

# 基于节点异构的双向查询 Chord 系统

周伟平, 刘卫国

(中南大学信息科学与工程学院, 长沙 410083)

**摘要:**资源的有效确定是 P2P 网络研究中一个关键问题。针对基本 Chord 和双向查询 Chord 路由算法存在路由表有较多冗余信息、没有考虑节点异构性的问题, 该文通过对路由表冗余信息的改进, 提出一种基于节点异构的双向查询 Chord 系统。实验结果表明, 该系统减少了平均路由跳数, 提高了资源查询效率。

**关键词:** P2P 网络; 基本 Chord; 双向查询; 路由

## Bidirectional Search Chord System Based on Heterogeneity of Peers

ZHOU Wei-ping, LIU Wei-guo

(School of Information Science and Engineering, Central South University, Changsha 410083)

**【Abstract】** It is a key issue to locate resources effectively in P2P network research. The routing tables of the basic Chord algorithm and the bidirectional Chord algorithm have the problems of information redundancy and regardless of peers' heterogeneity. According to the bidirectional Chord algorithm, this paper proposes a bidirectional Chord system based on different peers by considering the heterogeneity of peers and the improvement of information redundancy in the routing tables. Experimental results indicate that the system reduces the average search top and improves the search efficiency of resources.

**【Key words】** P2P network; basic Chord; bidirectional search; routing

### 1 概述

P2P 是一种在应用层上构建的逻辑网络, 具有非中心化、可扩展性、健壮性、高性价比、负载均衡的特点, 相对于传统的分布式系统, P2P 具有无可比拟的优势。近年来, P2P 技术得到了迅猛的发展, 广泛应用于多种网络服务。P2P 的资源愈来愈丰富, 高效率地定位所要查询的资源一直是 P2P 网络的关键问题。

CAN<sup>[1]</sup>, Chord<sup>[2]</sup>, Pastty<sup>[3]</sup>, Tapestry<sup>[4]</sup>等都是基于分布式哈希表(Distributed Hash Table, DHT)的结构化 P2P 系统。其中, Chord 算法的平均查询路径长度是  $1/2(\ln N)$  ( $N$  是 Chord 系统中的节点数), 具有很好的应用价值, 但是还存在影响其查询效率的问题: 其路由表中存在较多的冗余信息, 没有利用其中的前驱节点信息, 也没考虑到节点的异构性。双向查询 Chord<sup>[5]</sup>利用前趋节点信息对基本 Chord 进行了改进, 但在减少路由信息冗余和节点异构方面有待改进。

### 2 相关研究

在文献[2]中, Chord 系统中的每个资源关键字和节点分别拥有一个  $m$  bit 的标识符。关键字标识符  $K$  通过哈希关键字得到, 而节点标识符  $N$  则通过哈希节点的 IP 地址得到。所有节点按照其节点标识符从小到大(取模  $2^m$  后)沿顺时针方向排列在一个逻辑的标识圆环上(称为 Chord 环)。关键字标识符为  $K$  的资源存储在这样的节点上, 该节点的节点标识等于  $K$  或者在 Chord 环上紧跟在  $K$  之后, 这个节点被称为  $K$  的后继节点, 表示为  $\text{successor}(K)$ 。每个节点需要维护一个路由表, 称为指针表(finger table), 其中最多含有  $m$  个表项。路由表保存标识符空间中顺时针距当前点的距离为  $2^{i-1}$  ( $1 \leq i \leq m$ ) 的标识符及该标识符的后继节点。

