

分布式 Web 信息采集系统的设计与实现

金岳富, 范剑英, 冯 扬

(哈尔滨理工大学 测控技术与通信工程学院, 黑龙江 哈尔滨 150040)

摘 要: Web 信息的急剧增长, 给信息的有效使用带来了巨大挑战. 作为搜索引擎的重要组成部分, Web 信息采集扮演着越来越重要的角色. 文中从并行性、负载均衡、体系结构以及可扩展性这 4 个方面详细讨论了分布式 Web 信息采集系统的设计与实现. 经验证, 该分布式 Web 信息采集系统能够实现负载均衡, 并具有良好的可扩展性.

关键词: 分布式系统; Web 信息采集; 搜索引擎; 信息处理

中图分类号: TP315 **文献标志码:** A **文章编号:** 1007-2683(2010)01-0116-04

Design and Realization of Distributed Web Crawler

JIN Yue-fu, FAN Jian-ying, FENG Yang

(School of Measure-control Technology and Communication Engineering, Harbin University of Science and Technology, Harbin 150040, China)

Abstract: Great enhancement on the web information brings an enormous challenge to human on how to use it effectively. As a key part of search engine, web crawler plays a more and more important role. This paper introduces the design and realization of a distributed web crawler, covers the following four aspects: parallel, load balancing, architecture and expansibility. Practice shows that the distributed web crawler can achieve load balancing and have a good expansibility.

Key words: distributed system; web crawler; search engine; information processing

0 引 言

World Wide Web 作为一个庞大的分布式异构超文本数据库, 在提供丰富信息的同时, 也给人们在有效使用方面提出了巨大挑战. 为此, Web 搜索引擎技术应运而生. 作为搜索引擎的基础和组成部分, Web 信息采集系统发挥着举足轻重的作用, 并且随着应用的深化和技术的发展, 其应用范围也越来越广. 然而, Web 信息容量呈现爆炸性的增长, 单处理机采集系统受限于 CPU 的处理能力、磁盘存储的容量, 信息采集的速度和覆盖率越来越不能满足实际应用的需求. 针对这一情况, 本文设计和实现了一个

分布式 Web 信息采集系统, 以提高信息采集的范围和速度, 为搜索引擎等服务提供更好地支持.

1 Web 信息采集系统的基本原理

Web 信息采集 (Web Crawling) 是指通过 Web 页面之间的链接关系, 从 Web 上自动获取页面信息, 并且随着链接不断向整个 Web 扩展的过程^[1]. 这一过程主要由 Web 信息采集器 (WebCrawler) 来完成的. Web Crawler 通常由采集器 (spider)、采集控制 (Controller) 和原始网页库三部分组成, 如图 1 所示.

采集器 (spider) 负责从互联网上抓取网页, 从

收稿日期: 2009-02-23

作者简介: 金岳富 (1983—), 男, 硕士研究生, E-mail: jinyuefuzll@126.com;

范剑英 (1963—), 男, 教授, 硕士生导师.

中提取 URL 送入 URL 库,然后把网页保存到原始网页库.由于抓取一个网页通常需要秒量级的等待网络通信时间,因此为提高抓取速度,通常启动多个 spider 并行处理^[1]. Controller 负责 URL 的判断和分发,控制采集器抓取其它网页,反复循环直到 URL 库为空.

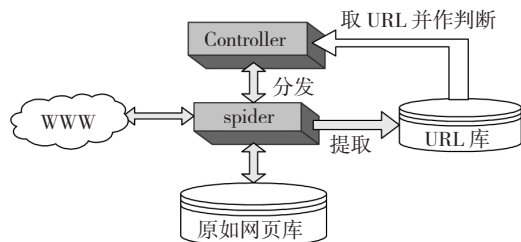


图1 Web Crawler 的基本组成

2 分布式 Web 信息采集系统的设计

分布式 Web 信息采集,就是指多个 Web 信息采集器在某种机制的协调下共同采集整个目标网页,也就是分布式技术在 Web 信息采集中的应用.本文介绍的是基于局域网的分布式采集系统,它具有这样一个特点:所有的采集器都分布在同一局域网内,它们之间通过高速且稳定的内连接通信.

