高技术通讯 2010 年 第 20 卷 第 3 期:239 ~ 245

doi:10.3772/j.issn.1002-0470.2010.03.004

广域网分布式爬虫中的 Agent 协同与 Web 划分研究①

许 笑② 张伟哲 张宏莉 方滨兴③

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 针对广域网环境下分布式 Web 爬虫的 Agent 协同和 Web 划分两个核心问题进行深入研究,提出了基于顾问服务的分布式 Web 爬虫系统模型,给出了详细的系统设计方案及 Agent 协同算法框架,并通过推导证明了顾问服务参与 Agent 协同能够使分布式爬虫系统承受相对较小的网络负载。提出了分布式 Web 爬虫 Web 划分的概念,围绕 Web 划分单元选取及 Web 划分策略,对 Web 划分的分类和实现进行了详细的讨论,并通过实验对多种 Web 划分方法进行了对比和评价,验证了广域网系统相对于局域网系统的优势,并发现运营商互连因素对爬虫系统性能的影响大于地理位置因素的影响。

关键词 分布式 Web 爬虫, Agent 协同, Web 划分, 顾问服务

0 引 言

分布式 Web 爬虫[1]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内部(本文称此时的 Web 爬虫为局域网分布式 Web 爬虫),或分布在广域网的不同地理位置和网络位置(本文称此时的 Web 爬虫为广域网分布式 Web 爬虫),每个 Agent 以多进程或多线程方式通过并发保持多个 TCP(传输控制协议)链接获取 Web 信息。部署在局域网内的分布式 Web 爬虫目前已得到广泛使用。较为著名的有 Google[2]、AltaVista 的 Internet Archive Crawler[3]及 Mercater[4]等,我国学术界也在该领域进行了一些工作,如北大天网[5]、Igloo[6,7]和 Chao[8]等。但是部署于局域网上的分布式 Web 爬虫受到带宽等因素的制约[9],即使系统中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。

广域网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及可扩展性好等优点,已经成为商业界和学术界的优选方案。在商业界,其思路一般是公司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(如文献[10]的个性化匿名检索),甚至将利润反馈给用户(如文献[11])。在学术方面,Cho[12]等人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,并提出广域网分布式爬虫的若干优点。UbiCrawer[13]扩展了文献[12]中的一些概念,并第一个声称可以支持广域网分布式平台。IPMicra[14]是第一个基于位置信息调度的广域网分布式爬虫,它利用地区(大洲级)互联网注册机构(Regional Internet Registries,RIRs)的注册信息生成 IP 地址层级结构在 Agent 间分配统一资源定位符(uniform resource locators,URL),但是 RIRs 所能提供的信息是相对有限的。SE4SEE[15]实现了基于网格的广域网分布式爬虫,但是其根据网页语言种类的调度策略仅适用于国土面积较小的国家。Apoidea[16]用 P2P 的方式在广域网上组织爬虫节点,但是由于 P2P 覆盖网的拓扑不匹配问题,为高效抓取网页造成了困难。

本文详细讨论了 Agent 协同和 Web 划分这两个分布式 Web 爬虫的核心问题,提出了一个新的基于顾问服务的分布式 Web 爬虫系统模型,给出了相应的 Agent 协同算法,并对其性能进行分析;之后,给出分布式 Web 爬虫的实验,对不同的 Web 划分方法下的实验结果进行对比分析,并对局域网和广域网

① 863 计划(2009AA01Z437),973 计划(G2005CB321806),国家自然科学基金(60703014),高等学校博士学科点专项科研基金(20070213044)和哈尔滨工业大学优秀青年教师培养计划(HITQNJS.2007.034)资助项目。
② 男,1983 年生,硕士,研究方向:网格计算,分布系统;E-mail:smilestor@gmail.com
③ 通讯作者,E-mail: bxfang@ict.ac.cn
(收稿日期:2008-11-24)

