

# 虚拟计算环境中的覆盖网技术

卢锡城 李东升

(国防科学技术大学并行与分布处理国家重点实验室 长沙 410073)

**摘 要** 互联网资源的成长、自治和多样等特性给资源的有效聚合带来了巨大挑战. 通过覆盖网动态组织互联网资源并支持资源的高效搜索, 是虚拟计算环境中资源按需聚合的重要途径之一. 文中概述了虚拟计算环境中覆盖网技术的研究进展. 针对互联网资源的成长性和自治性等特点, 阐述了基于 Kautz 图的高效覆盖网拓扑构造方法, 进而给出了适用于任意正则图的通用覆盖网拓扑构造方法; 针对互联网资源的多样性等特点, 提出了支持分组的覆盖网拓扑构造方法; 在此基础上, 阐述了基于覆盖网的高效区间搜索技术, 并对覆盖网拓扑的优化方法及其它复杂搜索技术进行探讨.

**关键词** 虚拟计算环境; 聚合; 覆盖网; 拓扑构造; 资源搜索

中图法分类号 TP393

## Overlay Technologies for Internet-Based Virtual Computing Environment

LU Xi-Cheng LI Dong-Sheng

(National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073)

**Abstract** Internet resources have the natural characteristics of growth, autonomy and diversity, which have brought great challenges to the efficient aggregation of these resources. Utilizing overlay to organize resources and support efficient resource discovery is an important approach to aggregate resources on-demand in Internet-based Virtual Computing Environment (iVCE). This paper gives an overview of research advances on overlay technologies in iVCE. To adapt to the growth and autonomy of Internet resources, an overlay topology construction mechanism based on Kautz graph is first introduced, and it is extended to a universal method for constructing overlay topologies based on arbitrary regular graphs. To adapt to the diversity of Internet resources, a grouped overlay topology construction mechanism is then proposed. Based on the overlay constructed above, an efficient range query scheme is presented, and the optimization methods of overlay topology as well as other complex query techniques are discussed.

**Keywords** Internet-based Virtual Computing Environment (iVCE); aggregation; overlay; topology construction, resource discovery

## 1 引 言

经过 40 年的发展, 互联网已成为现代社会的重要信息基础设施和无处不在的计算平台. 当前, 互联

网上汇聚了大量的计算资源、存储资源、数据资源和应用资源等各类资源. 随着国家信息化的推进, 经济、行政、科研、教育等各个领域都对互联网资源的共享和综合利用提出迫切的需求. 在这种背景下, 面向互联网的虚拟计算环境<sup>[1]</sup> (简称虚拟计算环境) 应

运而生. 虚拟计算环境建立在开放的互联网基础设施之上, 以资源的按需聚合与自主协同等为核心机制, 试图通过对互联网资源的虚拟化和自主化, 为终端用户或应用系统提供可信、透明的一体化服务, 实现有效的资源共享和便捷协作.

互联网环境及其资源的成长、自治和多样等自然特性, 给虚拟计算环境中资源的按需聚合带来了巨大挑战. 由于分布哈希表(DHT)覆盖网等技术<sup>[2]</sup>具有可扩展、自适应、自组织等良好特性, 通过覆盖网技术有效组织和管理大量动态的互联网资源, 成为虚拟计算环境中资源按需聚合的重要途径之一. 本文将以建立相对稳定的资源组织视图为目标, 对虚拟计算环境中的覆盖网技术展开阐述.

## 2 面向资源聚合的覆盖网技术

互联网是一个不断成长的开放系统, 其覆盖地域不断扩大, 大量分布异构的资源动态地更新与扩展, 资源的规模及其关联关系也在不断地成长变化, 导致资源管理的范围难以确定. 互联网资源的成长性和自治性等特性使得资源配置只能在变化中保持视图的相对稳定. 在开放、动态变化的互联网环境下, 既不可能也不必要获得传统意义下全局、时空一致的资源特征信息. 因此, 如何根据需求、基于局部信息, 有效组织互联网资源, 建立相对稳定的资源视图并支持任务完成, 是虚拟计算环境面临的重要挑战性问题<sup>[1]</sup>.

为支持资源的按需聚合, 在互联网资源之上, 虚拟计算环境建立了两个层次的视图, 如图 1 所示. (1)应用视图. 虚拟计算环境通过提供聚合模型和新型语言设施<sup>[3]</sup>, 实现应用视图与资源物理视图的解耦和动态绑定, 支持应用视图在资源动态变化时

的相对稳定; (2)组织视图. 虚拟计算环境通过提供高效资源覆盖网拓扑构造和资源搜索方法, 支持形成相对稳定的资源组织视图. 组织视图中组织的可以是物理资源, 也可以是虚拟化后的虚拟资源.

以建立相对稳定的资源组织视图为目标, 虚拟计算环境中的覆盖网技术主要包括两个组成部分 (如图 2 所示):

(1)覆盖网拓扑构造, 即根据局部信息, 在资源间建立一定的逻辑关联关系, 以实现互联网资源的有效组织. 针对互联网资源的成长性和自治性等特点, 我们首先设计了基于 Kautz 的高效覆盖网拓扑构造方法 FissionE<sup>[4]</sup>, 并将其扩展为可基于任意正则图的通用覆盖网拓扑构造方法. 针对互联网资源的多样性等特点, 我们进一步提出了支持资源分组的覆盖网拓扑构造方法. 本文第 3 节将对这些方法进行概要介绍.

(2)基于覆盖网的资源搜索, 即根据需求, 在覆盖网中有效地查找到符合条件的资源. 为满足多种应用需求, 虚拟计算环境需要提供精确匹配搜索、区间搜索、聚合搜索等多种搜索能力. 本文第 4 节将对这些方法进行讨论.

