利用二叉排序树改进结构化 P2P 模型

徐翠霞[1],崔玲玲[1],张家明[2]
XU Cui-xia[1],CUI Ling-ling[1],ZHANG Jia-ming[2]

1.潍坊学院 计算机与通信工程学院,山东 潍坊 261041
2.山东科技职业技术学院 信息网络系,山东 潍坊 261041
1.School of Computer and Communication Engineering,Weifang University,Weifang,Shandong 261041,China
2.Information Department,Shandong Vocational College of Science & Technology,Weifang,Shandong 261041,China
E-mail:wfucui@126.com

XU Cui-xia,CUI Ling-ling,ZHANG Jia-ming.Binary sort tree model for structured P2P improved mechanism.Computer Engineering and Applications,2009,45(36):101-104.

Abstract: P2P overlay network is a logical connection between the application layers consisting of networks,because of its easy to build,management flexibility,scalability,and in achieving a variety of applications on the Internet to play this important role. Based on the Chord algorithm this paper proposes a network topology model BBSTC,introduces the addition of network nodes and the routing algorithm from the Resource Locator,as well as the steps.Simulation experiments and analysis show that this program can significantly improve the success rate of search results and significantly reduce the routing hops of search.In the structured P2P environment,the strategy promotes the use and has certain research value.

Key words: Peer-to-Peer(P2P);Chord;resource locating

摘 要:P2P 覆盖网络是一种对等网之间的逻辑连接构成的应用层网络,由于其易于构建、管理灵活、可扩展性强,在实现互联网上的多种应用中发挥着重要的作用。在研究 Chord 算法的基础上提出了一个 BBSTC 网络拓扑模型,介绍了网络节点的加入和退出的路由算法以及资源定位的步骤,通过仿真实验和分析表明此方案可以显著改善搜索结果的成功率和大大减少搜索所需的路由跳数,在目前结构化 P2P 环境中,该策略有一定的推广利用和研究价值。

关键词:P2P;Chord;资源定位
DOI:10.3778/j.issn.1002-8331.2009.36.030 **文章编号:**1002-8331(2009)36-0101-04 **文献标识码:**A **中图分类号:**TP393.02

1 引言

随着网络的飞速发展,应用需求的不断增加,传统的 C/S 结构的网络不再能够完全满足人们的需要。在网络世界里越来越多的人需要互相交流,互相共享文件,共享信息资源,而 C/S 结构的网络并不能很好地满足大多数人的需求,P2P 网络应运而生,通过使用对等通讯的方式,人们不再完全依赖服务器就能实现信息交流和资源共享。不仅相当程度地减轻了服务器的负担,也使节点间通讯效率得到很大提升。

2005 年以后,P2P 流媒体直播技术逐渐成熟,也得到了越来越广泛的应用,尤其在中国发展更是迅速,PPLive、PPStre、QQLive 等商业系统赢得了用户的一致好评。客户机/服务器模式实质上是一种集中式体系结构,它在海量信息的组织、访问等方面存在着服务瓶颈、易于崩溃等缺点,各个对等节点的加入、退出通常十分频繁,因此采用合理的 P2P 网络模式,快速、准确地发现节点以及节点上的资源十分重要。最新的研究成果体现在采用分布式散列表(DHT)的分布式结构化网络模式。P2P 系统的应用层路由算法与网络层路由算法不同。基于 IP 层的路由算法形式多样,得到广泛应用的有两种:距离矢量路由算法和链路状态路由算法。网络层的拓扑结构主要由路由器组成,P2P 网络的路由通常与应用紧密相关,并不一定采用类似 IP 地址的层次结构路由,如何进行节点标识、消息目的地标识以及消息路由转发是 P2P 应用层路由要解决的最基本问题。一般情况下,P2P 网络利用自己的节点标识策略实现系统的路由,如 CAN。针对 P2P 的网络特性,良好的 P2P 网络路由算法需要满足以下基本要求:

(1)不依赖集中控制的分布式实现,避免过量的通信报文;

(2)每个节点均与一定数量的节点保持邻接关系;

(3)节点之间通过多个节点传递消息来完成通信;

(4)任何两个节点的消息通信的路由跳数,尽量维持在一个较小的数量级;