2.1 节点间 URL 的划分策略

分布式系统具有并行性的特点.并行性既可以是任务并行,也可以是数据并行.在本文中,任务并行是指在同一节点内让多个 spider 并行抓取网页以提高抓取速度,数据并行是指在节点间以怎样的粒度进行 URL 划分.

假定 $URLs = \{URL_1, URL_2, \dots, URL_n\}$ 代表所有目标网页的集合, n 表示并行采集系统的节点数.那么根据划分的粒度,节点间 URL 的划分可以有以下 3 种方案:

方案 1:以 url 为粒度,进行 URLs 到 n 个节点的映射.

方案 2:以主机为粒度,进行主机集合到 n 个节点,即到 $\{1, 2, \dots, n-1\}$ 的映射.

方案 3:更粗粒度的划分,将主机映射到一张逻辑表,再将逻辑节点映射到物理节点.

这 3 种方案都属于静态分配方式.静态分配方式就是将 URLs 按事先定义好的规则划分给各个采集器^[1].这种分配方式存在着一种跨区链接.跨区链接不一定能够被它所属的采集器发现,如果本采集器采集了则可能重复采集,如果不采集则

又可能漏采.为了避免重复采集或漏采集,通常将跨区链接进行转发,因此各个采集节点需要建立互连.

方案 1 实现很简单,但不具有方案 2 的一些优点.方案 2 其实是先将 URLs 这个大集合分解为各个互不相交的子集,然后再对这些子集作映射.由于 Web 上各个 WWW 服务器或主机是相对独立的(记 $Host(URIs)$ 表示主机集合),通常一个页面内属于同一 WWW 服务器的链接比较多,也就是说跨区链接数相对较少,并且属于同一 WWW 服务器的网页关联更为紧密,因此选用方案 2 可以减少各节点间 URL 的转发,减少通信量,同时可以保证属于同一主机的文件可以保存在同一节点下,这为后续的信息处理带来方便.

但方案 2 存在着这样一个不足:为实现跨区链接的转发,方案 2 中的每个节点必须维持一张任务划分表,这个任务表在所有节点中都是一致的,任务表中每个记录包含一个 WWW 主机及其所对应的节点标识.但 Web 上的 WWW 主机数以千万计,并且考虑到系统可扩展性方面,创建这样一个巨大的表需要占用较多的资源,也不利于更新.

方案 3 首先将主机集合 $Host(URIs)$ 通过某种哈希算法映射到一个一维数组上(数组中的元素用来保存所属物理节点的节点标识,数组的下标用来表示 $Host(URL)$ 经映射所得的值,称之为逻辑节点标识,这个一维数组也相应地称为逻辑表),然后将该数组以均分的形式划分给 n 个节点,其划分粒度是方案 2 的整数倍,因此具有方案 2 的优点,并且大大减小了任务表的规模.这样,每个节点只需保存这个规模较小的逻辑表,更加易于更新和维护.

通过对以上 3 种方案的分析 and 比较,在数据并行方面采用方案 3,即首先以主机为单位将所有可能的主机映射到一张逻辑表上,再将逻辑节点映射到物理节点,这样可以在减少节点间通信量的同时,做到以主机为单位进行原始网页的存储,并且任务表易于更新和维护.

2.2 负载均衡分析

各节点之间的负载均衡是提高系统效率的关键^[1].负载均衡主要通过处理机的选择和任务调度来实现.为降低负载均衡实现的难度,在分布式系统中使用了同构计算机.但由于每个任务的执行时间可能相差很大(受网页大小、网络状态波动等因素的影响),即便在同构的环境下,按照任务个数进行平均分配,这对负载均衡也是没有意义的.

考虑到节点间 URL 的划分策略,为实现负载均衡所采用的基本思路是采用同构的计算机成分分布式系统,再利用散列函数将 HOST(URLs) 映射到逻辑节点表,然后将逻辑节点平均分配给 n 个物理节点. 尽管不同 Web 站点含有的网页数量可能相差很大,每个网页的大小也可能存在较大的差异,但由于 HOST(URLs) 远远大于逻辑节点数,如果散列函数足够随机的话,则可以认为各节点负载均衡的可能性很大.