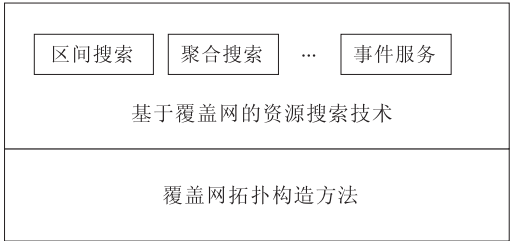


图 2 虚拟计算环境中的覆盖网技术

## 3 覆盖网拓扑构造方法

为适应互联网资源的成长性和自治性等特性, 虚拟计算环境中覆盖网拓扑构造方法需要具有良好的可扩展性、自组织性和适应性, 能够通过各节点的局部决策, 建立相对稳定的动态覆盖网拓扑, 并能有效适应系统规模的不断成长变化以及自治资源的动态加入或退出.

为支持资源的高效聚合, 虚拟计算环境中覆盖网还应具有较好的性能特性. 覆盖网性能评价的重要参数包括节点度数、网络直径以及负载平衡和拥塞特性等. 节点度数指覆盖网中每个节点需要维护的邻居节点的个数, 反映了覆盖网的维护开销; 网络直径则是指覆盖网中节点间通过覆盖网路由到达对

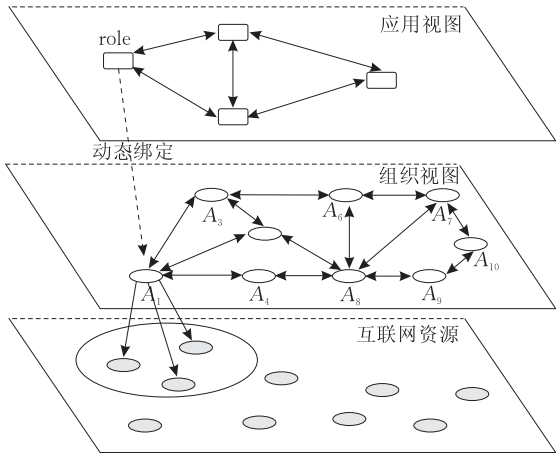


图 1 虚拟计算环境中的两种视图

方的路径长度的最大值,平均路由长度和网络直径则反映了覆盖网中消息路由的延迟开销.下面将概述虚拟计算环境中的覆盖网拓扑构造方法.

### 3.1 基于 Kautz 图的高效覆盖网构造方法

现有的很多 DHT 覆盖网<sup>[2]</sup>都是基于一些传统的静态拓扑图扩展而来的.在设计覆盖网<sup>[4]</sup>时,我们的思路是从分析静态拓扑图的特性分析入手,以获得具有良好特性的拓扑图,并以此为基础设计动态覆盖网的高效拓扑构造方法.

研究表明,在特定节点规模和节点度数下,存在较优网络直径的静态网络拓扑图,即 Kautz 图<sup>[4-5]</sup>. Kautz 图  $K(d, k)$  是一个节点出度和入度都为  $d$ 、网络直径为  $k$  的有向图.对  $K(d, k)$  中的每个标识为  $u_1 u_2 \cdots u_k$  的节点  $U$  (记为  $U = u_1 u_2 \cdots u_k$ ), 都有  $d$  条出边,即:对任意  $\alpha \in \{0, 1, 2, \cdots, d\}$  且  $\alpha \neq u_k$ , 节点  $U$  都有一条到节点  $V = u_2 u_3 \cdots u_k \alpha$  的出边. Kautz 图  $K(d, k)$  中各节点的标识都是基底为  $d$ 、长度为  $k$  的 Kautz 串,即串  $\xi = a_1 a_2 \cdots a_k, a_i \in \{0, 1, 2, \cdots, d\} (1 \leq i \leq k)$  且  $a_i \neq a_{i+1} (1 \leq i \leq k-1)$ . Kautz 串的特点是相邻字符不相同.可以证明在均匀路由负载情况下,采用优化的 Kautz 图路由算法, Kautz 图拓扑是拥塞可控(几乎无拥塞)的<sup>[4]</sup>.

Kautz 图具有良好的拓扑特性,但 Kautz 图是静态拓扑,无法满足互联网资源覆盖网的动态成长和节点自主加入退出等要求.因此,我们设计了高效的 DHT 覆盖网拓扑构造方法——FissionE<sup>[4]</sup>. FissionE 方法的基本思路是通过设计覆盖网的邻居拓扑规则和动态维护机制,使得动态的覆盖网拓扑在满足资源成长性和自治性等特性要求的同时,能够实现静态 Kautz 图的动态逼近,从而尽可能具有近 Kautz 图的良好特性.

与 Kautz 图节点规模是特定值不同, FissionE 覆盖网中资源节点规模可以是任意整数值.因此在 FissionE 覆盖网中,各节点的标识都是基底为 2 的 Kautz 串,但其标识长度可以是不同的,如图 3 所示.我们设计了“邻居拓扑规则”来指导 FissionE 覆盖网拓扑的构造和维护,以使得 FissionE 拓扑尽可能逼近静态 Kautz 图.“邻居拓扑规则”定义如下:对 FissionE 覆盖网中的任意节点  $U$  和  $V$ ,若节点  $U$  和  $V$  为邻居,则它们的节点标识长度之差不超过 1. 根据“邻居拓扑规则”和 Kautz 图的特性,设节点  $U$  的标识为 Kautz 串  $u_1 u_2 \cdots u_k$ , 则节点  $U$  的出边邻居的节点标识为  $u_2 u_3 \cdots u_k q_1 \cdots q_m (u_2 u_3 \cdots u_k q_1 \cdots q_m$  是基底为 2 的 Kautz 串,  $0 \leq m \leq 2$ ). 图 3 给出了 FissionE

覆盖网拓扑的示例.