(5)一定数量节点的失效不会影响系统可用性;

(6)各个节点应能够维持一致的网络拓扑信息;

(7)节点可以很容易(传递消息的数量少)地加入和离开系统,并满足一定的可扩展性;

作者简介:徐翠霞(1973-),女,副教授,主要研究方向:网络安全与理论;崔玲玲(1979-),女,讲师,主要研究方向:网络安全与理论;张家明(1979-),男,助教,主要研究方向:网络安全与理论。

收稿日期:2009-08-18 **修回日期:**2009-10-09

(8)网络在分割后能提供良好的恢复策略使系统从错误中恢复来提高系统的健壮性。

综上所述,P2P 网络路由算法应该考虑节点的命名、定位、加入、离开以及节点之间的邻接关系、系统容错性等。

2　网络拓扑结构

2.1　Chord 算法

Chord 实现的操作是给每一个节点和资源都分配一个关键字,而这个过程由相容哈希函数 SHA-1 来完成,相容哈希函数哈希节点的 IP 地址产生节点的 m 位关键字标识符,资源的标识符可以直接用哈希函数哈希该数据本身而得到 m 位的关键字标识符。所有节点按照其节点标识符从小到大(取模 2^m后)沿着顺时针方向排列在一个逻辑的标识圆环上(称为 Chord 环)。每个关键字 K 都保存在它的后继节点中,即节点标识符大于等于关键字 K 的第一个节点,记为 $successor(K)$。这样只要给定一个关键字 key,就可以将该关键字映射到网络中的一个节点上,因此可以通过节点的路由表实现数据的查找。

图 1 给出了一个 m=6 的 Chord 环,环中分布了 10 个节点,存储了 5 个关键字,节点标识前加上 N 而关键字前加上 K 以示区别。因为 $successor(10)$=14,所以关键字 10 存储到节点 14 上。同理,关键字 24 和 30 存储到节点 32 上,关键字 38 存储到节点 38 上,而关键字 54 则存储到节点 56 上。

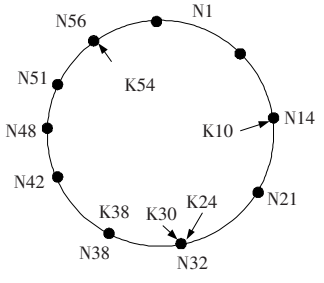


图 1　Chord

当网络中的参与节点发生变动时,上面的映射规则仍然要成立。为此,当某节点 n 加入网络时,某些原来分配给 n 的后继节点的关键字将分配给 n。当节点 n 离开网络时,所有分配给它的关键字将重新分配给 n 的后继节点。除此之外,网络中不会发生其他的变化。以图 1 为例,当标识为 26 的节点接入时,原有标识为 32 的节点负责的标识为 24 的关键字将转由新节点存储。

显然,为了能在系统中转发查询报文,每个节点要了解并维护 Chord 环上相邻节点的标识和 IP 地址,并用这些信息构成自身的路由表。有了这张表,Chord 就可以在环上任意两点间进行寻路。

2.2　二叉排序树

二叉排序树(Binary Sort Tree)或者是一棵空树,或者是具有下列性质的二叉树:(1)若它的左子树不空,则左子树上所有节点的值均小于它的根节点的值;(2)若它的右子树不空,则右子树上所有节点的值均大于它的根节点的值;(3)它的左、右子树也分别为二叉排序树。

根据上述的定义对二叉排序树进行中序遍历可得到一个节点的有序序列,满足二分查找的前提条件,这将大大降低路由查找的逻辑跳数。假定有序表的长度 n=2^h-1,则描述二分查找的判定树是深度为 h 的满二叉树。树中层次为 1 的节点有 1 个,层次为 2 的节点有 2 个,……,层次为 h 的节点有 2^{h-1} 个。假设表中每个记录的查找概率相等($p_i=\frac{1}{n}$),则查找成功时二分查找的平均查找长度

$$ASL_{bs}=\sum_{i=1}^{n}P_iC_i=\frac{1}{n}\sum_{j=1}^{h}j\cdot 2^{j-1}=\frac{n+1}{n}\mathrm{lb}(n+1)-1$$

