

P2P 环境下面向应用的查询服务模型

李建春¹, 赵宗渠², 黄道颖¹, 李健勇¹

(1. 郑州轻工业学院计算机与通信工程学院, 郑州 450002; 2. 河南理工大学计算机科学与技术学院, 焦作 454003)

摘要: 影响 DHT 结构的 P2P 网络性能的因素主要包括关键字的查询效率和维护网络稳定的代价。从分析查询效率和节点规模的关系入手, 提出面向应用查询服务(AOLS), 节点之间的逻辑关系采用二叉树结构, 给出网络自组织逻辑关系算法和消息路由算法。实验表明, AOLS 模型在查询效率和维护网络稳定代价 2 个方面都有较好的性能。

关键词: AOLS 模型; P2P 网络; DHT 算法; 二叉树

Application-oriented Lookup Service Model in P2P Environment

LI Jian-chun¹, ZHAO Zong-qu², HUANG Dao-ying¹, LI Jian-yong¹

(1. College of Computer and Communication Engineering, Zhengzhou Institute of Light Industry, Zhengzhou 450002;

2. College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003)

【Abstract】 Factors influencing performance of P2P network structured by DHT mainly include inquiry efficiency of key words and cost of maintaining the steady of network. This paper starts with analyzing the relation of inquire efficiency and nodes scale, proposes the Application-oriented Lookup Service(AOLS). The logic relation among nodes adopts binary tree. It provides the algorithm of the logic relation organization in network and messages routing algorithm between nodes. Experiment indicates ALOS's inquiry efficiency and cost of maintaining network's stability have better performance.

【Key words】 Application-oriented Lookup Service(AOLS) model; P2P network; DHT algorithm; binary tree

1 概述

现有的 DHT 系统都建立在整个 P2P^[1]网络之上, 节点之间的逻辑连接范围是整个 P2P 网络, 它们的路由算法的效率都是关于网络中节点规模 N 的函数^[2]。然而在许多 P2P 网络中, 针对每一项应用需要建立逻辑关系的节点可能是整个网络中的一部分节点, 关键字的内容和索引信息都发布在有限的节点范围之内, 因此查询范围不需要扩展到整个物理网络。例如, 在一个采用文件分片共享技术的 P2P 网络^[3]中共有 N' 个用户, 参与某一个文件共享的节点很可能是网络中的少数 N 个节点, 共享的文件片段可以用关键字来描述, 对这些关键字索引信息的维护和查询仅限于此 N 个节点即可。这时, 如果采用现有的 DHT 系统, 如: CAN^[4], Pastry^[5], Chord^[6], 整个 P2P 网络中的节点都需要参与查询服务, 不但造成了资源的浪费, 而且增加了操作的复杂性。

2 AOLS 系统模型

2.1 节点的逻辑组织结构

在面向应用查询服务 (Application-oriented Lookup Service, AOLS) 构造的系统中, 用户节点按照预先制定的原则逐个加入, 节点根据逻辑关系形成一个二叉树结构。除了第一个节点(本文称初始节点), 其他所有节点的加入过程类似一个二叉树中叶子节点分裂的过程, 使用如图 1 所示的二叉树结构来描述整个系统中节点的逻辑关系。二叉树上的节点代表用户在该位置时的状态, 其中, 叶子节点对应用户节点当前的状态, 非叶子节点代表用户在该位置时的状态。节点使用长度不固定的二进制编号 id_{node}^h , 其中, h 为节点所处的深度, $node$ 为节点名。用户加入系统时, 根据加入的位置分配编号, 初始节点深度为 1, 编号为 0, 由于其他所有节点都

是直接或者间接由初始节点分裂产生, 因此编号的第 1 位(最高位)都必须为 0。规定每一个节点的编号除了最高位的 0 之外称为有效编号(初始节点的有效编号也规定为 0), 为了说明方便, 在以后提到的编号指节点的有效编号, 节点在第 h 层的有效编号是 $h-1$ 位二进制编码。