Chord 系统模型如图 1 所示。

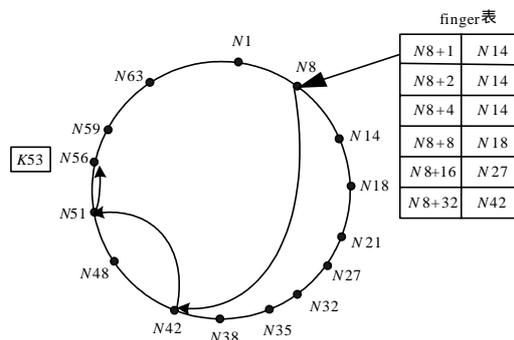


图 1 Chord 模型

文献[5]提出的双向查询 Chord 利用前趋节点信息, 从顺时针和逆时针 2 个方向进行资源查找。每个节点除了顺时针的查询路由表 finger 表, 还定义了一张逆时针路由表, 称为 R\_finger 表。表 1 是一个双向查询 Chord 中一个节点全部路由表的定义, 其中包括 Chord 协议中的 finger 路由表、successor 和 predecessor 路由项以及 R\_finger 路由表。

表 1 采用  $m$  bit 标识节点  $p$  的全部路由表

符号	定义
finger[k]	$(p+2^k) \bmod 2^m, 0 \leq k \leq m$
R_finger[k]	$(p-2^k+2^m) \bmod 2^m, 0 \leq k \leq m$
successor	节点沿顺时针方向的下一个节点 finger[0].node
predecessor	节点沿逆时针方向的下一个节点 R_finger[0].node

当节点发起查询时, 先通过判断关键字和节点之间位于

**基金项目:** 湖南省教育厅基金资助项目(湘教通[2004]344 号)

**作者简介:** 周伟平(1981 - ), 男, 硕士研究生, 主研方向: P2P 网络; 刘卫国, 教授

**收稿日期:** 2008-07-10 **E-mail:** wei725@126.com

Chord 环上顺时针和逆时针边的路由长度, 决定是选择顺时针还是逆时针方向查询。当顺时针方向的路由小于逆时针方向的路由, 从顺时针方向找起; 反之, 从逆时针方向找起。这样通过增加一次判断操作有效地减少了网络中信息的转发次数, 加快了查询速度, 其平均查询跳数是  $1/3(\ln N)$  ( $N$  是双向查询 Chord 系统中的节点数), 但它的路由表冗余信息比 Chord 系统多, 也没有考虑节点性能差异。

双向查询 Chord 模型如图 2 所示。

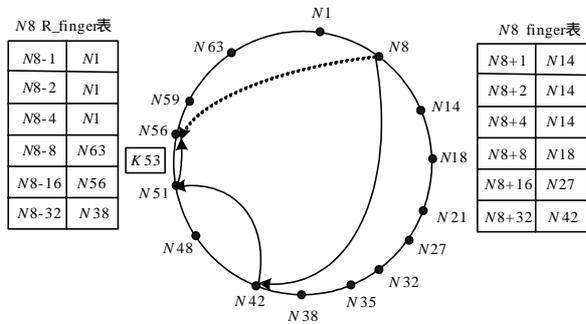


图 2 双向查询 Chord 模型

### 3 节点异构的双向查询 Chord 系统

#### 3.1 SChord 的形成

本系统利用节点性能的差异性, 引入了强节点的概念, 让强节点承担在主干环执行双向查询策略的任务, 同时改进了路由表的信息利用率, 达到提高资源查询效率的目的。采用了强节点的双向查询 Chord 系统是一个 2 层结构网络。

本文用  $C$  表示节点性能, 它主要由计算机节点运算能力  $A$ 、带宽  $B$  和节点在线时间  $D$  几个指标决定:

$$C = \alpha A + \beta B + \delta D$$

其中,  $\alpha$ 、 $\beta$ 、 $\delta$  都是常量系数, 可以根据系统节点性能的偏重进行设置, 且  $\alpha + \beta + \delta = 1$ 。

