ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.5, May 2010, pp.1067−1082
doi: 10.3724/SP.J.1001.2010.03725

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

广域网分布式 Web 爬虫*

许 笑+, 张伟哲, 张宏莉, 方滨兴

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

WAN-Based Distributed Web Crawling

XU Xiao+, ZHANG Wei-Zhe, ZHANG Hong-Li, FANG Bin-Xing

(School of Computer Science and Technology, Harbin Institue of Technology, Harbin 150001, China)

+ Corresponding author: E-mail: smilestor@gmail.com, http://pact518.hit.edu.cn

Xu X, Zhang WZ, Zhang HL, Fang BX. WAN-Based distributed Web crawling. *Journal of Software*, 2010, 21(5):1067−1082. http://www.jos.org.cn/1000-9825/3725.htm

Abstract: There are three core issues recognized for WAN-based distributed Web crawling systems: Web Partition, Agent collaboration and Agent deployment. Centering around these issues, this paper presents a comprehensive overview of the current strategies adopted by academic and business communities. The experiences, problems and challenges encountered by the WAN-based distributed Web crawlers are classified and discussed in depth. A summary of the current evaluation indicators is also given. Finally, conclusion and some suggestions for future research are put forward.

Key words: search engine; WAN-based distributed crawling; Web partition; agent collaboration; agent deployment

摘 要: 分析了广域网分布式Web爬虫相对于局域网爬虫的诸多优势,提出了广域网分布式Web爬虫的3个核心问题:Web划分、Agent协同和Agent部署.围绕这3个问题,对目前学术界和商业界出现的多种实现方案和策略进行了全面的综述,深入讨论了研究中遇到的问题与挑战,并论述了广域网分布式 Web 爬虫的评价模型.最后,对未来的研究方向进行了总结.

关键词: 搜索引擎;广域网分布式爬虫;Web划分;Agent协同;Agent部署

中图法分类号: TP393 **文献标识码**: A

搜索引擎作为互联网上一种有效的信息获取渠道,与电子邮件、即时通信并称为互联网三大基础应用,在

* Supported by the National Natural Science Foundation of China under Grant No.60703014 (国家自然科学基金); the National Basic Research Program of China under Grant No.G2005CB321806 (国家重点基础研究发展计划(973)); the National High-Tech Research and Development Plan of China under Grant No.2009AA01Z437 (国家高技术研究发展计划(863)); the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No.20070213044 (高等学校博士学科点专项科研基金); the China Postdoctoral Science Foundation under Grant No.20070410263 (中国博士后科学基金); the Heilongjiang Postdoctoral Foundation of China under Grant No.LBH-Z07108 (黑龙江省博士后资助); the Development Program for Outstanding Young Teachers in Harbin Institute of Technology of China under Grant No.HITQNJS.2007.034 (哈尔滨工业大学优秀青年教师培养计划)

Received 2008-09-27; Accepted 2009-09-03

人们的日常生活中发挥着重要的作用.然而,互联网的飞速发展使搜索引擎面临巨大的挑战.2008 年 1 月发布的《第 21 次中国互联网络发展状况统计报告》[1]显示,中国网站数量已达 150 万个,比去年同期增长了 66 万个,增长率达到 78.4%;中国总网页数为 84.7 亿个,年增长率达到 89.4%;网站总字节数已经达到 198 348GB.按照目前的统计数字,假设搜索引擎爬虫系统的网络接入总带宽为 100Mb/s,即使这些带宽被完全利用,仅下载中国的网页就需要近 200 天.如此巨大的数据量,使得对网页内容和链接关系的处理必须由多机并行完成.

分布式 Web 爬虫是由多个可并发获取 Web 信息的 Agent 构成的 Web 爬虫系统,每个 Agent 运行于不同的计算资源之上,这些资源或集中部署在同一个局域网(local area network,简称 LAN)内部,或分布在广域网(wide area network,简称 WAN)的不同地理位置和网络位置,每个 Agent 以多进程或多线程方式通过并发保持多个 TCP 链接获取 Web 信息.部署于 LAN 上的分布式 Web 爬虫受到带宽等因素的制约,已经不能对 Web 进行快速而有效的抓取.基于广域网的分布式爬虫实现方案具有多点接入总带宽较高、对 Internet 负载较小、容易实现就近高效抓取以及可扩展性强等优点,已经成为学术界、商业界和开源社区爬虫系统实现的优选方案.

广域网分布式爬虫融合了分布式系统、并行计算及网络测量等主题,具有很强的应用价值与理论研究意义.本文第 1 节概括总结广域网分布式爬虫近年来在商业界和学术界的发展现状.第 2 节~第 4 节详细讨论广域网分布式爬虫领域亟待深入研究的 3 个关键问题:Web 划分、Agent 协同和 Agent 部署,详细论述了解决这些问题的多种方法及策略,对其优、缺点进行评价.第 5 节给出目前流行的分布式爬虫的评价模型.最后,对全文进行总结,并指出未来研究的若干方向.

1 引 言

在分布式 Web 爬虫领域,商业界与学术界各自为战,许多优秀的实现方法不是源自于学术界,而是来自于一些公司.出于商业因素的考虑,公司成果一般不通过论文公开发表;而学术界的研究成果虽然公开,但是被大规模采用的并不多;另外,还有一些组织和个人以 GPL(GNU general public license)的方式开发和发布自己的系统.遗憾的是,这类系统也很少以论文形式发表.

部署在 LAN 上的分布式 Web 爬虫率先被提出,并得到广泛的使用.较为著名的有早期的 Google[2],AltaVista 的 Internet Archive Crawler[3],Mercater[4]等.但是,由于受到带宽等瓶颈因素的制约,此种系统即使软硬件的规模不断扩大,也只能获取全体 Web 信息中相对较小的一部分.为了解决上述问题,人们提出了部署于广域网环境的分布式 Web 爬虫.

1.1 相关工作

近几年来,商业界和开源社区出现了一些广域网分布式爬虫系统(或搜索引擎),其思路一般是公司或组织向用户提供爬虫程序.一方面,分布在各地的用户运行自己机器上的爬虫程序为公司提供数据;另一方面,公司为安装有爬虫的用户提供各种检索服务,如 Yacy(http://yacy.net/)的个性化匿名检索,甚至将利润反馈给用户(如 Faroo(http://www.faroo.com/)).在实现方面,这些系统有的是类似于 SETI@Home[5]那样的主从式结构(如 Majestic (http://www.majestic12.co.uk/)),属于有调度中心的 Agent 协同;有的是 P2P 方式进行分布式调度(如 Faroo),即无调度中心的 Agent 协同.这些系统的实现五花八门,但是由于发展时间较短,规模相对较小.

在学术方面,Cho 等人[6]首次给出了分布式爬虫的分类方法、评价指标等一系列基本概念,并提出基于广域网分布式爬虫与部署于 LAN 的系统相比,具有高可扩展性和减少 Internet 负载的优点,为广域网分布式爬虫的研究奠定了基础.UbiCrawer[7]扩展了文献[6]中的一些概念,并声称可以支持基于广域网的分布式平台.Dustin B 等人[8]对多种分布式爬虫进行了比较,提出广域网爬虫是解决爬虫系统带宽瓶颈的有效方法.Yahoo 研究院的 Baeza-Yates 等人[9]在其综述中将分布式爬虫定义为“原则上某些节点可以分布于不同的地理或网络位置”.2003 年后,很多研究开始关注广域网分布式爬虫,代表性的有,IPMicra[10]第一个基于位置信息调度的分布式爬虫,SE4SEE[11]实现了基于网格[12]的分布式爬虫,Apoidea[13]实现了基于 P2P 协议的完全分布式爬虫.

国内学术界对分布式爬虫研究得较少,代表性的有北京大学的天网搜索引擎[14]的爬虫系统,这是一个基于 LAN 的爬虫,已经开始商业化运作;上海交通大学的 Igloo 爬虫[15]实现了基于网格服务的分布式爬虫(IglooG),

网格的特性使其能够支持广域网部署.