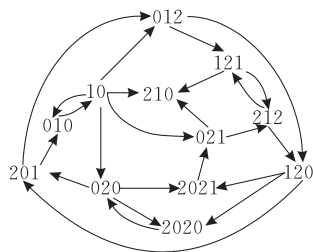


图 3 FissionE 覆盖网拓扑的示例

在 FissionE 覆盖网中,节点会由于自治性等原因不断加入或退出.为提高 FissionE 覆盖网拓扑的可扩展性和自适应性,我们设计了“局部优化”的覆盖网动态拓扑构造和维护机制,使得节点动态加入或退出时,都能够通过局部决策来维护 FissionE 覆盖网拓扑的相对稳定.该机制包括:

(1) 节点加入处理:当新节点  $P$  加入时,尽可能加入到覆盖网中节点标识“局部最短”的节点处(我们将其形象地称为覆盖网的“稀疏”处,如图 3 中节点 10 处),具体流程如下:节点  $P$  首先获知覆盖网中的一个随机节点  $W$ ,节点  $W$  发起一个更新消息.从节点  $W$  开始,若当前节点的标识长度比其邻居的长,则更新消息被转发给此邻居;更新消息会一直被转发到标识短的邻居,直到最终到达了标识“局部最短”的节点  $V$ ,即节点  $V$  没有一个邻居的标识比  $V$  的标识短.因此节点  $P$  加入到覆盖网中作为节点  $V$  的兄弟节点,然后重新设置  $P$  和  $V$  的标识,并更新相关节点的邻居关系.例如,在图 3 中,设节点  $P$  加入时获取的随机节点为节点 010,则更新消息会停止在节点 10;节点  $P$  会加入到覆盖网的节点 10 处,节点  $P$  的标识设置为 102,原标识为 10 的节点的标识改为 101,更新后的 FissionE 覆盖网拓扑如图 4 所示(图中虚线表示需更新的邻居关系).

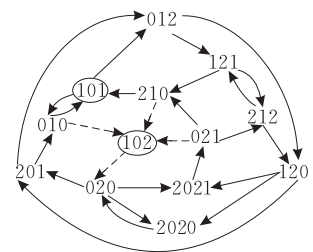


图 4 节点  $P$  加入后的覆盖网拓扑

(2) 节点退出处理:由标识“局部最长”的节点处(即覆盖网“密集”处)的节点接管,细节略<sup>[4]</sup>.

由上述过程可知,当节点加入或退出时,相关节点只需根据局部信息进行决策,就可维护 FissionE

覆盖网拓扑的相对稳定。

理论分析和实验表明, FissionE 覆盖网具有良好的性能。对节点规模为  $N$  的 FissionE 覆盖网, 其平均节点度数为 4, 网络直径小于  $2\log_2 N$ , 节点加入或退出时更新消息传播的节点数量小于  $\log_2 N$ , 并且 FissionE 覆盖网具有较好的负载均衡和低拥塞特性。与相同节点度数的 CAN 和 Koorde 相比, 在节点规模较大时, FissionE 具有较大的性能优势。

### 3.2 基于任意正则图的通用覆盖网拓扑构造方法

现有的很多覆盖网都是针对所基于的静态拓扑图, 设计专用的覆盖网拓扑构造方法。由于各种拓扑图众多, 我们希望能够设计通用的覆盖网构造拓扑方法, 以根据虚拟计算环境应用场景需求, 系统地比较基于各种拓扑图的覆盖网的优劣, 从而选择合适的覆盖网拓扑。虽然不同覆盖网的拓扑构造方法各不相同, 但是其本质均可以抽象为处理节点加入退出时的拓扑维护问题, 即: 设当前系统的覆盖网拓扑图为  $G$ , 如何设计分布式算法, 能够在节点加入或退出时以较小的代价得到新图  $G'$ , 同时尽可能维护原图的拓扑性质(如节点度数和路由算法等)。

针对该问题, 我们对 FissionE 覆盖网的拓扑维护机制进行扩展, 设计了适用于任意正则图的通用 DHT 覆盖网拓扑构造方法——DGO 技术<sup>[6-7]</sup>。DGO 技术受到了线图(Line Graphs, LG)迭代技术<sup>[8]</sup>的启发。线图迭代作为一种通用拓扑维护方法, 在 20 世纪 90 年代得到研究并应用于多处理器互连网络领域。但线图迭代是一种集中式算法, 需要全局拓扑信息和集中式控制, 因此不适用于虚拟计算环境中分布自治的覆盖网。

通用覆盖网拓扑构造方法所面临的主要挑战在于缺少具体的覆盖网拓扑图信息以及如何适应互联网资源的成长性和自治性等特点。针对上述挑战, 我们设计了一系列的机制和算法, 包括分布式线图(DL)迭代、节点合并与分裂等。

#### (1) 分布式线图(DL)迭代

不同拓扑图通常具有不同的节点命名方法, 为便于对通用覆盖网拓扑构造方法进行描述, 下面首先提出一种适用于任意图的统一节点命名机制。假设初始图  $G_0$  是一个具有  $N_0$  个节点的  $d$ -正则图。令  $X$  是一个具有  $N_0$  个字符的字符集并且每个字符的标识长度为 1, 则初始图  $G_0$  中的  $N_0$  个点将被依次命名为  $X$  中的  $N_0$  个字符, 并且对任意节点  $\alpha \in G_0$ , 有一个入边字符集合  $\zeta(\alpha) = \Gamma_{G_0}^-(\alpha)$  和一个出边字符集合  $\psi(\alpha) = \Gamma_{G_0}^+(\alpha)$ 。下文将简单地把  $G_0$  中的点依

次命名为  $0, 1, \dots, N_0 - 1$ , 如图 5(a) 所示。

DL 迭代的核心思想是边点变换, 即通过把原图中的边变换为点, 以得到新图。令  $u = u_1 u_2 \dots u_m$ ,  $v = v_1 v_2 \dots v_n$ ,  $m \geq n$ , 定义  $u \circ v = u_{m-n+1} v$ 。在一次 DL 迭代  $G_{i+1} = DL(G_i, v^{(i)})$  中, 负责点  $v^{(i)} \in V(G_i)$  的标识长度不大于其任意直接邻居的标识长度, 具体过程如下:

(i) 令新图  $G_{i+1}$  与原图  $G_i$  相同, 即  $G_{i+1} = G_i$ 。

(ii) 在新图  $G_{i+1}$  中删除点  $v^{(i)}$  以及  $v^{(i)}$  的所有入边和出边。

(iii) 对原图  $G_i$  中负责点  $v^{(i)}$  的每一条入边  $[u, v^{(i)}]$ , 在新图  $G_{i+1}$  中增加一个新点  $u \circ v^{(i)}$ 。

