中国科学: 信息科学 2011 年 第 41 卷 第 9 期: 1037–1053
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE CHINA PRESS

评 述

虚拟计算环境中的 DHT 拓扑构建技术研究综述

张一鸣[①②*], 卢锡城[①②], 李东升[①②]

① 并行与分布处理国家重点实验室 (PDL), 长沙 410073
② 国防科技大学计算机学院, 长沙 410073
* 通信作者. E-mail: ymzhang@nudt.edu.cn

收稿日期: 2010–06–01; 接受日期: 2010–08–25
国家重点基础研究发展计划 (批准号: 2011CB302601)、国家自然科学基金 (批准号: 60903205)、博士点基金 (批准号:20094307110008) 和全国优秀博士学位论文作者专项资金 (批准号: 200953) 资助项目

摘要 基于互联网的虚拟计算环境 (iVCE) 是一种新型网络计算平台. 互联网资源的成长性、自治性和多样性等自然特性给 iVCE 中的资源共享带来巨大的挑战. DHT 覆盖网 (简称 DHT) 具有可扩展、延迟低、可靠性高等优点, 是 iVCE 实现资源有效共享的重要途径之一. 拓扑构建是 DHT 的基础性关键技术, 实现了 DHT 的动态维护与消息路由等基本功能. 本文首先概述传统 DHT 的拓扑构建技术, 主要包括各种典型 DHT 的动态维护机制与消息路由算法、支持复杂查询的 DHT 索引构建技术, 以及支持管理域匹配的 DHT 分组构建技术等; 进而针对互联网资源的特点, 综述在 iVCE 中 DHT 拓扑构建技术的最新研究进展. 本文在最后对 DHT 拓扑构建技术的未来发展方向进行探讨.

关键词 虚拟计算环境 DHT 覆盖网 拓扑构建 分布式索引 灵活路由

1 概述

互联网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有效共享和综合利用提出迫切需求. 在这种背景下, 面向互联网的虚拟计算环境 [1](internet-based virtual computing environments, iVCE) 应运而生. iVCE 建立在开放的互联网基础设施之上, 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不断地成长变化, 资源管理的范围难以确定. 在动态变化的互联网环境下, 如何支持资源的有效共享, 是 iVCE 面临的重要挑战性问题. 由于分布式 Hash 表 (DHT, distributed Hash table) 技术 [2−7] 具有可扩展、自适应、自组织等良好特性, 通过 DHT 覆盖网 (简称 DHT) 组织和管理大量动态的互联网资源并形成相对稳定的资源组织视图, 成为 iVCE 实现资源按需聚合和有效共享的重要途径. 如图 1 所示, iVCE 中的 DHT 覆盖网主要包括动态维护与消息路由、资源查询、物理网络适配、负载均衡和安全 [8,9] 等功能.

在上述功能中, 动态维护与消息路由是 DHT 的核心关键技术. 其中动态维护是指在允许节点自主加入/退出的情况下, 采用分布式算法在节点之间建立逻辑上的连接关系, 形成一定的覆盖网拓扑.

引用格式: 张一鸣, 卢锡城, 李东升. 虚拟计算环境中的 DHT 拓扑构建技术研究综述. 中国科学: 信息科学, 2011, 41: 1037–1053

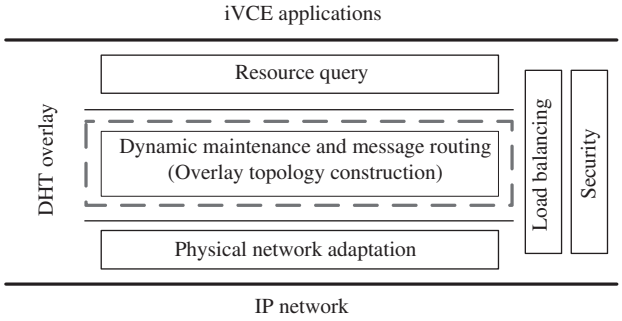

图 1　虚拟计算环境中的 DHT 覆盖网

Figure 1　DHT overlays in iVCE

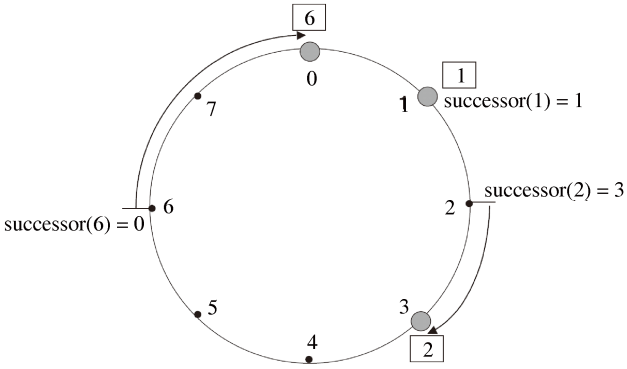

图 2　Chord 拓扑图示例

Figure 2　Example of Chord topology

消息路由是指根据消息的目标字符串, 把消息从源节点路由至目标节点. 由于动态维护和消息路由是紧密相关的, 因此通常被看作一个整体并统称为 DHT 拓扑构建技术 [10].

本文剩余部分的组织如下. 第 2 节介绍了各种典型 DHT 的动态维护机制与消息路由算法, 第 3 节介绍了支持复杂查询的 DHT 索引构建技术, 第 4 节介绍了支持管理域匹配和灵活路由的 DHT 分组构建技术, 第 5 节探讨了 DHT 拓扑构建技术的未来发展方向, 第 6 节总结全文.

2　基本拓扑构建技术

2.1　传统 DHT

(1) 基于环的 DHT.

Chord[11] 采用环作为其静态拓扑图, 所有节点根据标识的大小构成一个环形的拓扑结构, 如图 2[11] 所示. Chord 中每个节点都维护了一个路由表, 节点 n 的路由表的第 i 项是值 $n+2^{i-1}$ 在 Chord 环上的后继节点. 给定目标字符串 K, 在消息路由的过程中, 每个中间节点都会把消息转发到路由表中距离 K 最近且不超过 K 的邻居. Chord 的节点度数为 $O(\log_2 N)$, 路由延迟是 $O(\log_2 N)$, 平均路由延迟为 $1/2\log_2 N$, 动态维护开销为 $O(\log_2^2 N)$.

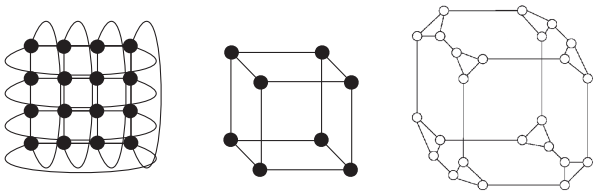

图 3 花环, 立方体和立方体连接环

Figure 3 Torus, hypercube, and CCC

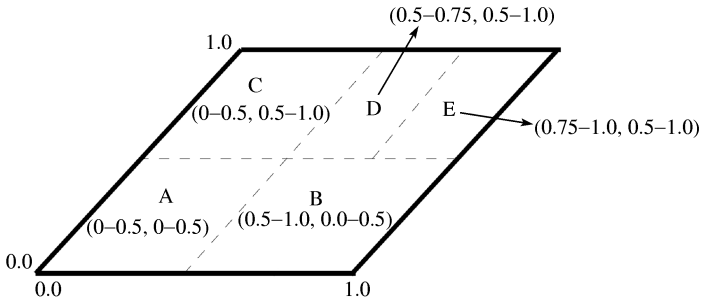

图 4 CAN 拓扑图示例

Figure 4 Example of CAN topology

为了提高路由性能, 有些研究提出在 Chord 的基础上增加额外的连接和路由信息, 主要包括 Kelips[12] 和 Accordion[13] 等.