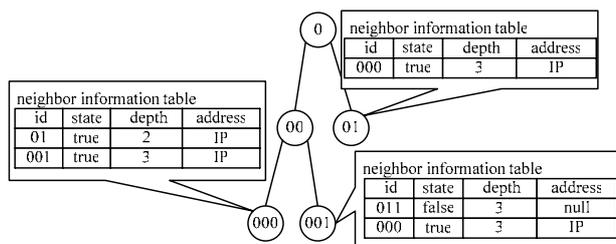


图 1 二叉树及邻居信息表示意图

2.1.1 关于二叉树上的一些定义

定义 1(分裂) 节点加入系统是选中并连接系统中已经存在的某个用户节点(叶子节点), 称此节点为父节点, 而新加入节点称为孩子节点。完成加入过程后, 父节点变成原节点的左孩子(2 个节点是同一个物理节点, 只是在系统中的部分逻辑关系发生变化), 而新加入节点处于原节点的右孩子位置, 它们的编号也要对应新位置, 这个过程称为分裂。

基金项目: 国家科技支撑计划基金资助项目(2006BAK01A38); 河南省杰出青年科学基金资助项目(0612000600); 河南省自然科学基金资助项目(0611052300)

作者简介: 李建春(1976 -), 男, 讲师、硕士, 主研方向: 计算机网络; 赵宗渠, 讲师、硕士; 黄道颖, 教授、博士; 李健勇, 副教授
收稿日期: 2009-01-25 **E-mail:** lijianchun@zzuli.edu.cn

定义 2(饱和度) 设在二叉树的第 h 层上, 能够容纳的节点数量 $N_h = 2^{h-1}$, h 层上已有的节点数量为 N'_h , 称 $S_h = \frac{N'_h}{N_h}$ 为二叉树上第 h 层的节点饱和度。深度为 l 的二叉树的饱和度定义为 $S_{tree} = \frac{N}{2^{l-1}}$, N 是已经加入系统的节点总数。饱和度都为 1 的二叉树称为饱和二叉树。

模型假定系统中现有的节点接收新节点的概率相同, 所以节点可能在本层饱和度小于 1 时接收新节点, 造成树出现多个未饱和层。AOLS 是通过规范节点加入时的行为, 使得新节点在加入时尽量选择低层的接收节点, 从而提高低层的节点饱和度。

定义 3(必要距离) 对于树上第 h 层任意 2 个节点 X, Y 之间, 编号位置相同(从第 1 位到第 $h-1$ 位)的值之间呈现互补关系的个数称为它们的必要距离, 记作 $D(X, Y)_h$ 。由于节点在 h 层的编号长度为 $h-1$, 因此 h 层上任意 2 个节点的必要距离不超过 $h-1$, 期望值是 $\frac{1}{2}(h-1)$ 。

定义 4(实际距离) 二叉树上的某个节点 X 按照路由算法将消息转发到另外一个节点 Y 时, 消息实际的转发次数称为 2 个节点之间的实际距离 $POS_D(X, Y)$, 它由 2 个节点的必要距离、模型的路由算法以及系统的状态决定。节点 X 到二叉树上所有其他节点的实际距离的平均值称为该节点是平均路由距离, 记作 $POS_D(X)$ 。二叉树上任意 2 个节点之间实际距离的期望值称为二叉树的平均路由距离, 记作 $POS_D(tree)$ 。

2.1.2 邻居信息表

节点的邻居是指和本节点建立直接逻辑关系的其他节点(相互知道对方的状态), 节点使用邻居信息表来保存所有邻居的状态(见图 1)。系统中的查询消息由源节点到达目标节点都是通过邻居之间的转发进行的, 邻居的选择根据模型制定的路由策略和邻居信息表中具体信息来确定。