1.2 分布式爬虫的基本结构和工作流程

由于爬虫要下载多个网页,而各个网页的下载过程之间依赖性较小,因此可以被并行化.为了高效地下载网页,爬虫程序一般被设计为多线程和多进程协同的方式,而分布式爬虫是将多个具有抓取网页功能的 Agent 分别部署于多个计算资源之上的爬虫程序.以下是分布式爬虫中每个 Agent 的大致工作流程(其中,左侧带*号的两行代码可能需要多机协同完成).为了突出 Agent 对 URL 的处理,算法描述省略了域名解析、对网页和 URL 的预处理以及解析网站的 Robots.txt 文件的过程.

URL Seen:用于存储已经抓取过的 URL.

URL 队列:用于存储待抓取的 URL.

输入:初始 URL 列表.

```
Agent (初始 URL 列表) {
    将初始 URL 列表中的 URL 放入 URL 队列;
    while (URL 队列不为空) {
        从 URL 队列中取出一个 URL;
        将 URL 存入 URL Seen;
        下载 URL 指向的网页;
        提取网页中含有的 URL;
        for (每一个新发现的 URL) {
            if (URL 应由本 Agent 负责) {
                if (URL 不在 URL Seen 中 && URL 不在 URL 队列中)
                    将 URL 放入 URL 队列;
            } else {
*               通过一定的 Web 划分方法选择负责当前 URL 的 Agent;
*               将 URL 发送至此 Agent;
    }   }   }
}
```

1.3 广域网分布式Web爬虫的优势和挑战

广域网分布式 Web 爬虫与基于 LAN 的分布式爬虫或称局域网爬虫相比具有诸多优势:

(1) 可扩展性

可扩展性是局域网爬虫的致命缺点,也是提出广域网分布式爬虫的主要原因.首先,广域网系统能够容纳更多的计算资源,拥有更多的网络接入点.理论上,整体吞吐量可以无限扩展;局域网爬虫因其计算资源数量受到 LAN 的限制,很难扩展到较大的规模,从而限制了系统整体吞吐量.其次,广域网系统是由若干个相对较小的机群甚至单机节点组成,这使得资源添加和系统维护都变得相对简单.如果能够进一步利用分布在 Internet 上的个人计算资源,则维护开销将大为降低;相比之下,在 LAN 中维护大规模机群的代价则非常昂贵,需要解决数据存储、系统互连、机架结构、电源、散热等诸多问题.

(2) 多网络接入点

爬虫在抓取网页时,HTTP 请求和下载网页的过程需要占用系统网络接入点的大部分带宽.对基于 LAN 的系统,随着机群规模的扩大,接入带宽将变为系统瓶颈.如果爬虫程序分布在不同的网络位置,就可以使用多个网络接入点,理论上可以获得相当于这些接入点加和的总带宽.并且随着网络接入点数量的增加,系统的总带宽也会相应增加,理论上带宽可以无限扩展.

(3) 减少对 Internet 的网络负载

爬虫程序在发出 HTTP 请求并下载网页时,大量数据报文的传播增加了 Internet 的负载,在一定程度上影响了 Internet 的服务质量.如果能够实现就近抓取,即布置在不同地域的分布式爬虫仅负责抓取距离自己相对较近的网站,则广域网分布式爬虫可以将系统带给 Internet 的网络负载控制在局部.而对于基于 LAN 的爬虫,由于其网络接入点单一,大量数据包要经过较长的路径才能到达目的地,从而给路径上的所有网络资源(如路由器、交换机、网关等)带来压力.

广域网尤其是 Internet 环境比局域网要复杂得多,系统一旦架设到广域网环境就会受到诸多限制.如何有效利用广域网资源同时又能消除广域网环境的不利影响,是广域网分布式爬虫研究所面临的重大挑战.本文针对当前广域网分布式 Web 爬虫的研究和实践,总结出这一领域的 3 个关键问题:

(1) Web 划分:如何将抓取 Web 这个巨大的任务切分成多份,交予系统中的多个 Agent 执行.

(2) Agent 协同:多个 Agent 之间应该如何进行协同工作,如何进行互联与通信.

(3) Agent 部署:如何利用现有硬件和网络资源构建广域网分布式爬虫系统.

这 3 个关键问题在广域网分布式 Web 爬虫研究中的层次结构如图 1 所示:最上层的 Web 划分强调的是逻辑问题,相当于决策层;最下层的 Agent 部署强调的是物理问题,它作为系统的基础是工程性很强的一层;Agent 协同则既涉及物理又涉及逻辑,包含了程序实现和网络环境分析等多方面的问题.

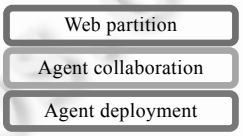


Fig.1 Hierarchical relationship of the three core issues of WAN-based distributed Web crawling

图 1 广域网分布式爬虫 3 个关键问题的层次关系

2 Web 划分

系统中各个 Agent 在抓取过程中会不断地发现新的 URL,而这些 URL 中存在大量的重复.如果将这些新 URL 直接交由发现它的 Agent 抓取,那么将会引起多个 Agent 下载相同的网页,从而引起重复工作,降低整体的网页抓取效率.因此,需要一种为各个 Agent 分配 URL 的策略,由此提出 Web 划分的概念.

2.1 Web划分的定义

定义 1(Web 划分集合和 Web 划分集合的分类). 设分布式 Web 爬虫由 N 个 Agent 组成,Web 上所有网页的集合为 W.对于 W 的子集的集合 $B=\{\beta_1,\beta_2,\beta_3,\ldots,\beta_N\}$,如果满足:

$$\beta_1\cup\beta_2\cup\beta_3\cup\ldots\cup\beta_N=W \text{ 且} |\beta_i\cap\beta_j|<\delta,\ i=1,2,\ldots,N,\ j=1,2,\ldots,N,\ i\neq j,$$

其中,δ是一个较小的整数,它表示各子集之间的交集应当最小化,则称集合 B 中的元素$\beta_i,i=1,2,\ldots,N$ 为一个 Web 划分集合.将 Web 分割为 Web 划分集合$\beta_1,\beta_2,\beta_3,\ldots,\beta_N$的过程称为 Web 划分集合的分类.

定义 2(Web 划分). 设分布式 Web 爬虫由 N 个 Agent 组成,Agent 的集合 $A=\{\alpha_1,\alpha_2,\alpha_3,\ldots,\alpha_N\}$,对于定义 1 中的集合 $B=\{\beta_1,\beta_2,\beta_3,\ldots,\beta_N\}$,称一一映射$\mu:A\to B,\alpha_i\to\beta_j(i=1,2,\ldots,N,j=1,2,\ldots,N)$为分布式 Web 爬虫的 Web 划分.

定义 3(跨分区链接和分区内链接). 如果链接 URL_x 属于 Web 划分集合β_i,链接 URL_y 属于 Web 划分集合β_j,而 URL_x 指向的网页上有一条指向 URL_y 的链接,这个链接就跨越了划分集合,称这种链接为跨分区链接(inter-partition link);如果链接没有跨 Web 划分集合,则称其为分区内链接(intra-partition link).

2.2 Web划分单元

Web 划分单元的选取是实现 Web 划分时必须考虑的问题.Web 划分单元是 Agent 在工作过程中所负责抓取的最小集合,凡是包含于划分单元的网页,全部由一个 Agent 负责抓取.用于 Web 划分单元的某些属性的集合称为划分属性,用于指导对 Web 划分单元的分类.这些属性可以来自 URL 字符串本身,也可以来自与 URL 相关

的某些事物,如网站 IP 地址、网页内容、第三方信息等.

根据广域网环境下实验的经验,广域网分布式系统在进行任务划分时粒度必须适当地大,以保证各个节点具有较高的计算通信比,尽量降低信息交换引发的时间开销.Web 划分单元对应任务粒度的概念,因此这样的结论同样适用于广域网分布式爬虫.下面讨论两个典型的 Web 划分单元(以下简称为单元),并对其划分属性及优缺点进行论述.

(1) 链接(URL)

URL 是 Web 爬虫研究中最小的 Web 划分单元,优点是简单、直观,缺点是粒度太细.由于 Web 上存在的链接比网站总数要多得多,对 URL 进行分类的工作量是十分巨大的.与主机名相比,URL 所携带的划分属性比较少,仅能显示文件类型等信息.

