的执行方向,箭头上的数字为执行的顺序. 可以看出沿着 1 - >2 - >3 - >4 - >5 - >6 - >7 - >8 - >9 - >10 - >11 - >12 的路径,恰好是前面描述的思想所对应的执行路径. 带有"X"标志的箭头表示不沿着该箭头所示方向进行探索,通常这种情况一般表示探索的链接将指向其他网站,一般来讲对网站拓扑的探测发现过程将只涉及到该网站自身,并不涉及到其他网站. 尽管并不对链接到的外部网站进行探索,但为了保存相应的信息,仍把外部网站的根节点信息作为网站拓扑结构的一部分,比如对于图中带有"X"标志的箭头所指向的那个灰色节点,也作为网站拓扑的一部分保存. 从上图的形式看,这种拓扑的探测发现十分类似于数据结构中树的深度优先遍历所给出的形式,这正是我们提出的基于深度遍历的 Web 站点拓扑结构获取策略名称的由来. 在对于树的深度遍历过程中,通常使用递归的方式进行,因此可以将上面的拓扑发现策略抽象为递归过程. 下面给出这种策略思想的简单描述,由于使用了递归形式,因此策略的描述十分简洁.

Algorithm: Get topology structure of a website

get_site_map(link,exclusive_links)

Input:

1) root_url of a website; #网站的根 url 链接;

2) exclusive_links; #无需进一步遍历的网站链接列表,比如图1中其他网站的根链接将在此列表中. Output:

网站拓扑结构(用恰当的数据结构或图形表示).

Steps:

Step1. 如果 link 没有被探测,则得到 link 所指向页面中包含的链接列表 link_list,列表中不应包含 link 和 exclusive_link 所包含的链接,同时将该 link 加入到已探测访问过得链接列表 visited_link_list 中;如果 link 已经被探测过,则不在对其探测,返回;

Step2. 对于 link_list 中的每个元素 a,继续调

用 get_site_map(a, exclusive_links).

应该指出的是,这种形式的策略与网络爬虫的工作方式十分相似. 网络爬虫是一个十分形象的名称. 通常来讲网络爬虫可以分为两类,一类是像搜索引擎提供商所设计的爬虫,这类爬虫会不断地在互联网中利用链接跳转,采集页面信息,返回后供搜索引擎建立相应的索引,这样当我们在引擎中输入文字进行搜索时,引擎就会根据输入对爬虫曾经得到的信息进行检索,找到接近我们搜索文字的相关内容并返回. 另一种爬虫是对明确指定的网站进行抓取的,这是最常见的一类爬虫. 利用这类爬虫可以对具体的目标网站进行数据抓取,获得所需要的信息. 这些信息通常都是可以公共访问的,利用程序设计爬虫来抓取的主要原因是可以节省人工浏览并保存信息的工作量. 比如,对于各类大型电商网站商品信息,房地产领域住房信息,各种机械,电子领域的部件信息,都可以设计爬虫有目的的抓取. 大数据时代,这种获取也是自建大数据集的一种有效手段. 本研究对网站拓扑获取所使用的技术,工作方式上类似于第二类爬虫.

3　拓扑获取策略的实现

对于已给出的策略,在实施上有很多问题需要考虑. 比如:

1)如何获得一个页面内所包含的所有链接的列表?

2)如何跟踪探测一个链接? 在 html 页面中通常使用 a 标签表示一个链接,一般来讲 a 的 href 属性就是需要进一步探测的链接,获得这个属性值后,就可以根据它进行适当的跳转;另一种常见的情况是,一个链接的跳转是通过点击它,触发相关的 js 代码实现的,那么如何设计一种探测方法同时兼顾这两种方式,是一个值得研究的技术问题,这将使我们设计的程序更具一般性,适应更加广泛的场合.

3)在 step1 和 step2 的描述中,只给出了一般的探测策略,但为了最后得到整个网站的拓扑结构的详细信息,必须在 step1 和 step2 的执行过程中同时将探测到的信息记录下来,那么采用何种数据结构或存储机制更为合适?

4)如何直观地展示拓扑结构? 如何将探测到的网站拓扑结构以图形的方式表示出来,获得更加直观的认识,也是一个十分值得研究的问题.