**定义 1** 强节点是 P2P 网络中性能较好(即  $C$  值较大)的节点。

**定义 2** 普节点是 P2P 网络中性能一般或比较差的节点。

对每一个节点的性能进行评估, 选择强节点作为一个簇的簇头, 选择普节点作为簇里的其他成员节点。簇里的普节点和强节点采用 Chord 的组织方式形成子环, 称为 SChord。各个簇头(即子环 SChord 的强节点)也按 Chord 的组织方式形成主干环, 称为 MChord。节点标志符和资源关键字标志符的获取与 Chord 系统相同, 都采用 DHT。

系统模型如图 3 所示。

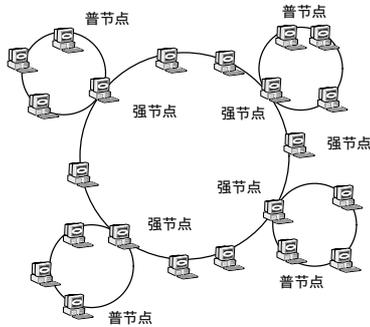


图 3 节点异构的双向查询 Chord 模型

#### 3.2 路由和数据定位

在本系统的 MChord 中, 各强节点之间的查找采用双向查找策略, SChord 中使用和 Chord 相同的顺时针单向查找,

则各强节点须保持 3 张路由表, 其中, 2 张为 MChord 上的顺逆方向路由表, 1 张为其所属 SChord 的路由表。而普节点只须保持其所属 SChord 的一张顺时针方向的路由表即可。

如图 1 所示的 Chord 路由表中, 对于节点标志符为 8 的当前节点  $N8$ , 在  $N8$  的路由表中, 随着  $i$  的增大,  $finger[i].interval$  呈 2 倍增长 ( $finger[i].interval$  表示  $finger$  指针表中第  $i$  项标志符和第  $i+1$  项标志符的空间间隔)。而相对  $2^m$  标志空间来说, 节点非常稀少并且基本是均匀分布的, 则当  $i$  较小时,  $finger[i].interval$  也较小, 中间的节点也很少, 甚至没有, 即前后相邻的多个指针完全相同, 造成了信息的大量冗余, 空间浪费严重。当  $2^m$  的环形空间节点数为 10 000 时, 其每张路由表的冗余信息大约为 57%。

由于 SChord 中的节点都是普节点, 性能一般或者较差, 因此为了不增加普节点的空间承受负担, 本系统中的 SChord 采取与 Chord 相同的路由策略, 冗余信息量与 Chord 基本相同。为了增加 SChord 路由的有效信息, 提高在 SChord 中的查询效率, 可用文献[6]的方法, 对其路由表的冗余信息进行改进, 只保留第 1 条路由信息, 删除其余的若干条冗余信息, 相应地, 在路由表不能到达的另一半 Chord 环上均匀地取若干个节点信息加进来, 则目标节点在另外一半时, 可直接跳到另一半上, 加快了查询速度。这在文献[6]中得证。

改进后的 Chord 模型如图 4 所示。

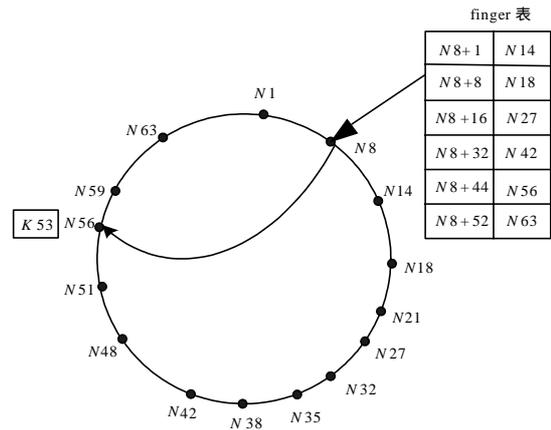


图 4 改进路由的 Chord 模型

在 MChord 中采取双向查找策略, 强节点在全局上拥有 2 张路由表, 正好可以利用强节点强大的性能加快各 SChord 之间的查询速度。但是如图 2 所示, 双向查询 Chord 路由表冗余信息量大约是 Chord 的 2 倍, 冗余度非常大, 同样需要进行重构, 增加有效信息, 进一步提高 MChord 上的查询效率。文献[6]只针对单向查询 Chord 中的路由表冗余信息给出了改进算法, 未考虑双向查询 Chord 中的路由表冗余信息, 因此, 本文针对双向路由表提出如下改进算法:

