基于 GNP 算法的分布式爬虫调度策略 *

刘 爽[1], 姜春祥[2], 张伟哲[1], 李 东[1], 张 鸿[3]
(1.哈尔滨工业大学 计算机科学与技术学院, 哈尔滨 150001; 2.国家计算机网络应急技术处理协调中心 黑龙江分中心,哈尔滨 150001; 3.国家计算机网络应急技术处理协调中心, 北京 100029)

摘 要: 针对分布式搜索引擎的任务调度及负载均衡问题,提出了基于 GNP 算法的分布式爬虫调度策略和负载均衡的方法。利用网络距离预估取代大规模的网络距离测量,不仅提高了系统的响应速度,还减少了系统对广域网造成的压力。通过在广域网上部署爬虫节点,构建分布式搜索引擎,应用该调度策略进行实验,验证了系统性能有较大提高。

关键词: 分布式爬虫; 任务调度; 负载均衡; 网络测量; 全局网络定位
中图分类号: TP309 **文献标志码**: A **文章编号**: 1001-3695(2010)02-0446-04
doi:10.3969/j.issn.1001-3695.2010.02.011

GNP-based scheduling strategy for distributed crawling

LIU Shuang[1], JIANG Chun-xiang[2], ZHANG Wei-zhe[1], LI Dong[1], ZHANG Hong[3]
(1. *School of Computer Science & Technology, Harbin Institute of Technology, Harbin* 150001, *China*; 2. *Heilongjiang Branch of National Computer Network Emergency Response Technical Team/Coordination center of China, Harbin* 150001, *China*; 3. *National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing* 100029, *China*)

Abstract: In order to solve task scheduling and load balancing problems of distributed search engines, this paper proposed a GNP-based scheduling strategy for distributed crawling and a load balancing method. Adopted internet distance estimating mechanism as a replacement for large-scale network distance measurement, which not only improved response time of the system, but also reduced WAN pressure caused by the system. Through deploying crawling nodes at WANs, built a distributed search engine, and implemented several scheduling strategies. The online experiment shows great improvement in system's performance.

Key words: distributed crawling; scheduling strategies; load balancing; network measurement; GNP(global network positioning)

0 引言

目前搜索引擎遇到的瓶颈问题在很大程度上是由其集中式结构造成的,一般的商用搜索引擎只有一个或多个大的数据中心,在本地执行全部的网页下载和索引工作。即使是地理上分布的搜索引擎,各个分系统之间也不能实现信息共享。受网络基础条件等方面的限制,随着网站数目与规模的迅速膨胀,集中式搜索系统逐步会被分布式搜索系统取代。分布式搜索引擎在广域网上部署爬虫节点,其优势主要表现在:a)分散系统的网络负载,同时在多个地理位置上工作的爬虫分别分担系统的网络总负载;b)减小对广域网增加的负载,选择最优的爬虫抓取网页,将会大大减少由于网页数据在广域网上路由造成的对广域网的负载;c)分布式地存储大规模数据,尤其是索引数据[1,2]。

如何将被获取站点调度至网络距离较近爬虫,同时保证系统负载均衡是分布式搜索引擎调度系统的重要研究内容。GNP 算法[3]通过将网络主机间的网络距离映射为欧几里德空间的坐标距离,可以有效地预估主机间网络距离,具有较高的准确性。因此,GNP 技术是分布式搜索引擎任务调度与负载均衡的基础。本文针对分布式搜索引擎调度系统提出了基于 GNP 算法的分布式爬虫调度策略,可以有效地提高调度系统的响应速度,同时减少系统对广域网的负载。

本文介绍了 GNP 算法和分布式搜索引擎常用的调度策略,详细讨论了基于 GNP 算法的分布式爬虫调度策略的基本思想和具体实现,并给出验证调度策略性能的实验。

1 相关研究

1.1 GNP 算法的基本思想

GNP 是最早被提出来的基于绝对坐标的网络距离预测算法[4],它把 Internet 模型化为几何空间(如三维的欧几里德空间),并把 Internet 上任何一个节点的位置都对应到这个几何空间的一点上去。这样,任何两个节点的网络距离都可以通过