(2) 主机(host)

以 URL 中的主机名(即 hostname,比如 URL:http://www.sina.com/index.html 的主机名为 www.sina.com)为 Web 划分单元,是大部分分布式 Web 爬虫的首选.相对于以 URL 为单元而言,本方法产生的跨分区链接较少.因为处于同一个主机的 URL 必然会被分配到同一个划分集合中;而在以 URL 为单元的情况下,这些 URL 可能会被分配到很多不同的 Web 划分集合中,这样,主机内部的链接也变成了跨分区链接.对主机名的一种延伸是域名,由于一个域名下可能拥有若干主机,因此域名是一种粒度更大的 Web 划分单元.主机所具有的划分属性主要有 IP 地址、网站类型(.cn,.org 等,后面将讨论)等.

除了以上两种单元以外,由于 RIRs(regional internet registries)的存在,通过主机的 IP 地址等信息还可以得到网站所在国家、地区及运营商等信息,给 Web 划分单元提供了更多的可选方案.

2.3 Web划分策略

根据定义 2,在系统中含有 N 个 Agent 的情况下,Web 划分的前提是找出 Web 全集的一个大小为 N 的子集(Web 划分集合)的集合.采用何种方法将所有 Web 划分单元分类成 N 个 Web 划分集合,并实现其与 N 个 Agent 的一一映射,构成了分布式 Web 爬虫的 Web 划分策略.

下面介绍目前已经提出的几种 Web 划分策略,对其原理和优、缺点进行详细论述.

(1) 基于随机哈希

基于随机哈希的方法是采用得最多的 Web 划分方法.最早的分布式爬虫系统大多是建立在对 URL 或主机名哈希的基础之上的.首先,这种方法非常容易计算,用于调度的系统开销较小;其次,由于哈希函数的随机性,保证了各个 Agent 间负载均衡;另外,这种将字符串映射为随机数的方法非常易于与采用 DHT 的 P2P 系统集成,如 UbiCrawler[7](并没有声称自己是 P2P 系统,但是最早使用了类 DHT 方法:consistent hashing[16]),Apoidea[13]等.

基于哈希的方法遇到的最大问题是,结构简单的哈希值无法体现出主机所具有的类型、地理位置、网络距离等信息,也就无法利用这些属性提高分类质量.

(2) 基于域名后缀及文件类型

有的爬虫根据主机或网站的域名后缀不同,将 Web 划分单元分配到不同的 Web 划分集合.比如,根据网站域名中诸如.net,.org,.com,.edu 这些表示组织性质的后缀进行分类;还可以根据 URL 字符串中的文件类型如.html,.mp3 等进行分类.以上两种分类方法更加注重对网页内容的分类.SE4SEE 提出根据表示语言类型或国家、区域的域名后缀,如.cn,.jp,.fr 等进行分类,这样不仅实现了按照网页内容分类,而且由于每种语言群体的地理分布基本都不相同,也部分地实现了按地理位置划分,为爬虫就近抓取创造了一定的条件.

这种方法的优点是,Web 数据在抓取时就已经进行了初步的分类,为以后的数据分析工作奠定了比较好的基础.但它仍然存在诸多缺陷:首先,并非每个 URL 或域名都遵守传统的后缀命名规范,如有的学校的域名就是.com 而不是大家普遍认同的.edu;同样,也有很多.cn(中文)后缀的网站其实含有大量英语内容;其次,由于各种类型的网站的数量或文件的数量分布不均,将造成系统中各个 Agent 的负载不均,比如,按照语言类型分类,小语种网站的数量非常少,而拥有诸如.cn,.de 这类域名后缀的网站数量则非常大.

跨越较大的地理范围和网络范围是广域网分布式系统天生的优势,可以利用这个优势实现 Agent 对网站的

就近抓取,即对每个网站由距离它最近的那个 Agent 来抓取.下面介绍的两种 Web 划分策略都以就近抓取为目的,并且均采用主机名作为 Web 划分单元.

(3) 基于地理位置

基于地理位置抓取是由就近抓取所想到的最直观的方法,即对每个网站由地理上距离它最近的那个 Agent 来抓取.比如,部署在法国的 Agent 只抓取法国境内的网站,部署在中国黑龙江省的 Agent 只抓取黑龙江省内的网站.这种方法具有一定的可行性,因为网络数据的传输都要经过物理线路,因此地理距离较远的两点间,数据传输时间也相对较长,尤其是跨国和跨大洲的线路.一种想法是按照地理位置的层级关系(如大陆、大洲、国家、省等)建立分类树,但前提是必须获得每个网站的地理位置信息.Exposto 等人[17]提出通过提取网页中的地理信息,再通过计算进行划分操作,从而实现对 URL 的分类.

仅以地理位置为参考也存在一定的局限性:

首先,网络上任意两台主机之间的路由路径长度并不与地理位置远近相符.网络运营商在进行路由选择时通常还要考虑到诸多商业因素的影响,从而不遵守地理就近的原则.比如,从哈尔滨工业大学教育网向北京网通发送数据时,路由路径甚至会经过广州的路由器.其次,通常网络链路的延迟分为固有延迟和可变延迟,固有延迟即信号在线路上传输所消耗的时间,本方法就利用了固有延迟的这一特点;可变延迟指数据包在传输沿途所经过的网络设备(如交换机、路由器、防火墙等)中消耗的时间,它主要取决于网络设备当时的负载情况,是一个不断变化的量.在特定网络环境下,可变延迟可能在总延迟中占有较大的比重.综上考虑,根据地理位置的方法对就近抓取的贡献是十分有限的.

(4) 基于网络位置

提到网络位置,一个很直观的想法就是利用网站的 IP 地址,因为 IP 地址本身就是一个带有层级关系的描述符,很容易建立起树状分类结构.但是,由于 IP 地址分配时的随机性,可能 IP 地址相近的两台主机在网络位置和物理位置上并不相近,甚至分布在两个不同的国家.所以,IP 地址并不能代表网络位置,单纯通过 IP 地址进行 Web 划分并不是理想的选择.另一个想法是利用网络中的自治域(autonomous system,简称 AS),这是一种由运营商或组织建立的 Internet 的子网.通常认为,自治域内的节点间网络互联质量较高.但是与 IP 地址一样,由于自治域并非网络距离上的概念,其构建过程综合了人力、物力、财力等诸多因素.所以,在同一个自治域中的节点并不能保证其网络距离相对较近.比如,中国的教育网(China Education and Research Network,简称 CERNET)只由一个自治域构成,由于全国各地的高校都接入该网,使得这个自治域与中国的国土面积一样庞大,难以保证自治域内节点间的高效通信.

IPMicra[10]提出利用 RIRs(regional internet registries)记录的运营商、子网等信息对 IP 地址进行分类,建立一个有层级关系的分类树.由于 RIRs 是非政府组织,没有强制执行的权利,所以它的记录完整性是一个很大的问题.在实验中我们发现,RIRs 的记录并不全面并且不能实时更新.利用 RIRs 这种第三方服务的好处是能够获得更多的信息指导 Web 划分,并且系统开发者不需要自己去寻找这些信息;但是,第三方服务通常提供的是比指导 Web 划分更加通用的功能,并且其提供的信息往往是不完备的,因此其达到的效果有一定的限度.

除了上述方法以外,基于网络位置的 Web 划分还有两个潜在的研究方向:网络拓扑和网络距离预测.

网络拓扑是网络的逻辑模型,通常以图的方式表示.网络中的主机和路由器对应图中的节点,主机和路由器之间的链路对应图中的边(有向或无向),链路上的带宽、延迟等参数对应边的权重.完成对网络拓扑图的划分实际上也就完成了 Web 划分,并且由于爬虫只关心网络中的 Web 服务器,不关心普通主机,因此相对于划分全网的拓扑图,为爬虫系统计算 Web 划分的工作量要小很多,虽然从绝对数量来说仍然很大.在实践上,一般采取 traceroute 等方法来获取网络拓扑;在学术上,还可以通过网络模拟器等方法进行研究,划分方法涉及图的划分和聚类等.哈尔滨工业大学开发的大规模网络拓扑结构自动发现系统[18]在这方面取得了一定的研究成果.