Kelips[12] 将覆盖网中的节点分成 k 个组, 每个节点通过 Hash 函数分配到某一个组中, 资源也通过一致性 Hash 算法发布到某一个组中, 从而各组中的节点和资源能够均匀分布. Kelips 的路由延迟为 $O(1)$, 节点度数和维护开销为 $O(N^{1/2})$. Accordion[13] 提出在给定带宽预算 (bandwidth budget) 下对节点度数进行动态调整, 以获得较好的路由性能. 在 Accordion 中, 上层应用首先指定一个总的带宽预算, 进而在该预算下每个节点尽可能多地增加邻居节点. Accordion 在不同的带宽预算下具有不同的路由性能: 在节点度数为 $O(\log_2 N)$ 时平均路由延迟为 $O(\log_2 N)$; 在节点度数为 $O(N)$ 时平均路由延迟为 $O(1)$.

(2) 基于多维花环或立方体的 DHT.

一个 d 维 k 元花环由 $N = k^d$ 个点组成, 如图 3 左图所示, 每个点由它的 d 维坐标向量来标识. d 维二元花环是一类重要的拓扑图, 通常被称为 d 维立方体, 如图 3 中图所示. 通过把 d 维立方体各点替换成包括 d 个点的环, 将得到 d 维立方体连接环 [14](cube-connected-cycle, CCC), 如图 3 右图所示, CCC 拓扑图可以看作是立方体的扩展.

CAN[15](content addressable network) 采用多维花环作为其静态拓扑图. CAN 的基本思想是构造一个虚拟的 d 维 Descartes 坐标空间, 覆盖网中各节点分别负责虚拟 d 维坐标空间中的一块区域. 每个资源映射到 d 维区域中的一点, 并且发布到负责该区域的节点上, 每个节点有 $O(d)$ 个邻居. 如图 4[15] 所示的 CAN 采用了二维 Descartes 坐标空间, 整个虚拟坐标空间由 5 个节点负责. CAN 的节点度数为 $O(d)$, 路由延迟为 $O(dN^{1/d})$, 动态维护开销为 $O(\log_d N)$.

Cycloid[14] 采用 CCC 图作为其静态拓扑, 如图 3 右图所示. Cycloid 模拟 CCC 图的路由算法,

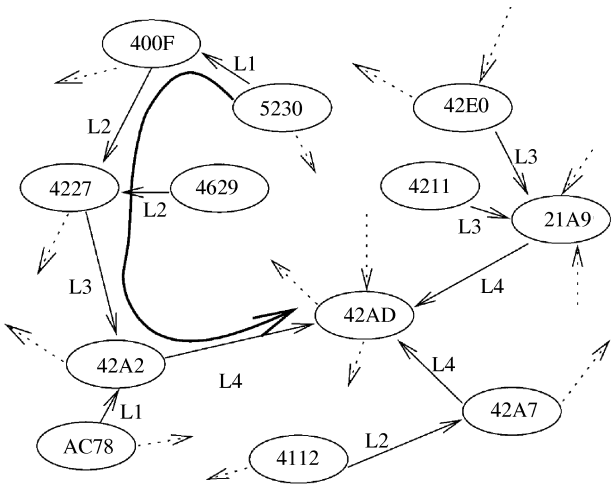

图 5　Plaxton 图示例

Figure 5　Example of a Plaxton graph

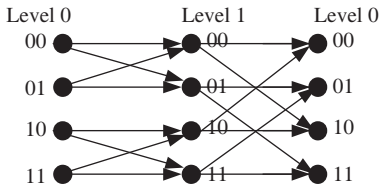

图 6　蝶网示例

Figure 6　Example of a butterfly graph

采用逐位匹配的方式进行路由. Cycloid 的节点度数为 $O(d)$, 路由延迟为 $O(\log_d N)$, 动态维护开销为 $O(\log_d N)$.

(3) 基于 Plaxton 图的 DHT.

在 Plaxton 图 [16] 中, 每个点的标识都是一个长度固定的位串. 各点间根据标识前缀匹配 (或后缀匹配) 的方式进行连接. 图 5[16] 是一个 Plaxton 图的例子.

Tapestry[17] 基于 Plaxton 图进行构建. Tapestry 节点只需维护邻居节点的路由信息. 在 Tapestry 中, 节点和资源的标识都是 160 位的二进制值, 使用基为 b 的字符串表示 (如 $b = 16$ 时为 16 进制 40 位字符串). 节点路由表中包括一个表项集和一个叶集. Tapestry 的节点度数为 $O(b \log_b N)$, 路由延迟为 $O(\log_b N)$, 动态维护开销为 $O(b \log_b N)$. Tapestry 的后续版本 Chimera[18] 采用了前缀匹配的方式实现动态维护和消息路由, 但是其本质是相同的.

Pastry[19] 与 Tapestry 类似, 也是基于 Plaxton 图进行构建. Pastry 采用基于前缀匹配的方式进行动态维护和路由. 每个 Pastry 节点维护了一个路由表, 一个邻居集和一个叶集. Pastry 的节点度数为 $O(b \log_b N)$, 路由延迟为 $O(\log_b N)$, 动态维护开销为 $O(b \log_b N)$. 在后续研究中人们提出了 Pastry 的各种改进版本, 如 MSPastry[20], Bamboo[21], OpenDHT[22], Kademlia[23] 等.

(4) 基于蝶网的 DHT.

一个 (k, r) 蝶网 [24] 包含 $n = kr^k$ 个点, 其中 k 和 r 分别被称为蝶网的直径和基. 点标识的形式为 $(x_0 x_1 \cdots x_{k-1}; i)$, 其中 i 表示点所处的层数, $0 \leqslant i \leqslant k-1$, $0 \leqslant x_0, x_1, \ldots, x_{k-1} \leqslant r-1$. 图 6 是一

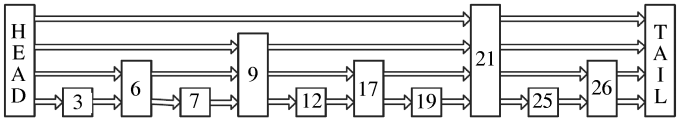

图 7 跳表示例

Figure 7 Example of a skip list

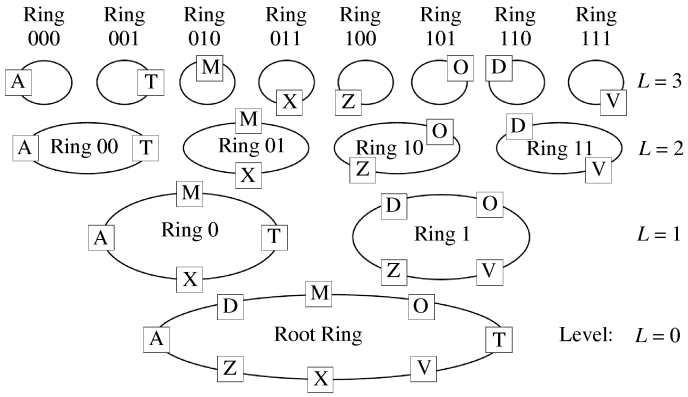

图 8 多层 SkipNet 结构示例

Figure 8 Example of a SkipNet structure

个 $(2,2)$ 蝶网的示例.

Viceroy[25] 基于蝶网进行构建. Viceroy 将所有节点组织成一个多层的环, 每个节点处于某一层中, 同一层的节点构成一个双向链表, 同时每个节点有 2 条到下一层中两个随机选取的节点的连接, 1 条到上一层节点的连接. Viceroy 的节点度数为 $O(d)$, 路由延迟和维护开销在大概率情况下为 $O(\log_d N)$, 但在某些情况下可能会达到 $O(\log_2 N)$.

Ulysses[24] 也是基于蝶网的 DHT. Ulysses 中节点的标识空间是一个 k 维空间, 每个节点负责 k 维空间中的一块区域. 在大概率情况下, Ulysses 的节点度数为 $O(\log_k N)$, 路由延迟和维护开销为 $O(\log_k N/\log_k \log_k N)$.

(5) 基于跳表的 DHT.

在跳表 [26] 中, 所有节点按照从小到大的顺序排序并构成一个有向链表. 图 7[27] 给出了一个跳表的示例.