收稿日期: 2009-05-20; **修回日期**: 2009-06-29 **基金项目**: 国家"973"重点基础研究发展计划资助项目(G2005CB321806);国家自然科学基金资助项目(60703014);高等学校博士学科点专项科研基金资助课题(20070213044);哈尔滨工业大学优秀青年教师培养计划(HITQNJS.2007.034)

作者简介:刘爽(1984-),女,辽宁辽阳人,硕士,主要研究方向为网络计算等(liushuang_0808@yahoo.com.cn);姜春祥(1966-),男,高级工程师,主要研究方向为网络安全等;张伟哲(1976-),男,副教授,博士,CCF 会员,主要研究为网络计算、网络安全等;李东(1967-),男,教授,博士,主要研究方向为并行计算、网络安全、计算机图形学等;张鸿(1976-),男,高级工程师,博士,主要研究方向为计算机网络、信息安全等.

两个节点间的被模型化的几何距离来估算[5]。

为了使 Internet 中的节点在几何模型中的坐标得以计算,GNP 算法提出了一种两阶段式的结构。在第一个阶段里,被称为路标节点(landmark)的一组位置分散的主机,在一个维度被预先设定的几何空间里首先计算它们自己的坐标;然后这些 landmarks 的坐标将作为参考结构,被传送给任何一个想要加入的节点。在第二个阶段里,任何一个拥有 landmarks 的坐标节点,都能够计算自己相对于这些 landmarks 的坐标。

GNP 的坐标计算主要使用爬山法,并用三角不等式原理保证网络距离预测的准确性。文献[3]描述了 GNP 算法的具体实现,实验表明在三维空间以上,GNP 的距离预测都有较高的准确率。

网络中主机节点间的距离可以通过 GNP 坐标按照欧几里德公式计算得到,计算值就可以作为实际网络距离的估计值。一旦得到一台主机在 GNP 坐标系中的坐标值,就可以立刻计算出该主机与坐标系中其他主机间的网络距离,而不需要进行实际测量,从而减少了主机和网络的负担,提高了网络距离测量的效率。

本文将使用 GNP 算法进行网络距离预测,根据广域网上爬虫与网站之间的网络距离预测值来估计爬虫下载该网页的速度,根据此结果进行分布式搜索引擎的任务调度。

1.2　分布式爬虫调度的主要策略及问题

分布式爬虫调度的策略主要有四种。

1)随机哈希调度[6]

最早的分布式爬虫系统大多是对 URL 或主机名哈希,在此基础上进行调度。这种调度策略非常容易计算,系统开销较小;同时,由于哈希函数具有随机性,可以保证爬虫间的负载均衡。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。但是随机哈希法不对分布式爬虫进行区分,不能有效地利用分布式爬虫抓取不同网站效率不同的特点。

2)根据网站的域名后缀及文件类型调度

这种调度方法根据网站域名后缀的不同,将具有相同域名后缀的网站分配给同一个或一组爬虫抓取。例如,根据网站的域名中诸如. net、. org、. com 等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、. jpg、. mp3 等进行分类。SE4SEE(South-East Europe search engine,东—南欧搜索引擎)还提出了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[7]。这种方法的优点是网页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,但还是存在诸多缺陷:a)并非每个 URL 或域名都是遵守传统后缀命名规范的,如有的学校域名就是. com 而不是大家普遍认同的. edu;b)由于各种类型的网站数量或文件数量分布不均,将造成系统中各个爬虫的负载不均,如按照语言类型分类中,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。

3)根据地理位置调度[8]

即就近抓取。对每个网站,由地理上距离它最近的爬虫抓取。例如,部署在法国的爬虫只抓取法国境内的网站,部署在中国黑龙江省的爬虫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,所以地理距离较近的两点,数据传输时间也相对较短。但是由于运营商的商业利益等因素,同一爬虫抓取地理上距离接近的不同网站可能要经过非常不同的路由器,真正的网络距离相差可能很大。

4)根据网络位置调度[9]