两种部署环境下的系统性能进行对比分析,验证了基于顾问服务的分布式 Web 爬虫协同算法的有效性。

1 系统模型

本文采用了基于顾问服务的分布式 Web 爬虫系统模型。它的结构介于有调度中心系统模型(如文献[14,15])和无调度中心系统模型(如文献[16])之间。在系统中设置顾问服务的思想源于 Internet 上的域名系统(domain name system,DNS)服务,Agent 在运行中会像网络浏览器查询 DNS 一样在必要时向顾问服务发起请求。系统如图 1 所示,共分为管理服务、爬虫、顾问服务、存储服务 4 个部分。

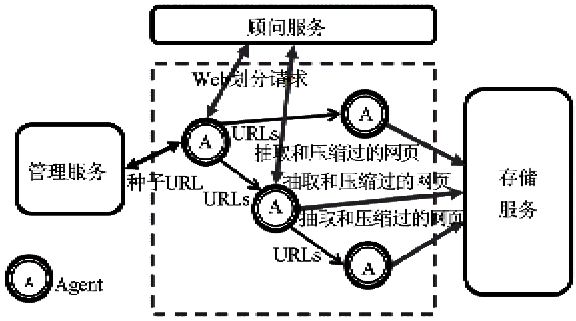

图 1 基于顾问服务的分布式 Web 爬虫系统组成

(1) 管理服务(Manager)

提供系统的管理界面,系统管理员通过管理服务下发抓取网页所需要的种子 URL。

(2) 爬虫(Agents)

此部分是由多个 Agent 程序共同组成的。Agent 部署于广域网之上,每个 Agent 具有单机版多线程爬虫的所有功能。Agent 能够与其他 Agent 通信,推送各自发现的 URL 到其他 Agent。Agent 还能够访问顾问服务,以获取关于 URL 的 Web 划分信息。在进行过一次成功的 URL 推送后,Agent 将会把这个记录在自身缓存一段时间(主机(host)缓存),从而减少访问顾问服务的次数。Agent 会定期将下载到的网页数据(经过信息抽取和压缩)写入存储服务。

(3) 顾问服务(Consultant Service)

一种提供 Web 划分依据的服务,Web 划分依据可以是动态的网络测量信息、地理位置信息、运营商信息等,相关划分策略在本文第 4 节将会详细讨论。顾问服务可以只部署在一个主机上,也可以多台主机组织成一个分布式服务,对 Agents 提供统一的接口。

(4) 存储服务(Storage Service)

一种提供数据存储的服务,与顾问服务一样可以部署于一台主机或多台主机上,对 Agents 提供统一的数据存储接口。

1.1 Agent 协同算法框架

系统中存在管理者、顾问、Agent 等多种角色,每种角色循环监听从系统中其他部分发送来的信息。由于真正的系统实现要相对复杂,我们只给出简化的算法框架,如图 2 所示。在 Agent 上我们省略了网页下载和网页存储的过程,把重点放在对 Agent 协同算法的描述上。基于顾问服务的 Agent 协同算法与有调度中心的 Agent 协同算法不同,顾问服务并不负责将 URL 推送到 Agent 的工作,推送 URL 的工作由 Agent 自己负责,从而大大降低了调度中心的负载,使其不再成为系统瓶颈。

```
Manager(){
  while (TRUE){
    if (管理员下发初始 URL){
      随机选择 Agent,向其发送 URL;
| | |
Agent (){
/* Agent 有一个 Daemon 进程进行网页的下载工作,定义这个进程为下载进程。这里不在详述*/
  while (TRUE){
    if (接收到来自 Manager 的 URL || Agent 自身发现新 URL){
    提取 URL 中的 host;
    if (host 缓存命中){
      从 host 缓存获得负责当前 host 的 Agent 的地址;
      发送 URL 至 Agent;
    }else{
      向 Consultant 查询负责当前 host 的 Agent;
      从 Consultant 获得负责当前 host 的 Agent 的地址;
      发送 URL 至 Agent;
    } }
    if (接收到来自其他 Agent 的 URL){
      发送 URL 至下载进程;
} } } }
Consultant (){
  while (TRUE){
    if (接收到来自 Agent 的带有 host 的查询请求){
*       调用 Web 划分算法,计算负责当前 host 的 Agent;
        发送划分结果至发出请求的 Agent;
} } }
```

