分布式网络爬虫设计

郭丙琴[1] 陈爱武[2]

（1.湖南科技学院 教学质量管理处，湖南 永州 425199；2.湖南科技学院 电子与信息工程学院，湖南 永州 425199）

摘 要：网络爬虫是互联网信息获取的重要工具之一，其性能的好坏直接影响到互联网信息检索的准确性，互联网信息复杂多变，造成传统方法的网络爬虫容易抓取到错误信息。论文在此基础上提出了一种并行和分布式技术进行设计，并通过招聘网页信息抓取的实验，实验结果证明该网络爬虫性能稳定，可以提升抓取信息的准确性。

关键词：分布式；网络爬虫；Python；搜索引擎

中图分类号：TP393 文献标识码：A 文章编号：1673-2219（2017）06-0021-02

DOI:10.16336/j.cnki.cn43-1459/z.2017.06.007

1 引 言

搜索引擎是基于一种网络爬虫技术来抓取Web网页、文档、图片、音频、视频等信息，并通过索引来组织这些信息，设计性能优良的网络爬虫是搜索引擎重要工作之一[1]。网络爬虫始于一张被称作种子的统一资源地址（URLs）列表，当网络爬虫访问这些统一资源定位器时，它们会甄别出页面上所有的超链接，并将它们写入一张“待访列表”，即所谓“爬行疆域”（Crawl Frontier），此疆域上的统一资源地址将被按照一套策略循环访问。如果爬虫在执行的过程中复制归档和保存网站上的信息，这些档案通常储存，使它们可以被查看。网络爬虫只能在给定时间内下载有限数量的网页，所以设计大容量体积的网络爬虫时需要优先考虑其下载，而互联网资源瞬息万变，网络爬虫下载的网页在使用前就可能已经被修改甚至是删除了。另外，服务器端软件所生成的统一资源地址数量庞大，所以网络爬虫难以避免的采集到重复内容，根据超文本协议“显示请求”（HTTP GET）的参数的无尽组合所返回的页面中，只有很少一部分传回唯一的内容等等，这些问题是网络爬虫设计所面临的基本问题[2]。

论文通过招聘网站的职位信息抓取的实验进行了设计和研究，并且对抓取的数据进行数据分析，解决数据的去重问题、分布式问题和web开发的整合问题等。

2 性能分析与模块设计

2.1 性能分析

由于网络爬虫是对互联网上数据进行处理，信息量非常大，因此对性能的要求是非常高的，所以在设计时一定要考虑其性能优化问题，文献显示[3,4]，在设计网络爬虫是需要考虑的主要性能包括数据库查询、程序执行的效率、网络IO流高低等，另外对挖取到的网页进行解析时，务必要保证解析结果的准确性，这样才能保证爬虫运行的稳定性，一旦解析出错，会导致大量的垃圾数据产生，极大的影响爬虫的运行效率。在爬虫运行过程中，一定要做到可监控，即使有意外的情况发生，也能及时的定位到错误。而且还可以通过监控页面的数据及时了解爬虫的工作性能并保证爬虫在每个时刻都达到最优的状态[5]。

2.2 模块构成

论文设计的网络爬虫（crawler）主要包含以下几个基本模块：扒取任务分配模块、任务执行模块（客户端）、页面解析模块、数据处理模块、运行监控模块，具体结构体系如图1所示。

根据图1所示的模块构成，网络爬虫运行流程是，首先手动创建多个搜索条件，每一个Session对应一个搜索条件，当网络爬虫客户端启动时会自动请求爬虫服务器（SERVER），服务器首先根据客户端的IP验证客户端申请的有效性；如果判断是非正常的客户端请求，服务端不返回任何数据。如果判断客户端请求正常，服务端会将客户端发过来的html页面进行GZIP解压（为减少网络流量，客户端从网站上下载的html页面会进行GZIP压缩后再发送给服务器），然后传给相应的callback函数进行处理。当callback处理完成，服务端根据客户端的申请生成一个当前Session的Queue，并将生成的Queue的信息返回客户端。论文设计的基于招聘网页信息网络爬虫Callback的工作模式分两种类型，列表页面和详情页面，列表页面在解析的时候会将所有详情页的url和职位（公司）的基本信息解析出来，然后对解析出来的url进行去重、有效性检查等处理，最后根据处理后的url生成详情页的Queue。详情页面在解析时主要是将职位（公司）的详细信息提取出来，主要包括职位名称、薪资、招聘信息发布时间、职位描述等信息。

收稿日期：2016－08－26

基金项目：2015年永州市科技计划项目（永科发[2015]9号No.22）。

作者简介：郭丙琴（1981－），女，广西鹿寨人，研究方向为信息技术处理。

3 系统分析与实现

论文的网络爬虫是针对招聘网站公司名称及职位信息而设计的，考虑到后期代码的扩展和维护，所以程序的整体框架采用面向对象的思维，在程序设计过程中，网络爬虫实现了一个爬虫的基类（BaseCrawler），这样在以后维护时如需要加入新的招聘网站的时候，只需要实现一个继承于BaseCrawler 的招聘网站专用的子类和对应的解析器。在设计 Resource 的时候，由于监控资源类和爬虫资源类功能上的差异性，所以将其分开设计，使用的框架要求每一个 Resource 绑定一个相关的 Model，而监控资源类和爬虫资源类两者绑定的 Model 存在一点冲突，所以最后在绑定 Model 的时候存在一些的不合理性，设计 Session 和 Queue 时，考虑到每一个搜索条件会产生不同的搜索结果，所以利用 Session 将各个搜索条件分开，让 Queue 来表示每一次的请求动作，这样设计方法便于在网络爬虫运行时及时发现 bug 保证程序结构的清晰度。

