

基于多环的 Chord 改进算法

李建军¹, 熊选东¹, 谭晓贞²

(1. 解放军信息工程大学电子技术学院, 郑州 450004; 2. 海军司令部航空管制处, 北京 100071)

摘要:通过分析 Chord 协议, 提出基于多环的 Chord 改进算法 MR-Chord。MR-Chord 采用多环和组相结合的结构, 在组内每个节点记录全组的路由, 组与组之间以递归算法相连成多个环。通过理论分析和仿真实验对 Chord 和 MR-Chord 进行比较, 结果表明, MR-Chord 使系统的性能和适应性更好, 路由表中的冗余很少。

关键词: Chord 协议; P2P 网络; 多环; 性能分析

Improved Chord Algorithm Based on Multi-ring

LI Jian-jun¹, XIONG Xuan-dong¹, TAN Xiao-zhen²

(1. Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004;

2. Navy Command Air Traffic Control Department, Beijing 100071)

【Abstract】 By analyzing Chord protocol, this paper proposes an improved Chord algorithm called MR-Chord(Chord Based on the Combination of Ring and Group), which is based on multi-ring. MR-Chord uses the structure combining multi-ring and group. Each node in the group records the whole routing of the group and the groups link into multi-ring with recursive algorithm. Analysis on theory and simulation results show that MR-Chord has better performance and adaptability of the system, and the routing tables have few redundancy.

【Key words】 Chord protocol; P2P network; multi-ring; performance analysis

1 概述

P2P 表示网络节点对等互联。它平行于网络上流行的客户端/服务器的主从互联模式, 实现了分布式资源利用与共享, 每个节点均可进行对等通信, 具备同时对信息内容进行接收、发送、存储、搜索等功能, 各节点对等协同完成任务。网络中的每个节点具有相同的地位, 既可以请求服务又可以提供服务, 同时扮演客户端和服务器的 2 种角色, 还具有路由器和高速缓冲存储器的功能。

P2P 网络按照网络中节点和资源的关系可以分为非结构化 P2P 网络和结构化 P2P 网络。非结构 P2P 网络是基于泛洪搜索的, 搜索较慢而且消耗大量的资源。结构化 P2P 网络是利用分布式哈希表(Distributed Hash Table, DHT)实现的, 搜索速度快, 产生的查询消息少。Chord 协议^[1]是 2001 年麻省理工学院(Massachusetts Institute of Technology, MIT)专门为结构化 P2P 应用设计的。它虽然具有算法简单、查找速度快等特点, 但仍存在以下不足, 从而影响了效率: (1)对网络拓扑的实际结构欠缺考虑, 使得逻辑上相邻的一跳在物理网络中可能经过很多跳数, 造成了相当大的网络延迟。(2)Chord 的路由表中存在严重的冗余。为了减少不必要的网络延迟、提高路由的效率, 许多学者对 Chord 做了进一步的研究。其中, Echord^[2], TCS-Chord^[3], Achord^[4]等都通过不同的方式对 Chord 进行了改进, 但是这些改进局限性较大、实用性较差, 而且都无法消除 Chord 路由表中大量的冗余。另一些研究只是简单地在路由表存在冗余时以其他节点来替换, 或者引入超级节点。

本文提出了一种结合多环和组的算法 MR-Chord(Chord Based on the Combination of Ring and Group), MR-Chord 利用组构成环, 由于组在线时间远大于节点, 因此可以大幅减少

路由表的冗余。同时通过调节组的大小来适应网络的动态变化, 提高查找效率。而且可以在其上应用任意减少网络延迟的改进方法。

2 相关工作

2.1 Chord 协议

Chord 协议通过相容哈希函数把指定的关键字 *Key* 映射到对应的节点 *Node* 上。在 Chord 中, 节点及所有的关键字都被同一个哈希算法映射到一个 *m* 位的标识符(一个节点的标识符是它的 IP 地址或其他信息经过哈希映射得到的)。其中, *m* 的取值必须足够大, 才能使 2 个节点或 2 个关键字的哈希值相同的可能性忽略不计。

在 Chord 系统中, 每个节点按照其标识符大小依次顺序连接, 最后首尾相接形成一个环。对节点的 IP 地址或其他信息进行哈希得到节点的标识符 *Node Key*, 用同样的方法, 哈希资源的关键字可以得到资源的标识符 *key*。每个资源根据自己的 *key* 值被分派到某个节点上, 这个节点叫作 *key* 的 successor 节点, 用 *successor(key)* 表示, 它是 Chord 环上沿顺时针方向第 1 个标识符大于等于 *key* 的节点。