网络距离预测是一种建立网络端到端模型的思路.最早进行网络距离预测的是 IDMaps[19,20]项目.它采用类似三角不等式的方法,通过分布于网络中的若干 Tracer 估算出 Internet 中任意两个 IP 之间的网络距离.但是,这种对端到端距离的估算难以完成对全网的建模.于是,网络坐标的概念应运而生.网络坐标研究的目标是将

Internet 映射为多维几何空间,为 Internet 上的每个主机分配一个几何空间坐标,使各个主机间的坐标距离尽可能地与其实际网络距离相同.最早提出网络坐标概念的是 GNP[21]项目,后来又涌现出 NPS[22],PIC[23],LightHouse[24],Vivaldi(引文)等系统,进一步优化了网络坐标系统的性能、精度和健壮性.对于网络坐标来说,Web 划分就是将几何空间中的坐标点划分为多个集合,也是一个聚类的过程.但是,网络坐标方法具有网络拓扑方法所不能比拟的优势:(1) 建模开销小,生成网络坐标需要的网络通信量远远小于生成网络拓扑时用到的通信量;(2) 端到端的网络距离估算速度快,仅需要通过一个基于坐标的计算公式进行少量数学运算,而网络拓扑方法需要在图中做一次路径搜索.该方法的缺点是生成坐标时进行的网络测量较少,得到的网络距离值没有网络拓扑方法精确.

2.4 Web划分实现方法

一个实际可运行的分布式 Web 爬虫对以上提出的几种 Web 划分策略需要选择程序的实现方法.Web 划分的程序实现可以分为动态划分和静态划分两大类.程序选择哪种实现方法,取决于爬虫系统的设计目标、开发方已有资源等因素.

(1) 静态划分

所有 Web 划分集合在系统启动前就已经划分完毕,在爬取过程中,当发现新 URL 时,爬虫根据静态配置就可以判断出这个 URL 属于哪个划分集合.通常在实现中,系统中所有 Agent 会共用一个静态划分配置文件,这个文件要么存储在一个中心节点,要么在各个 Agent 上都有一份一致的拷贝.

静态划分的优点是容易实现、系统内通信负载较小、抓取效率较高;缺点是不能应付如 Agent 意外退出等特殊事件,不能进行动态负载均衡,可扩展性也不强.要实现静态划分的系统通常已经拥有了所要抓取的大部分网站的信息,这些信息可能来自于预先探测以及其他爬虫系统的抓取结果.

(2) 动态划分

系统通过计算动态决定新发现的 URL 属于哪个 Web 划分集合,这样的系统通常需要一个调度中心节点来进行 Web 划分操作.有的系统(如 Apoidea[13])摒弃了中心控制节点,并实现了动态划分.这样的系统在系统架构的选择上局限性比较大,通常采用目前比较成熟的 DHT 算法(如 Chord[25]等)实现.

动态划分的优点是适应性强,能够应付 Agent 的意外退出等特殊情况,并且能够做到动态负载均衡;缺点是动态划分算法会带来大量系统内网络通信,提高了系统负载.实现动态划分既要实现 Web 划分策略又要兼顾系统服务质量,是一个非常有趣的研究点.

无论采用何种实现方法,Web 划分的实现都需要由系统中的多个 Agent 共同协作完成,由此引出 Agent 协同问题.

3 Agent 协同

分布式爬虫内部多 Agent 间的高效协同,是分布式 Web 信息获取中另一个亟待深入研究的关键问题.分布式 Web 爬虫 Agent 之间协同的方法和策略称为 Agent 协同模式.Agent 协同模式的选择对爬虫系统的设计意义重大.同时,Agent 交互过程中的通信策略及 URL 交换方案也是非常重要的研究问题.

3.1 Agent协同模式

目前,分布式 Web 爬虫的 Agent 协同模式主要有 4 种实现方式:独立获取、有调度中心的协同、无调度中心的协同及 Agent 自主与调度中心辅助相结合.

(1) 独立获取

系统中的每个 Agent 从各自预先设定的种子 URL 开始独立地下载网页,并不进行任何协同和 Web 划分.这种工作方式实现非常简单,一种可行的方案是在世界各地部署多个 Agent,每个 Agent(通常是一个机群)独立工作,由于各个 Agent 所在区域不同,其抓取策略也可以不同,以迎合当地的地域特点.但是,由于 Web 链接结构的复杂性,该方法造成不同 Agent 下载相同网页,从而大大增加了网页重复率,浪费了资源.该方案已被一些商业

系统所采用.

(2) 有调度中心的协同

系统需要通过调度中心为各个 Agent 分配要抓取的 URL.Agent 在抓取过程中发现新的 URL 以后,将新 URL 交给调度中心,由调度中心选择合适的 Agent,并将 URL 推送给这个 Agent.集中式是基于 LAN 的分布式爬虫的首选,也是很多基于 WAN 的分布式爬虫的实现方式.调度中心的存在,一方面使得对全系统的精确控制成为可能,同时也有利于系统统一的分析 Web 链接之间的关系,为网页重要程度的分析打下基础;另一方面,调度中心也引发了单机负载瓶颈及单点失效等问题,影响了系统的可扩展性.

(3) 无调度中心的协同

该方法摒弃了调度中心,将计算 Web 划分及推送 URL 的功能都赋予了 Agent.为了保证所有 Agent 的调度一致性,目前使用此种协同模式的分布式 Web 爬虫都是基于 DHT 实现.DHT 无中心的特性使得系统能够在 Internet 上扩展,但是 DHT 往往存在逻辑网络与真实网络之间的拓扑一致性问题,即哈希值近的两点其实际网络距离可能非常远,从而引发低带宽和高延迟等问题,给高效抓取网页带来了困难.Loo[26]和 Singh[13]等人提出利用一些动态搜集的网络信息弥补系统在这方面的不足;在 P2P 领域,还有一些研究[27,28]试图通过建立网络位置与 DHT(如 Chord[25],CAN[29]等)之间的映射关系,从根本上解决拓扑一致性问题,但是这些研究并没有在分布式爬虫领域得到应用.

(4) Agent 自主与调度中心辅助相结合

该方法结合了前两种方法的特点.这种系统中存在调度中心节点,但是与有调度中心的系统相比,调度中心的作用仅仅是对 Agent 的工作进行辅助.如同 DNS 系统及 DNS 缓存的原理,调度中心提供原始的 Web 划分知识.Agent 在未获得知识的情况下,将向调度中心询问 URL 的调度方法,同时将这些信息存入自身知识库.随着 Agent 所处理过的 URL 数量的增多,其自身知识库的信息量也不断增加,从而可以脱离调度中心,自主计算 Web 划分.本方法的优点是,与无调度中心系统相比加入了调度指导者,与有调度中心系统相比减少了系统内的通信量;缺点是难以使各个 Agent 上的知识保持一致,可能造成各个 Agent 的 Web 划分结果互相矛盾,因此必须花费一定的开销进行 Agent 间同步.

3.2 Agent间通信策略

由于跨分区链接的存在,使得 Agent 必须将不属于自己管辖的 Web 划分集合的 URL 直接或间接(通过调度中心再进行一次下发操作)地推送到其他合适的 Agent,从而引发 Agent 间通信.虽然 Agent 之间、Agent 与调度中心(如果有的话)之间不仅仅进行关于跨分区链接的通信,但是这种通信占据了系统内部通信量的大部分,因此相关研究都是以跨分区链接交换为中心进行的.

根据对跨分区链接处理方式的不同,Cho 等人[6]将分布式 Web 爬虫的 Agent 间通信策略分为以下 3 种:

(1) Firewall 方式

本方法适用于以 Agent 独立获取方式运行的系统.系统中,每个爬虫各自下载自己承担的 Web 划分,爬虫之间不互相交换跨分区链接,即如果爬虫发现跨分区链接则直接将其丢弃.在这种模式下,爬虫之间不会产生跨分区链接交换,在 Web 划分集合之间没有交集的情况下也不会出现网页重复下载,但是很有可能丢失部分网页.因为在抓取过程中,所有的跨分区链接都被丢弃了,而很可能有些网页只有通过这些跨分区链接才能访问 得到.

(2) Cross-Over 方式

本方法也是一种 Agent 独立获取的实例.系统中的每个爬虫各自下载自己承担的 Web 划分,当爬虫发现跨分区链接时不是丢弃,而是继续沿链接向下抓取.与 Firewall 方式相同,爬虫之间不互相交换跨分区链接.在这种模式下,爬虫之间同样不会产生 URL 交换,但是很可能出现网页重复下载.比如,A 爬虫发现了指向网页 W 的跨分区链接,B 爬虫也发现了指向网页 W 的分区内链接,那么 A 爬虫和 B 爬虫将都会下载网页 W.让人乐观的是,这种模式的系统下载的总的不重复的网页数要比在 Firewall 模式下下载到的多,因为下载过程中没有一个链接被丢弃.但是在这种模式下,最后得到的所有网页在各个爬虫上的分布很可能与最初的 Web 划分大相径庭,给网页的去重和分类工作造成了麻烦.