2.3 分布式系统的体系结构设计

各节点之间的负载均衡是提高系统效率的关键^[1]. 负载均衡主要通过处理机的选择和任务调度来实现. 在分布式系统设计中,一般有两种体系结构可供选择: 主控/节点 (Host/Node) 结构和无主控 (Hostless) 结构,如图 2、图 3 所示^[1].

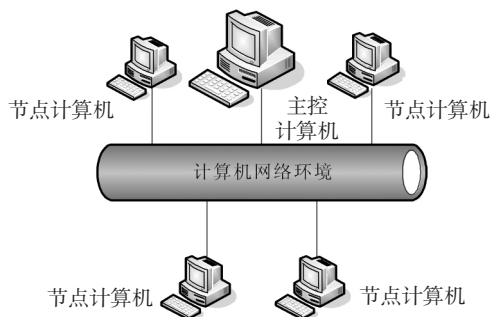


图2 主控/节点结构

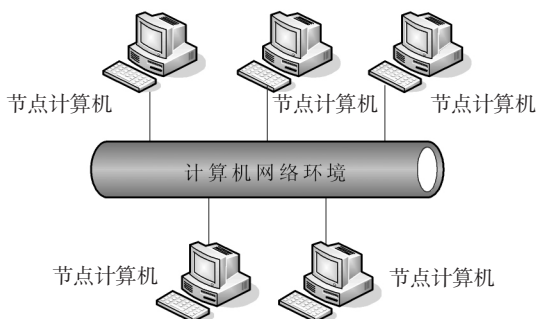


图3 无主控结构

通过并行性及负载均衡两方面的分析,发现各节点的任务是相同的,都是负责采集划分给本节点所有主机上的网页,并且对不属于本节点的 URL 进行转发. 但任务划分以及任务划分表的更新与维护都需要专门的主控程序进行控制,因此适宜采用主控/节点结构. 但任务划分或更新完成后,各节点不需要主控节点的参与就能自行工作,此时系统的体系结构实质上是一种无主控结构. 因此,该分布式 Web 采集系统的体系结构如图 4 所示.

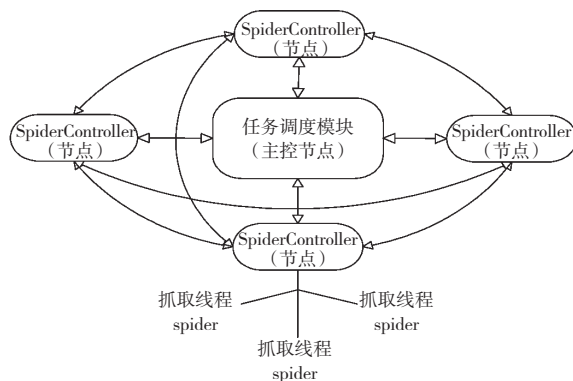


图4 分布式 Web 采集系统的体系结构

2.4 可扩展性设计

可扩展性就是要求分布式系统能够随用户需求或任务规模的变化作相应的更改,以满足不同时期不同任务规模对系统在功能和配置上的不同需求. 对 Web 采集系统来说,就是要求系统在运行过程中能够添加和删除节点机器,删除节点可以视为增加节点的逆过程.

在并行系统中,主控节点通过静态分配方式划分采集目标的 URL,使得每个节点都保存一份一致的任务表,这是系统具有可扩展性的前提. 增加一个节点,为保证系统负载均衡,势必要进行任务重新划分. 主控节点通过运算确定每个节点需要交出的逻辑节点,然后将这些逻辑节点对应的物理节点标识 (通常为节点 ID) 置为新增节点的节点标识,然后统一更新各节点所保存的逻辑节点表. 进行任务重新划分后,通常会导致以下两个问题,为描述方便,记原节点为 A,新增节点为 B:

问题一: 节点 A 任务池中的一部分 URL 将不再属于节点 A

问题二: 属于新节点 B 的 URL,如何确定是否已访问

问题一的解决方法: 将这部分 URL 视为跨区链接进行转发.