SkipNet[27] 基于跳表进行构建. SkipNet 采用双重地址空间: 名字空间和数字 ID 空间. 节点名称和资源名称映射到名字空间, 而它们的 Hash 值则映射到数字 ID 空间, 如图 8[27] 所示. SkipNet 的节点度数, 路由延迟和动态维护开销均为 $O(\log_2 N)$.

(6) 基于 de Bruijn 图的 DHT.

de Bruijn 图 $B(d,D)$[28] 是一个有向图, 其中各点的标识是基为 d, 长度为 D 的字符串. 每个点 $u = u_1u_2\cdots u_D$ 有 d 条出边: 对任意 $\alpha \in \{0,1,\ldots,d-1\}$, 点 u 有一条到点 $v = u_2u_3\cdots u_D\alpha$ 的出边. 图 9 给出了一个 de Bruijn 图的示例 $B(2,2)$.

Koorde[29] 基于 de Bruijn 图, 采用类似于 Chord 的动态维护等机制维护 DHT 拓扑. Koorde 的节点邻居关系模拟了 de Bruijn 图. Koorde 的节点度数为 $O(1)$, 路由延迟为 $O(\log_2 N)$; 经过扩展后其节

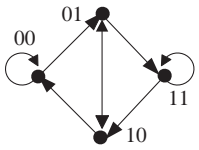

图 9　de Bruijn 图示例

Figure 9　Example of a de Bruijn graph

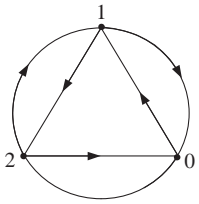

图 10　Kautz 图示例

Figure 10　Example of a Kautz graph

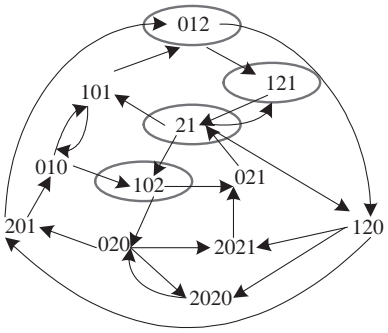

图 11　FissionE 拓扑与路由示例

Figure 11　Example of FissionE topology and routing

点度数为 $O(\log_2 N)$, 路由延迟则减小为 $O(\log_2 N/\log_2\log_2 N)$. D2B[30] 基于 de Bruijn 图进行构建. 在大概率情况下, D2B 的节点度数和路由延迟均为 $O(\log_2 N)$. 其他基于 de Bruijn 图的 DHT 包括 ODRI[31] 和 Broose[32] 等.

2.2　iVCE 中的 DHT 拓扑构建技术

上述研究均未能在节点度数和路由延迟之间取得令人满意的折中. 本小节将介绍 iVCE 中的两种基本拓扑构建技术.

(1) FissionE.

在 Kautz 图 $K(b,k)$[33] 中, 各节点的标识都是基底为 b, 长度为 k 的 Kautz 串, 即串 $\xi = a_1a_2\cdots a_k$, 其中 $a_i \in \{0,1,2,\ldots,b\}$, $1 \leqslant i \leqslant k$ 且 $a_i \neq a_{i+1}$, $1 \leqslant i \leqslant k-1$. Kautz 串的特点是相邻字符不相同. 在 Kautz 图 $K(b,k)$ 中, 节点出度和入度都为 b, 网络直径为 k. 对 $K(b,k)$ 中每个标识为 $u_1u_2\cdots u_k$ 的节点 u (记为 $u = u_1u_2\cdots u_k$), u 都有 b 条出边: 即对任意 $\alpha \in \{0,1,2,\ldots,b\}$ 且 $\alpha \neq u_k$, 节点 u 都有一条到节点 $v = u_2u_3\cdots u_k\alpha$ 的出边. 图 10 给出了一个 Kautz 图 $K(2,1)$ 的示例.

FissionE[34] 使用 Kautz 图 $K(2,D)$ 作为静态拓扑, 每个节点的标识都是一个基为 2 的 Kautz 串. 对 FissionE 中的任一节点 $u = u_1u_2\cdots u_d$, 其出边邻居为所有标识为 $v = u_2u_3\cdots u_Dq_1\cdots q_m$ $(0 \leqslant m \leqslant 2)$ 的节点, 如图 11[34] 所示. FissionE 的入度为 2, 平均出度为 2, 网络直径小于 $2\log_2 N$, 平均路由延迟小于 $\log_2 N$, 动态维护开销小于 $3\log_2 N$.

(2) DLG 变换.

为了简化面向 iVCE 应用需求的 DHT 设计, 我们在文献 [35] 中提出一种适用于任意 d 正则图 (所有点的出度和入度均为常数 d)[36] 的通用 DHT 拓扑构建技术: 分布式线图 (DLG, distributed line

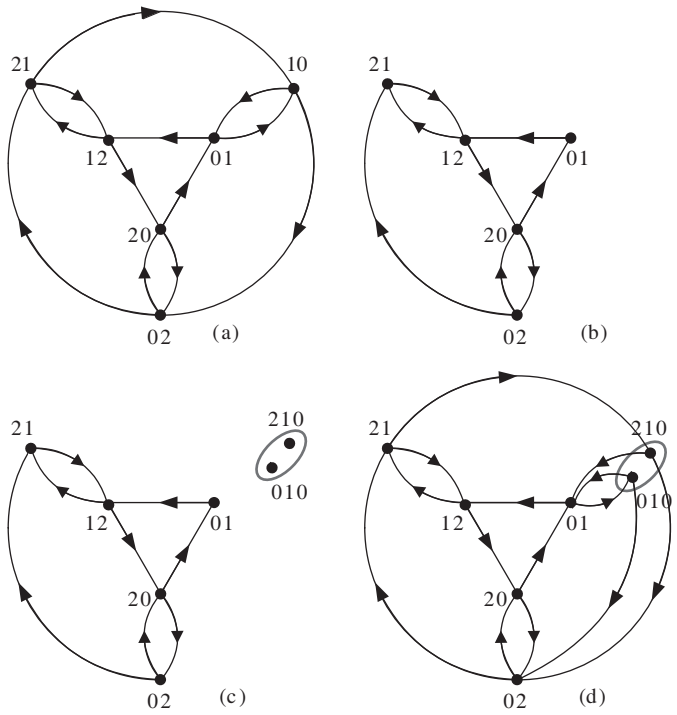

图 12 DLG 动态维护示例

Figure 12 Example of DLG dynamic maintenance

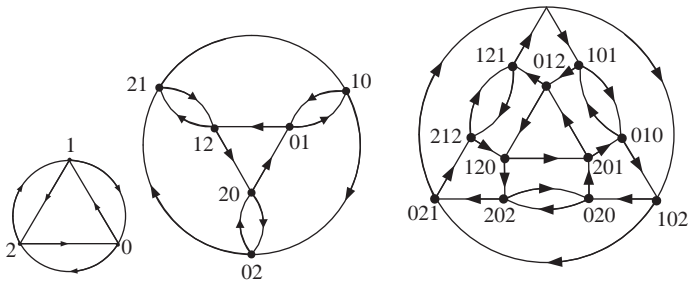

图 13 基于 Kautz 图构建的 DLG-Kautz (DK)

Figure 13 DLG-Kautz (DK) based on kautz graphs

graph) 变换.

DLG 变换的基本思想是 “边 – 点变换”, 即把原拓扑图中的边变为新拓扑图中的点. 例如, 假设当前 SKY 的拓扑图如图 12(a) 所示, 令 $|u|$ 表示节点 u 的标识长度. 当有新节点 p 加入时, 它首先寻找一个节点 u 满足对 u 的任意邻居 v 有 $|u| \leqslant |v|$. 假设在本例中选择 $u = 10$. 加入处理过程如下:

(i) 删除节点 u 以及 u 的所有入边和出边, 如图 12(b) 所示;

(ii) 原图中节点 u 的每一条入边 [21, 10] 和 [01, 10] 分别变为一个节点 210 (对应新节点 p) 和 010 (对应节点 u), 如图 12(c) 所示;

(iii) 为 u 和 p 生成新的入边和出边, 如图 12(d) 所示.