对任意的 n,当 n 较大时,可有下列近似结果

$$ASL_{bs}=\mathrm{lb}(n+1)-1$$

可见,二分查找的效率比顺序查找高,于是提出基于二叉排序树的 Chord 算法 BBSTC(Based on Binary Sort Tree Chord),将拓扑中所有物理节点的集合记为 Peers,BBSTC 拓扑中的节点称为虚拟节点,每个虚拟节点与 Peers 中一个节点对应,拓扑结构分两层,上层是由 Chord 算法生成的网络拓扑图,考虑到路由查找的效率在该层的节点数量与事先规定的参数值($refmax$)有关。当上层的节点数量达到参数值($refmax$)后,新增的节点将按照二叉排序树的性质分别增加在以上层节点 $Nodei$ ($0<i<refmax$)为根节点的左子树或右子树叶子节点,如图 2 所示。

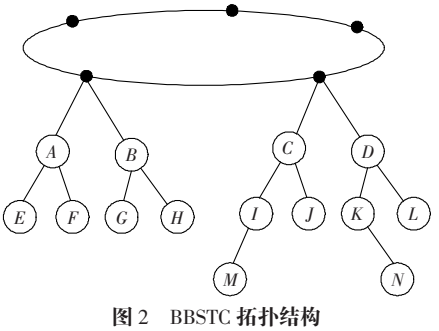


图 2　BBSTC 拓扑结构

在 BBSTC 拓扑结构的基础上,有如下定义。

定义 1　节点标识(NodeID)采用具有哈希值均匀分布的哈希函数 SHA-1 完成,相容哈希函数哈希节点的 IP 地址产生节点的 M 位二进制标识符。

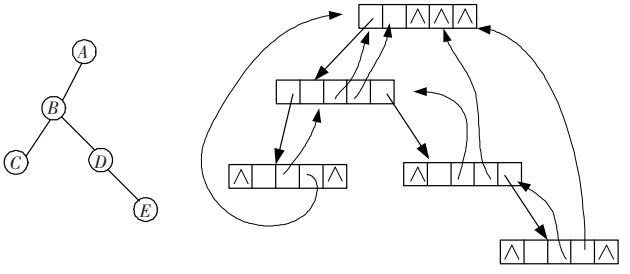
定义 2　资源的标识符可以直接用哈希函数哈希该资源的关键字本身而得到 L 位($L>M$)的二进制关键字标识符,资源的标识符是代表该资源关键字的索引,跟资源标识符相对应的是存放该资源的物理 IP 地址。资源关键字索引的存放规则是:该资源标识符的前 M 位与节点标识相同。满足上述条件的节点负责管理此二进制串标识的空间。

2.3　改进的二叉排序树存储结构

BBSTC 拓扑结构在理论上对资源定位搜索分两个步骤:首先在上层节点中搜索,然后选择合适的节点作为二叉排序树的根节点在下层节点中搜索,通过搜索树的父子关系就可以在树中进行高效而确定的路由。然而实际应用中,由于父子层次关系,一方面路由往往总是从上至下执行,另一方面由 $Nodei$ 左子树的叶子节点到 $Nodei$+1 右子树的叶子节点的访问,总是要经过几层父节点后到达上层的 $Nodei$ 转发到 $Nodei$+1 再进行搜索树。这就导致上层节点被访问的次数远大于底层节点,父节点被访问的次数多于儿子节点,这不但导致上层节点负载过重,而且增加了资源定位搜索的逻辑路由跳数,从而降低路由效率。可以从二叉树的存储结构入手,选择一种合理的机制,使得不同根节点的子树之间可以很快地跳转减少网络直径。

二叉树的链式存储结构是比较常用的。二叉树的节点由一个数据元素和分别指向其左、右子树的两个分支构成,则表示二叉树的链表中的节点至少包含 3 个域:数据域和左、右指针

域。为了便于找到节点的双亲，则还要在节点结构中增加一个指向其双亲节点的指针域。该文将增加一个指针域指向所在二叉树的根节点 *Nodei*，这样，即使是最底层的叶子节点发起查询请求，则通过一跳就可以在整个 BBSTC 拓扑的最上层进行不同 *Nodei* 二叉排序树根节点间的转发。如图 3 所示。