(3) Exchange 方式

本方法是被广泛接受的主流方法,适用于有调度中心及无调度中心的系统.系统中的每个爬虫各自下载自己承担的Web划分,当爬虫发现跨分区链接时,将跨分区链接发送给承担这个链接所在Web划分的爬虫.发现跨分区链接的那个爬虫并不沿着这个链接继续抓取.这种模式与前两种相比,基本不会出现网页重复下载,并且不会丢失链接,但是其最大的缺点是实现复杂并且会产生大量的系统内通信.因此在本方式的设计中,既要实现爬虫间互相通信,又要控制爬虫间的通信量以减小系统负载.

3.3 跨分区链接交换量最小化

Cho等人[6]还指出,采用Exchange模式工作的分布式爬虫可以采用批量交换和去重“著名网页”的方法减小Agent 间的跨分区链接交换量.下面结合我们的经验对这两种方法进行介绍.由于跨分区链接实际上就是 URL,以下均简称为URL.

(1) 批量交换

Agent 发现跨分区链接后并不是立即将其发送出去,而是等跨分区链接积攒到一定的数量或者到达了一定的时间间隔后再批量发送,一次通信发送多个URL(即跨分区链接),发送时需要把积攒的URL先根据Web划分分类,不同的分类发送给不同的Agent.这里,对于积攒的URL的存储方法,可以使用内存存储、磁盘文件存储以及内存与磁盘文件相结合的存储.

批量交换有两个优点:(1) 减少总的通信次数.如果把一次 Agent 间通信当作一次会话,那么批量通信产生的会话次数要少于非批量通信(即每个URL单独通信).(2) 由于URL需要先在Agent本地缓存一段时间,如果在这段时间内又发现了相同的链接,则可以直接在缓存中去重,以避免或减少 URL 的重复发送.我们把这个过程称为URL交换中的“发送方去重”.

发送方去重的作用是有限的,因为其范围仅限于Agent缓存的尚未发送的那些URL(即跨分区链接).如果第一次发现的链接 W 已经被发送,那么后来发现的 W 将不会被去重,仍然造成重复发送.由于URL发送方的URL缓存大小、发送时间间隔的有限性以及Web链接结构的随机性,造成在Agent之间交换的URL中必然存在大量无法避免的重复,这就要求URL接收方必须有能力将重复接收到的URL去重,这称为URL交换中的“接收方去重”.接收方去重比起发送方去重更能彻底地去除URL重复现象,由此可见,接收方去重是为避免URL重复而必须实现的功能.

另外,批量交换在实现中需要考虑是等URL积累到一定数量后才发送,还是每过一个固定的时间间隔才发送.前者每次可以发送较多的URL,但是如果长时间凑不齐规定的URL数量会导致长时间无法进行URL交换;后者由于没考虑到URL数量,所以有可能一次只发送很少的URL,多浪费了带宽,还可能一次发送太多的URL,造成瞬间网络负载过大.较好的办法是两者同时使用,即等 URL 积累到一定数量后才发送,但是如果到达了固定的间隔时间还没有凑齐规定的URL数量,就将现有的URL全部发送.或者反过来,即每过一个固定的时间间隔才发送,但是如果间隔时间到来之前 URL 数量就已经达到上限,则不等时间到来直接发送.数量和时间这两个参数必须根据系统规模、吞吐能力、网络环境等情况进行调优.

(2) 去重“著名网页”

指向特定网页的链接称为该网页的“进入链接(incoming links)”.研究表明,Web 上的进入链接的数量符合Zipf 分布,即只有少数网页拥有大量的进入链接,而大部分网页的进入链接数很少,也就是引用量很小.这里,我们把进入链接多的网页称为“著名网页”.如果能够对“著名网页”的URL进行较好的去重,则能在一定程度上减少Agent间的URL交换量,因为大量URL(跨分区链接)都是指向这些“著名网页”的.

“著名网页”可以根据网站知名度采用手工配置的方法,但是手工方法很可能漏掉许多网页,所以可以要求Agent在工作过程中实时地发现“著名网页”.一种实现方案是:在每个Agent上都建立一个独立的“已发送URL缓存”,用来缓存在以前一段时间内发送过的跨分区链接.在这个缓存上,可以按照发现次数将所有URL进行排队,发现次数最多的URL(比如达到某个上限值),将被视为是指向“著名网页”的URL,从而可以考虑用特殊的方式对这个URL进行去重.在具体实现中,可以将URL在缓存中的寿命与URL的发现次数挂钩.这样,URL越“著

名”,它在缓存中存在的时间就越长.爬虫每发现一个跨分区链接,就把链接拿到“已发送 URL 缓存”进行匹配,如果能够匹配则不发送,这样就可以保证“著名网页”URL 的较少发送.由于爬虫每隔一定时间就会将已抓取的全部网页再次抓取一遍(即内容更新),所以还有一种方法是利用前一次或几次抓取时的记录总结出“著名网页”,并用于指导下一轮抓取.

4 Agent 部署

Agent 部署是广域网分布式爬虫系统的底层问题,本节针对真实系统的部署方案展开论述.

4.1 使用可控平台和专用网络

广域网分布式爬虫系统可以架设于科研机构和企业自己所拥有的可控平台之上.这样的企业级基础设施通常拥有较高的节点间互联速度,各个节点的使用情况较为可控,节点的加入和退出会有全局登记和通知,由专人负责.这些特性使得爬虫程序在设计时可以采用相对简单的策略,比如采用静态划分、采用有调度中心的 Agent 协同等.

为了提高系统性能,可以通过专用网络而不是 Internet 对分布在各处的 Agent 进行互联.很多科幻作品中常有这样的情节:在宇宙中航行的飞船可以通过一种“超空间”,快速地从宇宙的 A 位置跳跃到 B 位置,跳跃所消耗的时间远远小于航行 A 位置与 B 位置之间直线距离所用的时间.这种“快速通道”在物理世界中还难以实现,但是在 Internet 上却是确确实实存在的现象.使用专用网络,理论上就可以实现这样的“超空间”.专用网络虽然跨越了较大的物理距离,但是网络通信效率很高.由于专用网络中节点比较少,因此也不会出现像 Internet 那样的网络拥塞.如果把这一特性加以利用,比如 Agent 下载网页使用 Internet,而 Agent 间通信以及其他数据传输使用专用网络,则会大大提高系统效率.

我们曾经利用 CNCERT 的专用网络测量了北京与广州之间的网络延迟,结果如图 2 所示.该专用网络上的节点之间可以通过专用网络互联(使用私网 IP),也可以通过 Internet 互联(使用公网 IP).对比图上方的一组延迟值是使用私网 IP 在专用网络上获得的,下方的一组延迟值是使用公网 IP 在 Internet 上获得的.实验结果显示,Internet 上的延迟值几乎达到了专用网络延迟值的 10 倍.