问题二的解决方法: 一种方法是新增节点 B 为避免重复采集或陷入死循环,在进行 URL 分发时要事先判读该 URL 是否已访问. 由于新增节点 B 的任务来自于其它所有节点,也就是说已访问 URL 的信息分布在其余各节点上,这可以先在本节点进行判断,当本节点返回结果为未访问时,再将该 URL 转发给原先所属的节点进行判断. 另一种方法是在进行任务重新划分时,将该节点划分出去的主机所对应的已访问信息一并转移给新增节点. 这两种解决

方法都不是很理想,第一种方法显然较为繁琐,并且当再增加一节点时,管理上的开销就更大.第二种方法存在较大的通信量,即便对每个 URL 采用 MD5 算法,每个 URL 仍需用 16 个字节表示,而且在保存已访问信息时,需要按主机分开保存,这种方式不仅造成数据结构复杂,而且占用资源多.

通过上面分析,发现可扩展性差的关键因素是已访问 URL 信息分布在各节点上.考虑到整个系统是基于局域网的分布式信息采集,在局域网内通信具有高速稳定的特点,因此将各节点判断 URL 是否已访问这部分功能抽离出来,如图 5 所示的 UrlServer,用来集中统一地管理已访问信息.

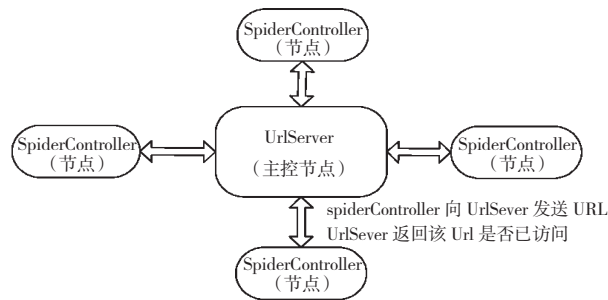


图 5 UrlServer 的结构图

UrlServer 的工作过程其实比较简单,就是接收各节点发送过来的 URL 信息,然后在 UrlMd5 库中判断该 URL 是否已访问.如果未访问,则向该节点返回该 URL,并在 UrlMd5 库中记录已访问信息,通常使用布隆过滤器来实现.

3 实验验证

为了验证该方案,进行了两个实验,实验中的主控节点采用 1G 内存、2GHz 的机器,节点采用 512M 内存、1.7Hz 的机器.

3.1 负载均衡的验证

选取了 12 个站点,采用 3 个节点并行采集 1 h,实验的结果如表 1 所示.

表 1 负载均衡的结果

节点 ID	原始网页库大小	网页总数	采集速度
1	1.14GB	58 336 个	16.2 页/s
2	1.06GB	56 587 个	15.7 页/s
3	0.97GB	55 431 个	15.4 页/s

从表 1 可以看出,该系统各节点每小时搜集的网页总数和原始网页库的总量大致相当,能够较好地实现负载均衡,并且平均采集速度达到 15 页每秒

以上.另外,对存储下来的原始网页库进行了基于模板的信息提取,提取的结果如图 6 所示:

id	name	phone	address	intro
1	上蚝佳	13077566814	河东路237号	(MEMO)
2	农夫烤鱼		河东路233号	(MEMO)
3	石浦海鲜	13355711013	河东路239号	(MEMO)
4	乾隆航	0571-88836777	湖野北路仓基新村	(MEMO)
5	茶桔便	0571-88083167	沈塘桥路114号	(MEMO)

图 6 对 edushi 原始网页库进行信息提取的结果

从提取的结果来看,存储下来的网页库能为进一步的信息处理提供充分支持.

3.2 可扩展性的验证

先以实验 1 的环境运行 30 min,然后增加一新节点,再让整个系统运行 30 min.实验二的结果如表 2 所示.

表 2 可扩展性的结果

节点 ID	原始网页库总量/GB	网页数/个	采集速度/页/s
1	1.07	56 317	15.9
2	1.01	54 963	15.5
3	0.93	54 253	15.3
4	0.52	28 413	16.3

从表 2 可以看出,增加新节点,系统仍能很好的工作,并且在负载均衡方面表现出良好的性能.跟实验 1 比较,前 3 个节点采集的网页个数减少了,这是因为在任务重新划分时,为保持数据的同步,主控节点控制每个节点暂停数据的抓取,另一原因是原先的每个节点都要对划分出去的 URL 进行转发.