Lchild	Data	Parent	Root	Rchild

图 3　节点的存储结构

3　BBSTC 的节点加入

节点加入分为上层 Chord 环加入和下层二叉排序树加入两种情况，节点的 IP 地址被哈希函数 SHA-1 哈希成 *M* 位二进制标识符，按大小顺序地散列到上层 Chord 环上，当节点的数量超过事先规定的参数值 *refmax*-1 后，进行下层二叉排序树的构造。在二叉排序树中新插入的节点一定是新添加的叶子节点，并且是最后一个节点的左孩子或右孩子节点。根据二叉排序树的定义得知，左子树的各节点值小于根节点小于右子树各节点，所以，按照中序遍历二叉排序树可得到一个关键字的有序序列。也就是说，一个无序序列可以通过构造一棵二叉排序树而变成一个有序序列，构造树的过程即为对无序序列进行排序的过程。不仅如此，从上面的介绍可以知道，每次插入的新节点都是二叉排序树上新的叶子节点，则在进行插入操作时，不必移动其他节点，仅需改动某个节点的指针，由空变为非空即可。这就相当于在一个有序序列上插入一个记录而不需要移动其他记录。这表明，二叉排序树既拥有类似于二分查找的特性，又采用了链表作存储结构，比较适宜网络节点频繁加入、退出的情况，能有效降低网络抖动量（网络抖动量 Churning-Count 是节点加入或退出时所影响的节点数目）。可以看出，在节点加入的时候，网络抖动只涉及到该节点的父节点，网络抖动量 *Churning Count*=1，在节点退出的时候分几种不同的情况，网络抖动量有所不同。

4　BBSTC 的节点退出

对于二叉排序树来说，删除树上的一个节点相当于删除有序序列中的一个记录，只要在删除某个节点之后依旧保持二叉排序树的特性即可。假设在二叉排序树上被删节点为 *P*，其双亲节点为 *F*，且不失一般性，可设 *P* 是 *F* 的左孩子。如图 4 所示。

（1）若节点 *P* 为叶子节点，即 *PL* 和 *PR* 均为空树，由于删除叶子节点不破坏整棵树的结构，则只需修改其父节点的指针即可。网络抖动量为 1。

（2）若节点 *P* 只有左子树 *PL* 或者只有右子树 *PR*，此时只要令 *PL* 或 *PR* 直接成为其父节点 *F* 的左子树即可。显然，作此修改也不破坏二叉排序树的特性。网络抖动量为 1。

（3）若节点 *P* 的左子树和右子树均不空。显然，此时不能像上面那样作简单处理。在删除节点 *P* 之后，为保持其他节点之间的相对位置不变，可以令 *P* 的左子树为 *F* 的左子树，而 *P* 的右子树为 *P* 的左子树的最右边的叶节点的右子树。如图 5 所示，*P* 的右子树 *PR* 成为 *S* 的右子树，只需要改动节点 *F* 和节点 *S* 的指针即可。显然，这种情况下网络抖动量仅为 2。

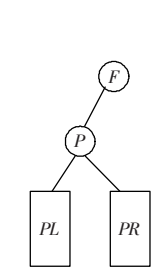


图 4　叶子节点退出

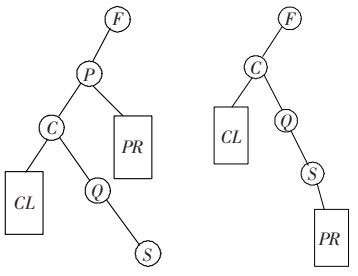


图 5　非叶子节点的退出

（4）BBSTC 拓扑的上层 Chord 的节点 *Nodei* 退出若其有左右子树，则其左右子树分别成为独立的二叉排序树，满足 *Nodei* 的最大数量等于 *refmax*。

5　资源搜索

通过前面的网络节点加入和网络节点退出算法已经形成了二叉排序树，而构建二叉排序树的目的就是为了便于资源的搜索。现将资源搜索算法描述如下：

通过上面的节点加入算法可知，在 BBSTC 网络的上层 Chord 环中每个 *Nodei* 节点负责管理一定区域的节点 IP 地址的哈希散列值，因为采用的是相容哈希函数 SHA-1 产生均匀的哈希值，又因为在事先已经设置好 *Nodei* 的最大值 *refmax*，所以每个 *Nodei* 节点的管理区间为 $2^m/refmax$。