(iv) 对新图  $G_{i+1}$  中的每一个新点  $u \circ v^{(i)}$ , 增加入边  $[u, u \circ v^{(i)}]$ 。

(v) 对新图  $G_{i+1}$  中的每一个新点  $u \circ v^{(i)}$ , 增加出边  $[u \circ v^{(i)}, w]$ , 其中  $w$  满足  $w \in \Gamma_G^+(v^{(i)})$ 。

图 5 给出了 DL 迭代的一个示例。

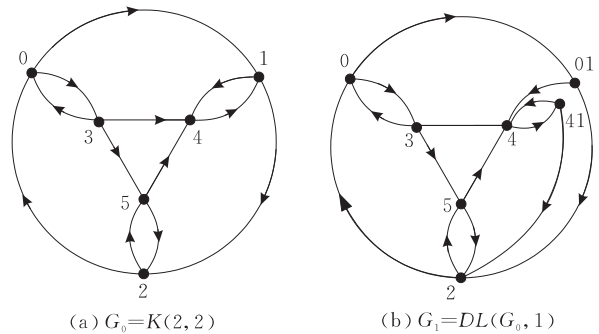


图 5 DL 迭代示例

#### (2) 节点合并与分裂

令初始正则图的基为  $d$ , DL 迭代算法的每次迭代将增加  $d-1$  个点, 从而在  $d > 2$  时无法直接应用于覆盖网的拓扑构造。针对该问题, 我们通过逻辑节点的合并与分裂机制, 支持以任意  $d$  为基的正则图。

令图  $G$  是以  $d$  为基的 DL 图, 图  $G' = DL(G, v)$ , 负责点  $v = v_1 v_2 \dots v_m$ 。可以证明, 在图  $G'$  中将增加  $d$  个新点  $\alpha v_1 v_2 \dots v_m$ , 其中  $\alpha \in \zeta(v_1)$ 。点合并过程如下。前一半点合并为一个新节点  $s$ ; 后一半点合并为一个新节点  $t$ 。节点  $s$  (或  $t$ ) 的入边为合并前各点的入边的并, 节点  $s$  (或  $t$ ) 的出边为合并前各点的出边的并, 节点  $s$  (或  $t$ ) 所包含的点的个数被称为节点  $s$  (或  $t$ ) 的秩, 记为  $\llbracket s \rrbracket$  (或  $\llbracket t \rrbracket$ )。上述操作被称为合并(merge)操作, 记为  $G'' = Merge(G', v)$ 。

点分裂可以看作是点合并的逆过程。如果在 DL 迭代  $G' = DL(G, v)$  前负责点  $v$  的秩大于 1, 那



么一次分裂操作  $G' = Split(G, v)$  将取代此次 DL 迭代。分裂操作将把负责点  $v$  所包含的点(以及各点的边)分成两份。负责点  $v$  将得到前一半点;一个新节点将得到后一半点。显然分裂操作后新图  $G'$  比原图  $G$  增加了一个节点。

显然, DGO 技术仅需要负责点  $v$  的直接邻居信息, 因此可以应用于 DHT 覆盖网的拓扑构造。令  $d$  为初始正则图的基,  $N_0$  为初始正则图的节点数,  $D_0$  为初始正则图的直径,  $N$  为系统当前节点数。可以证明: 基于 DGO 技术构建的覆盖网的节点出度为  $d$ , 节点入度在 1 和  $2d$  之间, 网络直径小于  $2(\log_d N - \log_d N_0 + D_0 + 1)$ , 节点加入退出维护开销为  $O(\log_d N)$ , 每次节点加入退出时最多有  $3d$  个节点需要更新其路由表。

由于 FissionE 覆盖网是基于 Kautz 图  $K(2, k)$  设计的, 其平均节点度数是 4。应用 DGO 技术, 我们可设计基于任意 Kautz 图  $K(d, k)$  的 DHT 覆盖网 DGO\_Kautz(简称 DK)。DK 覆盖网平均节点度数为  $2d$ , 网络直径小于  $2\log_d N + 2$ , 具有较好的性能。因此 DK 覆盖网可根据需要, 将平均节点度数设置为所需的数值。

### 3.3 支持分组的覆盖网拓扑构造方法

现有 DHT 覆盖网通常假设各节点是同构的, 并且使用同一路由算法对所有消息进行路由。然而, 虚拟计算环境中的互联网资源通常是异构的, 在计算能力、信誉、稳定性和管理域从属关系等各方面都存在广泛差异。传统 DHT 覆盖网难以很好地适应互联网资源的多样性等特点<sup>[1]</sup>。

针对虚拟计算环境中的资源多样性特点, 我们在 DK 覆盖网的基础上提出一种支持分组的覆盖网技术——Glax<sup>[9]</sup>。Glax 支持上层应用根据资源属性的不同对覆盖网中的节点进行分组, 进而允许在消息路由过程中采用灵活的路由策略, 主要包括:

(i) 目标指定 (Destination-Specified, DS) 路由。给定一个组  $X$  和关键字  $K$ , DS 路由将把消息路由至组  $X$  中负责关键字  $K$  的唯一节点;

(ii) 路径受限 (Path-Constrained, PC) 路由。给定关键字  $K$ , 设消息产生于组  $X$  中的任一点, PC 路由将该消息路由至组  $X$  中负责关键字  $K$  的唯一节点, 且消息路由都是在组  $X$  中进行。

图 6 给出了覆盖网中节点分组的示例。在图 6 中, 覆盖网中的节点形成了两个组(组  $X$  和组  $Y$ ), 节点  $A$  同时处于组  $X$  和组  $Y$  中。下面我们介绍 Glax 的方案。

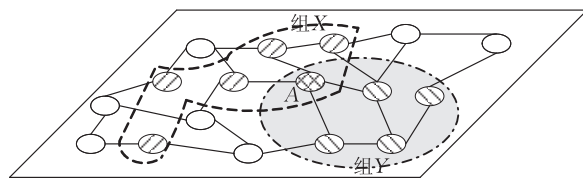


图 6 节点分组示例(虚线包括的范围表示一个组)