图 2 基于顾问服务的分布式 Web 爬虫 Agent 协同模式算法框架伪代码

1.2　系统性能分析

本节重点分析加入顾问服务后 Web 爬虫系统内部的节点间通信量与有调度中心和无调度中心两种模型下的节点间通信量的差异。设一个 Agent 负责若干个 host 的抓取,这些 host 所包含的全部网页中共有 n 个链向非自身 host 的链接,而这些链接平均分别属于 m 个 host。设 Agent 间批量推送的数量上限为 α,及 Agent 与顾问服务间批量推送的数量上限为 β,则在顾问服务参加调度的情况下,Agent 与其他 Agent 通信的次数为 $\mu\frac{n}{\alpha}$,其中 μ 是一个系数,$0<\mu\leqslant 1$,实际上并不是所有 n 个链接都需要发送到其他 Agent,其中有一部分会被划分到 Agent 本地抓取,因此我们设这部分 URL 的百分比为 μ。考虑到 Agent 上有 host 缓存,在所有 host 都无法在缓存命中的情况下,Agent 与顾问服务的通信次数最多为 $\lambda\frac{2m}{\beta}$,$0<\lambda\leqslant 1$,同样乘以系数是因为可能有一部分 host 实际上是由 Agent 自己负责的,2 倍是因为 Agent需要向顾问服务发出查询,顾问服务还要向 Agent作出应答,查询和应答实际上是两次数据传输过程,因此算做两次通信。Agent 的总通信次数最多为

$$\mu\frac{n}{\alpha}+\lambda\frac{2m}{\beta} \tag{1}$$

其中,必然有 $n>m$,且 n 和 m 经常相差很大,可以认为 $n>>m$。由于同样是网络通信的数据量上限,α 与 β 的值基本上相当,可以认为它们相等。

对于有调度中心系统,由于所有 URL 需要调度中心来下发,所以总体来说 URL 在网络中总共被推送了两次,第一次是由 Agent 到调度中心,第二次是由调度中心再到 Agent。则一个 Agent 的总通信次数相当于

$$\frac{2n}{\alpha}=\frac{n}{\alpha}+\frac{n}{\alpha}>\mu\frac{n}{\alpha}+\lambda\frac{2m}{\beta} \tag{2}$$

对于无调度中心系统,由于在 Agent 自身就可以进行 Web 划分,所以每个 URL 只需要通信 1 次,就是从 Agent 推送到另一个 Agent,则一个 Agent 的总通信次数相当于

$$\mu\frac{n}{\alpha} \tag{3}$$

综上所述,顾问服务参与 Agent 协同的情况下,Agent 的通信次数小于有调度中心的系统,大于无调度中心的系统。但是由于通常 $n>>m$,所以在通信次数上已经十分接近于分布式调度。在下载网页所造成的网络负载恒定不变的情况下,这意味着,顾问服务参与 Agent 协同能够使分布式爬虫系统承受相对较小的网络负载。

2　Web 划分

分布式 Web 爬虫中各个 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,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。由此本文提出 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 A 的子集的集合 $B=\{\beta_1,\beta_2,\beta_3,\cdots,\beta_N\}$,如果满足

$$\beta_1\cup\beta_2\cup\beta_3\cup\cdots\cup\beta_N=W$$

且

$$|\beta_i\cap\beta_j|<\delta$$
$$i=1,2,\cdots,N;j=1,2,\cdots,N;i\neq j$$

(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)

称集合 B 中的元素 $\beta_i(i=1,2,\cdots,N)$ 为一个 Web 划分集合。称将 Web 分割为 Web 划分集合 $\beta_1,\beta_2,\beta_3,\cdots,\beta_N$ 的过程为 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 $A=\{\alpha_1,\alpha_2,\alpha_3,\cdots,\alpha_N\}$,对于定义 1 中的集合 $B=\{\beta_1,\beta_2,\beta_3,\cdots,\beta_N\}$,称一一映射 $\mu:A\to B,\alpha_i\to\beta_j(i=1,2,\cdots,N;j=1,2,\cdots,N)$ 为分布式 Web 爬虫的 Web 划分。