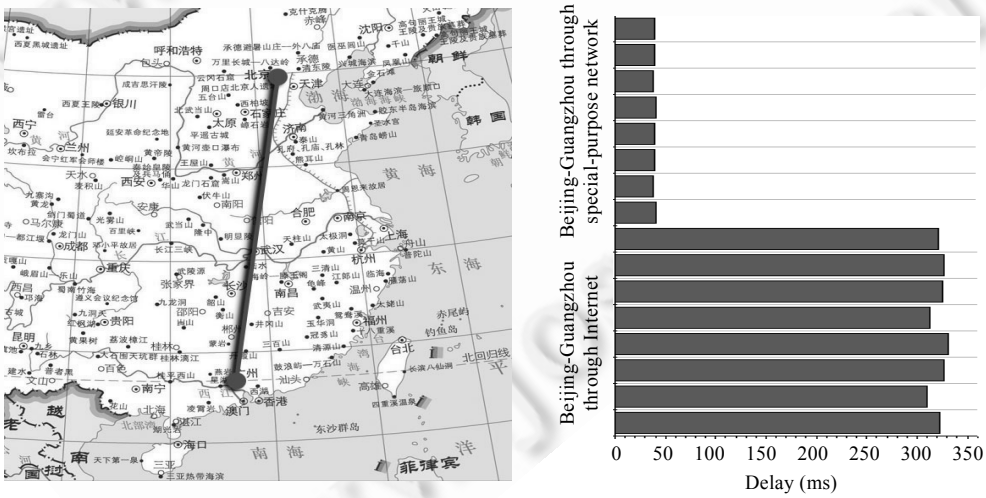


Fig.2 Special purpose network vs. Internet on Beijing-Guangzhou network delay

图 2 北京到广州之间专用网络延迟与 Internet 延迟对比图

可控平台和专用网络带来的最大缺点是维护开销较大.对于广域网分布式系统,更是需要企业同时在多个物理位置维护多组这样的可控平台.因此,一些小规模的公司和研究团体只能寻求更加节省开销的解决方案.另一个缺点是部署环境受限:首先,如何找到空间安放设备是一个复杂的问题;其次,由于当地网络结构的限制,可

控平台的网络接入点环境很难达到预期的效果.在实践中,我们甚至遇到过数据包在一座大楼中跨越多个网络自治域又返回原自治域的情况.很多情况下,这类问题不是技术所能解决的.

4.2 利用普通网络用户的资源

利用互联网用户的闲散资源是分布式计算的重要目标,同样也可以应用到广域网分布式爬虫系统中.类似于已经非常成熟的 P2P 文件共享应用,一些系统的部署环境不是基于专用平台,而是私人拥有的个人电脑.网络用户从特定网站下载安装程序并安装后,即可将其个人电脑变成爬虫系统中的一个 Agent,参与到整个爬虫系统的工作中来.由于每个用户都拥有自己独享的网络带宽,加入的 Agent 越多,系统的总带宽就越大,理论上可以无限扩展.

一个特别的例子是 Faroo(http://www.faroo.com/),它提出了更加新颖的爬虫实现方案.在该系统中,安装在用户个人电脑中的爬虫程序的日常工作并非抓取网页,而是监听用户的浏览器动作,将用户在其浏览器上浏览过的网页存储下来进行处理.这种实现方法大大降低了系统给 Internet 带来的网络负载,免去了 Web 划分的工作,但是也使得用户的搜索范围只能集中在“所有 Faroo 用户所浏览过的网页”这样一个有限的集合中.之后,相应的 Rank 工作也是根据 Faroo 用户对网页的点击量确定的.

这种利用 Internet 草根资源的系统为 Web 爬虫的发展开辟了一条新的道路,但是仍然面临一些严重的问题:

(1) 难以吸引网络用户加入系统

不同于文件共享系统,爬虫系统并不能带给加入者直接的好处.另外,由于大部分互联网用户上网需要缴费,运行爬虫这种带宽占用较多的程序反而会给加入者带来经济上的额外开销.因此,各种系统采用五花八门的方式提供鼓励机制.比如,Faroo 为加入者反馈公司利润;Majestic(http://www.majestic12.co.uk/)设立全球加入者贡献排名;YaCy(http://yacy.net/)只允许加入者享有查询索引的权利,并提供个性化定制搜索;甚至有人提出把贡献带宽和计算资源作为一种“义务劳动”等等.但是到目前为止,还没有哪个系统发展到大型商业搜索引擎的规模.

(2) 部署环境不可控,程序设计复杂

因为系统通常面临 Agent 频繁加入/退出的情况,所以必须保证系统的健壮性和容错能力;对于无调度中心的系统,还要尽量避免覆盖网断裂为多个独立子网的问题.来自网络黑客的攻击也是一个需要考虑的方面.另外,系统设计在面临复杂性问题的同时,还要不断地权衡容错能力与工作效率的矛盾.由于爬虫所执行的任务并没有一个精确的描述,对数据总体的完整性要求不高,一个可行的办法是在保证抓取覆盖率的前提下适当地降低容错能力,允许一定量数据的丢失.

5 评价模型

Cho 等人[6]在其综述中已经详细给出了诸如重复率、覆盖率、网页质量等传统的爬虫评价标准.但是由于广域网分布式爬虫与传统爬虫系统相比具有很多独特的属性,因此仅凭借传统标准并不能完整地描述出新系统的性能.本文结合已有的研究工作给出如下几种针对广域网分布式爬虫系统的评价标准:

5.1 节点-资源距离

组成广域网分布式爬虫系统的各个节点(即爬虫)分布于 Internet 的若干位置.如果部署在不同位置的爬虫都能负责抓取距离自己相对较近的网站,那么将会大幅缩短下载 Web 数据所需要的时间;同时,如第 1.3 节所述,这种就近获取的方案还能降低系统对 Internet 施加的负载.就近获取主要通过 Web 划分方法实现.

设系统中含有的 Agent 的集合 $A=\{\alpha_1,\alpha_2,\alpha_3,\ldots,\alpha_N\}$,目标网站的集合 $W=\{\omega_1,\omega_2,\omega_3,\ldots,\omega_M\}$,经过某种 Web 划分后,Agent$\alpha_i$ 负责获取网站ω_j.定义α_i 与ω_j 之间的“距离”为节点-资源距离(peer-resource distance,简称 PRD),标记为 $L(\alpha_i,\omega_j)$.全系统中,节点-资源距离的数量与目标网站的数量同为 M.全系统的就近获取能力可以通过 M 个节点-资源距离的分布或节点-资源距离的算术平均值体现.系统实现中,“距离”常常代表地理距离、网络延迟或下载速率等信息.例如,文献[26]中采用网络延迟(round trip time,简称 RTT)作为节点-资源距离的度量,并通过分

析节点-资源距离的分布评价多种 Web 划分策略的优劣(如图 3 左图(即文献[26]中的 Figure 12)所示).文献[10]则采用网页的下载速率作为度量,评估的是基于地理位置的 Web 划分(如图 3 右图(即文献[10]中的 Fig.5)所示).

总体而言,最直观的“距离”就是下载速率,这也是系统最直接的目标之一.但是,这个数值是一个后验知识,只有在真正对网站进行抓取时才能测量得到.在先验知识中,最易于获得的是网络延迟(或 RTT),它可以通过 ping,http-ping(http://www.coretechnologies.com/products/http-ping/)等简单方法来获得.但是,网络延迟处于不断的变化中,需要长期测量才能获得稳定的数值.在研究中,采用网络延迟作为“距离”度量的方法更易于在模拟器上进行实验.也有一些网络延迟数据集,如 p2psim(http://pdos.csail.mit.edu/p2psim/)的 King(http://pdos.csail.mit.edu/p2psim/kingdata/)数据集等可供使用.

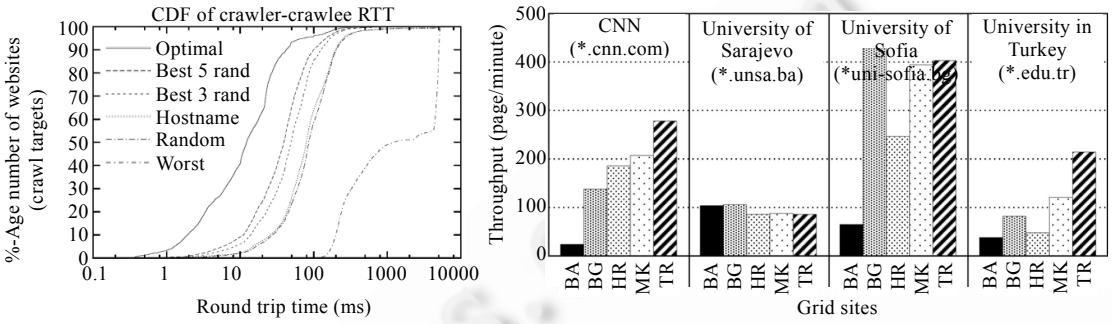


Fig.3 Evaluations of the peer-resource distances

图 3 节点-资源距离评价方法

5.2 总通信量

如前所述,目前对总通信量的研究都是面向 Agent 间跨分区链接交换的.当然,系统还存在全局控制信息、Agent 间同步、消息路由、容错消息等也需要在网络上传输,但是大部分对待这些通信的思路都是尽量减少和避免,没有评价标准.