2.2 Chord 的路由分析

根据 Chord 的原理, 现假设 Chord 环的节点数为 2^m , 而网络中的实际节点数为 2^n ($m \gg n$), 并且所有节点均匀分布在 Chord 环上。于是任意相邻节点间的间距为 $\Delta = 2^{m-n}$ 。若其中一个节点为 N_0 , 则之后的 2^{n-1} 个节点可以表示为 $N_0 + \Delta$, $N_0 + 2\Delta$, ..., $N_0 + 2^{n-1}\Delta$; 其路由表为 $(N_0 + 2^{i-1}) \bmod 2^m$

作者简介: 李建军(1980 -), 男, 硕士研究生, 主研方向: 网络安全, P2P 网络; 熊选东, 研究员; 谭晓贞, 工程师

收稿日期: 2009-05-20 **E-mail:** lijianjun2006@gmail.com

($1 \leq i \leq m$)。可以看出, 当 $2^{i-1} \Delta$ 时, $(N_0 + 2^{i-1}) \bmod 2^m = (N_0 + 2^{i-1}) \bmod 2^m = \dots = (N_0 + 2^{i-1}) \bmod 2^m = N_0 + \Delta$ 。显然这些是 N_0 路由表中的冗余信息, 因此, 冗余度为 $(m-n)/m \times 100\%$ 。因为一个 Chord 系统建立后, m 值是固定的, 所以随着 n 的增大, 冗余度逐渐减小, 而 P2P 网络中的节点数量肯定小于 2^m , 因此, Chord 的路由表中必然会产生不同程度的冗余。

3 MR-Chord

3.1 网络结构

定义 1 组 在 MR-Chord 中, 从 0 到 2^m 每 2^{m-k} 个标识构成一个组, $(Node\ Key)/2^{m-k}$ 的值称为组号 Gid 其中 $Node\ Key$ 是节点的标识; m 为标识符的位数; k 为常数。其构成如图 1 所示。

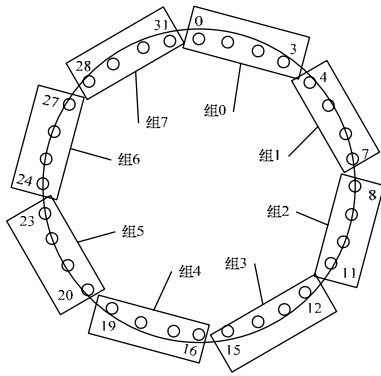


图 1 MR-Chord 中的组

定义 2 基本环 在 MR-Chord 中, $(Node\ Key)/2^{m-k}$ 的余数相同的节点构成的环叫基本环, 该余数称为环号 Rid 。

在 MR-Chord 中, 在环上平均分成 2^k 个组, 每个组中的节点保存组内所有节点的信息。而组与组之间构成了 2^{m-k} 个基本环。如果一个组中没有某个基本环对应的节点, 可以选用组内性能高的节点来代替, 这样可以更好地利用高性能的节点资源, 又可以保证基本环上的路由完整。如果在某个组内一个节点也没有, 则指向下一组中对应的节点。

MR-Chord 中节点的路由表是根据自己所在的基本环和组共同构成的, 如图 2 所示。同时 MR-Chord 维护一个本组所有节点的指针表(circle 表)。

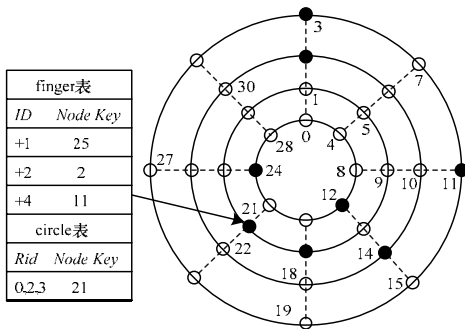


图 2 MR-Chord 中的路由表

3.2 MR-Chord 查找算法

当某个节点要查找资源时, 首先将该资源的关键字哈希后得到 m 位的 Key , 然后 Key 除以 2^{m-k} 得到组号 Gid 。其后过程如下:

(1) 判断 Gid 是否与该节点的组号相同, 如果是, 则从 circle 表中查找到与 Key 对应的节点, 结束查找; 否则, 转步骤(2)。

(2) 查找 finger 表, 判断 Gid 是否与某个 ID 相同, 如果是, 则把这个查询请求转发给该 ID 对应的节点, 转步骤(1); 否则, 转步骤(3)。