2.2　Web 划分策略

在本系统中,我们采用 host(即主机名)作为 Web 划分单元。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,采用何种方法区分这 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。由于顾问服务被设计成像 DNS 那样的网络服务,其内部实现细节对系统的其他部分是透明的,因此可以在顾问服务上实现多种 Web 划分策略。目前我们已经实现了两种 Web 划分策略。这两种策略均属于静态 Web 划分方法。实现动态 Web 划分方法是未来的研究重点。

(1) 根据网站的地理位置进行 Web 划分(以下简称为地理划分)。将 Agent 所在的城市作为每个 Web 划分集合的中心,目标网站距离哪个 Agent 最近,就被划分到该 Agent 对应的划分集合中。

(2) 根据网站的接入运营商进行 Web 划分(以下简称为运营商划分)。将 Agent 所接入的网络运营商作为每个 Web 划分集合的中心,如果目标网站与某 Agent 接入相同的运营商网络,它就被划分到该 Agent 对应的划分集合中。

实现"就近抓取",是促使我们从地理位置和运营商出发的原因。距离的度量是"就近抓取"的关键问题。在"就近"的概念上,地理位置是物理世界的距离度量,运营商信息可以认为是网络虚拟世界的距离度量。根据以往的经验,我们认为在同一运营商网络中的两 IP 间通信,其效率一般高于跨运营商的两 IP 间通信,即使两 IP 之间的地理距离较近。

3 实 验

此次实验的数据采集于 2008 年,实验环境是基于正在建设中的"虚拟计算环境试验床",它是由国家计算机网络应急技术处理协调中心(CNCERT/CC)和哈尔滨工业大学(HIT)协作建设的,以国家计算机网络应急技术处理协调中心的网络基础设施及计算资源为基础的,开放、安全、可控的大规模虚拟计算环境实验平台。该平台的设备分布于全国多个省市,并能够接入多家运营商网络,为我们提供了真实的 Internet 实验环境。

3.1 Web 划分策略的性能对比

本实验的目的是对比广域网部署方式下,Web 划分策略分别采取地理划分和运营商划分对爬虫性能造成的影响。部署方案如下:

A. 地理分布部署,地理划分(geo distribute, geo partition):Agent 分别部署在北京、哈尔滨、广州三地,随机挑选接入的运营商;

B. 地理分布部署,运营商划分(geo distribute, isp partition):Agent 分别部署在北京、哈尔滨、广州三地,分别接入联通 UNICOM、教育 CERNET、电信 TELECOM 三个运营商;

C. 运营商分布部署,运营商划分(Beijing ISP distribute, ISP partition):Agent 全部部署在北京,但是分别接入联通 UNICOM、教育 CERNET、电信 TELECOM 三个运营商。

为了尽量避免 Internet 环境动态变化造成的影响,我们将实验时间定为 1h,各种部署方案都在一天中的同一时间段内运行(上午 11 点到 12 点)。每种部署方案中,三组 Agent 所在的网络接入点带宽均为 10Mb/s,总接入带宽为 30Mb/s,接入网络均为 Internet。抓取目标网站均为国内站点,且分别来自北京、哈尔滨和广州三个城市,考虑到运营商划分,事先把没有接入三个运营商网络的网站剔除。实验结果如图 3 所示。各部署方案的下载网页数和平均下载速率如表 1 所示。