分布式爬虫之间可能需要交换信息以进行协同工作,比如在 Exchange 方式下工作的爬虫需要按一定的时间间隔进行跨分区链接交换.为了评价这种信息交换的通信量,文献[6]中定义总通信量(communication overhead)为每下载一个网页所需交换的跨分区链接(未去重)的平均数量.设 E 是在抓取过程中 Agent 之间交换的跨分区链接所引发的通信数量,N 是总共抓取到的网页数,则总通信量为

$$Communication\ Overhead=E/N \tag{1}$$

如果一个分布式爬虫总共下载了 1 000 个网页,并且在下载过程中各个爬虫之间总共交换了 3 000 个跨分区链接,那么总通信量=3000/1000=3.在 Exchange 方式下工作的分布式爬虫应该尽量减小总通信量;而在非 Exchange 方式下工作的分布式爬虫由于爬虫之间不交换跨分区链接,因此总通信量为 0.文献[26]在不同的系统规模下测量了多种基于 DHT 的 Web 划分策略的总通信量(如图 4 所示,引自文献[26]的 Fig.5).由于文献[26]中的系统采用了 DHT 方法,因此爬虫-爬虫之间的通信并不是直接一步到达,而是要通过一个在 DHT 覆盖网上的路由过程.

关于总通信量,UbiCrawler[7]提出了一个非常有意义的理论:设每个网页平均有λ个链向其他网站的链接(当然,这是平均值),Agent 抓取 n 个网页就会产生λn 个需要交换的 URL.由于λn 个 URL 不一定全部需要在网络上传递,即不都是跨分区链接,有的可能在 Agent 本地处理,所以需要给λn 加一个系数.这个系数可以由 Agent 的承载能力决定.定义每个 Agent 的任务承担能力(capacity)(比如负责的主机数)为 C_α,从而对一个 Agent 有如下不等式成立:

$$\lambda n\frac{\sum_{\alpha\neq\bar{\alpha}}C_\alpha}{\sum_\alpha C_\alpha}<\lambda n \tag{2}$$

其中,α表示系统中任意存活的 Agent,$\bar{\alpha}$ 表示抓取了这 n 个网页的那个 Agent.系数实际上代表需要在网络上传递的跨分区链接在λn 个 URL 中所占的百分比.由上式可以看出,需要通过网络交换的跨分区链接数量上限与 Agent 数量是相互独立的,这个数量仅依赖于 Agent 下载的网页数和网页中包含的链接数.这就意味着,虽然从经验上来看,Agent 数量越多,引发的网络负载越重,但实际上,网络负载加重的真实原因是因为系统下载的网页数量的增加,而不是 Agent 数量的增加.所以,单纯 Agent 数目庞大并不能造成系统瓶颈.

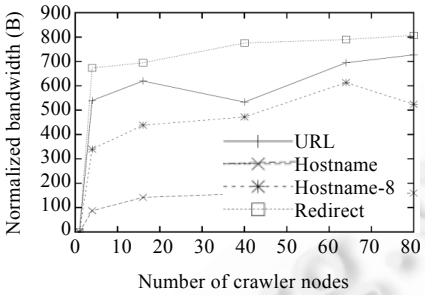


Fig.4 Evaluation of the communication overhead

图 4 总通信量的评价方法

5.3 可扩展性

对于分布式爬虫的可扩展性评价方法,学术界一直没有一致的意见,且提出的评价方法也非常稀少.较有代表性的是 UbiCrawler 提出的在系统规模不同(如 Agent 数量不同,或者每个 Agent 上的线程数不同)的情况下,使用比较每线程每秒下载的网页数的方法评价可扩展性.设系统中含有的 Agent 的集合 $A=\{\alpha_1,\alpha_2,\alpha_3,\ldots,\alpha_N\}$,每个 Agent 上的线程数分别为 $t_1,t_2,t_3,\ldots,t_N$,每个 Agent 的网页下载速率分别为 $v_1,v_2,v_3,\ldots,v_N$,则系统的可扩展性(scalability)可用以下公式来衡量:

$$Scalability=\frac{\sum_{i=0}^{N}v_i}{|A|} \tag{3}$$

另一种线程级的定义为

$$Scalability=\frac{\sum_{i=0}^{N}v_i}{\left|\sum_{j=0}^{N}t_j\right|} \tag{4}$$

以上公式揭示出作为一个可扩展性高的系统,应该保证每个线程的工作不会受到线程数增加的影响,即随着系统规模的扩大和通信量的增加,每个线程的性能不会减弱.文献[13]根据公式(3)给出可扩展性的评价(如图 5 的左图(即文献[13]中的 Fig.6)所示);文献[7]分别根据公式(3)和公式(4)进行了评价(分别如图 5 的中图和右图(即文献[7]中的 Figure 3)所示),但是线程级的评价是针对单机而非全系统.

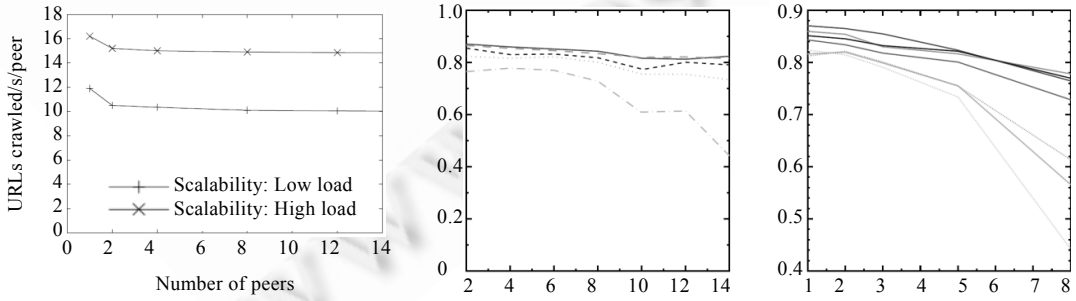


Fig.5 Evaluation of the scalability

图 5 可扩展性的评价方法

目前的定义完全从程序吞吐量角度评价可扩展性,给出了对不同规模系统进行比较的办法,但是并没有考虑到前面叙述的非技术层面的因素,因此还有待于进一步完善.

6 总结及未来研究

我们对全文各节进行总结,并针对各个问题提出未来研究中应该给予更多关注的研究点.表 1 总结了几种较为典型的广域网分布式 Web 爬虫的实现方法,以供读者参考.

Table 1 A summary of typical WAN-based distributed Web crawlers

表 1 几个典型广域网分布式 Web 爬虫的总结

	Web partition unit	Web partition strategy	Web partition implementation	Agent collaboration mode	Agent communication strategy	Agent deployment
UbiCrawler	Host	Consistent hashing	Dynamic	No central coordinator	Exchange	Controlled research platform
YaCy	N/A	N/A	N/A	No central coordinator	N/A	Resources contributed by Internet users
Faroo	Not needed	Not needed	Not needed	No agent collaboration	Not needed	Resources contributed by Internet users
Majestic	URL	N/A	N/A	No agent collaboration	Firewall	Resources contributed by Internet users
IPMicra	Host	Network position from RIRs	Static	Need central coordinator	N/A	Controlled research platform
SE4SEE	Host	Domain name	Static	No agent collaboration	Firewall	Controlled grid infrastructure
Apoidea	Host	DHT based random hash & network proximity	Dynamic	No central coordinator	Exchange	Controlled research platform
IglooG	URL	Random hash	Dynamic	Need central coordinator	Exchange	Controlled research platform

(1) Web 划分

为系统中各个 Agent 高效而合理地分配 URL 是 Web 划分所要解决的主要问题.确定 Web 划分方法的过程包括选取 Web 划分单元、选取 Web 划分策略以及选取 Web 划分实现方法.对于不同的系统实现目标,可以采取不同的组合.对 Web 的动态划分是一个非常有意义的研究方向;如何利用网络拓扑和网络坐标的研究成果为广域网分布式爬虫服务以实现就近抓取,将是一个创新性的研究点.