3.1 爬行策略实现

网络爬虫程序的初衷是尽可能遍历每一搜索条件所有的页面，所以论文采用广度优先算法的理论，广度优先算法理论上能够覆盖更多的节点，在扒取招聘网站信息的时候，先根据特定搜索条件将所有职位列表下载下来，然后再依次下载相关的职位详情。

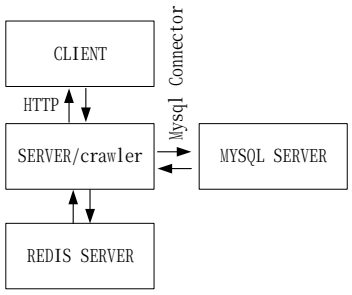

图 1.Crawler 模块组成

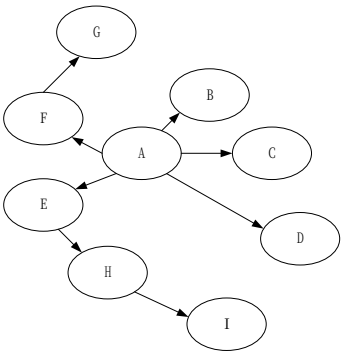

图 2.URL 节点分布图

如图 2 所示，假如 A 代表初始 URL，BCDEF 为根据 A 节点获取的 5 个 URL，G 和 HI 为根据 F、E 获取的 3 个 URL，以此类推。那么这些 URL 获取的顺序就是 ABCDEFHIG 这样一个顺序，当通过 E 节点获取到 H 节点的 URL 之后，并不会马上进行下载，而是先解析同 E 在同一层中的 F 节点对应的 URL，当这一层 URL 全部下载完后，再开始下一层 URL 下载。

3.2 Server 端并行方案实现

我们采用 gunicorn 结合 gevent 的方式实现并行访问机制，虽然多线程的方案能够减少进程创建时候带来的开销，但是对于临界资源的访问控制等变得更加的复杂，需要考虑的因素更多，这样导致开发的难度大大提升。gunicorn 和 gevent 支持 pip 直接下载安装，只需两句简单的 shell 命令即可安装完成，为了结合 gevent 使用，我们只需要在爬虫的启动程序中加入代码即可实现。

网络爬虫对招聘网站信息监控页面曲线如图 3 所示。

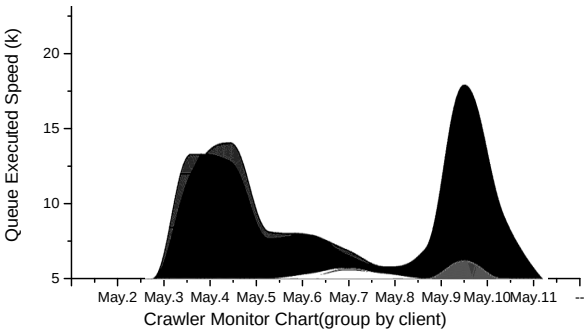

图 3.监控页面曲线图

3.3 爬行策略优化

爬虫一般正常运行了一段时间后，扒取速度就会变得越来越慢，经大量实验证明这种速度变慢是由于服务器的接口的性能所产生的问题，网络爬虫在建表的时候，生成联合索引，所以导致查询速度非常慢，另外，在搜索强制加载的 Queue 的时候，可能就要进行文件排序，这种文件排序会导致搜索速度降低， 我们在进行策略优化时只需将代码中的搜索条件改为唯一。

结束语

论文基于分布式方法进行网络爬虫的设计，并具体针对招聘网站的信息进行实验，实验结果显示论文设计的网络爬虫性能稳定，符合互联网和大数据搜索引擎的基本要求。但存在多个客户端同时工作的时候，可能会导致 Mysql 的死锁（deadlock），虽然不会影响爬虫的正常运行，但是对爬虫的运行效率会造成一定程度的影响。

参考文献：

[1]李勇,韩亮.主题搜索引擎中网络爬虫的搜索策略研究[J].计算机工程与科学,2008,(3): 4-6.

[2]刘金红,陆余良.主题网络爬虫研究综述[J].计算机应用研究,2007,(10):26-29.

[3]孙立伟,何国辉,等.网络爬虫技术的研究[J].电脑知识与技术,2010, (15):4112-4115.

[4]唐波.网络爬虫的设计与实现[J].电脑知识与技术,2009,(11):2867-2868.

[5]詹恒飞,杨岳湘,等.Nutch 分布式网络爬虫研究与优化[J].计算机科学与探索, 2011,(1):68-74.

（责任编校：宫彦军）