在 AOLS 模型构造的系统中, 节点 X 在二叉树的第 h 层上拥有 $h-1$ 个邻居, 记做邻居节点集 Ω_x , 与所有邻居的必要距离为 1。邻居信息表中存储了所有邻居的相关属性信息, 如 id_{node}^h 代表当前所知该邻居节点的编号; 状态 $state$ 是布尔型的值, $false$ 代表该邻居节点未加入系统, $true$ 代表该邻居节点已经加入系统; 邻居节点当前所处的层数(深度) $depth$ 以及连接地址 $address$ (如 IP 地址等等)。

邻居信息表在节点加入网络时被初始化, 包含了所有邻居的属性信息。当节点接收新节点加入网络时, 在邻居信息表中加入新节点的信息。新节点加入网络后, 要通知已经存在的邻居节点更新它们的邻居信息表中相关的信息项。

2.2 节点的加入

伴随着节点的加入, 在二叉树结构上表现出增宽和增高的过程。节点的加入直接影响系统的状态, 进而影响关键字索引信息的分布状况。AOLS 是通过初始化和维护节点的邻居信息表, 以及关键字索引的重定位, 这 2 种主要的操作来保证查询路由算法时间复杂度为 $O(\lg N)$ 。本节主要介绍新节点加入 DHT 系统的过程, 包含了系统结构的变化、节点信息的维护以及关键字索引信息重定位的过程。

(1)邻居优先策略。二叉树上任一节点 X , 只有同时具备以下特点时才能作为父节点接收节点 Y 加入: 1) X 的所有邻居状态都要为 $true$ 2) X 的所有邻居节点在树上现处的深度不低于 X 现在所处深度。当节点 X 的某个邻居节点尚未加入系

统时, 在二叉树上重新挑选节点 Z 作为候选父节点, Z 需要满足这样的条件: 一旦接收 Y , Y 恰好成为 X 刚才检查到 $state$ 属性为 $false$ 的邻居; 当节点 X 的邻居都已经加入系统而某一个邻居 X' 深度小于 X 时, 选择 X' 作为新的候选父节点接收 Y 。

根据邻居优先策略, 第 h 层的节点 X 接收到新节点加入申请时, 首先要检查自己的 $h-1$ 个邻居节点, 只有这些节点都已经存在, 并且各自至少位于 h 层时(各自拥有 $h-1$ 个已经加入系统的邻居), X 才能选择采用分裂的方式接收新节点。通过这种策略, 提高了二叉树较低层的饱和度。需要说明的是, 这是以牺牲节点加入的响应时间为代价的。

(2)维护邻居信息表。原来处于 h 层的父节点 X 接收新节点后达到树的第 $h+1$ 层, 更新自身的属性信息, 其中编号 $id_x^{h+1} = 2 \cdot id_x^h$, 发送 $(h-1)$ 个消息通知其邻居节点在本地邻居信息表中更新相关信息项, 将原有的邻居信息表的副本传递给子节点, 并且在邻居信息表中加入子节点的信息。

(3)初始化邻居信息表。子节点 Y 加入网络之后, 根据父节点的属性信息生成自身的属性信息, 其中编号 $id_y^{h+1} = 2 \cdot id_x^h + 1$ 。子节点根据从父节点继承的邻居信息表生成本地的邻居信息表, 表中包含父节点以及父节点原有邻居节点的子节点, 通过向这些节点发送状态查询消息可以得到这些节点的属性信息, 如果新节点处于 $h+1$ 层, 需要发送并接收 $2(h-1)$ 个状态查询消息。