(2) Agent 协同

Agent 间如何进行互联和通信以实现多 Agent 之间的高效协作,是 Agent 协同所要解决的主要问题.要实现跨广域网的 Agent 协同,需要精心设计 Agent 协同模式和 Agent 间通信策略,并尽量减小 Agent 间的通信量.在未来的研究中,Agent 自主与调度中心辅助相结合的方式仍有知识描述、Agent 知识库间同步等问题需要解决;对无调度中心协同的研究需要与 P2P 算法相结合,并设法解决逻辑网络与物理网络的拓扑一致性问题.DHT 与网络坐标相结合可能是一个理想的选择.

(3) Agent 部署

Agent 部署更加注重工程方面的问题,系统构建者可以采用可控平台和专用网络资源搭建系统底层,也可以寻求利用普通网络用户资源的方法.研究可以关注不可控部署环境下爬虫系统的健壮性和容错能力等问题.

(4) 评价标准

目前为止,没有一个系统采用了完全相同的评价标准,并且对于容错能力等还没有具体的评价方法.

致谢 在此,我们向对本文的工作给予建议和帮助的老师和同学表示感谢.

References:

[1] CNNIC. The 21st statistical survey report on the Internet development in China. 2008 (in Chinese). http://www.cnnic.net.cn/uploadfiles/pdf/2008/1/17/104156.pdf

[2] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 1998, 30(1-7):107−117. [doi: 10.1016/S0169-7552(98)00110-X]

[3] Burner M. Crawling towards eternity—Building an archive of the World Wide Web. Web Techniques Magazine, 1997,2(5):37−40.

[4] Heydon A, Najork M. Mercator: A scalable, extensible Web crawler. World Wide Web, 1999,2(4):219−229. [doi: 10.1023/A:1019213109274]

[5] Korpela E, Werthimer D, Anderson D, Cobb J, Lebofsky M. SETI@HOME—Massively distributed computing for SETI. Computing in Science & Engineering, 2001,3(1):78−83. [doi: 10.1109/5992.895191]

[6] Cho J, Garcia-Molina H. Parallel crawlers. In: Proc. of the 11th Int'l Conf. on World Wide Web. New York: ACM Press, 2002. 124−135.

[7] Boldi P, Codenotti B, Santini M, Vigna S. Ubicrawler: A scalable fully distributed Web crawler. Software-Practice & Experience, 2004,34(8):711−726.

[8] Boswell D. Distributed high-performance Web crawlers: A survey of the state of the art. 2003. http://www.cs.ucsd.edu/~dboswell/PastWork/WebCrawlingSurvey.pdf

[9] Baeza-Yates R, Castillo C, Junqueira F, Plachouras V, Silvestri F. Challenges in distributed information retrieval. In: Proc. of the Int'l Conf. on Data Engineering (ICDE). Washington: IEEE Computer Society Press, 2007.

[10] Papapetrou O, Samaras G. IPMicra: An IP-address based location aware distributed Web crawler. In: Proc. of the 5th Int'l Conf. on Internet Computing (IC 2004). 2004. 694−699.

[11] Cambazoglu BB, Karaca E, Kucukyilmaz T, Turk A, Aykanat C. Architecture of a grid-enabled Web search engine. Information Processing and Management, 2007,43(3):609−623. [doi: 10.1016/j.ipm.2006.10.011]

[12] Foster I, Kesselman C, Wrote; Jin H, Yuan PP, Shi K, Trans. The Grid 2: Blueprint for a New Computing Infrastructure—Application Tuning and Adaptation (2nd ed.). Beijing: Publishing House of Electronics Industry, 2004 (in Chinese).

[13] Singh A, Srivatsa M, Liu L, Miller T. Apoidea: A decentralized peer-to-peer architecture for crawling the World Wide Web. In: Proc. of the SIGIR 2003 Workshop on Distributed Information Retrieval. 2004. 126−142.

[14] Li XM, Yan HF, Wang JM. Search Engine: Principle, Technology and System. Beijing: Science Press, 2005 (in Chinese).

[15] Ye YM, Yu S, Ma FY, Song H, Zhang L. On distributed Web crawler: Architecture, algorithms and strategy. Acta Electronica Sinica, 2002,30(12A):2008−2011 (in Chinese with English abstract).

[16] Karger D, Lehman E, Leighton T, Levine M, Lewin D, Panigrahy R. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In: Proc. of the ACM Symp. on Theory of Computing. New York: ACM Press, 1997. 654−663.

[17] Exposto J, Macedo J, Pina A, Alves A, Rufino J. Geographical partition for distributed Web crawling. In: Proc. of the 2005 Workshop on Geographic Information Retrieval. New York: ACM Press, 2005. 55−60.

[18] Jiang Y, Hu MZ, Fang BX, Zhang HL. An Internet router level topology automatically discovery system. Journal of China Institute of Communications, 2002,23(12):54−62 (in Chinese with English abstract).

[19] Francis P, Jamin S, Jin C, Jin Y, Raz D, Shavitt Y, Zhang L. IDMaps: A global Internet host distance estimation service. IEEE/ACM Trans. on Networking, 2001,9(5):525−540. [doi: 10.1109/90.958323]

[20] Francis P, Jamin S, Paxson V, Zhang LX, Gryniewicz DF, Yin YX. An architecture for a global internet host distance estimation service. In: Proc. of the 8th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM'99). Washington: IEEE Computer Society Press, 1999. 210−217.

[21] Ng TSE, Zhang H. Towards global network positioning. In: Proc. of the ACM SIGCOMM Internet Measurement Workshop. New York: ACM Press, 2001. 25−29.

[22] Ng TSE, Zhang H. A network positioning system for the Internet. In: Proc. of the USENIX Annual Technical Conf. Berkeley: USENIX Association, 2004. 141−154.

[23] Costa M, Castro M, Rowstron A, Key P. PIC: Practical Internet coordinates for distance estimation. In: Proc. of the Int'l Conf. on Distributed Systems. Washington: IEEE Computer Society Press, 2004.

[24] Pias M, Crowcroft J, Wilbur S, Harris T, Bhatti S. Lighthouses for scalable distributed location. In: Proc. of the 2nd Int'l Workshop on Peer-to-Peer Systems (IPTPS 2003). Berlin, Heidelberg: Springer-Verlag, 2003. 278−291.

[25] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proc. of the 2001 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM Press, 2001. 149−160.

[26] Loo BT, Cooper O, Krishnamurthy S. Distributed Web crawling over DHTs. Technical Report, CSD-4-1305, Berkeley: Technical Department of Electrical Engineering and Computer Sciences, University of California, 2004.

[27] Doi K, Tagashira S, Fujita S. Proximity-Aware content addressable network based on Vivaldi network coordinate system. In: Proc. of the 5th Int'l Workshop on Databases, Information Systems and Peer-to-Peer Computing. 2007.

[28] Nikolaos E, Athanasios C, Spyros D, Odysseas K. L-CAN: Locality aware structured overlay for P2P live streaming. In: Proc. of the 11th IFIP/IEEE Int'l Conf. on Management of Multimedia and Mobile Networks and Services: Management of Converged Multimedia Networks and Services. Berlin, Heidelberg: Springer-Verlag, 2008. 77−90.

[29] Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content addressable network. In: Proc. of the 2001 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM Press, 2001. 161−172.

附中文参考文献:

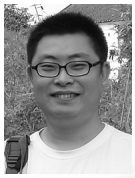
[1] CNNIC.第 21 次中国互联网络发展状况统计报告.2008.

[12] Foster I,Kesselman C,著;金海,袁平鹏,石柯,译.网格计算(第二版).北京:电子工业出版社,2004.

[14] 李晓明,闫宏飞,王继民.搜索引擎:原理、技术与系统.北京:科学出版社,2005.

[15] 叶允明,于水,马范援,等.分布式 Web Crawler 的研究:结构、算法和策略.电子学报,2002,30(12A):2008−2011.

[18] 姜誉,胡铭曾,方滨兴.一个 Internet 路由器级拓扑自动发现系统.通信学报,2002,23(12):54−62.



许笑(1983－),男,山东淄博人,博士生,主要研究领域为网格计算,分布式系统.

张宏莉(1973－),女,博士,教授,博士生导师,CCF 会员,主要研究领域为网络安全,网络计算.

张伟哲(1976－),男,博士,副教授,CCF 会员,主要研究领域为网络计算,网络安全.

方滨兴(1960－),男,博士,教授,博士生导师,中国工程院院士,CCF 高级会员,主要研究领域为网络安全,网络计算.