以上的 4 个问题是将要关注的主要问题,但并不是全部问题,在实现的过程,还有会遇到很多其他的问题需要解决. 对与上面列出的几个问题,都是从程序设计角度提出的,因此接下来就需要结合某种语言来解决这些问题,实现相应的功能. Java, C#, C + +, C 等主流语言都可以作为选择,本研究中使用 Python[3] 作为实现语言,主要原因是 Python 的动态语法机制更加灵活简洁,同时 Python 有强大的支持库可以方便地调用,加快开发的效率. 本研究所提供的拓扑获取策略并不依赖于具体的语言,其他动态类型脚本语言如 PHP 和 Ruby 等也可作为实现语言,方式也都类似. 下表 1 给出了利用 Python 实现时,上述四个问题所涉及的解决技术. 在拓扑可视化方面,使用了 Graphviz[4] 作为工具,将 Python 探测到的拓扑信息显示出来.

表 1　针对各问题的解决方案

	解决方案	说明
问题 1	Beautifulsoup[5] (Python)	使用该库可以方便地根据需求提取页面中的信息.
问题 2	Selenium[6] (Python)	一个可以模拟各种浏览器的库,利用该库可以利用模拟点击的方式来进行链接的探测,有效地解决了问题 2.
问题 3	Python 面向对象技术(Python)	在跟踪拓扑信息的时候,可以设计链接类,对每一个链接都用一个链接对象来表示,利用对象的属性保存相应的信息,达到保留拓扑结构线索的作用,有效地解决了问题 3.
问题 4	Graphviz	在运行结束时,可将各链接对象的信息抽取出来,然后把这些信息组成字符串,供 Graphviz 作图.

需要说明的是,对于问题 3 中链接类的设计涉及了许多的细节,比如每个链接对象都需要保存其父节点链接,保存其子链接列表,而链接类还要有变量来保存全局已经访问过的链接和链接对象,这样当拓扑获取完成时,可以根据对象间的这些链接线索穿成一条线,完成对整体拓扑结构的获取.

4　实　验

本节应用上节所设计的程序进行网站拓扑的获取,目标是哈尔滨商业大学的官方网站. 图 2 是网站的首页.

图2　网站首页内容

该网站是一个以新闻咨询为主的网站,网站包含很多链接跳转到学校的各个职能部门,对于这些部门的网站都是独立的,因此在拓扑获取的过程中将只记录这些部门首页的链接,将其作为整体拓扑的一部分. 对于各部门内部的拓扑将不再探测追踪. 如果对这些子部门内部的拓扑结构也感兴趣,则可继续探测,将探测到的拓扑结构和整体拓扑利用首页链接相连接即可.

将实现前面拓扑策略的程序文件命名为 get_site_map. py. 使用的方式是利用 Windows 命令行 CMD 程序或 Linux 的终端,进入到该文件的目录,然后输入命令 python get_site_map. py http://www.hrbcu. edu. cn/启动即可. 其中,python 将启动本机安装的 python 解释器,get_site_map. py 是文件名称,参数"http://www. hrbcu. edu. cn/"是待探测网站的主页链接. 程序将运行一段时间,在程序最后,为了可视化地展示拓扑结构,将得到的拓扑信息组织成了 graphviz 绘图程序所能识别的结构形式,这将生成一个 dot 文件,里面包含有网站拓扑结构的信息,比如可命名文件为 graph. dot. 文件内容如下.

```
digraph G {
"http://www. hrbcu. edu. cn/" -> "http://3w. hrbcu. edu. cn/";
"http://www. hrbcu. edu. cn/" -> "http://en. hrbcu. edu. cn/";
"http://www. hrbcu. edu. cn/" -> "/Category_28/Index. aspx";
"http://www. hrbcu. edu. cn/" -> "/Category_40/Index. aspx";
"http://www. hrbcu. edu. cn/" -> "/Category_41/Index. aspx";
"http://www. hrbcu. edu. cn/" -> "/Category_93/Index. aspx";
"http://www. hrbcu. edu. cn/" -> "/Category_117/Index. aspx";
"http://www. hrbcu. edu. cn/" -> "/Category_118/Index. aspx";
"http://www. hrbcu. edu. cn/" -> "/Category_38/Index. aspx";
…
}
```

然后同样在命令行或终端环境下,导航到文件所在目录,运行 graphviz -Tjpeg graph. dot -o graph. jpg 便可在相同目录下生成 grapg. jpg,这个图片就是网站的拓扑结构图,图3中给出了部分截图.