直观的想法就是利用网站的 IP 地址,因为 IP 地址本身就具有层级关系,很容易建立树状结构。但是由于 IP 地址分配时的随机性,IP 地址并不能代表网络位置。另一个想法是利用网络中的自治域(AS),但是自治域的构建过程综合了人力、物力、财力等诸多因素,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。

目前用分布式爬虫对网站进行测量来计算网络坐标方面的工作很少,而将其结果应用在分布式爬虫调度上的研究几乎没有。

2　基于 GNP 的分布式爬虫调度策略

2.1　调度策略的基本思想

分布在广域网上的爬虫抓取不同网站的性能差异巨大,笔者认为网络距离较近的爬虫与网站之间的通信时间较短,爬虫下载网页的速度也相对较快。

分布式搜索引擎的调度器首先测量若干组选定爬虫节点之间网络时延,汇总后建立 GNP 坐标系统;然后通过坐标值计算其余网站与爬虫节点间的网络时延;最后挑选时延值最小的 K 个爬虫作为调度对象。分布式爬虫调度系统的模块图如图 1 所示,这里忽略了爬虫模块的内部实现。

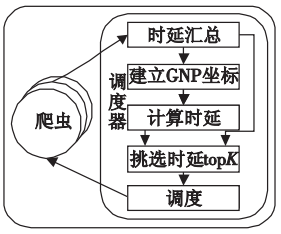

图1　基于GNP算法的分布式爬虫调度系统模块图

调度策略的基本思想是利用 GNP 算法,通过测量较少组 Web 节点(爬虫和 Web 网站)之间的网络距离,估算大量节点间的网络距离,利用网络距离来预测爬虫下载网页的速度进行调度,从而有效地减少了大规模网络测量的时间开销及其对广域网造成的负载。具体做法如下:

a)以网络时延为距离,选择若干爬虫节点作为路标节点,所有路标节点彼此双向测量网络时延;普通爬虫节点测量到所有路标节点的网络时延。

b)利用路标节点间的网络时延构建 GNP 基本坐标系;通过普通爬虫节点到路标节点的网络时延完善 GNP 坐标系。

c)需要确定调度策略的网站加入 GNP 坐标系。路标节点测量到该网站的网络时延,确定该网站在 GNP 坐标系中的坐标值。

d)利用欧几里德公式计算网站到普通爬虫节点的估计网络时延。

e)统计少数测量时延(网站到路标爬虫节点的网络时延)和大量计算时延(网站到普通爬虫节点的网络时延),从中挑

选与该网站的网络时延最短的若干个爬虫作为它的调度结果。

f)根据调度结果抓取网站。

2.2　调度方案的具体实现

2.2.1　算法设置

1)关于路标节点的选择　路标节点必须全部是完全可控制的节点,因为路标节点之间的距离需要双向测量以准确地计算出基本的 GNP 坐标系。在本系统中从爬虫节点中选出部分节点作为路标节点。由于 Internet 上的网站并不是可控节点,对其进行的测量只能是单向测量,不能选择网站作为路标工点。

通常选择具有区域代表性的一组爬虫作为坐标节点。一个实际的选择方案是根据爬虫的地理位置确定它的区域性,认为地理上分布均匀的坐标节点彼此间的网络距离也比较均匀,更能代表空间坐标系,由此构建的基础坐标系更加可靠。但是爬虫的部署会受到各种因素影响,所以所选的路标节点的分布未必会完全均匀。另一种理想方案是根据爬虫的网络距离进行划分,选择划分集合的中心节点作为路标节点。根据实际的可操作性,本系统采用第一种选择方案,选择不同城市、不同运营商的若干爬虫作为路标节点。

2)关于网络时延值的准确性　为了得到比较准确的网络时延值,简单直接的方法是对一个网站或爬虫进行多次时延测量,取多个网络时延的平均值作为准确时延值。但是时延值不绝对稳定,对同一个目标的多次测量中,存在数目不定的由丢包造成的过大时延值,这些值是时延的噪声,不能简单地取所有时延值的平均值。