应用 DLG 变换技术可以容易地得到一系列 “DHT 族”, 我们把基于不同正则图 X, 应用 DLG 变换技术构建的 DHT 称为 DLG-X. 图 13–15 分别给出了基于 Kautz 图, de Bruijn 图和蝶网构建的

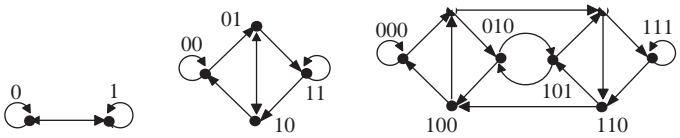

图 14　基于 de Bruijn 图构建的 DLG-de Bruijn (DdB)

Figure 14　DLG-de Bruijn (DdB) based on de Bruijn graphs

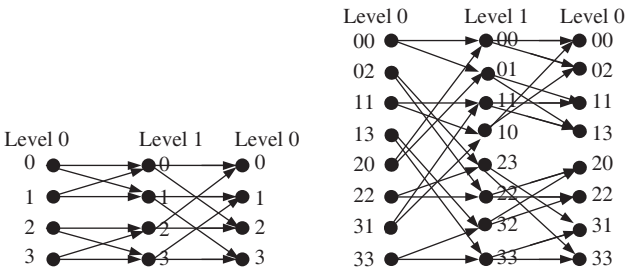

图 15　基于蝶网构建的 DLG-Butterfly (DBF)

Figure 15　DLG-Butterfly (DBF) based on butterfly graphs

DHT 拓扑图示例: DLG-Kautz (DK), DLG-de Bruijn (DdB) 和 DLG-Butterfly (DBF).

令 d, N_0, D_0 分别为初始正则图的基, 节点数和直径, N 为当前 DHT 覆盖网的节点数, 则应用 DLG 变换技术构建的 DHT 的节点出度为 d, 节点入度在 1 和 $2d$ 之间, 平均节点入度为 d, 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 $3d$ 个节点需要更新路由表.

3　支持复杂查询的 DHT 索引构建技术

3.1　支持区间查询的 DHT 索引构建技术

(1) 分层索引.

分层索引构建技术基于现有 DHT 实现区间查询, 无需改变下层的拓扑结构和消息路由算法.

文献 [37] 在 Chord 的基础上通过位置敏感的 Hash 算法 (locality sensitive Hashing, LSH) 来获得属性值区间的标识, 并基于 Chord 的消息路由构建分布式索引. LSH 算法返回的查询结果只能在一定概率下符合查询条件, 不能确定性地返回满足查询条件的所有资源.

Squid[38] 采用 Hilbert 空间填充曲线 (space-filling curve, SFC) 技术, 将资源的多个属性值映射到 Chord 环上的一点, 然后通过 Chord 方法进行资源发布. Squid 方法的每一步搜索都会引起 Chord 中的一次 DHT 路由, 因此其查询延迟和消息开销较大.

SkipNet[27] 采用双重地址空间, 并把节点名称和资源名称直接映射到名字空间. SkipNet 能够支持单属性区间查询. 在 SkipNet 的基础上, SCRAP[39] 采用 SFC 技术提供了多属性区间查询能力.

PHT[40] 提出一种类似于二叉树的前缀 Hash 树 (prefix hash tree) 索引结构, 进而支持多种复杂查询. 例如在进行区间查询时, 首先将多维区间转换成二进制字符串的集合, 然后在前缀 Hash 树中进行由根向下的搜索. 由于前缀 Hash 树中的每一步搜索都会引起一次 DHT 路由, 因此其查询延迟和消息开销都比较大.

(2) 定制索引.

PTree[41] 在多个分布的节点之间建立类似于 B+ 树的分布式索引结构, 进而提供了单属性区间查询能力, 其平均节点度数为 $O(d\log_d N)$, 平均区间查询延迟为 $O(\log_d N)$.

Mercury[42] 为每种资源属性建立一个虚拟 Hub 索引结构, 同一个 Hub 中的全部节点形成一个环形的覆盖网拓扑, 每个节点属于其中的某一个虚拟 Hub, 负责此虚拟 Hub 中一段连续的属性值. 在 Mercury 中, 节点度数为 $O(d)$, 区间查询延迟为 $O(\log_d^2 N/d)$.

SWORD[43] 在 PlanetLab 中提供支持区间查询的索引服务, 每个节点参与多个 DHT 网络, 每个 DHT 网络负责一个属性. MAAN[44] 与 SWORD 类似, 为每种属性在覆盖网中构建一个基于 Chord 的子 DHT.

其他定制索引构建技术包括 P-Ring[45], LHT[46], BATON 及其改进方法 [47,48] 等.

3.2 支持聚合查询的 DHT 索引构建技术

聚合查询 (aggregation query) 是指对一组节点某些属性的聚合信息 (如 count, sum, max, average 和 median 等) 的查询, 在数据管理、数据共享和缓存, 以及 DNS 解析等系统中有着广泛应用 [49].

分布式概略系统信息服务 [50](distributed approximative system information service, DASIS) 基于 Kademlia[23] 提出一种聚合树结构. 给定字符串 s, DASIS 支持对标识前缀为 s 的节点个数的查询.

Cone[51] 基于 Chord[11] 实现了 Max 聚合树. 每个节点记录了树中同层与其标识互补 (例如标识 000 的互补标识为 001) 的节点的 IP 地址. 两个标识互补的节点通过直接交互来确定它们所代表的子树的 Max 值, 并聚合到上层以其标识为前缀的节点. 在第 0 层的根节点将得到全局的 Max 值.

SDIMS[52] 基于 Pastry[19] 实现了聚合树并支持多种聚合查询功能. 在 SDIMS 中, 每个节点的本地信息以元组 ⟨ 属性类型, 属性名称, 属性值 ⟩ 的形式保存. 每个属性类型关联一种聚合函数. 每个 ⟨ 属性类型, 属性名 ⟩ 对应一个聚合树, 以 $k = \text{Hash}(\text{属性类型, 属性名})$ 为根, 按照 Pastry 的路由层次进行组织.

除了区间查询和聚合查询, 近年来研究者们还对基于 DHT 的其他类型的复杂查询进行了初步研究. 例如, 文献 [53] 提出了一种分布式 Top-k 查询方法 (对查询结果进行排序, 返回最优的 k 个结果), 文献 [54] 研究了 DHT 中 JOIN 查询 (类似于数据库中两个表的 JOIN 操作) 的实现方式, 文献 [55] 研究了使用类 SQL 语句表达 DHT 复杂查询的方法, 文献 [56,57] 研究了 DHT 中的 Skyline 查询等. 但是, 这一类研究相对较少, 查询性能也较低 [49].

3.3 iVCE 中的 DHT 索引构建技术

(1) 分区树.

我们在文献 [42] 中基于 FissionE 提出一种分区树索引结构, 使得属性值连续的资源能够获得相近的标识, 从而映射到相同或相关的节点上. 图 16 给出了分区树 $P(2,4)$ 的一个例子. 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.

基于分区树, 我们提出一种多属性区间查询方法 Armada, 无论查询区间的大小或属性值个数的多少, Armada 都能确保在一定的延迟内返回所有结果. Armada 的平均查询延迟小于 $\log_2 N$ (N 为节点数), 最大查询延迟小于 $2\log_2 N$, 多属性区间查询的平均消息开销约为 $\log_2 N + 4n - 4$ (n 为返回搜索结果的节点数目).

