

# EIR: 具有超级结点的非结构化 P2P 中多维数据搜索框架研究 \*

曾碧卿<sup>1</sup> 陈志刚<sup>2</sup> 陈恒法<sup>1</sup> 吴锐珍<sup>1</sup>

(华南师范大学 南海校区计算机系 广东南海 528225)<sup>1</sup> (中南大学 信息科学与工程学院 长沙 410083)<sup>2</sup>

**摘要** 在具有超级结点的非结构化 P2P 系统中,研究了复杂多维数据的查询搜索策略,提出了一个应用于具有超级结点的非结构化 P2P 网络的综合框架,在该框架中,能够实现对多维数据共享、索引以及查询等操作的处理。以 R\*-tree 索引树为基础,提出了一种能够应用于 P2P 的扩展 R\*-tree 索引树,即 EIR-tree 树,研究了系统中集群信息的收集与维护、EIR-tree 树的构建与维护等方法 and 措施。

**关键词** 非结构化 P2P, 超级结点, 多维数据, 数据查询, 搜索框架

## EIR: The Search Framework of Multi-dimensional Dataset on Unstructured P2P Network with Super-peer

ZENG Bi-Qing<sup>1</sup> CHEN Zhi-Gang<sup>2</sup> CHEN Heng-Fa<sup>1</sup> WU Rui-Zhen<sup>1</sup>

(Department of Computer Engineering, Nanhai Campus, South China Normal University, Nanhai 528225)<sup>1</sup>

(College of Information Science and Engineering, Central South University, Changsha 410083)<sup>2</sup>

**Abstract** The query and search policy of supporting complex query over multi-dimensional data is researched on the unstructured P2P network with super-peer, and a comprehensive framework is proposed to apply for sharing, indexing, and querying multi-dimensional dataset. Based on the indexing tree of R\*-tree, a new adaptive indexing tree named Extended Indexing R\*-tree (EIR-tree) is proposed. All the methods such as the information collected and maintained and the EIR-tree constructed and updated are discussed.

**Keywords** Unstructured P2P, Super-peer, Multi-dimensional dataset, Data query, Search framework

### 1 多维数据查询搜索概述

近年来,在支持复杂多维数据查询搜索方面的研究已经取得了一些进展。文[1]中提出了一棵 P2PR-tree 树,它是对 P2P 框架中 R\*-tree 的扩充。然而,在非结构化 P2P 网络中一个非常重要的问题是维护一棵动态的 R-tree 树,这个难题尚未得到有效的解决。为了使动态 R-tree 树能够正常工作,需要很大的维护成本,因为在完成每一个维护操作的过程中,都需要通过消息在 P2P 系统的节点之间进行通信。在文[2]中提出了一个基于跳跃图的索引结构<sup>[3]</sup>,它是 SkipNet<sup>[4]</sup>在处理多维数据时的扩充。Tanin 与 Harwood 在文[5]中提出了一个基于四元组树 quad-tree 的技术,该技术能够支持范围查询,然而,由于它的应用和研究都是基于 Chord 的,因此,这就使得它仅适合于结构化 P2P 网络系统,然而对于非结构化 P2P 中的多维数据搜索尚缺乏相应的深入研究。

在各种多维数据访问方法中,R\*-tree 树<sup>[6]</sup>是最早 R-tree 的一个变种,它已经为工业界、研究机构和学术团体广为接受,R\*-tree 树是 B\*-tree 树在多维数据应用中的扩展。实质上,在 Oracle、IBM Informix 及 DB2 等数据库中对多维数据进行索引时,主要是通过 R\*-tree 树来实现的。如图 1 所示的是一棵索引了一组节点 {a, b, c, ..., k} 的 R\*-tree 树,并假设每一个节点的容量为 3。它是对在空间上靠得比较近的点(如: e, f, g)进行聚簇,并作为同一个层次叶子节点连接到一个节点(如: E6)中,并以一个最小边界矩形(Minimum Bounding Rectangle, MBR)来表示。然后,按照同样的原理,对空间中的节点不断地进行着递归聚簇,直至达到这棵树的最顶层,在树的顶层是一个唯一的树根节点。