#### (1) 支持分组的路由表设计

在 DK 中, 每个节点的平均度数为  $2d$ 。为支持组的创建和组相关的路由策略, 在 Glax 中需要为节点路由表增加组相关的邻居信息。首先给出如下定义。

对任意两个 Kautz 串  $\alpha = u_1 u_2 \cdots u_m$  和  $\beta = v_1 v_2 \cdots v_n$ ,  $\alpha$  和  $\beta$  的 Kautz 匹配度<sup>[8]</sup>  $M(\alpha, \beta) = i$  是指  $i$  的最大取值 ( $0 \leq i \leq \min(m, n)$ ), 使得对任意的  $j$  ( $1 \leq j \leq i$ ) 有  $u_{m-i+j} = v_j$ 。令节点  $A$  的标识为  $\alpha = u_1 u_2 \cdots u_m$ , 节点  $B$  的标识为  $\beta = v_1 v_2 \cdots v_n$ , 则节点  $A$  和  $B$  的 Kautz 距离为  $D(A, B) = \min(m, n) - M(\alpha, \beta)$ 。令 Glax 中节点标识长度的最大值为  $n_{\max}$  位, 通过把每个节点的标识向后补足  $n_{\max}$  位(补 0), 那么所有节点的标识将形成一个类似于 Chord 环<sup>[10]</sup> 的稀疏 Kautz 环。与 Kautz 距离相对应, 我们把任意两个节点在环上的距离称为“环距离”, 简称为距离。

假设节点  $u = u_1 u_2 \cdots u_n$  属于组  $X_1, X_2, \cdots, X_m$ , 那么节点  $u$  的路由表中将包括一个 Kautz 邻居表和一个叶表。Kautz 邻居表共有  $h$  层, 第  $i$  层 ( $0 < i \leq h$ ) 的邻居为所有满足条件  $D(u, v) = i$  的节点  $v$ , 并且记录节点  $v$  属于哪个(些)组。在节点  $u$  的叶表中, 对每个组  $X_i$  维护一个包含  $2f$  个叶节点的叶集, 其中包括  $f$  个与  $u$  顺时针距离最近的节点以及  $f$  个与  $u$  逆时针距离最近的节点, 称为组  $X_i$  的叶集。

#### (2) 灵活路由

基于上述路由表, Glax 可以容易地实现各种不同的路由策略。

PC 路由。假设一个组  $X$  内的 PC 路由消息  $K = k_1 k_2 \cdots k_n$  到达一个组  $X$  节点  $u = u_1 u_2 \cdots u_n$ 。节点  $u$  首先检查  $K$  是否在组  $X$  叶集内, 如果是, 则把消息转发至距  $K$  最近的叶节点。否则, 将检查在邻居表中是否有组  $X$  节点, 如果有, 则把消息转发至与目的节点的 Kautz 距离最近的节点。否则, 将在组  $X$  叶集中选择与目的节点 Kautz 距离最近的节点(如果有多个节点具有相同的 Kautz 距离, 则选择距离目的节点最近的节点), 并把消息转发给该节点。

DS 路由。给定组  $X$  和关键字  $K$ , DS 路由可以通过如下步骤实现: (i) 路由至组  $X$  中任意一点  $u$ ;

(ii) 从点  $u$  开始把消息路由至组  $X$  内负责关键字  $K$  的唯一一节点, 显然上述第(ii)步可以通过一次 PC 路由完成, 而上述第(i)步可以通过 DHT 操作来实现<sup>[11]</sup>; 组  $X$  内的节点使用 DHT 的 put 操作(以组名为关键字)把自身信息发布到覆盖网; 路由时通过 DHT 的 get 操作得到组  $X$  中任意一点的路由信息。

Glaz 通过支持覆盖网中节点的分组, 能够较好地适应互联网资源的多样性特点, 能够使上层应用在性能、可靠性和安全性等多方面获益。

(1) 性能。DS 路由允许上层应用把负载(服务或数据)均衡地分配到一组满足应用需求的节点上, 从而提高应用的性能。例如, DS 路由支持计算需求较大的应用选择一组计算能力强的节点提供计算服务, 或者支持存储密集型的应用选择一组硬盘空间大的节点提供存储服务。而传统 DHT 覆盖网不具有控制负载分配的能力。并且由于同一管理域内的节点间通常具有较低的延迟, 若按照管理域关系进行分组, 可促使节点间的消息路由被尽可能限制在管理域内, 从而具有较高的路由性能。

(2) 可靠性。DS 路由允许上层应用把负载分配到一组相对稳定的节点上, 从而提高系统的可靠性。例如, 对覆盖网之上的存储应用, 若存在部分不稳定的节点, 那么上层应用通过把数据存储在一组具有较长会话时间(session time)的节点上, 可以显著提高系统的可靠性。另一方面, 如果节点的分组与下层的互联网域名结构相匹配, 即按照管理域从属关系进行分组, 那么一个管理域之外的节点失效显然不会影响到该域内的 PC 路由。

(3) 安全性。DS 路由通过把数据保存在一组信誉高的节点或一个可信的管理域, 可以显著提高系统的安全性。PC 路由能够保证一个管理域内的数据访问通信永远不会被该管理域外的节点所截获, 从而可进一步提高安全性。

### 3.4 覆盖网拓扑的优化方法

上一节阐述的支持节点分组实际是对覆盖网拓扑的一种优化。节点分组技术具有较大的灵活性, 可以根据应用需要, 在具有特定属性的节点间建立组, 支持目标指定路由(DS 路由)或路径受限路由(PC 路由)。但节点分组技术需要增加节点度数, 从而增大了覆盖网的维护开销。除节点分组外, 我们根据虚拟计算环境的需求和互联网资源的特点, 对覆盖网拓扑进行了一系列的其它优化, 主要包括物理网络匹配、节点度数自适应和震荡(churn)优化