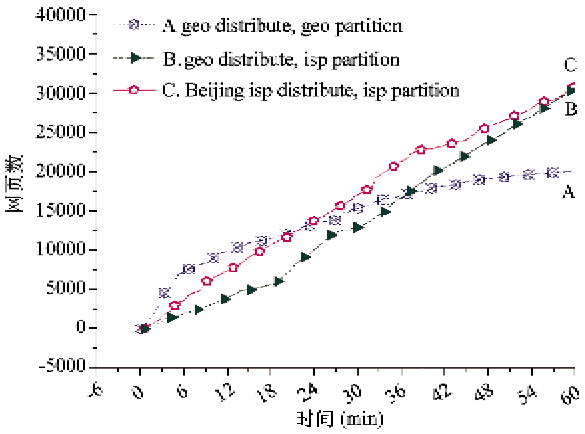

图 3 三种 Web 划分策略的实验结果对比

表 1 三种 Web 划分集合分类方法的平均下载速率统计

部署方式	30min 内下载网页数	30min 内平均下载速率(网页数/s)	60min 内下载网页数	60min 内平均下载速率(网页数/s)
A	15316	8.51	20108	5.59
B	14825	8.24	35439	9.84
C	19686	10.94	35152	9.76

A 方案(地理分布部署,地理划分)全过程的变化非常特别,下载速率一度非常高,根据表 1 前 30min 的平均速率 8.51 网页每秒也在所有部署方案中居第二位,但是之后的性能不断下降,最后只抓到了最低的 20108 个网页。对 A 方案的 3 个 Agent 各自数据的分析发现,每个 Agent 都出现了下载速率下降的现象。因为在其他部署方案中并没有出现这种现象,所以这并不是运行 Agent 的主机过载所致。反观 B 方案(地理分布部署,运营商划分),其 Agent 也是分布于北京,哈尔滨,广州三地,但是该方案按运营商进行 Web 划分后,前 60min 的性能与 C 方案(运营商分布部署,运营商划分)相当。这是运营商网络的互连问题造成的,由于各运营商网络之间互连的接口是运营商网络的访问瓶颈,跨运营商网络

的通信性能会比较低。A 方案中,Agent 虽然在地理上距离目标网站较近(实际上是在一个城市里),但是他们之间的通信有的跨越了不同的运营商网络。而 B 方案中,Agent 与目标网站不一定都在一个城市,但是通信双方都处于同一个运营商网络内。据此我们得出这样的结论:对于国内网站地理位置近并不能代表网络距离近,即国内网站按照地理位置进行 Web 划分性能并不高。这个结论并不意味着地理位置这个因素可以忽略,由于信号在线路上的传输必须经过一定的时间,所以地理位置差异造成的通信效率差异是实际存在的。

C 方案在实验中性能最好,根据表 1 前 30min 的平均下载速率 10.94 网页每秒在各种方案中最高,前 60min 的下载速率 9.76 网页每秒在各方案中居第二名,与第一名仅差 0.08 网页每秒。此方案中,虽然 Agent 部署在北京,但是地理因素的影响会比 B 方案小些,因为北京在地理位置上恰好处于哈尔滨和广州之间,即较小的地理位置因素影响以及按运营商调度使得 C 方案显示了较好的性能。这说明对于国内网站,运营商互连因素相对于地理位置因素对分布式爬虫的性能影响更大些。

3.2　局域网和广域网上部署的性能对比

实验的目的是对比本系统在局域网(LAN)上部署与在广域网(WAN)上部署的性能差异。在局域网上部署时对 Web 采用随机划分策略。部署方案如下:

A. 所有 Agent 集中部署于北京全部接入联通网(Unicom Beijing, LAN);

B. 所有 Agent 集中部署于北京全部接入电信网(Telecom Beijing, LAN);

C. 所有 Agent 集中部署于哈尔滨全部接入教育网(Cernet Harbin, LAN);

D. 采用 3.1 节中表现最好的方案作为广域网部署方案与以上 3 个方案进行对比(WAN)。

由于当时实验环境有带宽限制(10Mb/s),所以在局域网上部署的情况下,这个局域网的接口带宽就只有 10Mb/s;而在广域网上部署时,3 组 Agent 所在的网络接入点各为 10Mb/s 带宽,总带宽就是 30Mb/s。因此我们在数据对比中将局域网上部署的网页数乘以 3,以此作为 30Mb/s 带宽下最理想的系统性能数据。抓取的目标网站集合与 3.1 节的相同。实验结果如图 4 所示。各部署方案的下载网页数和平均下载速率如表 2 所示。