从图3中可以清楚的看到,网站形式为三级层次,这和一般的新闻咨询类网站风格是一致的. 该图中有两个链接"http://djgz. hrbcu. edu. cn"和"http://kxfz. hrbcu. edu. cn/"分别跳转至不同的子部门,因此只保留了入口链接而并没有继续的探测. 如果需要可进一步探测,然后将获得到的拓扑结构信息在上述链接处和整体结构相链接即可. 同时可以看到,有的二级节点连接着大量的三级子节点,这些二级枢纽节点在提升网络性能和提高安全性的角度都是需要关注的对象.

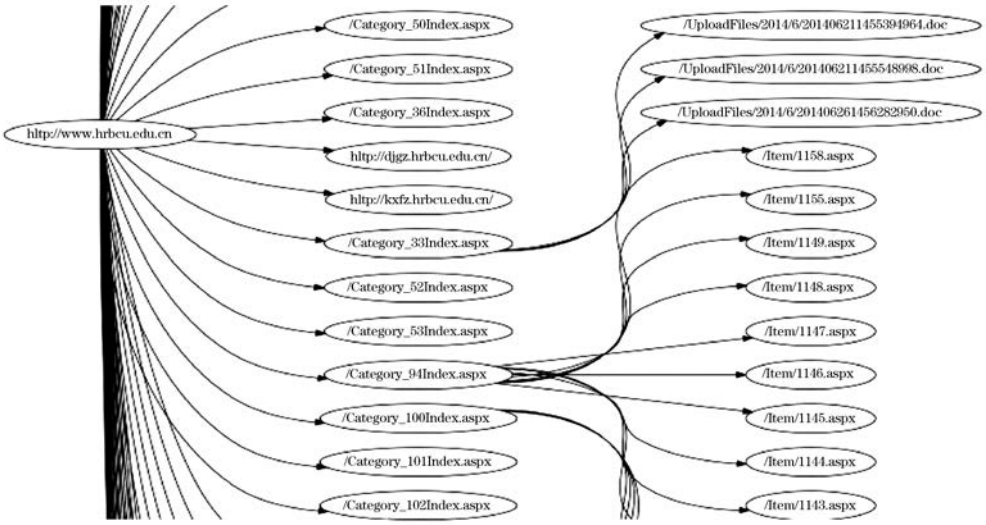

图 3　网站拓扑结构部分截图

前面提到过,在探测的过程中如果将页面信息同时保存下来,即可实现网络爬虫的功能. 对于本实验也可在探测拓扑时同时保存所探测的网页. 图4 给出了保存网页文件夹的截图,其中每一个网页文件的名称为时间戳加上两个随机数字,利用下划线相连接,这是因为如果探测速度过快,可能导致同一个时间戳对应多个页面的情况,因此利用时间戳加上两个随机数的形式,可以最大可能地避免这个问题. 如果将网页文件名作为链接对象的一个属性值保存,也可建立起这些文件间的关联.

图 4　抓取到的网页文件

5　结　语

本文给出了一种 Web 站点拓扑获取的策略,并利用 Python 语言加以实现,得到了一种通用的,适合于各种类型网站的拓扑获取方法. 通过使用拓扑获取的方式对网站进行研究,可以在网站分类,性能评测,安全性防护方面给出有价值的建议. 文中使用 selenium 进行链接的追踪是一种通用性的考虑,因为通过模拟点击追踪链接的方式,可以无需关心链接的具体形式. 如果不考虑通用性,有的站点可以使用 asyncio,aiohttp[7] 等进行快速地探测和抓取. 文本设计的拓扑获取程序,经过简单地改动可以变成一个爬虫,在探测拓扑的同时保存相关的网页,为进一步分析提供基础.

参考文献:

[1] 法新社. 全球互联网网站数量破 10 亿[J]. 中国教育网络, 2014(10): 4.

[2] 国家互联网应急中心. 中国互联网站发展状况及其安全报告(2015)[EB/OL]. http://www.isc.org.cn/zxzx/ywsd/listinfo-31792.html, 2015-03-20.

[3] MARK J J. A Concise Introduction to Programming in Python [M]. United Kingdom: Chapman & Hall/CRC Textbooks in Computing, 2012.

[4] JOHN E, EMDEN G, LEFTERIS K, *et al*. Graphviz—Open Source Graph Drawing Tools [J]. Graph Drawing Lecture Notes in Computer Science, 2002, 2265: 483-484.

[5] VINEETH G N. Getting Started with Beautiful Soup [M]. Birmingham: Packt Publishing, 2014.

[6] BURNS D. Selenium 2 Testing Tools: Beginner's Guide [M]. Birmingham: Packt Publishing, 2012.

[7] JAN P. Parallel Programming with Python [M]. Birmingham: Packt Publishing, 2014.