等<sup>[12]</sup>。下面我们进行简要介绍。

#### (1) 物理网络匹配

覆盖网是架构在物理网络之上的逻辑网络, 覆盖网中节点间的逻辑拓扑可能会与底层的实际物理网络拓扑存在不匹配, 从而导致节点间消息路由的物理延迟较大。针对这一问题, 我们基于网络坐标技术, 研究设计了覆盖网的物理网络匹配优化方法。由于处于互联网同一 AS 域内节点间的通信延迟通常比处于不同 AS 域中节点间的通信延迟要小, 该方法首先根据节点所处的 AS 对节点进行分组, 然后通过网络坐标技术, 对组内的节点设置网络坐标。在覆盖网拓扑构建时, 该方法优先选择网络坐标接近的节点作为邻居, 并在运行时动态调整各节点在覆盖网拓扑中的位置, 以减小覆盖网拓扑连接的平均物理延迟。

#### (2) 节点度数自适应

互联网资源在带宽和处理能力等方面存在较大的多样性, 传统 DHT 覆盖网通常假设各节点的节点度数是同构的, 未能充分利用互联网资源能力的多样性特点。因此, 我们研究设计了节点度数自适应的覆盖网拓扑优化方法。在该方法中, 各节点根据自身的带宽能力, 设定自身的节点度数上限。新节点加入时, 首先形成传统的覆盖网拓扑(如 FissionE 和 DK), 然后新节点根据自身剩余节点度数, 根据特定的随机化规则, 不断尝试向覆盖网中的远程节点请求增加新的快捷连接请求, 直到剩余节点度数接近零。远程节点在接收到快捷连接请求后, 根据自身能力决定是否同意接收该连接, 并通知请求连接的节点。通过节点度数的自适应优化, 可有效利用高带宽节点的网络带宽, 降低覆盖网网络直径和路由延迟, 并增强覆盖网的容错能力。

#### (3) 震荡(churn)优化

在虚拟计算环境的一些应用中, 部分互联网资源节点由于自主性等原因频繁加入或退出, 节点震荡(churn)频率高, 动态性强, 会引起覆盖网拓扑维护开销的显著增加, 甚至造成覆盖网拓扑的不稳定和分割。为此, 我们研究设计了基于虚拟节点的震荡优化方法, 以减小覆盖网中的高频震荡节点的影响。在该方法中, 覆盖网中的节点是逻辑的虚拟节点, 在物理上可能映射到多个物理节点。每个物理节点加入或退出后, 其相关的虚拟节点只要仍有其它物理节点相对应, 虚拟节点间的逻辑拓扑可不变动。每个物理节点被分配一个终身性的标识(PID), 其震荡发生一定时间间隔后, 邻居节点才进行拓扑维

护处理. 在虚拟节点构成上, 根据物理节点的历史在线时间状况, 设计算法预测其可能的震荡频率, 将同一虚拟节点以较大概率同时映射到较稳定节点和高频震荡的节点.

4 基于覆盖网的资源搜索技术

为支持资源的有效聚合, 虚拟计算环境需根据应用的各种需求, 高效搜索到所需的资源. 下面将首先介绍基于覆盖网的区间搜索技术, 然后对其它复杂搜索技术进行简要讨论.

4.1 区间搜索技术

区间搜索, 即搜索属性值处于某一连续区间(或空间)内的资源对象, 是虚拟计算环境中一种重要的资源搜索方式. 例如, 搜索符合条件“ $100\text{GB} \leq \text{Storage} \leq 200\text{GB}$ ”的计算资源; 搜索满足条件“ $35 \leq \text{Age} \leq 40$  且  $5000 \leq \text{Salary} \leq 10000$ ”的数据资源对象等都是区间搜索.

为支持资源区间搜索, 我们提出了基于覆盖网的资源区间搜索技术 Armada<sup>[13]</sup>. Armada 架构于 FissionE 覆盖网之上, 通过设计保序命名机制和高效率的区间搜索处理算法, 可有效支持单属性和多属性区间搜索. 限于篇幅, 下面概述 Armada 中的单

属性区间搜索技术.

(1) 单属性保序命名机制

在 FissionE 覆盖网中, 资源对象会根据其 ObjectID, 通过覆盖网路由发布到节点标识是 ObjectID 前缀的覆盖网节点上. 例如, 在图 4 所示的 FissionE 覆盖网中, 所有 ObjectID 为 102\* 的资源对象都会发布到节点 102 上. 资源对象的发布, 可根据需要发布资源对象本身或其元信息. 根据覆盖网的这一特性, Armada 设计了保序命名机制, 其目的是使得属性值接近的资源对象(Object)能够获得命名空间中邻近的资源对象标识(ObjectID), 从而被发布到覆盖网中相同或相关的节点上, 以便于支持资源区间搜索.

设资源对象的属性值空间为  $[L, H]$ , 我们首先构造资源对象属性值的分区树  $P(2, k)$ . 图 7 给出了分区树  $P(2, 4)$ . 分区树的所有叶节点的标识正好与所有基底为 2、长度为  $k$  的 Kautz 串一一对应. 我们将资源对象的属性值空间  $[L, H]$  按照如下方式划分到分区树上: 根节点表示整个属性值空间  $[L, H]$ , 分区树中每个节点都均分其父节点表示的属性值子区间. 例如, 在图 7 中, 根节点表示整个属性值区间  $[0, 3]$ , 节点 A, B, C 是根节点的子节点, 分别表示属性值子区间  $[0, 1]$ ,  $(1, 2]$ ,  $(2, 3]$ .