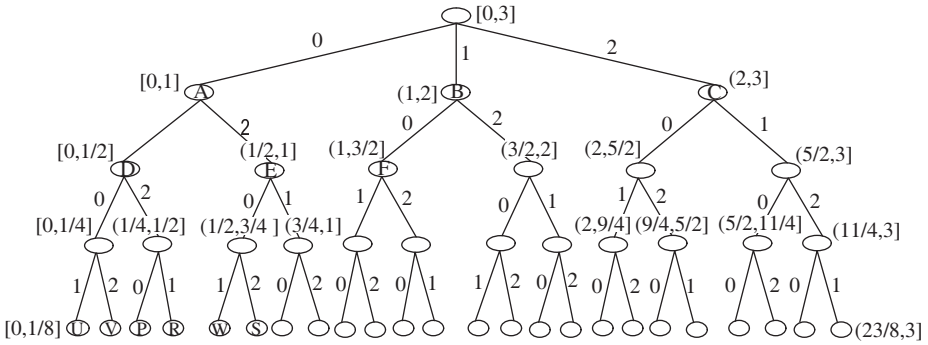

图 16　分区树示例

Figure 16　Example of a partition tree

(2) 平衡 Kautz 树.

在给定节点度数的前提下, Armada 是国际上查询延迟最低的区间查询方法. 但是, Armada 基于历史信息调整资源的分布, 从而在负载动态变化的情况下其动态负载平衡性能较差. 针对上述问题, 我们在文献 [58] 中基于 DK 的基础上提出一种高效的分布式索引构建技术: 平衡 Kautz 树 (BK 树).

BK 树通过 Z 曲线 [59] 实现了资源空间到节点空间的映射, 并基于 PHT 技术 [40] 设计了高效的资源信息索引结构. BK 树中的每个内部节点有 d 个子节点, d 被称为 BK 树的基. 资源 X 将映射到 BK 树的一个叶节点上, 该叶节点负责 X 所在的资源空间. 为保持负载平衡, 每个叶节点最多容纳 MAX 个资源. 例如, 当处于第 h 层的叶节点 A 所对应的资源数超过 MAX 时, A 将生成 d 个子叶节点, 并把所有资源分配给各子叶节点.

BK 树可以支持区间查询, 聚合查询和 Skyline 查询等多种复杂查询. 例如, 我们基于 BK 树提出一种延迟有界的区间查询方法 ERQ[58], 其最大查询延迟小于 $\log_d N(2\log_d \log_d N + 1)$, 平均消息开销约为 $O(\log_d N \log_d \log_d N + \beta N)$, 其中 β 为查询区间和资源空间大小的比值.

4　支持灵活路由的 DHT 分组构建技术

4.1　支持管理域匹配的 DHT 分组构建技术

DHT 下层的 IP 网络是典型的层次化 (hierarchical) 结构, 并通过分级的管理域 (administrative domain) 进行组织. 为实现 DHT 与下层 IP 网络的匹配, 研究者们提出在 DHT 中模拟分级域结构, 每个节点均属于最底层的一个管理域, 进而支持满足路径局部性 (path locality) 和路径收敛性 (path convergence)[60] 的管理域路由 (domain-aware routing), 如图 17 所示.

在 SkipNet[27] 的路由过程中, 消息从底向上依次在每一层的某个环中传递, 从而所有中间节点都在起点和终点的最小公共域内. 因此, SkipNet 中的路由满足路径局部性 (但是不满足路径收敛性).

Canon[61] 基于 Chord[11] 实现了管理域结构. 在 Canon 中, 最底层域内节点的连接关系与 Chord 完全相同. 在路由过程中, Canon 总是使用逐渐增大的域所对应的环连接, 因此 Canon 中的消息路由满足路径局部性; 给定目标消息 K, 每个域中距离 K 最近的后继节点是 K 在该域的收敛点.

ADHT[60] 基于 Pastry[19] 实现了管理域结构. 在 ADHT 中, 每个节点针对整个覆盖网维护了一个与 Pastry 类似的路由表, 以及针对所参加的每一级管理域维护了 $2f$ 个叶邻居. 由于大部分路由是基于叶节点进行的, 因此 ADHT 的路由效率较低. ADHT 使用 DHT 的 put/get 接口实现管理域的查

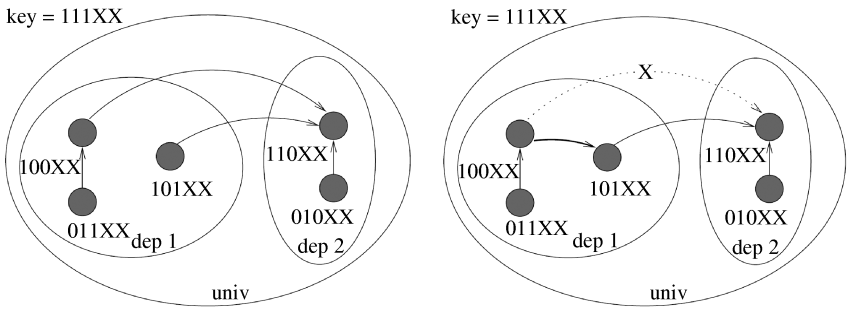

图 17 路径收敛性示例

Figure 17 Example of path convergence

找: 任意域 X 的节点把地址发布 (put) 到负责 Hash(X) 的节点, 寻找域 X 的节点则从负责 Hash(X) 的节点查找 (get) 域 X 中任意节点的地址.

4.2 任意组的创建

文献 [62] 对 Chord[11] 进行改进, 提出一种支持任意组创建的 DHT: 缩减 Chord (diminished chord 或 D-Chord). 给定组 X, 令 $\langle a,b\rangle$ 表示地址 $(a_0 + b/2^a) \mod 1$, 其中 $a_0 = \text{Hash}(X)$, $1 \leqslant a$ 且 $1 \leqslant b \leqslant 2^a$. 需要说明的是, 由于 D-Chord 中的地址范围为 $[0,1)$, 这里的 "mod 1" 表示取小数, 例如 $1.2 \mod 1 = 0.2$, $2.5 \mod 1 = 0.5$. 假设在 DHT 中有 N 个节点, 节点标识长度为 n, D-Chord 通过使用地址 $\langle a,b\rangle$ 的前驱节点来模拟 $\langle a,b\rangle$, 得到一个具有 2^n 个虚节点的稠密 (fully populated) 环. D-Chord 进而构建一个组 X 的成员树, 其中点 $\langle a,b\rangle$ 是点 $\langle a+1, 2b-1\rangle$ 和点 $\langle a+1, 2b\rangle$ 的父节点, 并且树中每个点 u 保存了 u 的右子树中距离 u 最近的属于组 X 的节点的地址. D-Chord 支持如下类型的路由: 给定目标字符串 K 和组 X, 路由至 Chord 环上 K 的第一个属于组 X 的后继节点. 具体过程如下: (i) 首先路由至 K 的后继节点 s; (ii) 然后从 s 开始沿着成员树向上寻找, 直到找到第一个 s 的属于组 X 的后继节点, 该节点即为 K 的第一个属于组 X 的后继节点.

4.3 iVCE 中的 DHT 分组构建技术

针对 iVCE 资源的多样性特点, 我们在上述研究的基础上提出支持多种灵活路由的 DHT 分组构建技术, 下面将分别进行介绍.

(1) G-TAP.

我们在文献 [63] 中基于 Tapestry[17] 提出一种分组 DHT 构建技术: G-TAP (grouped tapestry). G-TAP 首先利用 Tapestry 表项集中各表项的多选择性为每个组嵌入一个子 DHT 结构 (sub-DHT), 使得各组能够被标识和分辨; 然后为每个组创建一个组成员信息汇聚 (group membership rendezvous, GMR) 树, 使得查找任意组的负载能够被均衡地分配到一组汇聚点. G-TAP 支持多种灵活的路由方式.

(i) 目标指定路由 (destination-specified routing, DS routing): 给定一个组 X 和目标字符串 K, DS 路由将把消息路由至组 X 中负责 K 的唯一节点, 如图 18 左图所示.

(ii) 路径受限路由 (path-constrained routing, PC routing): 给定目标字符串 K 并且假设消息产生于组 X 中的一个节点, PC 路由将把消息路由至组 X 中负责 K 的唯一节点, 如图 18 右图所示.