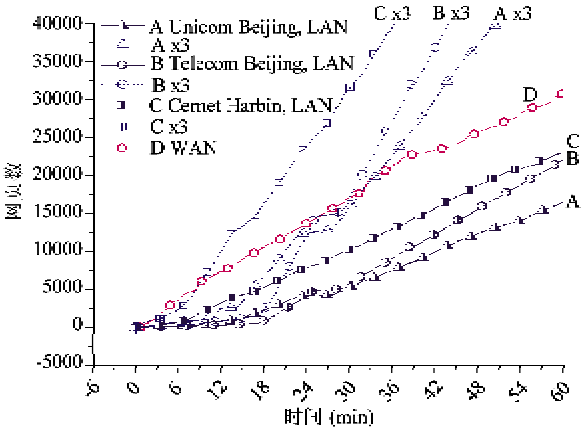

图 4　系统在 LAN 和 WAN 上部署的实验结果对比

表 2　系统分别在 LAN 和 WAN(Internet)上部署时的平均下载速率统计

部署方式	30min 内下载网页数	30min 内平均下载速率(网页数/s)	60min 内下载网页数	60min 内平均下载速率(网页数/s)
A	5190	2.88	16446	4.57
B	5704	3.17	22015	6.12
C	10349	5.75	23071	6.41
D	19686	10.94	35152	9.76
A×3	15570	8.65	49338	13.71
B×3	17112	9.51	66045	18.35
C×3	31047	17.25	69213	19.23

从图 4 可以看出,网页下载量曲线并不平滑,我们分析认为引起这种现象的原因有两个:(1)Agent 到目标网站的链路不稳定,带宽的抖动比较大;(2)由目标网站链接结构导致,由于使用了相同的目标网站集合,几条曲线都出现了相似的斜率变化。

从表 2 中可以看出,前 30min,D 方案(广域网部署)的平均下载速率与 A×3(北京联通局域网部署数值乘以 3),B×3(北京电信局域网部署数值乘以 3)相当,接近理想情况;前 60min 广域网部署与局域网上部署相比仍具有很大优势,但是 LAN 部署乘以 3 的性能已经达到了广域网部署方案的 2 倍,我们分析认为这个差距是由于 Agent 间通信造成的,根据文献[3]的分析,Agent 间通信量会随全系统下载的总网页数的增大而增大,而 D 方案中 Agent 间通过 Internet 进行通信,也造成了一定的时间开销,从而使系统性能低于理想情况。

表 2 中还需要说明的一点是在爬虫下载过程中,前 30min 的平均下载速率与前 60min 的下载速率并不相等。其中的规律是 LAN 上的部署方案其前 60min 的平均速率高于前 30min 的平均速率;而

WAN 上部署的方案则与之相反。其主要原因在于，LAN 部署方案下，起初的下载速率比较低，之后，随着下载获得的新的 URL 数量的不断增加，其下载的网页数将越来越多。WAN 部署方案下，由于系统具有带宽上的优势，起始时就能够到达很高的下载速率(几乎是 LAN 部署方案的 2 倍)，以至于前 30min 就已经下载了非常多的网页，到第 60min 时，爬虫所发现的 URL 很多都是已经下载过的链接了。因此 WAN 部署方案下，前 60min 的平均下载速率比前 30min 有所下降。