R\*-tree 树能够有效地响应各种类型多维数据的查询请

求,特别是对范围查询更加有效。对于一个给定的查询窗口 q(如图 1 中的阴影部分所示),范围查询算法将会对所有处于阴影部分的数据或者是与查询对象 q 有交叉的数据都进行查询。使用文献<sup>[7]</sup>中所提出的算法,就能够对范围查询请求进行有效的处理,这样的处理过程是从 R\*-tree 树的根节点开始的。对于任何需要查询的数据来说,只要其所在的最小边界矩形(MBR)与查询窗口相交,那么它的子树就会得到递归查询。如果在查询过程中遇到了叶子节点数据,那么与查询窗口相交的最小边界矩形(MBR)中的所有数据对象都已经得到了查询,而与查询窗口不相交叉的数据(如: E3)则不会得到任何查询。

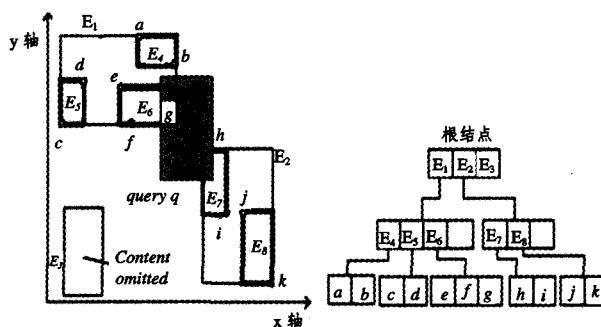


图 1 R\*-tree 树实例

R\*-tree 树能够有效地支持的另一类重要查询策略是最近 k 个邻居节点查询(kNN)。给定一个查询点 q 及其参数 k ( $\geq 1$ ),一个 kNN 查询算法将会按照到查询点 q 的最短路径来查询树中可能的 k 个数据对象。在文[8]中, D. Tsoumakos 和 N. Roussopoulos 等提出了一个分支与边界(branch-

\* 基金项目:国家自然科学基金资助项目(60573127)。曾碧卿 博士,副教授,CCF 会员,研究方向为网络计算、分布式处理、并行 I/O;陈志刚 教授,博士,博士生导师,研究方向为普适计算、分布式处理。

and-bound)算法,在该算法中,它是以深度优先的方式来遍历  $R^*$ -tree 树,在文[1]中,Mondal A. 与 Yilifu 对该算法进行了改进。分支与边界遍历算法是从根节点开始遍历的,它是根据被查询的数据对象到查询点  $q$  的最短路径 (mindist) 来对查询数据对象进行排序的,具有较小 mindist 值的子树将首先得到查询。在这种处理过程中不断地进行着反复递归,直至达到了树的叶子层,这样能够查询到排在前面的  $k$  个候选数据对象。假设离查询点  $q$  最远的待查询数据对象的 mindist 值是  $nn\_mindist$ ,在回溯至树的最顶层的遍历过程中,仅只有小于  $nn\_mindist$  的数据对象能够得到访问。现以如图 2 所示的例子来说明该算法,该查询对象树与如图 1 所示基本相同。