(4)关键字索引信息的迁移。当一个节点加入系统时, 除了要发布自身的关键字信息, 还应该负责维护系统中原有的部分关键字索引信息。AOLS 中采用最长前缀的分配策略, 即节点只保存前缀与节点编号相同的关键字, 而关键字总是由其前缀相同的节点中编号最长的节点保存。例如, 二叉树上位于第 5 层的节点 $X(1101)$ 分裂前保存所有前缀为 1101 的关键字的索引信息, 当它分裂后出现了 11010 和 11011 2 个节点, 这部分关键字索引信息也应该相应地分成两部分, 分别由这 2 个节点保存。由于 11010 就是 X 本身, 而 11011 是它的子节点, 因此只需要将前缀为 11011 的信息在分裂时传递给子节点。由此可见, 关键字的迁移是在节点分裂时进行的。

2.3 二叉树具有的一些性质

消息的传递是在二叉树的节点之间进行的, 通过对这种逻辑结构有关性质的研究, 可以从理论上分析路由算法的效率。

性质 1 第 h 层, 节点 X (编号为 $k_1 k_2 \dots k_{h-1}$) 的所有邻居节点的编号是与它有且只有一位互补值的 $h-1$ 位编码。

证明:

(1)网络中的原始节点第一次分裂接收的子节点是该节点在第 2 层的唯一邻居, 第 2 层上的 2 个节点编号分别为 0 和 1, 满足题设。

(2)假设节点 X 在 h 层的所有的邻居节点(X_1, X_2, \dots, X_{h-1}) 的编号是与 X 有且只有一位互补的 $h-1$ 位编码, 那么当 X 接收一个子节点进入 $h+1$ 层时, 它的邻居增加了该子节点, X 和它邻居编码在第 $h+1$ 层上都左移一位, 一位互补关系不变, 而新加入邻居的编号是 X 的新编号最后一位由 0 改成 1。所以在 $h+1$ 层, X 和它的所有邻居仍然具有题设的性质, 证毕。

因此, 节点 X 与 Ω_x 中的所有邻居节点的必要距离都为 1。

性质 2 父节点和子节点邻居信息表中对应的节点也是

父子关系，其中子节点的邻居是父节点对应邻居的子节点。此性质由子节点初始化邻居信息表的过程可证。

性质 3 若二叉树中第 h 层为饱和层，那么在 h 层上任意 2 个节点 X, Y 之间传递信息，通过 $D(X, Y)_h$ 次转发可以到达目标节点。

证明：设第 h 层上任意的 2 个节点 X, Y ，且 $D(X, Y)_h = k$ ，其中 $0 < k < h-1$ 。那么一定存在 $X_1 \in \Omega_x$ ，使得 $D(X_1, Y)_h = k-1$ ，同样存在 $X_2 \in \Omega_{x_1}$ ，使得 $D(X_2, Y)_h = k-2$ ，最后在节点 X_{k-1} 的 $\Omega_{x_{k-1}}$ 中找到节点 Y 称 $X, X_1, X_2, \dots, X_{k-1}, Y$ 为节点 X 到 Y 的转发路径，如果信息在转发路径上可以通过 $D(X, Y)_h$ 次转发由源节点 X 到达目标节点 Y ，称此转发路径为最小转发路径。

寻找最小转发路径举例：

设 $X_{id} = (k_1, k_2, \dots, k_j, \dots, k_{h-1})$ ， $Y_{id} = (\bar{k}_1, \bar{k}_2, \dots, \bar{k}_j, \dots, \bar{k}_{h-1})$ ， k_j 是 X 和 Y 最高的互补位，则必然存在 $X_1 \in \Omega_x$ ， $X_{1id} = (k_1, k_2, \dots, \bar{k}_j, \dots, k_{h-1})$ ，而 $D(X_1, Y)_h = k-1$ 。同理一共通过 k 次转发，消息可以到达目标节点。可以看出，每一次转发，都是在邻居信息表中寻找转发节点与目标节点最高互补位相同的邻居节点作为下一个转发节点，通过这种方式选择的转发节点形成一条最小转发路径。