从实验数据的整体来看，本系统在 Internet(广域网)上多网络接入点部署的性能优于局域网上部署单一网络接入点的性能。

4 结论

广域网分布式爬虫越来越多地受到学术界、商业界和开源社区的关注。本文实现了一种能够运行于广域网环境下的分布式 Web 爬虫。在系统设计上，提出了基于顾问服务的 Agent 协同模式，给出了算法框架，并推导出顾问服务参与 Agent 协同能够使分布式爬虫系统承受相对较小的网络负载。本文还提出了 Web 划分的概念，并对其进行了探讨。对顾问服务我们首先实现了两种静态 Web 划分策略：按照网站的地理位置进行 Web 划分和按照网站的接入运营商进行 Web 划分。在实验部分，本文通过在 Internet 环境下对几种部署方案的对比实验，验证了系统对基于 Internet 的分布式环境的适应能力，并分析了几种部署方案在性能上的异同。我们发现运营商网络之间的互连对爬虫性能的影响超过地理位置差异的影响。本文还通过对局域网和 Internet 两种部署环境下的系统性能对比，证明本系统在 Internet 上部署的性能优于局域网上部署的性能。我们的后续工作包括开发具有动态 Web 划分功能的顾问服务，更深入地研究运营商互连与地理位置对分布式 Web 爬虫性能的影响，及研究和实现无调度中心的 Agent 协同等。

参考文献

[1] Baeza-Yates R, Castillo C, Junqueira F, et al. Challenges in distributed information retrieval. In: Proceedings of the International Conference on Data Engineering (ICDE), Istanbul, Turkey, 2007

[2] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. In: Proceedings of the 7th International World Wide Web Conference (WWW), Brisbane, Australia, 1998. 107-117

[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

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

[6] Liu F, Ma F Y, Ye Y M, et al. (2005). IglooG: A distributed web crawler based on grid service. *Web Technologies Research and Development* (*APWeb* 2005), 2005, 3399 : 207-216

[7] 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

[8] 蒋宗礼，赵钦，肖华等. 高性能并行爬行器. 计算机工程与设计，2006，27(24)：4762-4766

[9] Dustin B. Distributed High-performance Web Crawlers: A Survey of The State of the Art. http://www.cs.ucsd.edu/~dboswell/PastWork/WebCrawlingSurvey.: UCSD, 2003

[10] Christen M. YaCy Peer-To-Peer Web Search. http://yacy.net/: YaCy, 2003

[11] Garbe M. FAROO P2P Web Search. http://www.faroo.com: FAROO, 2007

[12] Cho J, Garcia-Molina H. Parallel crawlers. In: Proceedings of the 11th International World Wide Web Conference, New York, NY, USA, 2002. 124-135

[13] Boldi P, Codenotti B, Santini M, et al. Ubicrawler: A scalable fully distributed web crawler. *Software, Practice and Experience*, 2002, 34(8): 711-726

[14] Papapetrou O, Samaras G. IPMicra: An IP-address based location aware distributed web crawler. In: Proceedings of the 5th International Conference on Internet Computing (IC 2004), Las Vegas, USA, 2004. 694-699

[15] Cambazoglu B B, Karaca E, Kucukyilmaz T, et al. Architecture of a grid-enabled Web search engine. *Information Processing and Management*, 2007, 43 (3): 609-623

[16] Singh A, Srivatsa M, Liu L, et al. Apoidea: a decentralized peer-to-peer architecture for crawling the world wide web. In: Proceedings of the Special Internet Group on Information Retrieval (SIGIR) Workshop on Distributed Information Retrieval, Toronto, Canada, 2003. 126-142

Research on Agent collaboration and Web partition in WAN-based distributed Web crawlers

Xu Xiao, Zhang Weizhe, Zhang Hongli, Fang Binxing
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract

This paper focuses on agent collaboration and Web partition, the two core issues in WAN-based distributed crawling. First, a new consultant-service-based agent collaboration method and the corresponding system model are proposed. The new method has a lower communication overhead than the central-coordinator-based crawling systems and exploits location proximity better than the ones based on Distributed Hash Table (DHT). Second, the detailed definitions of Web partition are presented. The selection of Web partition unit and the Web partition strategy are discussed. The experiment under the real Internet environment shows that WAN-based distributed Web crawling systems have better performance than the LAN-based ones. The experiment also shows that the impact of Internet service providers interconnectivity on the system performance is greater than that of the geographical locality.

Key words: distributed Web crawler, Agent collaboration, Web partition, consultant service