4 结 语

本文从并行性、负载均衡、体系结构以及可扩展性这 4 个方面,详细讨论了分布式 Web 信息采集系统的设计,并实现了一个基于局域网的分布式 Web 信息采集系统,最后通过实验对负载均衡和可扩展性进行了验证.目前,该系统已用于上海旺城网络科技有限公司的 Web 信息采集.实验和实践表明,该系统具有良好的负载均衡和可扩展性,能满足中小型企业 Web 信息采集的需求,为后续的 Web 信息处理提供充分支持.

参 考 文 献:

[1] 周 赞. 垂直搜索引擎 Spider 技术的研究与应用 [D]. 广州: 中山大学, 2007.
[2] 陈言敏, 肖诗斌. 基于网络技术的 Web 信息采集系统 [J]. 现代图书情报技术, 2007.

本文对于在媒体广告终端中使用矢量字体完成滚动字幕的方法进行了探讨. 通过使用 FreeType 作为字体引擎,很好的满足了滚动字幕中对字体显示的要求,并且在实际应用中取得了很好的效果.

参 考 文 献:

- [1] 王 竹. 展望多媒体广告在国际互联网中的应用 [J]. 云南大学学报·自然科学版, 2006, 28(S2): 164-167.
- [2] 彭现美. 医疗广告存在的问题及出路 [J]. 医学与哲学、人文社会医学版, 2007, 28(9): 78-80.
- [3] 邱文中. 互联网广告类型及其广告媒体特性研究 [J]. 新闻界, 2006, (6): 157-158.
- [4] 刘 冰. 平面媒体的经营创新 [J]. 中国记者, 2007, (7): 78.
- [5] 姚辉学, 付永忠, 殷苏民. TrueType 字体数控雕刻 [J]. 工程图学学报, 2007, (5): 148-152.
- [6] 唐英敏, 张艳霞, 吕肖庆. 基于汉字构形的 TrueType 字库压缩方法 [J]. 微电子学与计算机, 2007, 24(6): 52-55.
- [7] 杨建红, 刘 蓉, 余泽太. TrueType 字体在图形图像处理软件中的应用 [J]. 武汉大学学报: 工学版, 2004, 32(6): 110-112, 136.
- [8] 朱开玉, 宋国辉, 燕存良. CAPP 系统中动态创建 TrueType 字体的研究 [J]. 计算机工程, 2006, 32(8): 20-22.
- [9] David Turner. The FreeType2 Tutorial [EB/OL]. [2007-04-030]. <http://www.freetype.org>.
- [10] 于明俭, 陈向阳, 方 汉. Linux 程序设计权威指南 [M]. 北京: 机械工业出版社, 2001.
- [11] 孙晓辉, 陈 晓, 王 春, 等. 在嵌入式浏览器中使用 TrueType 矢量字库 [J]. 电视技术, 2007, 31(8): 120-121, 150.
- [12] Neal Bedard. CC1100 Data Sheet and System Designer's Guide. [2006-10-13]. <http://www.MicroSystems.com>.

(编辑:王 萍)

(上接第 119 页)

- [1] 李树青, 崔北亮. 基于个性化信息推荐服务的 Web 搜索引擎技术综述 [J]. 情报杂志, 2007, 26(8): 18-21.
- [2] 吴丽辉, 王 斌, 张 刚. 一个个性化的 Web 信息采集模型 [J]. 计算机工程, 2005, 31(22): 35-39.
- [3] 杜成龙, 龙绍军. 分布式系统开发优化策略 [J]. 电脑知识与技术, 2007, 3(18): 6-9.
- [4] 吴丽辉, 王 斌, 余智华. 一种通用 Web 信息采集系统的设计与实现 [J]. 计算机工程, 2005, 30(3): 11-15.
- [5] 李晓明, 闫宏飞. 搜索引擎 - 原理、技术与系统 [M]. 北京: 科学出版社, 2004.
- [6] 李盛韬, 成 陵. 分布式 Web 信息采集系统的研究与设计 [J]. 计算机工程与应用, 2003, 16: 20-24.
- [7] 张麒麟, 邵春福, 等. 基于分布式并行算法的动态交通流分配研究 [J]. 北方交通大学学报, 2002, 26(5): 4-9.
- [10] BRIN S, PAGE L. The Anatomy of a Large-Scale Hypertextual Web Search Engine [J]. Computer Networks and ISDN Systems, 1998, 30(1): 107-117.

(编辑:王 萍)