本文的办法是计算网络时延的统计加权平均值。首先根据时延值的分布确定一个统一的模值;再将所有时延值按前述统一的标准取模;然后对概率最大的三个取模整数值对应的时延值计算平均值,即可得到比较准确的时延值。由这种方法计算出的平均值称为统计加权平均值,它有效地排除了时延序列中的噪声,保证了平均值计算的准确性。计算网络时延的统计加权平均值的公式可以表示为

$$\bar{X}=\frac{\sum_{i=1}^{n} m_{1i}+\sum_{j=1}^{p} m_{2j}+\sum_{k=1}^{q} m_{3k}}{n+p+q}$$

其中:m_{1i}、m_{2j}和 m_{3k}分别是最大的三个取模整数值对应的三组时延值,m_{1i}有 n 个,m_{2j}有 p 个,m_{3k}有 q 个;$\bar{X}$是统计加权平均值。

2.2.2　应用层网络时延的测量

本文考虑采用应用层的测量来代替传统网络层的测量作为坐标点的距离量度:a)系统更关心的是网页的下载速率,它更加直接,并且本身就包含了网络延迟、丢包率等问题;b)测量下载速率是单端的,不需要采用双端式测量工具,即不需要对方服务器安装应答程序作配合。即使是最简洁的 ICMP 的 ping 操作,也可能因为防火墙、代理服务器等问题受到阻断(在本文的实验中,大约有 30% 的网站无法 ping 通)。

考虑到正常运行的网站必定会打开 80 端口,所以可以测量网站对于 HTTP 请求的响应时间,将这个时延近似作为网络时延,称以这种原理工作的测量程序为 httpping。

不同于网络层的 ping 操作,httpping 操作使用 TCP/IP 应用层协议 HTTP 请求—响应来工作,其本身就包含了网络延迟、丢包率等。与单纯的网络层测量相比,它能够更准确地反映分布式搜索引擎中爬虫与网站之间的距离,因为搜索引擎关心的是爬虫在应用层上的网页下载速率。

将 httpping 测量的距离作为 GNP 的输入,改变坐标系维度和路标节点数,得到 20% 和 30% 去噪率下的时延估计值的准确率如表 1、2 所示。观察发现,当路标节点数固定,坐标系维度变化时,估计准确率随着维度的增大(忽略维度为 2 时的情况,维度过小时,建立坐标系的准确度下降程度比较大)有略微的提升,但幅度不大;当维度固定,路标节点数变化时,估计准确率随着路标数的增大而显著提升。这些规律与 ping 在 GNP 算法下的结果一致,所以笔者认为将 httpping 的测量作为网络时延的测量手段是可行的。

表 1　GNP 坐标系路标数固定、维度变化下的估计准确率

维度(9 个路标)	去掉 30% 噪声的准确率	去掉 20% 噪声的准确率
2	0.684 679	0.493 873
3	0.737 595	0.599 625
4	0.740 472	0.602 797
5	0.741 811	0.604 769
6	0.742 344	0.605 146
7	0.743 145	0.605 081
8	0.744 649	0.607 787

表 2　GNP 坐标系维度固定、路标数变化下的估计准确率

维度和路标数	去掉 30% 噪声的准确率	去掉 20% 噪声的准确率
6 维 9 个路标	0.742 344	0.605 146
6 维 7 个路标	0.505 910	0.299 740
4 维 7 个路标	0.505 902	0.299 708
4 维 5 个路标	0.420 972	0.076 933

2.2.3　负载均衡的实现

负载均衡的一种实现方式是随机散列法。当有一个服务请求时,从若干相同的备选服务器中随机选择其一对请求进行响应,从而避免单台服务器的负载过重。许多 DNS 服务器都采用这种随机的方法实现负载均衡。

为了使分布式搜索引擎实现负载均衡,确定一个网站的调度方案时,选择坐标系中距离该网站最近的多个爬虫作为备选调度节点,而不是只取最近的一个爬虫作为单一调度节点。这样,当需要抓取一个网站时,对该网站对应的若干个备选调度爬虫进行随机,选一个爬虫抓取该网页。这个方法可以在保证抓取效率的前提下,一定程度地实现系统的负载均衡。