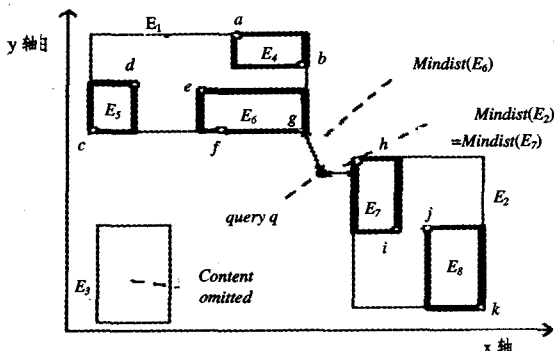


图 2 kNN 实例

假如现在欲查询找到距离查询点  $q$  最近的 2 级邻居节点中的数据,查询是从  $R^*$ -tree 中的根节点开始进行的,从查询点  $q$  到所有相邻数据对象节点的 mindist 值已经计算出来了。对象  $E_2$  由于具有最小的 mindist 值,因此它将首先得到访问,接下来, $E_2$  的叶子层节点  $E_7$  将得到访问, $h$  和  $i$  是最近邻居节点中的待查询节点。当前的  $nn\_mindist$  值是从  $h$  到  $q$ ,现在查询处理中将回溯  $R^*$ -tree,而  $E_8$  由于它的 mindist 值比  $nn\_mindist$  值要大,所以没有访问到。接下来要访问  $E_1$  中的节点,接着以类似的方式进行查询,找到了点  $g$ , $g$  取代  $h$  的原因是由于它更靠近  $q$ 。当前的  $nn\_mindist$  值调整为从  $g$  到  $q$ ,根据 mindist 值的大小,其它节点与数据均没有得到访问,最后的结果是点  $g$ 。在文[6]中,Beckmann N. 与 Kriegel H. P 提出了一个改进的算法,该算法使用堆以最快的方式遍历了  $R^*$ -tree 树。

## 2 P2P 系统拓扑结构

在详细介绍和讨论多维数据搜索框架之前,现在对研究中所需使用的 P2P 网络索引结构进行概述。如图 3 所示的拓扑结构为具有超级结点的 P2P 网络模型,该模型是由超级结点和普通节点所组成的集群系统结构,其中  $S_1$  和  $S_2$  等节点是一些超级结点,它们充当了本地服务器的角色,接受普通节点所提交的数据请求,而  $C_1$  和  $C_2$  等普通节点连接到超级结点上,向这些具有较强计算和处理能力的超级结点提交查询请求,待超级结点完成了查询请求后,它们就能够收到相应符合要求的查询结果。这个模型由于考虑了 P2P 系统中不同节点计算能力的异构性以及网络不同节点间带宽的差异,因此,该模型具有一定的可操作性和通用性。在文[9]中,Ala'a Qasim Al-Namiy 和 Faris S. Majeed 从正反两个方面对超级结点网络进行了全面的研究,并为系统设计者们在设计具有超级结点的 P2P 系统时制定了一些有用的规则。

本研究中也假定 P2P 系统中的节点具有相应的计算能

力,节点相互之间能够有效地进行通信、构建多维数据索引并且能够在节点本身数据集中对查询请求进行有效的查询处理。由于  $R^*$ -tree 树是应用于多维数据的主要索引方法,对本系统中的每一个节点都使用  $R^*$ -tree 树来对其自身数据进行索引,这个索引过程是在该节点加入到 P2P 网络系统之前就已经完成了。

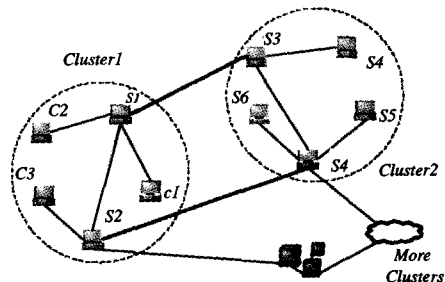


图 3 一个具有超级结点的 P2P 网络系统

由于以下两个方面的原因<sup>[2,10]</sup>,系统中的超级结点需要具备一定的冗余:

(1)如果 P2P 系统中的一个超级结点因为某种原因而发生了故障,此时,应该有其它的超级结点来为集群系统提供服务,以避免系统运行的瘫痪;

(2)超级结点之间能够相互共享系统的负载。虽然一个普通节点在某一时刻只能连接到一个超级结点上,但在研究过程中,集群系统还是应提供一个以上的超级结点,以形成超级结点的冗余。

## 3 EIR-tree 树思想的提出及其构建

通过对系统中哪些保存有语义上接近的数据的节点进行分组和聚簇,可以构建集群,这种数据上的接近是通过数据矩形,即最小边界矩形来衡量的。通过递归地划分数据空间,每一个集群在语义空间上均占据了一个区域,当一个新节点准备连接到该区域中的超级结点以加入到 P2P 网络系统中来时,它就会被路由到集群系统中进入相应的矩形区域。集群系统之间路由策略类似于 CAN 中所使用的路由机制,在这些路由机制中,既不考虑节点存储空间细节问题,也不使用任何的哈希类算法。在每一个集群内部,超级结点使用特殊的  $R^*$ -tree 树,即扩展索引树 (Extended Indexing  $R^*$ -tree, EIR-tree),索引了为所有节点共享用的数据。