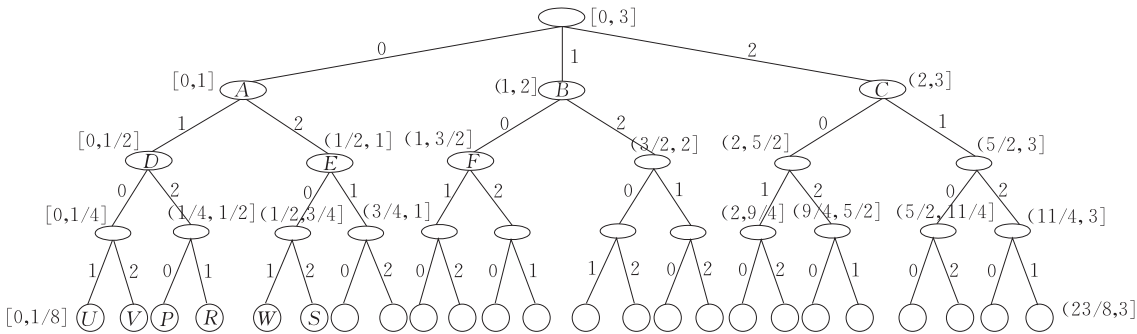


图 7 属性值区间  $[0, 3]$  的分区树  $P(2, 4)$

基于分区树  $P(2, k)$ , 可以设计单属性保序命名算法 *Single\_hash*: 设资源对象  $O$  的属性值为  $c (c \in [L, H])$ ,  $c$  会处于分区树某叶节点所表示的子区间中; 设该叶节点的标识为 Kautz 串为  $S$ , 则  $Single\_hash(c) = S$ , 即  $O$  的 ObjectID 被设置为  $S$ . 例如, 在图 7 所示的分区树中, 若资源  $O$  的属性值为 0.6, 由于 0.6 处于分区树叶节点 0201 (即叶节点 W) 表示的子区间中, 因此资源  $O$  的 ObjectID 被设置为 0201. 我们证明了 *Single\_hash* 算法具有区间保序性质.

(2) 单属性区间搜索处理算法

设节点  $P$  需搜索属性值处于区间  $[a, b]$  中的资源对象, 令  $LowT = Single\_hash(a)$ ,  $HighT = Single\_hash(b)$ . 由于 *Single\_hash* 算法具有区间保序特性, 所有属性值在区间  $[a, b]$  中的资源对象的 ObjectID 都处于从  $LowT$  到  $HighT$  的 Kautz 区间中, 因此所有属性值在区间  $[a, b]$  中的资源对象都被发布到覆盖网中负责该 Kautz 区间的所有节点上.

我们基于 FissionE 覆盖网拓扑, 设计了前向路由树, 前向路由树的每一层中相邻节点负责的 Kautz 区间都是相邻的. 前向路由树实现了区间搜

索路径与底层覆盖网拓扑的高效匹配,能够有效支持对覆盖网中负责任一属性值子区间的全部目标节点的高效并发搜索。

分析和实验表明, Armada 的区间搜索延迟是有界的(delay-bounded),即无论单属性区间搜索或多属性区间搜索, Armada 的平均搜索延迟都小于  $\log_2 N$ , 最大搜索延迟小于  $2\log_2 N$ , 并达到常量度数 DHT 覆盖网区间搜索技术搜索延迟的下界  $O(\log N)$ . Armada 的搜索消息开销少, 其单属性区间搜索的平均消息开销约为  $\log_2 N + 2n - 2$  ( $n$  为返回搜索结果的节点数目), 接近常量度数覆盖网区间搜索技术消息开销的下界  $O(\log N) + n - 1$ .

## 4.2 其它搜索技术

除区间搜索外, 虚拟计算环境还提供了基于覆盖网的  $K$  邻近搜索、聚合搜索、模糊搜索和多关键字搜索等多种复杂搜索能力, 以满足多种应用的需求。下面对前三者进行简要讨论。

### (1) $K$ 邻近搜索( $K$ -NN)技术

$K$  邻近搜索是指在覆盖网中搜索与给定属性值最接近的  $K$  个资源对象。以上述区间搜索技术中的维序命名机制为基础, 虚拟计算环境中设计采用了基于回溯的  $K$  邻近搜索算法<sup>[14]</sup>。该算法的基本思路是首先定位到负责给定属性值的节点, 接着以该节点为中心, 沿搜索路径回溯到前向路由树的第  $\lceil \log_2 K \rceil$  层, 然后沿前向路由树向前广播, 并根据需要进行微调回溯, 直到最终搜索到合适的  $K$  个资源对象。

### (2) 聚合搜索技术

聚合搜索<sup>[15]</sup>是指在覆盖网中对一组资源对象的某个(或多个)属性的统计信息(如 Count、Sum、Max 和 Average 等)的搜索。基于 FissionE 覆盖网拓扑, 我们设计了覆盖网的动态聚合树, 并由聚合树各层的中间节点依次将本节点的信息以及从子节点收到的聚合信息聚合后发送给其父节点, 有效实现了资源的聚合搜索。但在现有的聚合搜索技术中, 不同聚合搜索之间的优化较少, 每个聚合搜索是分别独立地进行, 在覆盖网中同时进行多个连续的聚合搜索请求会产生大量的网络通信负载。因此, 未来将针对大量连续聚合搜索的场景, 通过研究多聚合搜索请求的合并和共享方法, 提高系统聚合搜索的整体性能。

### (3) 模糊搜索技术

模糊搜索是指搜索满足与给定关键字模糊匹配条件的资源对象。为支持模糊搜索, 在资源对象发布

时, 首先借鉴传统信息检索技术, 将资源对象的关键字进行变换, 使之变换成多个部分关键字, 然后按照多关键字属性对资源对象进行发布。在资源搜索时, 对搜索请求进行类似变换, 对由此产生的多关键字进行搜索, 并在搜索过程中对搜索请求进行合并和消减, 以减少搜索开销。但目前虚拟计算环境中设计采用的模糊搜索技术产生的消息开销和延迟较大, 仍需进行优化, 以在搜索准确性和搜索开销方面同时获得良好特性。

## 5 结束语

基于覆盖网技术的互联网资源动态组织和高效搜索, 是虚拟计算环境中支持资源按需聚合的关键技术之一。本文首先介绍了基于 Kautz 图的高效覆盖网拓扑构造方法, 进而给出了适用于任意正则图的通用覆盖网拓扑构造方法, 并提出了支持分组的覆盖网拓扑构造方法。在此基础上, 给出了基于覆盖网的高效区间搜索技术, 并对覆盖网拓扑优化方法以及  $K$  邻近搜索等复杂搜索技术展开了探讨。