2.4 AOLS 中的路由算法

根据性质 3，AOLS 中消息的路由过程是转发节点 X_i 与目标节点 Y 之间必要距离逐步逼近的过程，直到消息被转发到目标节点。假设二叉树是深度为 l 的饱和二叉树，共有 $N = 2^{l-1}$ 个节点。根据定义 4 和性质 3，二叉树的平均路由距离等于 $\frac{1}{2} \lg N$ ，与节点规模相同的 Chord 路由算法效率相同。

上面分析的是一种特殊二叉树情况，在实际中这种情况出现的概率非常小，因此需要研究二叉树中有部分不饱和层的情况。在不饱和层上按性质 3 的描述进行消息转发过程中可能出现失败(将未加入网络的邻居节点选作下一个转发节点，或者目标节点未加入网络时)。显然，查询失败的概率和本层饱和度有关，即随饱和度的增大而减少。

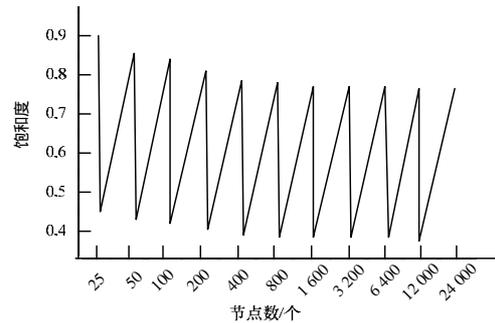
由于在节点加入时采用邻居优先策略，因此第 $h-1$ 层的节点饱和度必定大于第 h 层。在消息转发失败时采用父节点代替传送的方法：第 h 层上的节点 X 向同层目标节点 Y 转发信息时，如果下一个转发节点 Z 不存在，那么 X 就将该信息交给它的父节点 X' 代为传送给 Y 的父节点 Y' ，然后由 Y' 将该信息交给 Y 节点。该方案保证了信息传送的可靠性，但同时增加了传送过程中信息的转发次数。

消息从源节点到目标节点过程中，如果不出现转发失败的情况，2 个节点之间的实际距离等于必要距离；如果出现转发失败的情况，实际距离会大于必要距离。转发的失败概率和转发节点的有效邻居数量有关，二叉树上消息转发平均失败概率和二叉树的饱和度有关，因此二叉树的平均路由距离主要由节点的必要距离和二叉树的饱和度决定，理论分析复杂度为 $O(\lg N)$ ，根据模拟实验的数据，它是略大于 $\frac{1}{2} \lg N$ 的数值。

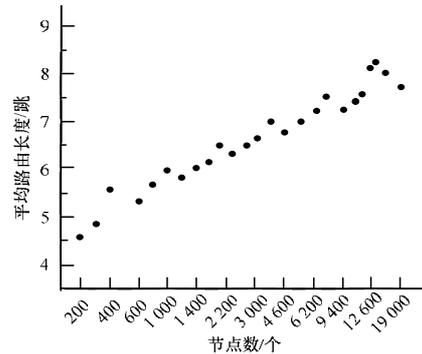
该算法具有精确查找的特点，源节点在使用前缀截取法计算目标节点编号时，不能确定目标节点是否已经加入，或者已经将关键字信息交由其他节点维护。采用上述算法，允许用户最初计算的目标节点与最后找到的目标节点不同，提高了查找的可靠性。

3 二叉树饱和度的变化情况

二叉树的饱和度 $S_{tree} = \frac{N}{2^{l-1}}$ 受节点规模 N 以及树的深度 l 的影响，是随着节点规模变化的锯齿曲线函数，如图 2(a)所示。二叉树深度由 $l-1$ 变成 l 之后的一段时间内，深度不变而节点数增加，饱和度随之增大；当饱和度增加到一定程度 S_{max}^l 时 (S_{max}^l 表示深度为 l 的二叉树中接收新节点使得树深度变为 $l+1$ 的节点数量占全部叶子节点数量的 $\frac{1}{2}$ 时该二叉树的饱和度)，若第 l 层某一个节点接收新节点使得树深度增加，此时 $S_{tree} \approx 0.5 S_{max}^l$ ，然后重复上述过程。实验数据表明，第 l 层的 S_{max}^l 随 l 的增大而减小，但逐渐趋于稳定。二叉树具有比较好的节点容纳能力，深度为 l 的二叉树能够容纳的节点数实际大约在 $(0.4 \times 2^{l-1}, 0.8 \times 2^{l-1})$ 区间内。