3　实验与分析

本文设计的分布式搜索引擎包括 30 个爬虫,分别分布在北京(10 个)、上海(10 个)、广州(9 个)和哈尔滨(1 个)。考虑到表 1 和 2 呈现出的规律,调度系统建立坐标系时,采用 9 个不同城市不同运营商的爬虫作为路标节点(北京:电信 1 个、联通 1 个、网通 1 个;上海:电信 1 个、联通 1 个、网通 1 个;广州:电信 1 个、联通 1 个;哈尔滨:教育网 1 个),建立 8 维坐标系。为了实现负载均衡,为每个网站选择 3 个最优的爬虫作为备选调度节点。

为了测量整个系统的吞吐量,笔者精选了国内分布在各省市自治区的 203 个较大型的网站。让系统用三种调度方式抓取这 203 个网站。第一种调度方式是随机调度法(random),不对 30 个爬虫作任何区分,对每个网站,随机选择 30 个爬虫中的任意的一个进行抓取。第二种调度方式就是本文介绍的基于 GNP 算法的调度算法,每个网站选择最优的 3 个爬虫随机调度,这里将这种调度方法叫做 Top3 最优调度法(Top3)。第三种调度方式也是基于 GNP 算法的调度算法,但只选择最优

的 1 个爬虫作为调度节点,因而叫做 Top1 最优调度法(Top1)。图 2 显示了分布式搜索引擎在三种调度方式下抓取 203 个网站的吞吐量(throughput)。

如图 2 所示,以几个时间点为例,在第 6 min,Top3 调度的系统吞吐量比随机调度的系统吞吐量高出 436.5 MB,而 Top1 调度的系统吞吐量比随机调度高出 411.1 MB;在第 12 min,Top3 调度的系统吞吐量比随机调度的系统吞吐量高出 767.1 MB,而 Top1 调度的系统吞吐量比随机调度高出 674.5 MB。总体来看,随着系统运行时间的增加,Top1 和 Top3 调度的系统吞吐量与随机调度的系统吞吐量的差值越来越大。随机调度的系统吞吐量增长率大约为 198.8 MB/min,而 Top3 调度的系统吞吐量增长率大约为 270.5 MB/min,比随机调度高出 36.1%;Top1 调度的系统吞吐量增长率大约为 242.6 MB/min,比随机调度高出 22.0%。可以看出,Top1 和 Top3 调度明显优于随机调度,显示出按照网络延迟进行爬虫调度的优势。

直观来说,Top1 调度法选择最优的惟一一个爬虫调度,Top3 调度法随机选择最优的三个爬虫中的一个调度,那么 Top1 调度法就应该好于 Top3 调度法,但根据图 2 显示,Top1 调度的系统性能反而略低于 Top3 调度的系统性能,这说明根据网络时延的关系预测网页下载速率的关系虽然有一定可行性,但并不能做到完全准确。网络时延只是影响网页下载速率的众多因素之一,虽然这个因素的影响因子很大,但不能决定一切。

从 203 个网站中随机挑选一个网站 www.hljtele.com 作单机吞吐量的测试,分布式搜索引擎只抓取该网站。这里只比较随机调度法(random)和 Top3 最优调度法(Top3)的吞吐量(throughput),其结果如图 3 所示。

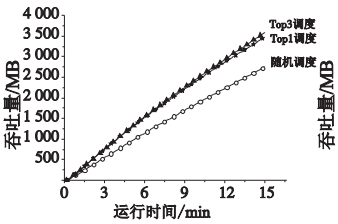

图2　系统吞吐量性能比较

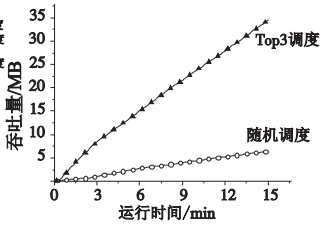

图3　单机吞吐量性能比较

图 3 显示的两种调度方法的性能比较与系统吞吐量的性能比较结果相似,随机调度的单机吞吐量增长率大约为 0.45 MB/min,Top3 调度的系统吞吐量增长率大约为 2.38 MB/min,Top3 调度方式的系统吞吐量增长率约为随机调度的 5 倍。Top3 最优调度法应用在 www.hljtele.com 上的效果非常明显,大大提升了爬虫系统对该网站的抓取效率。