未来工作将针对虚拟计算环境中资源聚合的需求, 进一步优化覆盖网拓扑构造方法, 提供基于覆盖网的其它复杂搜索能力(如相似搜索、Top-k 搜索<sup>[16]</sup>和 Skyline 搜索<sup>[17]</sup>等), 并在广域实验床上对系统进行全面的测试和完善。

## 参 考 文 献

- [1] Lu Xi-Cheng, Wang Huai-Min, Wang Ji. Internet-based Virtual Computing Environment (iVCE): Concepts and architecture. Science in China, Series F: Information Science, 2006, 49(6): 681-701  
(卢锡城, 王怀民, 王戟. 虚拟计算环境 iVCE: 概念与体系结构. 中国科学(E辑), 2006, 36(10): 1081-1099)
- [2] Stephanos Androutsellis-Theotokis, Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. ACM Computing Surveys, 2004, 36(4): 335-371
- [3] Lu Xi-Cheng. Research on the Mechanisms of On-Demand Aggregation and Autonomic Collaboration of Internet-Based Virtual Computing Environment. Development Report of Chinese Computer Science and Technology in 2007. Beijing: Tsinghua University Press, 2008(in Chinese)  
(卢锡城. 虚拟计算环境聚合与协同机理研究. 中国计算机科学技术发展报告 2007. 北京: 清华大学出版社, 2008)
- [4] Li Dong-Sheng, Lu Xi-Cheng, Wu Jie. FISSIONE: A scalable constant degree and low congestion DHT scheme based on Kautz graphs//Proceedings of the IEEE INFOCOM 2005. Miami, Florida, USA, 2005: 1677-1688



- [5] Panchapakesan G, Sengupta A. On a lightwave network topology using Kautz digraphs. *IEEE Transactions on Computers*, 1999, 48(10): 1131-1138
- [6] Zhang Yi-Ming, Liu Ling, Li Dong-Sheng, Lu Xi-Cheng. Distributed line graphs: A universal framework for building DHTs based on arbitrary constant-degree graphs//*Proceedings of the ICDCS 2008*. Beijing, China, 2008
- [7] Zhang Yi-Ming, Li Dong-Sheng, Lu Xi-Cheng. A universal maintenance mechanism for structured overlays. National University of Defense Technology, Changsha: Technical Report PDL-08-01-02, 2008
- [8] Fiol M A, Llado A S. The partial line digraph technique in the design of large interconnection networks. *IEEE Transactions on Computers*, 1992, C-41(7): 848-857
- [9] Lu Xi-Cheng, Zhang Yi-Ming, Li Dong-Sheng. Designing DHT overlay with flexible groups. National University of Defense Technology, Changsha: Technical Report PDL-08-01-16, 2008
- [10] Stoica I, Morris R, Karger D et al. Chord: A scalable peer-to-peer lookup service for Internet applications//*Proceedings of the ACM SIGCOMM2001*. New York, 2001: 160-177
- [11] Yalagandula P, Dahlin M. Administrative autonomy in structured overlays//*Proceedings of IPTPS 2006*. LNCS. CA, USA, 2006
- [12] Li Dong-Sheng, Lu Xi-Cheng. Topology-aware optimization in structured overlays. National University of Defense Technology, Changsha: Technical Report PDL-07-12-18, 2008
- [13] Li Dong-Sheng, Cao Jian-Nong, Lu Xi-Cheng et al. Efficient range query processing in peer-to-peer systems. *IEEE Transactions on Knowledge and Data Engineering*(to appear, available at IEEE Computer Society Digital Library, <http://doi.ieeecomputersociety.org/10/1109/TKDE.2008.99>)
- [14] Li Dong-Sheng, Lu Xi-Cheng. Efficient  $K$ -nearest neighbor search in DHT-based overlays. National University of Defense Technology, Changsha: Technical Report PDL-08-03-24, 2008
- [15] Yalagandula P, Dahlin M. A scalable distributed information management system//*Proceedings of the ACM SIGCOMM 2004*. Portland, USA, 2004
- [16] Akrivi Vlachou, Christos Doukeridis, Kjetil Nørvg, Michalis Vazirgiannis. On efficient Top-k query processing in highly distributed environments//*Proceedings of the Sigmod 2008*. Vancouver, BC, Canada, 2008
- [17] Wang Shi-Yuan, Ooi Beng Chin, Tung Anthony K H, Xu Li-Zhen. Efficient Skyline query processing on peer-to-peer networks//*Proceedings of the ICDE 2007*. Istanbul, Turkey, 2007
- [18] Bridges W G, Toueg S. On the impossibility of directed Moore graphs. *Journal of Combinatorial Theory, Series B*, 1980, 29: 330-341
- [19] Zhang Yi-Ming, Li Dong-Sheng, Lu Xi-Cheng. Scalable distributed resource information service for Internet-based virtual computing environment. *Journal of Software*, 2007, 18(8): 1933-1942(in Chinese)  
(张一鸣, 李东升, 卢锡城. SDIRIS: 虚拟计算环境中可扩展分布式资源信息服务. *软件学报*, 2007, 18(8): 1933-1942)



**LU Xi-Cheng**, born in 1946, professor, Ph. D. supervisor, member of the Chinese Academy of Engineering. His main research interests include computer network, parallel and distributed processing, etc.

**LI Dong-Sheng**, born in 1978, Associate professor. His research interests include computer network and distributed computing, etc.

## Background

With the fast development and wide application of computing and network technologies, Internet has become an important information infrastructure and pervasive computing platform in modern society. Different from the traditional resources, resources over Internet have such natural characteristics as growth, autonomy and diversity, which have brought great challenges to efficient sharing and comprehensive utilization of these resources. The Internet-based Virtual Computing Environment (iVCE) is proposed to virtualize Internet resources and abstract them as autonomic entities, and

then provide harmonious, trustable and transparent services for end-users and applications by the way of on-demand aggregation and autonomic collaboration. Distributed Hash Table (DHT) based overlay technologies are used as an important approach to aggregate resources on-demand in iVCE. And this paper gives an overview of research advances on overlay technologies in iVCE. This work is supported in part by the National Basic Research Program of China (973 Program) under Grant "Research of aggregation and collaboration mechanisms in the virtual computing environment".