EIR-tree 树的主要思想是:它是一种特殊的  $R^*$ -tree 树结构,在该树结构中,除了为所有节点所共享使用的哪些数据索引表是存储在超级结点中之外,实际的被索引过的哪些原始数据仍然是存储在集群系统中的各个节点上。

EIR-tree 树具有  $R^*$ -tree 树应有的查询处理能力,但是它在空间上会有更好的效率。当构建 EIR-tree 树索引时,为了节省超级结点的存储空间和计算资源,并不是所有节点中的数据都会插入到 EIR-tree 树中并迁移到超级结点上,仅仅只有少量的、汇集了节点数据的最小边界矩形 (MBRs) 会插入到 EIR-tree 树中。在集群系统中,EIR-tree 树向超级结点提供了一些数据信息,一个超级结点在接收到一个数据查询请求时(如范围查询或者是最近  $k$  个邻居节点查询等),它会将这个查询请求转发到相应的集群系统中。例如:一个范围查询请求会被转发到集群系统中的相应区域,这个区域一般是基于范围查询所应落入的矩形区域。相应的集群就会对该查询请求进行处理,集群系统既会在本地数据集中进行数据搜索,也会将这些数据查询扩大到邻近集群系统中。相

应的集群系统都会对查询进行处理,并将查询得到的结果返回到查询源节点中。

#### 4 P2P 网络系统中集群信息的收集及维护

一个 P2P 集群系统就是由一组运行中的节点所组成的集合,其中包括超级节点和普通节点,超级节点充当本地服务器,通过与其在地理位置上相邻相近的节点及所存储的数据而构成集群,这样的拓扑结构是本论文研究 P2P 系统多维数据搜索查询策略的基础架构。

不失一般性,对于任何一组多维数据,研究中假设其在每一维上都有一个域值范围,如:  $[0, 10k]$ , 这里,  $k$  值可以取某一个整数。如果在数据的任意一维上,有数据值超出了所设定的范围,那么可以依据一定的比例对其进行合理的调整。通过对数据矩形区域及其中的数据进行分析与研究,系统能够对处于域中的每一个节点进行唯一准确的定位。在这种情况下,一个节点的位置也表明了它在共享数据矩形区域中的位置,而研究中则使用矩形区域的矩心来代表处在该域中的节点。

当 P2P 系统在启动运行时,系统中只有一个集群。当系统中所拥有的节点数量超过了它的容量时,该集群区域就会分成大小相等的两个子区域,而每一个子域则构成了一个新的集群系统,对于域的拆分处理过程与 CAN 中的方法相同。在每一个新的子域中,根据各个节点的计算能力来选取新的超级节点。下面以图 4 为例来进行详细的分析和说明,假设该集群的容量是 4 个节点,而且它是二维的。在系统中最初恰好只有 4 个节点,其中 P1 是一个超级节点,如图 4(a) 所示,为简单起见,在每一个集群系统中只设定一个超级节点。当节点 P5 加入到该系统中以后,整个系统被分割为两个子域,如图 4(b) 所示。在图 4(b) 下边的子域中,按照相应的规则,选取 P4 为超级节点。当有新节点加入到系统中时,它们将会分配到与其数据相适应的一个矩形区域中,随着更多新节点的加入,一个子域系统还可以进一步分为更多的子域,如图 4(c) 所示。

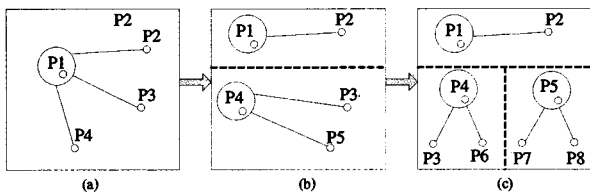


图 4 P2P 系统中域的分割

在数据存储集合中,集群信息机制尽量使相邻相近的节点归入到同一个或者是相邻的集群中,这将极大地改善同一个集群中数据位置的局部性。使用与 CAN 相同的路由机制,一个  $d$  维的系统空间可以分割为  $n$  个相同的子域(即集群),这时平均的路由路径长度为  $(d/4)(n^{1/d})$ ,每个集群保持了  $2d$  个相邻的邻居。