综上,从系统吞吐量的实验中可以看出,基于 GNP 算法的最优调度方法相对于传统方法有一定的优势。

4　结束语

GNP 算法是一种有效的进行网络估计的方法。它将广域网模拟为几何坐标系,并赋予广域网中的主机节点惟一的坐标值,将主机间网络距离的测量转换为坐标系中节点间距离的计算。由于 GNP 的实用性,本文提出将其应用于分布式搜索引擎的爬虫调度上,并首次尝试用应用层网络距离测量代替传统的网络层距离测量。分布式搜索引擎中不同的爬虫节点抓取不同网站的性能各异,调度算法需要得到抓取某个网站的最优爬虫,因而需要测量所有爬虫到该网站的距离。基于 GNP 算法的调度策略是对直接测量法的改进,它可以较准确地通过少量的测量值计算出所有网络时延,有助于提高抓取任务的响应速度,同时减少了对广域网流量的占用,进而提高整个系统的服务质量。本文给出的实验充分证明了基于 GNP 算法的调度策略相较于传统方法有比较明显的优势。

参考文献:

[1] BAEZA-YATES R, CASTILLO C, JUNQUEIRA F, *et al*. Challenges in distributed information retrieval[C]//Proc of International Conference on Data Engineering. Istanbul, Turkey: IEEE CS Press, 2007.

[2] BOSWELL D. Distributed high-performance Web crawlers: a survey of the state of the art[EB/OL]. (2003)[2009-05-15]. http://www.cs.ucsd.edu/dboswell/PastWork/WebCrawlingSurvey.pdf.

[3] NG T S E, ZHANG Hui. Towards global network positioning[C]//Proc of the 1st ACM SIGCOMM Conference on Internet Measurement. New York: ACM Press, 2001:25-29.

[4] FRANCIS P, JAMIN S, PAXSON V, *et al*. An architecture for a global internet host distance estimation service[C]//Proc of IEEE INFOCOM'99. New York: ACM Press, 1999: 210-217.

[5] 柯怡, 林宇, 金跃辉,等. GNP 算法与基于 GNP 的全局负载均衡技术[C]//第九届全国青年通信学术会议论文集. 2004.

[6] KARGER D, LEHMAN E, LEIGHTON T, *et al*. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web[C]//Proc of the 29th Annual ACM Symposium on Theory of Computing. New York: ACM Press, 1997: 654-663.

[7] CAMBAZOGLU B, KARACA E, KUCUKYILMAZ T, *et al*. Architecture of a grid-enabled Web search engine[J]. Information Processing and Management, 2007, 43(3):609-623.

[8] EXPOSTO J, MACEDO J, PINA A, *et al*. Geographical partition for distributed Web crawling[C]//Proc of the Workshop on Geographic Information Retrieval. New York: ACM Press, 2005:55-60.

[9] GOVINDAN R, TANGMUNARUNKIT H. Heuristics for Internet map discovery[C]//Proc of IEEE INFOCOM Conference. Tel Aviv, Israel: IEEE Press, 2000:1371-1380.

(上接第 445 页)

[4] SHAHBAZIAN E, ROGOVA G, VALIN P. Data fusion for situation monitoring, incident detection, alert and response management[M]. Washington DC: IOS Press, 2005.

[5] KOES M, SYCARA K, NOURBAKHSH I. A constraint optimization framework for fractured robot teams[C]//Proc of the 5th International Joint Conference on Autonomous Agents and Multi-agent Systems. New York: ACM Press, 2006:491-493.

[6] STROMBERG D, ANDERSSON M, LANTZ F. On platform-based sensor management[C]//Proc of the 5th International Conference on Information Fusion. [S.l.]: ISIF & IEEE Press, 2002:600-607.

[7] KNOLL A, MEINKOEHN J. Data fusion using large multi-agent networks: an analysis of network structure and performance[C]//Proc of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Las Vegas: IEEE Press, 1994:113-120.

[8] SMETS P. Decision making in the TBM: the necessity of the Pignistic transformation[J]. International Journal of Approximate Reasoning, 2005, 38(2):133-147.

[9] [EB/OL]. http://www.fira.net/.