(a)饱和度与系统规模的关系



(b)不同系统规模时的平均路由长度

图 2 系统规模与饱和度及平均路由长度关系图

3.1 维护系统动态平衡的代价

二叉树中一个节点状态的更改会引起其他节点维护的邻居信息表中的信息项发生变化，模拟实验中使用新节点加入引起系统中消息传递次数作为衡量维护系统动态平衡的代价，包括了用于选择父节点以及用于更新二叉树状态的消息传递次数。在深度为 l 的二叉树中，这个代价取决于新节点选择的最初候选父节点所在二叉树中的位置。设最初候选父节点位于第 h 层，找到真正接收新节点的父节点位于第 m 层，则复杂度为 $(h-m) + 2m = h+m = O(\lg N)$ ，即二叉树中节点加入时，只需 $O(\lg N)$ 个消息便可使得系统趋于稳定。

3.2 平均路由距离

系统稳定时，查询消息由源节点到目标节点的转发次数体现了该模型路由算法的效率，使用二叉树的平均路由距离 $POS_D(tree)$ 作为树上任意 2 个节点之间期望的路由距离。二叉树上任意节点之间实际距离受到两方面影响：首先是受到节点的必要距离影响，由于最初源节点对目标节点的信息一

无所知,消息只能通过邻居节点转发,必要距离等于消息由源节点到目标节点的最少转发次数;另外转发次数还要受到二叉树状态的影响,当转发过程中计算出下一中转节点失效时,需要使用父节点代替的方法,增加了消息转发次数。

一个新节点 X 加入系统后,对二叉树中的平均路由距离的影响体现在两方面:如果 $POS_D(X)$ 大于 $POS_D(tree)$ 时,则该节点的加入使得 $POS_D(tree)$ 变大,称为正向影响;另一方面,新节点加入后如果使得二叉树饱和度增大,就会减少原有节点在直接转发消息过程中的失败率,从而降低了二叉树的平均路由距离,称为逆向影响。

二叉树深度由 l 增长到 $l+1$ 的过程中,新加入的节点 X 位于 l 层时,节点带来的正向影响会大于逆向影响,因此 $POS_D(tree)$ 变大;新节点低于 l 层时,逆向影响大于正向影响, $POS_D(tree)$ 减少。这个期间 $POS_D(tree)$ 或变大或减少,但由于二叉树在 l 层的接收节点能力逐渐增加, $POS_D(tree)$ 总体呈现增长趋势。当 $POS_D(tree)$ 增加到一定值时,新节点的正向影响减弱,而逆向影响增加,使得 $POS_D(tree)$ 总体呈现减小趋势,直到二叉树的深度变成 $l+1$,然后重复这个过程。

实验模拟了不同规模的二叉树中,每个节点平均进行 500 次随机的查询操作的情况,实验结果证实了正向和逆向 2 种现象对 $POS_D(tree)$ 的影响,尤其当二叉树的深度变大时更为明显。图 2(b)描述了节点规模变化时的二叉树平均路由距离。从图中的数据可以看出,二叉树的 $POS_D(tree)$ 的值略大于 $\frac{1}{2} \lg N$ 。