#### 5 多维数据查询搜索综合框架和 EIR-tree 树的构建与维护

在每一个集群中,通过普通节点的协作与配合,超级节点可以在系统中进行数据的索引与查询处理。为了不使超级节点过载,并在系统中实现负载均衡,系统中将一部分索引与查询处理任务交给普通节点去执行,下面将讨论如何在一个集群内构建一棵 EIR-tree 树索引,并且当系统中有新节点加入或者是运行中的节点离开系统时,能够对这个索引表进行维

护。多维数据查询搜索综合框架的核心就是构建一棵 EIR-tree 树,该综合框架具有以下两个突出的特点:

(1) 具有较强计算能力的若干个节点一起协同工作,完成对多维数组的索引与查询处理;

(2) 为了使整个 P2P 网络系统具有良好的可扩展性,并在系统中实现负载均衡,系统中的普通节点需要参加完成部分的计算任务。

##### 5.1 构建 EIR-tree 树的思路与方法

在 P2P 系统的每一个集群环境中,超级节点除了使用一棵  $R^*$ -tree 树来精确地索引其自身的数据外,在每一个超级节点上都还维护有一棵 EIR-tree 树,它能够索引其自身以及系统中每一个节点上的数据。超级节点中的  $R^*$ -tree 树与普通节点中的  $R^*$ -tree 树作用相同,然而,EIR-tree 树通过对最小边界矩形(MBRs)进行索引,对每个普通节点及超级节点中的数据进行汇总、并形成数据摘要,从而显示出它在多维数据查询搜索中的重要作用。这些特殊的最小边界矩形(MBRs)是通过在每个节点上维护的  $R^*$ -tree 树中已有的 MBRs 进行进一步的聚簇而得到的。现在通过一个实例来说明构建 EIR-tree 树的处理过程,EIR-tree 树的构建如图 5 所示,为了达到有效的比较,仍然使用与如图 1 中相同的数据来进行分析。

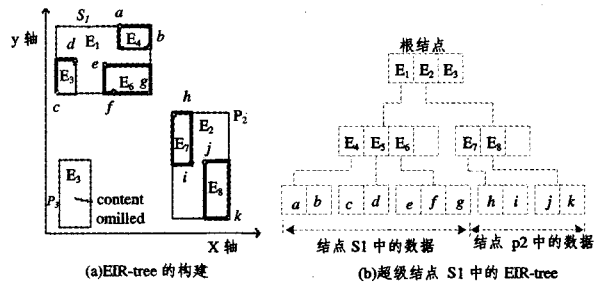


图 5 EIR-tree 树示例

假设在集群系统中有三个节点,它们分别是 S1, P2 和 P3,其中 S1 是一个超级节点。为简便起见,假定系统中只有一个超级节点,当然该例子也能很容易地扩展到具有两个超级节点的情况。S1 中连接有节点  $\{a, b, \dots, g\}$ , P2 中连接有节点  $\{h, i, j, k\}$ 。在这两个节点中,都是使用  $R^*$ -tree 树来对数据进行索引,如图 5 所示。假设最初在系统中只有一个节点 S1, P2 通过连接到 S1 而加入到系统中来, S1 应该通知 P2 系统中能够容纳最小约束矩形 MBRs 的数量,并以  $m$  表示。如:假设  $m=2$ ,则表示系统中允许普通节点向超级节点发送更多的信息,但是也将会导致超级节点更多的空间开销。假设  $m=1$ ,那么 P2 在它的  $R^*$ -tree 树中应该重组当前的 MBRs,以形成一个更大的 MBR。由于在根层次 E2 上只有一个 MBR,它将会发送到 S1 上,同时, S1 也需要使用它的  $R^*$ -tree 树来执行同样的任务。S1 选择 E1 的 MBR,同样,假定在节点 P3 中又有另一个 MBR。使用以上三个 MBRs, S1 可以开始构建它自己的 EIR-tree,如图 5(b) 所示,在该图中树根节点是存储于节点 S1 上的 EIR-tree 树,而实际数据是驻留在所有节点中。在 EIR-tree 树中的每一项都有一个额外的位,它是用来标识数据是存储在本地还是存储在其它节点上。如果在集群系统中存在多个超级节点的话,如同 S1 中构建 EIR-tree 树的操作将会在所有的超级节点上进行,以得到多个具有同样内容的 EIR-tree 树的复制,以形成冗余。