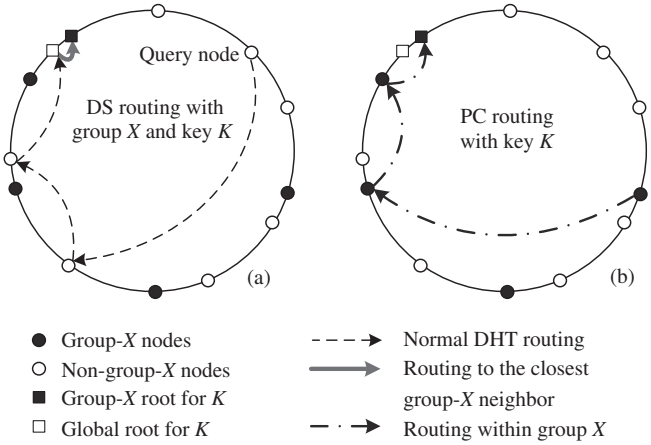

图 18　灵活路由示例

Figure 18　Example of flexible routing

(iii) 传统路由: 与传统结构化覆盖网中的消息路由相同, 没有指定目标节点所属的组或限制路由路径.

基于 DS/PC/传统路由, G-TAP 可以容易地实现任意组的创建与查找, 并且支持路由过程中的路径局部性和收敛性, 从而能够使上层应用在性能、可靠性和安全等多方面获益,

(2) G-DK.

我们在文献 [64] 中提出一种基于 DK[35] 的 DHT 分组构建技术 G-DK. 为了支持任意组的创建, G-DK 需要为节点路由表增加组相关的邻居信息. 我们首先给出如下定义.

定义 1　Kautz 匹配度. 对任意两个 Kautz 串 $u = u_1u_2\cdots u_m$ 和 $v = v_1v_2\cdots v_n$, u 和 v 的 Kautz 匹配度 $M(u,v) = i$ 是指 i 的最大取值, $0 \leqslant i \leqslant \min(m,n)$, 使得对任意的 j $(1 \leqslant j \leqslant i)$ 有 $u_{m-i+j} = v_j$.

例如, $M(10\underline{121}, \underline{121}20) = 4$, $M(10\underline{121}, \underline{121}20) = 3$.

定义 2　Kautz 距离. 令节点 $u = u_1u_2\cdots u_m$, 节点 $v = v_1v_2\cdots v_n$, 则从节点 u 到节点 v 的 Kautz 距离为 $D(u,v) = |v| - M(u,v)$.

例如, 在如图 12(d) 所示的 DK 拓扑图中, 令 $u = 20$, $v = 210$, 那么有 $|v| = 3$, $M(u,v) = 0$, 从而 $D(u,v) = 3$.

在 DK 中各节点的标识长度可能随着节点加入/退出覆盖网而发生变化, 从而给各组的维护带来困难. 针对该问题, 我们假设 G-DK 覆盖网所支持的节点标识长度的最大值为 λ 位, 每个节点 $u = u_1u_2\cdots u_m$ 将产生一个长度为 λ 的环标识 u'. 首先通过 KHash 算法 [10], 为节点 u 产生一个长度为 $\lambda - |u|$ 的前缀. 由于在 Kautz 串中相邻字符不相同, 因此需要产生一个长度为 $\lambda - |u| + 1$ 的 Kautz 串 $v = \mathrm{KHash}(u, d, \lambda - |u| + 1) = v_1v_2\cdots v_{\lambda-|u|+1}$. 如果 $v_{\lambda-|u|+1} \neq u_1$, 那么节点 u 的环标识 $u' = v_2\cdots v_{\lambda-|u|+1} \bullet u$; 否则 $u' = v_1v_2\cdots v_{\lambda-|u|} \bullet u$, 其中 "$\bullet$" 为字符串连接操作符. G-DK 覆盖网中所有节点的环标识形成一个类似于 Chord 环 [11] 的稀疏 Kautz 环. 例如, 图 19(a) 给出了图 12(a) 所示的 DK 拓扑对应的 Kautz 环, 图 19(b) 给出了图 12(d) 所示的 DK 拓扑对应的 Kautz 环, 其中各节点环标识中带有下划线的部分为该节点的原始标识. 基于上述结构 G-DK 能够容易地实现 DS 路由、PC 路由和传统的 DHT 路由.

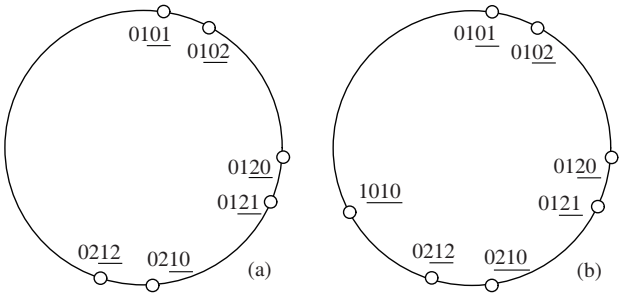

图 19 Kautz 环示例

Figure 19 Example of Kautz ring

5 未来工作的展望

5.1 最优状态效率折衷

由于 DHT 的路由延迟与节点度数紧密相关, 研究者们将 DHT 在节点度数和路由延迟之间的折衷称之为 "状态效率折衷"[65](state-efficiency tradeoff). 文献 [66] 证明, 对于节点度数为 $O(d)$ 和 $O(\log_d N)$ 的 DHT, 其网络直径的下界分别为 $\Omega(\log_d N)$ 和 $\Omega(\log_d N/\log_d \log_d N)$. 另一方面, 在最大出度为 d, 节点数为 N 的任意网络中, 网络直径 D 由 Moore 界 [34] 所限定: $D \geqslant \lceil \log_d(N(d-1)+1) \rceil - 1$. 我们注意到 Moore 界仅在集中式网络中才能达到, 因此 DHT 网络直径的下界应高于 Moore 界. 在给定平均节点度数 (或最大度数) 下 DHT 网络直径的确定下界, 将是未来工作中需要研究的一个重要学术问题.

5.2 振荡环境下的多目标协同优化

振荡环境下的多目标协同优化互联网资源的自治特性使得节点可能会频繁地加入/退出覆盖网, 形成一定程度的振荡. 然而, 在大多数现有 DHT 中节点路由表由其静态拓扑图特性决定, 与系统振荡频率 (即节点加入/退出频率) 无关. 近年来, 一些研究者对系统振荡进行研究. 例如, 文献 [13] 提出根据指定的带宽预算设置节点度数以获得较低的路由延迟, 文献 [67] 提出使用弹性路由表 (elastic routing table) 调整节点入度以获得较好的负载平衡特性等. 然而, 上述研究在针对某一特定目标进行优化的同时忽略了其他系统属性. 因此, 未来的一个重要工作是: 在振荡环境下通过反馈控制 [68], 动态调整概率路由表等方法, 实现针对系统多个属性的多目标协同优化.

5.3 高效复杂查询支持

现有研究通常认为多个查询是独立进行的, 对不同复杂查询之间关系的研究较少. 在 DHT 中进行多个连续 (continuous) 查询时将产生大量网络通信负载, 甚至导致网络瘫痪 [69]. 因此, 下一步的研究重点将是: 考虑不同查询之间的关系, 在存在大量连续查询的情况下, 通过把现有的集中式优化技术应用到分布式的 DHT 环境, 实现不同查询间的有效共享, 进而提高系统的可扩展性.

目前对基于 DHT 的其他类型复杂查询 (如 Top-k 查询, Skyline 查询和 JOIN 查询等) 的研究还相对较少. 我们注意到所有基于 DHT 的复杂查询在数据库技术中均有与之对应的 SQL 查询, 因此,

未来的一个重要研究方向将是: 结合目前数据库技术中 SQL 查询的实现方法, 在分布式的 DHT 环境中高效地实现各种类 SQL 查询.