(1) 分别标志出节点顺时针 2 张路由表中的冗余信息为  $m+1$  条、 $n+1$  条; 分别删除 2 张路由表中除第 1 条路由信息外的其他冗余信息。

(2) 在节点顺时针路由表加入有效信息时, 必须考虑该节点在环上顺时针方向无法直接到达的另一半且在逆时针路由表中没有的  $m$  个节点, 这些所取节点应该是均匀分布的。

(3) 在节点逆时针路由表中加入有效路由信息时, 该节点在环上逆时针方向无法直接到达的另一半且在顺时针路由表中没有的  $n$  个节点也必须是均匀所取的。

这样, 改进后的 MChord 的路由命中率有所提高。

改进后的双向查询 Chord 如图 5 所示。

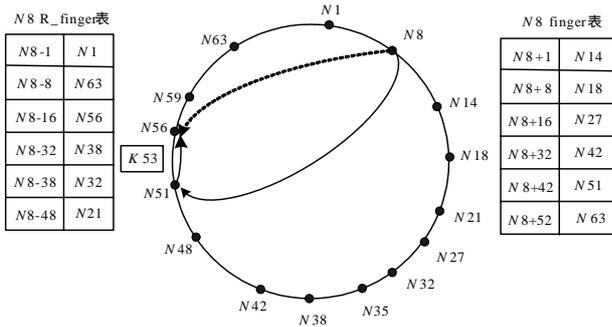


图 5 改进路由的双向 Chord 模型

节点查找资源过程如下：

(1)节点发起查询，在该节点所在的 SChord 上按照单向路由策略进行查找，如果找到所需资源，转到(4)；如果没有找到，转到(2)。

(2)将查询信息发给查询节点所在 SChord 环上的强节点，由该强节点在 MChord 环上按照双向查询路由策略查询资源节点所在 SChord 环上的强节点。

(3)当找到资源节点所在 SChord 环上的强节点时，在该 SChord 环上进行单向路由查找。

(4)将资源所在节点信息按查询路线逆向返回给查找节点，查找过程结束。

#### 4 实验和分析

实验中使用 BRITE<sup>[7]</sup>作为拓扑产生器，选择按照 bottom-up 类型参数生成 2 层拓扑结构，拓扑生成的方法是自底向上的。路由结构是使用 Waxman 模式，节点按照 heavy-tailed 的方法分配到 SChord 中，在此基础上用 C++ 实现 Chord 系统。

##### 4.1 SChord 的规模对性能的影响

本文用 BRITE 生成系统节点总数是 10 000。在系统节点总数保持不变的情况下，对系统中 SChord 中的节点个数分别取 1, 10, 100, 1 000, 10 000 进行多次实验，得到的结果如图 6 所示。

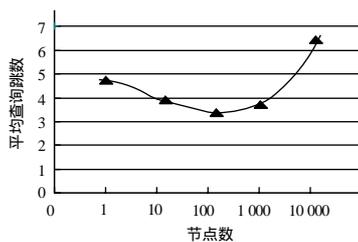


图 6 SChord 的节点数对查询性能的影响

当规模为 1 时，则回归到了双向查询 Chord 系统，性能也与其相同。当规模为 10 000 时，则回归到 Chord 系统，性能也与其相近。当规模是 100 时，性能最优。