在实际运行的系统中,每一个节点中所得到的总 MBRs

(下转第 91 页)

**结束语** 本文中,我们提出了一种基于混沌映射产生动态S盒的方法。通过迭代混沌映射,产生整数序列 $\{0, 1, \dots, m-1\}$ 的一个置换。利用混沌映射具有对初始值和控制参数敏感的特征,微小地改变初始值和控制参数,将得到整数序列的不同置换,从而达到动态生成S盒的目的。利用我们提出的算法,生成了10000个S盒并对这些S盒的性能进行分析,结果表明,绝大部分的S盒具有很好的性质,能抵御差分密码分析、线性密码分析等现代密码分析技术的攻击。同时,由于S盒是动态生成,提高了对抗各种密码分析技术攻击的能力。

**参考文献**

- 1 Forre R. The strict avalanche criterion: spectral properties of Boolean functions and an extended definition. In: Advances in cryptology. Proc of CRYPTO-88. Berlin: Springer-Verlag; 1989
- 2 Detombe J, Tavares S. Constructing large cryptographically strong S-boxes. In: Advances in cryptology, Proc. of CRYPTO92, Lecture notes in computer science; 1992

- 3 Jakimoski G, Kocarev L. Chaos and cryptography: block encryption ciphers. IEEE Trans Circ Syst-I, 2001, 48(2): 163~170
- 4 Kocarev L, Jakimoski G. Logistic map as a block encryption algorithm. Phys Lett A, 2001, 289: 199~206
- 5 Yi Xun, Tan C H, Siew C K. A new block cipher based on chaotic tent maps. IEEE trans. Circuits and Systems-I, 2002, 49(12): 1826~1829
- 6 Li Shujun, Chen Guanrong, Mou Xuanqin. On the Security of the Yi-Tan-Siew Chaotic Cipher. IEEE Trans. Circuits and Systems-II; Express Briefs, 2004, 51(2): 665~669
- 7 Pieprzyk J, Finkelstein G. Towards Effective Non-linear Cryptosystem design. IEEE Proceedings, Part E: Computers and Digital Techniques, 1988, 135: 325~335
- 8 Adams C, Tavares S. Good S-boxes Are Easy to Find, Advances in Cryptology. In: Proc. of CRYPTO'98, Lecture Notes in Computer Science, 1989. 612~615
- 9 Webster A, Tavares S. On the Design of S-boxes, Advances in Cryptology. In: Proc. of CRYPTO'85, Lecture Notes in Computer Science, 1986. 523~534
- 10 Webster A. F, Tavares S E. On the design of S-boxes. In: Advanced in Cryptology; Crypto'85 Proc. Springer, 1986

(上接第61页)

数量 $m$ 需要在集群系统内部的超级结点间进行协商处理,就 $m$ 的一般值而言,为了在所有的数据中形成 $m$ 个MBRs,每一个节点在所有的对象中要进行 $k$ 种方式的聚簇。本论文研究中,也对系统中节点上已有的 $R^*$ -tree树进行调整,以便能够更有效地执行聚簇操作。

在 $R^*$ -tree树的维护过程中,如果在节点和数据插入处理中出现溢出,那么系统中针对 $R^*$ -tree树的溢出处理机制就会自动运行。在维护EIR-tree树的过程中,当有MBRs插入到一棵EIR-tree树中时,可只对 $R^*$ -tree树算法进行少量修改,就能够应用于超级结点中以维护一棵EIR-tree树。当一个节点加入到系统中时,与插入 $m$ 个矩形对象相同的是,系统运行的总体成本就会增加,不过,对于一些较小的 $m$ 值来说,所增加的成本非常小。在S1与P1中的 $R^*$ -tree树如图6所示。