4 结束语

本文介绍了一种在 P2P 环境下提供查询服务的模型 AOLs,它按照节点应用范围的不同在原有网络之上构造多个

overlay 网络,通过划分网络提高了查询服务的性能,同时保持了良好的系统特征。模型提供同类节点之间自组织逻辑结构的功能,节点之间的逻辑关系形成类似二叉树的结构。在拥有 N 个节点的这种结构中,每个节点维护其他 $O(\lg N)$ 个节点的状态,通过 $O(\lg N)$ 次路由将消息传送到目标节点。因此,使用 AOLs 构造的 overlay 网络具有较好的可扩展性。

参考文献

- [1] Wang Chonggang, Li Bo. Peer-to-Peer Overlay Networks: A Survey[D]. Hong Kong, China: Department of Computer Science, Hong Kong University of Science and Technology, 2003.
- [2] Ratnasamy S, Shenker S, Stoica I. Routing Algorithms for DHTs: Some Open Questions[C]//Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems. Berlin, Germany: Springer-Verlag, 2002: 174-179.
- [3] Pouwelse J P, Garbacki P, Epema D H J, et al. The Bittorrent P2P File-sharing System: Measurements and Analysis[C]//Proc. of the 4th Int'l Workshop on Peer-to-Peer Systems. Ithaca, Greece: Springer-Verlag, 2005: 205-216.
- [4] Ratnasamy S, Francis P, Handley M, et al. A Scalable Content-addressable Network[C]//Proc. of SIGCOMM'01. [S. l.]: ACM Press, 2001.
- [5] Rowstron A, Druschel P. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems[C]//Proc. of IFIP/ACM International Conference on Distributed Systems Platforms(Middleware). Heidelberg, Germany: [s. n.], 2001: 329-350.
- [6] Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-to-Peer Lookupservice for Internet Applications[C]//Proc. of SIGCOMM'01. [S. l.]: ACM Press, 2001.

编辑 任吉慧

(上接第 89 页)

采用多线程技术,对多个数据源同时并行处理。(3)合理安排查询优先级。对大数据量、耗时的查询,不在同一数据库的可并行优先查询;对同一数据库的查询,小查询优先进行,否则大查询将导致数据库整体性能下降。查询执行优化的目标就是使查询执行所行使用的系统资源尽量地少,从而降低整个系统开销。

4 结束语

查询分解是异构数据集成系统中的一个重要问题,一个好的查询分解算法能有效地提高数据集成系统的性能。本文研究的查询分解算法,思路简单,易于实现,对基于 XQuery 的异构数据集成系统具有较强的适用性。基于 FLWR 表达式层次的 XQuery 查询优化方法适用范围广,可与其他 XQuery 优化方法结合使用,达到更优的目的。从查询分解实验可见,本文的不足之处在于 XQuery 查询仅限于 FLWR 结构的查询,对包含其他复杂表达式的 XQuery 查询的处理方法未进行讨论。下一步研究工作是对本文提出的 XQuery 分解算法进行扩充,使其能处理更为复杂的 XQuery 查询,同时深入

分析多种数据源查询能力之异同,设计一套规范的查询能力描述算法和集成查询优化方法。

参考文献

- [1] Levy A Y, Rajaraman A, Ordille J J. Querying Heterogeneous Information Sources Using Source Descriptions[J]. VLDB, 1996, 3(6): 251-262.
- [2] Lenzerini M. Data Integration: A Theoretical Perspective[C]//Proc. of the 21st ACM SIGMOD Symposium on Principles of Database Systems. New York, USA: ACM Press, 2002.
- [3] 甄玉钢,刘璐莹,康建初.基于 XML 的异构数据库集成系统构架与开发[J].计算机工程,2006,32(2): 85-88.
- [4] 方长江,肖宗水,夏晓忠.异构数据源集成技术在军事中的应用[J].计算机工程与设计,2008,29(2): 510-512.
- [5] Park J H, Kang J H. Optimization of XQuery Queries Including for Clauses[C]//Proc. of the 3rd International Conference on Internet and Web Applications and Services. Washington D. C., USA: IEEE Computer Society Press, 2007.

编辑 金胡考