(3) 查找 finger 表中 ID 最大但不超过 Gid 的第 1 个节点, 并将这个查询请求转发给该节点, 然后转步骤(1)。

4 性能分析及仿真实验

4.1 性能分析

由于 MR-Chord 算法仅对 Chord 中环的构成方式进行改进, 因此继承了 Chord 算法简单、实现方便等特点。同时因为没有引入超级节点, 所以不会存在过分依赖超级节点等缺点。下面从路由表冗余、最大跳数和实现代价等方面进行分析。

(1) 路由分析。从 MR-Chord 的网络结构可以看出, finger 表中的指针实际上是指向组的, 即只要组中至少存在一个节点, 那么所有的 finger 表都不会存在冗余。由于节点都是均匀分布在环上的, 组内没有一个节点的概率很小, 因此 MR-Chord 的 finger 表中基本不存在冗余。而同组的节点是记录其他所有节点的, 因此, circle 表不存在冗余的问题。

(2) 最大跳数。从 MR-Chord 的查找算法可以看出, 当目标节点与查找节点在同一组时, 只需要一跳就可以定位到目标节点。而当目标节点与查找节点不在同一组时, 通过 finger 表查找到对应组后, 如果 finger 表中该组对应的节点就是目标节点, 则最多需要 k 跳达到目标节点; 否则, 最大跳数为 $k+1$ 。由于 k 值肯定小于节点数为 2^n 的 Chord 的最大跳数 n , 因此 MR-Chord 的最大跳数肯定小于 Chord 等一般的结构化 P2P 系统。

(3) 代价分析。由于 Chord 环中的每个节点都存在崩溃的可能, 因此 Chord 环中的节点必须定期触发更新过程, 最终达到更新各个节点中错误 finger 表项的目的, 以保持 Chord 环的正确性以及摒弃错误的路由项。但是由于 MR-Chord 中组与组之间通过多个环相连接, 即使某个节点失效, 同组的其他节点也可以代替它完成跳转, 具有更好的容错性, 因此 MR-Chord 更新过程的时间间隔可以更长。

由于节点需要记录同组内的其他所有节点, 在更新过程中需要占用大量的带宽, 因此在网络变化频繁时, 如果组内节点数量太多, 还是会带来性能上的影响。因此, k 值应根据具体的网络情况进行调整, 既要保证性能的提高, 又要将带宽控制在可接受的范围内。

4.2 仿真实验

本文选用的仿真器是 MIT 开发的开源软件 P2Psim, 它用 C++ 语言编写, 工作在 Unix/Linux 平台上。为了使模拟数据尽可能真实, 仿真时采用的网络拓扑结构完全采自 Internet 实际数据。这些数据来自华盛顿大学计算机系提出的 King^[5] 数据集。在对 P2Psim 进行改造使其能够实现 MR-Chord 并获得需要的数据后, 采用 King 方法提供的 2 048 个节点进行仿真。令 $k=7, m=16$, 分别记录查找成功率、平均查找跳数以及 finger 表的冗余度。结果表明, MR-Chord 可以将查询成功率保持在 99.5% 左右。这证明对 Chord 的改进并没有降低 Chord 的查找成功率。

MR-Chord 与 Chord 的平均跳数比较如图 3 所示。可以看出, 由于 MR-Chord 的查找过程可以在更小的基本环上或者组内进行, 因此其平均查找跳数明显少于 Chord。另外, 随着节点的增多, Chord 的平均跳数增长迅速, MR-Chord 却增长缓慢。这是因为在线节点数的增加对 MR-Chord 来说组

数并没有发生变化,而 MR-Chord 中的查找只需要组间跳转,所以跳数变化也不会很明显。

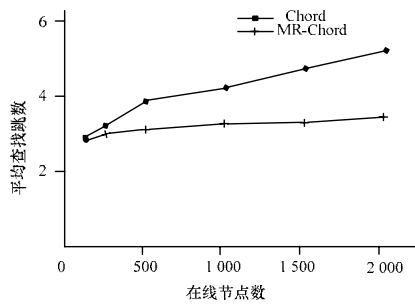


图3 MR-Chord与Chord的平均跳数比较

由于 MR-Chord 中的 finger 表只有在存在没有一个节点的组时才有冗余,因此只要在线节点能够保证每个组至少存在一个节点,那么 MR-Chord 的 finger 表就没有冗余。如图 4 所示,随着节点的增多,Chord 的冗余度虽然在减少,但是始终存在。而 MR-Chord 由于节点的增加,没有节点的组的数量急剧减少,冗余度明显降低,几乎为 0。