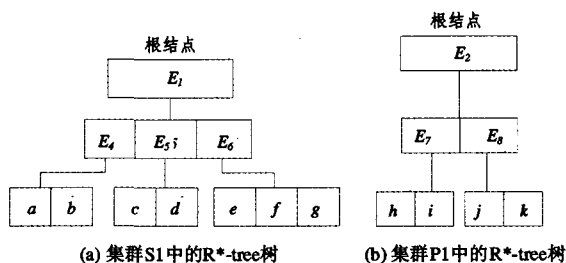


图6 在S1与P2中的 $R^*$ -tree树

**5.2 EIR-tree树的维护方法**

EIR-tree树的维护是由系统中某个节点的加入或者离开而产生的,当一个普通节点离开系统时,仅仅只需要在超级结点的EIR-tree树中对这种情况进行标记即可。但是当一个超级结点离开系统时,系统需要选取一个新的超级结点来代替原来离开系统的超级结点,以行使相应的功能。

当一个普通节点离开系统时,EIR-tree树应得到及时更新,以反映脱离系统节点中的共享数据不能再响应数据访问请求的实时信息。离开系统的节点中的所有MBRs都应从EIR-tree树中删除。当一个超级结点将要离开系统时,需要对所有处于活动状态的普通节点进行分析,主要是比较它们的计算能力和带宽等性能,从而推选出一个节点来作为新超级结点。新超级结点将会从已经存在的超级结点中获取

EIR-tree树的复制信息,将所有已经存储于本地的哪些数据的标识位信息置为失效,删除EIR-tree树中有关已经离开系统的超级结点信息,并将标识自身节点中数据信息的标识项置为本地标志。在P2P网络系统中,如果在某一时刻只有一个超级结点,而这时它失效了,那么这个集群系统就被破坏了,所有的节点将会重新来构建P2P网络。由于P2P系统的构建和运行强调超级结点的冗余,因此,发生这种情况的可能性非常少。当一个具有较强计算能力的节点加入到集群系统中以后,它将会取代系统中计算能力最弱的哪个超级结点,从而使得系统中的超级结点发生改变,并进行相应信息的迁移,如EIR-tree树信息的更新与迁移等。

**结论** 为了提高P2P系统对查询复杂多维数据的有效处理能力,增强系统的可扩展性,论文中首先研究了一个对多维数据能够进行共享、索引以及查询的综合框架,基于该框架,通过对 $R^*$ -tree的改进,研究并提出了EIR-tree树,它是一种具有最大空间范围的扩展索引 $R^*$ -tree。EIR-tree由于索引了簇簇空间中所有节点上的数据,能够有效处理诸如范围查询以及在最近 $k$ 个邻居节点上进行数据查询等复杂的查询处理。

**参考文献**

- 1 Mondal A, Yilifu, Kitsuregawa M. P2PR-tree: An R-tree-based Spatial Index for Peer-to-Peer Environments. In: EDBT Workshop, 2004
- 2 Zhang C, Krishnamurthy A, Wang R. SkipIndex: Towards a Scalable Peer-to-Peer Index Service for High Dimensional Data: [Tech. Report]. Princeton Univ., 2004
- 3 Aspnes J, Shah G. Skip Graphs, SODA, 2003
- 4 Harvey N, et al. SkipNet: A Scalable Overlay Network with Practical Locality Properties. USITS, 2003
- 5 Tanin E, Harwood A. A Distributed Quadtree Index for Peer-to-Peer Settings. ICDE, 2005
- 6 Beckmann N, Kriegel H P, Schneider R, et al. The  $R^*$ -tree: An Efficient and Robust Access Method for Points and Rectangles. SIGMOD, 1990
- 7 Bharambe A R, Agrawal M. Mercury: Supporting Scalable Multi-Attribute Range Queries. SIGCOMM, 2004
- 8 Tsoumakos D, Rousopoulos N. Adaptive probabilistic search in peer-to-peer networks. In: Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03), 2003
- 9 Ala'a Qasim Al-Namiy, Majeed F S. Improving Query Answering in Peer-to-Peer Data Searching. In: Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05), 2005
- 10 Gaede V, Günther O. Multidimensional Access Methods. ACM Computing Surveys, 1998, 30(2)