6 结束语

虚拟计算环境 (iVCE) 的目标是实现互联网中分布、异构、自治、动态、大规模资源的有效共享. DHT 是实现上述目标的重要途径. 我们设计了基于 Java 的程序设计语言 Owlet 及其编程环境 [70], 向 iVCE 上层应用提供基本的 DHT 拓扑构建功能的支持. 目前, Owlet 语言已经内嵌了 FissionE 和 DLG-HyperTree (应用 DLG, 基于 HyperTree 图的 DHT) 两种覆盖网的基本实现, 并结合资源动态绑定和 XML 数据模型等, 实现了上层应用逻辑与资源共享逻辑的解耦. Owlet 向上层应用主要提供了节点加入/退出覆盖网、自定义消息、资源信息发布等接口函数, 上层应用开发者通过调用 Owlet 接口函数即可使用相应的动态维护与消息路由功能. 在未来工作中, 我们将在 Owlet 语言中进一步实现对面向复杂查询的 DHT 索引, 以及面向灵活路由的 DHT 分组等功能的支持.

参考文献

1 Lu X C, Wang H M, Wang J. Internet-based virtual computing environment (iVCE): concepts and architecture. Sci China Ser F-Inf, 2006, 49: 681–701

2 Lazowska E D, Patterson, D A. Distributed computing, Science, 2005, 308: 757

3 Hoffman D, Novak T, Venkatesh A. Has the Internet become indispensable? Commun ACM, 2004, 47: 37–42

4 Schoder D, Fischbach K. Peer-to-peer prospects. Commun ACM, 2003, 46: 27–29

5 Taylor I J. From P2P to Web Services and Grids. London: Springer-Verlag, 2005. 20–23

6 Balakrishnan H, Kaashoek M F, Karger D, et al. Looking up data in P2P systems. Commun ACM, 2003, 46: 43–48

7 Castro M, Costa M, Rowstron A. Debunking some myths about structured and unstructured overlays. In: Proceedings of the 2nd NSDI. Boston: USENIX Press, 2005. 85–98

8 Daswani N, Molina H G, Yang B. Open problems in data-sharing peer-to-peer systems. ICDT 2003

9 Theotokis S A, Spinellis D. Survey of peer-to-peer content distribution technologies. ACM Comput Surv, 2004, 36: 335–371

10 Li D S. Research on peer-to-peer resource location in large-scale distributed systems. PhD Thesis. Changsha: National University of Defense Technology, 2005. 21–22

11 Stoica I, Morris R, Karger D R, et al. Chord: a scalable peer-to-peer lookup service for Internet applications. IEEE ACM Trans Net, 2003, 11: 17–32

12 Gupta I, Birman K, Linga P, et al. Kelips: building an efficient and stable P2P DHT through increased memory and background overhead. In: Kaashoek M F, Stoica I, eds. Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03). Berkeley: Springer, 2003. 160–169

13 Li J, Stribling J, Morris R, et al. Bandwidth-efficient management of DHT routing tables. In: Proceedings of the 2nd NSDI. Boston: USENIX Press, 2005. 99–114

14 Shen H Y, Xu C Z, Chen G H. Cycloid: a scalable constant-degree p2p overlay network. Perform Eval, 2005, 63: 195–216

15 Ratnasamy S, Francis P, Handley M, et al. A scalable content addressable network. In: SIGCOMM 2001. San Diego: ACM Press, 2001. 161–172

16 Plaxton C G, Rajaraman R, Richa A W. Accessing nearby copies of replicated objects in a distributed environment. In: Proceedings of SPAA. Newport: ACM Press, 1997. 311–320

17 Zhao B Y, Huang L, Stribling J, et al. Tapestry: a resilient global-scale overlay for service deployment. IEEE J Sel Area Comm, 2004, 22: 41–53

18 CURRENT Lab. Chimera V1.20. http://current.cs.ucsb.edu/projects/chimera

19 Rowstron A, Druschel P. Pastry: scalable, decentralized object location and routing for large-scale peer-to-peer systems. In: Guerraoui R, ed. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware). Heidelberg: Springer, 2001. 329–350

20 Castro M, Costa M, Rowstron A. Performance and Dependability of Structured Peer-to-Peer Overlays. Technical Report MSR-TR-2003-94, Microsoft Research. 2003

21 Rhea S, Geels D, Roscoe T, et al. Handling churn in a DHT. In: Proceedings of USENIX Annual Technical Conference. Boston: USENIX Press, 2004. 127–140

22 Rhea S, Godfrey B, Karp B, et al. OpenDHT: a public DHT service and its uses. In: Guérin R, Govindan R, Minshall G, eds. Proceedings of ACM SIGCOMM. Philadelphia: ACM Press, 2005. 73–84

23 Maymounkov P, Mazieres D. Kademlia: a peer-to-peer information system based on the xor metric. In: Druschel P, Kaashoek M F, Rowstron A I, eds. Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS'02). Cambridge: Springer, 2002. 53–65

24 Kumar A, Merugu S, Xu J, et al. Ulysses: a robust, low-diameter, low-latency peer-to-peer network. In: Proceedings of ICNP 2003. Atlanta: IEEE Press, 2003. 258–267

25 Malkhi D, Naor M, Ratajczak D. Viceroy: a scalable and dynamic emulation of the butterfly. In: Proceedings of PODC. Monterey: ACM Press, 2002. 183–192

26 Pugh W. Skip lists: a probabilistic alternative to balanced trees. In: Dehne F, Sack J, Santoro N, eds. Workshop on Algorithms and Data Structures. Ottawa: Springer, 1989. 437–449

27 Harvey N J A, Jones M B, Saroiu S, et al. SkipNet: a scalable overlay network with practical locality properties. In: Proceedings of USITS 2003. Seattle: USENIX Press, 2003

28 de Bruijn N G. A combinatorial problem. Koninklijke Nederlandse Akademie van Wetenschappen P, 1946, A49: 758–764

29 Kaashoek F, Karger D. Koorde: a simple degree-optimal distributed hash table. In: Kaashoek M F, Stoica I, eds. Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03). Berkeley: Springer, 2003. 98–107

30 Fraigniaud P, Gauron P. D2B: a De Bruijn based content-addressable network. Theor Comput Sci, 2006, 355: 65–79

31 Loguinov D, Kumar A, Rai V, et al. Graph-theoretic analysis of structured peer-to-peer systems: routing distances and fault resilience. In: Feldmann A, Zitterbart M, Crowcroft J, et al., eds. Proceedings of ACM SIGCOMM 2003. Karlsruhe: ACM Press, 2003. 395–406

32 Gai A T, Viennot L. Broose: a practical distributed Hash table based on the de Bruijn Topology. In Caronni G, Weiler N, Shahmehri N, eds. Proceedings the International Conference on Peer-to-Peer Computing. Switzerland: IEEE Computer Society, 2004. 167–174

33 Kautz W H. The design of optimum interconnection networks for multiprocessors. In: Architecture and Design of Digital Computer. USA: Springer, 1969. 249–277

34 Li D S, Lu X C, Wu J. FISSIONE: a scalable constant degree and low congestion DHT scheme based on Kautz graphs. In: Proceedings of IEEE INFOCOM. Miami: IEEE Computer Society, 2005. 1677–1688

35 Zhang Y M, Liu L, Li D S, et al. Distributed line graphs: a universal framework for building DHTs based on arbitrary constant-degree graphs. IEEE ICDCS 2008. 152–159

36 Li D S, Cao J N, Chan K, et al. Delay-bounded range queries in DHT-based peer-to-peer systems. In: Proceedings of ICDCS 2006. Lisboa: IEEE Computer Society, 2006

37 Gupta A, Agrawal D, Abbadi A E. Approximate range selection queries in peer-to-peer systems. In: Proceedings of the 1st Biennial Conference on Innovative Data Systems Research (CIDR). Asilomar, 2003

38 Schmidt C, Parashar M. Enabling flexible queries with guarantees in P2P systems. IEEE Internet Comput, 2004, 8: 19–26

39 Ganesan P, Yang B, Molina H G. One torus to rule them all: multidimensional queries in P2P systems. In: Amer-Yahia S, Gravano L, eds. Proceedings of WebDB'04. Paris: ACM Press, 2004. 19–24