每个节点可以清晰地知道自己在 BBSTC 网络的上层还是二叉排序树上，为了算法描述不失一般性，现将发起查询请求的节点指定为二叉排序树的中间节点。此节点首先将查询请求关键字哈希后的关键字标识符和该节点标识符比较大小，判断关键字标识符是否在该子树中，如果不在这个区间则要和根节点 *Nodei* 节点标识符作比较，判断是否在该节点所在的二叉排序树中。若在管理区间内则按照上面提到的二叉排序树的性质进行搜索，若请求关键字标识符超过了 *Nodei* 管理的区间范围，*Nodei* 将按照传统的 Chord 算法转发到下一个 Chord 节点 *Nodei*+1，收到搜索请求的根节点首先判断该请求关键字标识符是否在管理区间内，决定是否继续转发还是在以 *Nodei*+1 节点为根的二叉排序树中查找。

通过上面的描述可知资源查找主要发生在二叉排序树中，所以影响查找长度的因素主要在于二叉排序树的查找分析。假设在含有 $n(n>0)$ 个关键字的序列中，i 个关键字小于第一个关键字，$n-i-1$ 个关键字大于第一个关键字，则由此构造而得的二叉排序树在 n 个记录的查找概率相同的情况下，其平均查找长度为：

$$P(n,i)=\frac{1}{n}[1+i*(p(i)+1)+(n-i-1)(P(n-i-1)+1)]$$

其中，$P(i)$ 为含有 i 个节点的二叉排序树的平均查找长度，则 $P(i)+1$ 为查找左子树中每个关键字时所用比较次数的平均值，$P(n-i-1)+1$ 为查找右子树中每个关键字时所用比较次数的平均值。又假设表中 n 个关键字的排列是“随机”的，即任一个关键字在序列中将是第 1 个，或第 2 个，……，或第 n 个的概率相同，且 $P(0)=0$，$P(1)=1$，可推得：

$$P(n)\approx 2(1+\frac{1}{n})\ln n$$

由此可见,在随机的情况下,二叉排序树的平均查找长度和 $\log n$ 是等数量级的。

6 仿真实验

为了评估该文提出的策略,设计了仿真实验。实验所需网络拓扑用 BRITE 生成,仿真程序用 Java 编写。图 6 给出了 BBSTC 算法和 Chord 算法的比较图,网络节点也从 512 个节点开始一直增加到 8 192 个节点,分别比较这些状态下网络的平均延迟。从图中可以看出,BBSTC 算法比 Chord 算法的延迟要小,这是因为逻辑跳数相对减少了,这样延迟也相应地减少。

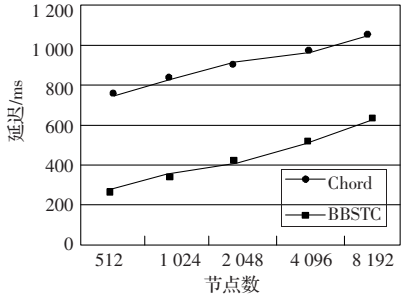


图 6 BBSTC 的平均延迟

图 7 给出了查找的成功率,网络中节点的个数从 100 到 1 000 变化,每次增加 100,通过仿真程序测试网络节点数量改变后,成功查找到资源的数目,为了使数据尽量真实,节点每变化一次,反复测试 10 次,记录每次得到的成功查找到资源的数目,求平均值作为最后的结果。从图中可以很容易看出,随着网络规模的增大,BBSTC 模型的查找成功率稳定在 85%。

7 结束语

提供了基于二叉排序树的结构化 P2P 覆盖网络拓扑

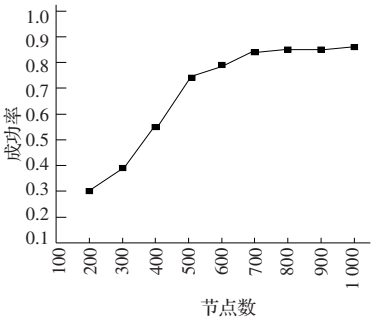


图 7 BBSTC 的查找成功率

BBSTC 的构造方法,详细分析了网络中节点的加入和退出的操作步骤,以及资源搜索的路由算法。通过仿真验证在资源搜索方面的性能优于传统的结构化网络,二叉排序树的网络拓扑结构的简单性决定了其应用布置的方便性和可用性,在目前结构化 P2P 环境中,该策略有一定的推广利用和研究价值。