为了使实验更具有一般性，分别取总节点数为 2 000, 4 000, 6 000, 8 000 进行仿真，每次实验对 SChord 的规模进行多种设置，结果与图 6 相似，当其规模为  $\sqrt{N}$  时 ( $N$  是系统的总节点数)，系统具有较好的性能。这是因为如果 SChord 中节点规模过大，则强节点个数过少，无法发挥双向查询的快速特性；如果 SChord 中节点规模过小，则 SChord 的个数过多，会增加 SChord 之间的查询，反而降低整个系统的查询速度。当规模是  $\sqrt{N}$  时，既能使强节点的性能得以发挥，又

使得大多数的查询在各自的 SChord 中发生。

#### 4.2 与 Chord 系统、双向 Chord 系统的比较

本文用 BRITE 分别生成系统节点个数 1 000, 2 000, 3 000, 4 000, 5 000, 6 000, 7 000, 8 000, 9 000, 10 000 进行多次实验，每次实验中其 SChord 规模大小取总节点的平方根近似值，对其进行仿真，结果如图 7 所示。

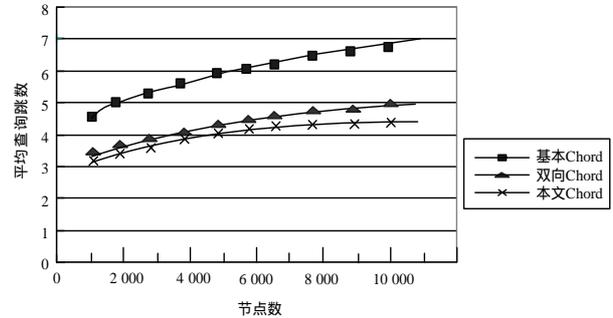


图 7 平均查询跳数对比图

可以看出，当系统总节点数较少时，本系统在 MChord 上的双向路由表项不是很多，对路由表冗余信息的改进虽然能提高 MChord 的查询命中率，但并不明显，所以，系统整体查询性能与双向查询 Chord 系统差不多。当节点规模逐渐增大时，MChord 上的双向路由表项逐渐增多，添加进来的有效信息也在增多，并且查询多数发生在改进路由后的 SChord 中，因此，系统的性能相对后两者有着明显的改进，说明系统有着较好的扩展性。

#### 5 结束语

现有研究在改进 P2P 资源查询效率时，只是单独考虑了节点性能差异、路由表冗余信息改进以及双向查询，本文将这三者综合考虑，提出了一种基于节点异构的双向查询 Chord 系统，给出了一种针对双向查询 Chord 路由表冗余信息的改进算法。实验证明本系统可以减少平均路由跳数，提高资源查询效率。由于本系统是基于 DHT 的，因此能方便地移植到实际 P2P 中。

#### 参考文献

- [1] Ratnasamy S, Francis P, Handley M, et al. A Scalable Content-addressable Network[C]//Proc. of ACM SIGCOMM'01. San Diego, California, USA: ACM Press, 2001: 161-172.
- [2] Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications[C]//Proceedings of ACM SIGCOMM'01. San Diego, California, USA: ACM Press, 2001.
- [3] Rowstron A, Druschel P. Pastry: Scalable, Distributed Object Location and Routing for Large-scale P2P Systems[C]//Proc. of International Conference on Distributed Systems Platforms (Middleware). Heidelberg, Germany: Springer, 2001.
- [4] Zhao Ben, Kubiawicz J D, Joseph A D. Tapestry: An Infrastructure for Fault-resilient Wide-area Location and Routing[R]. University of California, Technical Report: UCB/CSD-01-1141, 2001.
- [5] Ganesan P, Manku G S. Optimal Routing in Chord[D]. Stanford, USA: Stanford University, 2004.
- [6] 姜守旭, 韩希先, 李建中. 一种改进的 chord 路由算法[J]. 计算机应用, 2006, 26(4): 919-921.
- [7] Medina A, Lakhina A, Matta I, et al. Brite: A Network Topology Generator[EB/OL]. (2001-08-10). <http://www.cs.bu.edu/brite/>.