40 Chawathe Y, Ramabhadran S, Ratnasamy S, et al. A case study in building layered DHT applications. In: Guérin R, Govindan R, Minshall G, eds. Proceedings of ACM SIGCOMM. Philadelphia: ACM Press, 2005. 97–108

41 Crainiceanu A, Linga P, Gehrke J, et al. PTree: a P2P index for resource discovery applications. In: Feldman S, Uretsky M, Najork M, et al., eds. Proceedings of WWW 2004. New York: ACM Press, 2004. 13–19

42 Bharambe A R, Agrawal M, Seshan S. Mercury: supporting scalable multi-attribute range queries. In: Yavatkar R, Zegura E, Rexford J, eds. Proceedings of SIGCOMM 2004. Portland: ACM Press, 2004. 353–366

43 Oppenheimer D, Albrecht J, Patterson D, et al. Distributed resource discovery on planetlab with SWORD. In: Proceedings of the 1st Workshop on Real Large Distributed Systems (WORLDS'04). Santa Fe: USENIX Press, 2004

44 Cai M, Frank M, Chen J, et al. MAAN: a multi-attribute addressable network for grid information services. In: Stockinger H, ed. Proceedings of the 4th International Workshop on Grid Computing (Grid'2003). Phoenix: IEEE Computer Society, 2003. 184–191

45 Crainiceanu A, Linga P, Machanavajjhala A, et al. P-Ring: an efficient and robust P2P range index structure. In: Chan C, Beng Ooi C, Zhou A, eds. Proceedings of SIGMOD 2007. Beijing: ACM Press, 2007. 223–234

46 Tang Y, Zhou S. LHT: a low-maintenance indexing scheme over DHTs. In: Proceedings of IEEE ICDCS 2008. Beijing: IEEE Computer Society, 2008. 141–151

47 Jagadish H V, Ooi B C, Vu Q H. Baton: a balanced tree structure for peer-to-peer networks. In: Böhm K, Jensen C, Haas L, eds. Proceedings of VLDB 2005. Trondheim: ACM Press, 2005. 661–672

48 Jagadish H V, Ooi B C, Tan K L, et al. Speeding up search in peer-to-peer networks with a multi-way tree structure. In: Chaudhuri S, Hristidis V, Polyzotis N, eds. Proceedings of SIGMOD 2006. Chicago: ACM Press, 2006. 1–12

49 Risson J, Moors T. Survey of Research Towards Robust Peer-to-Peer Networks: Search Methods. Technical Report UNSW-EE-P2P-1-1. 2004

50 Albrecht K, Arnold R, Gahwiler M, et al. Aggregating information in peer-to-peer systems for improved join and leave. In: Caronni G, Weiler N, Shahmehri N, eds. Proceedings of P2P Computing 2004. Zurich: IEEE Computer Society, 2004. 227–234

51 Bhagwan R, Varghese G, Voelker G M. Cone: Augmenting DHTs to Support Distributed Resource Discovery. Technical Report. San Diego: University of California, 2003

52 Yalagandula P, Dahlin M. A scalable distributed information management system. In: Yavatkar R, Zegura E, Rexford J, eds. Proceedings of SIGCOMM 2004. Portland: ACM Press, 2004. 379–390

53 Cao P, Wang Z. Efficient top-K query calculation in distributed networks. In: Chaudhuri S, Kutten S, eds. Proceedings of PODC 2004. Newfoundland: ACM Press, 2004. 206–215

54 Huebsch R, Chun B, Hellerstein J M, et al. The architecture of PIER: an internet-scale query processor. In: Proceedings of CIDR 2005. Asilomar, 2005. 28–43

55 Zhang Y M, Li D S, Lu X C. Scalable distributed resource information service for Internet-based virtual computing environment(in Chinese). J Softw, 2007, 18: 1933–1942

56 Wu P, Zhang C, Feng Y, et al. Parallelizing skyline queries for scalable distribution. In: Ioannidis Y, Scholl M, Schmidt J, et al., eds. Proceedings of EDBT 2006. Munich: Spinger Press, 2006. 112–130

57 Chen L, Cui B, Lu H, et al. iSky: efficient and progressive skyline computing in a structured P2P network. In: Proceedings of IEEE ICDCS 2008. Beijing: IEEE Computer Society, 2008. 160–167

58 Zhang Y M, Liu L, Li D S, et al. DHT-based range query processing for web service discovery. In: Proceedings of IEEE International Conference on Web Services 2009 (ICWS'09). Los Angeles: IEEE Computer Society, 2009. 477–484

59 Jagadish H V. Linear clustering of objects with multiple attributes. In: Garcia-Molina H, Jagadish H, eds. Proceedings of ACM International Conference on Management of Data (SIGMOD 1990). Atlantic City: ACM Press, 1990. 332–342

60 Yalagandula P, Dahlin M. Administrative Autonomy in Structured Overlays. Technical Report, University of Texas at Austin. 2006

61 Ganesan P, Gummadi K, Molina H G. Canon in G major: designing DHTs with hierarchical structure. In: Proceedings of IEEE ICDCS 2004. Tokyo: IEEE Computer Society, 2004. 263–272

62 Karger D, Ruhl M. Diminished chord: a protocol for heterogeneous sub-group formation in peer-to-peer networks. In: Voelker G, Shenker S, eds. IEEE IPTPS 2004. 288–297

63 Zhang Y M, Li D S, Chen L, et al. Enabling routing control in a DHT. IEEE J Sel Area Comm, 2010, 28: 28–38

64 Zhang Y M. Tag-Based path diversity in structured peer-to-peer networks. Technical report, 2010

65 Ratnasamy S, Shenker S, Stoica I. Routing algorithms for DHTs: some open questions. In: Druschel P, Kaashoek M, Rowstron A, eds. Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS'02). Cambridge: Springer Press, 2002. 45–52

66 Xu J, Kumar A, Yu X X. On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks. IEEE J Sel Area Comm, 2004, 22: 151–163
67 Shen H Y, Xu C Z. Elastic routing table with probable performance for congestion control in DHT networks. In: Proceedings of ICDCS 2006. Lisboa: IEEE Computer Society, 2006
68 Zhang H Y, Liu B H, Dou W H. Design of a robust active queue management algorithm based on feedback compensation. In: Feldmann A, Zitterbart M, Crowcroft J, et al., eds. Proceedings of ACM SIGCOMM 2003. Karlsruhe: ACM Press, 2003. 277–285
69 Huebsch R, Garofalakis M, Hellerstein J M, et al. Sharing aggregate computation for distributed queries. In: Chan C, Beng Ooi C, Zhou A, eds. Proceedings of SIGMOD 2007. Beijing: ACM Press, 2007. 485–496
70 National Laboratory for Parallel and Distributed Processing. Owlet Programming Language. http://owlet-code.sourceforge.net

Survey of DHT topology construction techniques in virtual computing environments

ZHANG YiMing[1,2*], LU XiCheng[1,2] & LI DongSheng[1,2]

1 *National Laboratory for Parallel and Distributed Processing (PDL), Changsha* 410073, *China;*
2 *School of Computer, National University of Defense Technology, Changsha* 410073, *China*
*E-mail: ymzhang@nudt.edu.cn

Abstract The Internet-based virtual computing environment (iVCE) is a novel network computing platform. The characteristics of growth, autonomy, and diversity of Internet resources present great challenges to resource sharing in iVCE. The DHT overlay (DHT for short) technique has various advantages such as high scalability, low latency, and desirable availability, and is thus an important approach to realizing efficient resource sharing. Topology construction is a key technique for structured overlays that realizes basic overlay functions including dynamic maintenance and message routing. In this paper, we first introduce the traditional techniques of DHT topology construction, focusing mainly on dynamic maintenance and message routing of typical DHTs, DHT indexing techniques for complex queries, and DHT grouping techniques for matching domain structures. We then present recent advances in DHT topology construction techniques in iVCE taking advantage of the characteristics of Internet resources. Finally, we discuss the future of DHT topology construction techniques.

Keywords virtual computing environments, DHT overlays, topology construction, distributed indexing, flexible routing