参考文献:

[1] Qu C,Nejdl W,Kriesell M.Cayley DHTs-a group-theoretic framework for analyzing DHTS based on Cayley graphs[C]//Cao J.The Second International Symposium on Parallel and Distributed Processing and Applications.Berlin:Springer-Verlag Press,2004:914-925.

[2] Shi S.Making Peer-to-Peer keyword searching feasible using multi-level artitioning[C]//3rd International Workshop on Peer-to-Peer Systems,Feburuary 2004:400-408.

[3] PUB180-1 Secure Hash Standard[S].National Institute of Standards and Technology FIPS,2005-04-07.

[4] Stocia I,Morris R,Karger D,et al.Chord:A scalable peer-to-peer lookup service for Internet applications[J].Computer Communication Review,2001,31(4):149-160.

[5] 朱承,刘忠,张维明,等.结构化 P2P 网络中基于流言传播的负载均衡[J].通信学报,2004,25(4):31-40.

(上接 90 页)

$$\Phi\left(\sqrt{N}\cdot\sqrt{\frac{\rho_1^2+\rho_2^2}{1-(\rho_1^2+\rho_2^2)}}\right)=\Phi\left(\sqrt{N}\cdot\sqrt{\frac{\rho_0^2}{1-\rho_0^2}}\right)>$$

$$\Phi(\sqrt{N}\cdot|\rho_0|)\approx 99.8\%$$

(3)由于 $K_0\neq K, K_0\neq K'$,将 LA0 和 LA0 的逆分别用于算法 1 得到 26 比特密钥值后,利用 LA1、LA2 和算法 2 可得到另外 9 比特密钥值。所以,利用改进算法 2M 可以得到 35 比特密钥的值,减少了剩下需要穷举的密钥比特数。

3 结束语

给出了两个较好的线性逼近,它们的相关系数分别为最佳线性逼近相关系数的 0.8 倍和 0.6 倍,而且它们涉及完全相同的密钥但不同的明密文。结合多重线性逼近中的相应算法,给出了攻击 DES 的改进算法 2M。在等量明密文对的条件下,利用此改进算法 2M 可以多得到 9 比特密钥的值。

参考文献:

[1] Matsui M,Yamagishi A.A new method for known plaintext attack of FEAL cipher[C]//Advance in Cryptology Eurocrypt'92.Berlin:Springer-Verlag,1992:81-91.

[2] Matsui M.Linear cryptanalysis method for DES cipher[C]//Advance in Cryptology Eurocrypt'94.Berlin:Springer-Verlag,1994:398-409.

[3] Matsui M.The first experimental cryptanalysis of the data encryption standard[C]//Advance in Cryptology Crypto'94.Berlin:Springer-Verlag,1994:1-11.

[4] Kaliski Jr B S,Robshaw M J B.Linear cryptanalysis using multiple approximations[C]//Advance in Cryptology Crypto'94.Berlin:Springer-Verlag,1994:252-267.

[5] 冯登国,裴定一.密码学导引[M].北京:科学出版社,1999.

[6] Selcuk A A.On probability of success in linear and differential cryptanalysis [EB/OL].http://www.cs.bilkent.edu.tr/~selcuk/teaching/cs519/LC_DC.pdf.

[7] 吕述望,张如文.一类 Feistel 密码的线性分析[J].电子与信息学报,2003,25(9):1237-1242.

[8] 孙林红,叶顶锋,吕述望.多重线性密码分析的改进[J].通信学报,2002,23(5):83-88.

[9] 王建华,怀进鹏.多重线性密码分析中线性逼近方程的构造[J].计算机工程与应用,2007,43(8):118-120.

[10] 卫宏儒.RAINBOW 分组密码的线性密码分析[J].应用数学学报,2008,31(2):193-197.