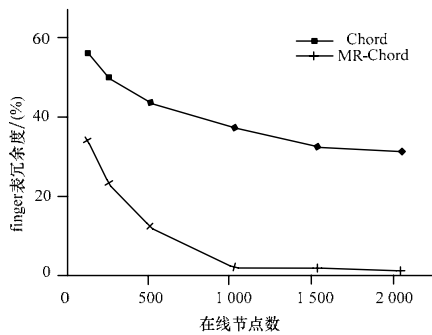


图4 MR-Chord与Chord的finger表的冗余度比较

最后将节点失效率从 1% 增大到 10%, 分别记录 Chord 与 MR-Chord 的查找失效率。从图 5 可以看出, MR-Chord 在节点失效的情况下查找失效率明显低于 Chord。这是因为组内的节点记录与组内其他所有节点相同,而且组与组之间通过多个环相连接,所以查找到的目标路径不只一条,查找失效率必然很低。

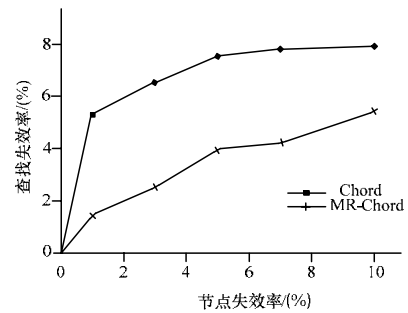


图5 MR-Chord与Chord的查找失效率比较

5 结束语

本文提出的 MR-Chord 将组内记录所有路由与组间递归查找相结合,减少了路由表的冗余,降低了查找时间和平均查找跳数,同时算法简单、易于实现。下一步的工作是研究 MR-Chord 的自适应性,使其可以根据网络的变化自动调节 k 值,从而达到带宽和查找效率的平衡。

参考文献

- [1] Stocia I, Morirs R, Karger D, et al. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Application[C]//Proceedings of the ACM SIGCOMM'01. [S. l.]: ACM Press, 2001.
- [2] Ratnasamy S, Handley M, Karp R, et al. Topologically-aware Overlay Construction and Server Selection[C]//Proc. of the IEEE INFOCOM'02. New York, USA: IEEE Press, 2002.
- [3] Liu Ye, Yang Peng, Chu Zi, et al. TCS-Chord: An Improved Routing Algorithm to Chord Based on the Topology-aware Clustering in Self-organizing Mode[C]//Proceedings of the 1st International Conference on Semantics, Knowledge and Grid. Beijing, China: [s. n.], 2005.
- [4] Dao Lehai, Kim J W. AChord: Topology-aware Chord in Anycast-enabled Networks[C]//Proc. of 2006 International Conference on Hybrid Information Technology. Cheju Island, Korea: IEEE Press, 2006.
- [5] Gummadi K P, Saroiu S, Gribble S D. King: Estimating Latency Between Arbitrary Internet End Hosts[C]//Proceedings of SIGCOMM'02. Marseille, France: [s. n.], 2002.

编辑 张帆

(上接第 115 页)

可以看出,在目标不丢失的前提下,适当提高 t_s 有利于提高节点的利用效率,降低无效唤醒次数,且网络对高速目标的追踪效率明显高于低速目标。但是当目标速度比较快时,应当将 t_s 严格控制在式(1)的范围内,否则容易失去目标。

在 $k=0$ 的情况下,本文对目标感应半径内节点的唤醒比率进行了统计, SAT 算法至少比唤醒感应半径内所有节点的算法节约了 47% 的能量,在主节点能连续更替,且相邻 2 次扫描所得的主节点为邻居时,该比率可以达到 71%。

5 结束语

本文提出一种无线传感器网络中的速度自适应追踪算法,实验表明,该算法具有优良的目标追踪能力,当目标运动曲线变化剧烈时仍然能自适应地调整节点的唤醒区域。与基于预测的目标跟踪算法相比,省去了大量的数据计算和通信,提高了网络整体寿命。

参考文献

- [1] Zhang Wensheng, Cao Guohong. DCTC: Dynamic Convoy Tree-based Collaboration for Target Tracking in Sensor Networks[J]. IEEE Trans. on Wireless Communications, 2004, 3(5): 1689-1701.
- [2] Yang H, Sikda B. A Protocol for Tracking Mobile Targets Using Sensor Networks[C]//Proceedings of the 1st IEEE Workshop on Sensor Network Protocols and Applications. Anchorage, Alaska, USA: [s. n.], 2003.
- [3] Jeong J, Hwang T, He Tian. MCTA: Target Tracking Algorithm Based on Minimal Contour in Wireless Sensor Networks[C]//Proc. of INFOCOM'07. Anchorage, Alaska, USA: [s. n.], 2007.
- [4] Rudafshani M, Datta S. Localization in Wireless Sensor Networks[C]//Proc. of ACM IPSN'07. Cambridge, Massachusetts, USA: [s. n.], 2007.

编辑 顾姣健