文章编号: 1000-6893(2003)05-0447-05

SBN: 一种新的 Peer-to-Peer 覆盖网络构造协议

唐 焱, 胡正国
(西北工业大学 计算机科学与工程系, 陕西 西安 710072)

SBN: A New Peer-to-Peer Overlay Network Construction Protocol

TANG Yan, HU Zheng-guo
(Department of Computer Science and Engineering, Northwestern Polytechenical University, Xi'an 710072, China)

摘 要: 提出了一种新的动态模拟蝶形网络的 P2P(Peer to Peer)覆盖网络组织结构强蝶形网络 (Strong Butterfly Network, 简称 SBN), 论证了其基本的网络特性, SBN 可以以常数级的度达到对数级的路径长度, 或者增加到对数级的度达到接近优化的路径长度。与目前其他的 DHT(分布式哈西表) 相比, SBN 能够表现因特网的多样性, 并利用这种多样性提供更好的性能和数据可靠性。与同样是基于蝶形网络的 Viceroy 项目相比, SBN 具有较优异的性能, 同时更具有良好的可扩展性。

关键词: 分布式系统; 覆盖网络; 强蝶形网络; Peer to Peer; 多样性
中图分类号: TP31 文献标识码: A

Abstract: A novel way to simulate Butterfly network in the P2P overlay construction, named SBN(Strong Butterfly Network), is presented, and its basic network properties are discussed. SBN can be configured flexibly to be a constant degree network with logarithmic hop counts or near optimal hop counts network with logarithmic degree. And above all, SBN accords with the heterogeneousness of the Internet, and can take advantage of this heterogeneousness to improve performance and data reliability. Compared with Viceroy Project, SBN is better in performance and flexibility.

Key words: distributed system; overlay network; strong butterfly network; peer to peer; heterogeneous

目前流行的客户/服务器结构随着分布式技术和网络技术的进一步发展出现了新问题: 因特网用户规模的日益庞大, 服务器成为系统性能和安全性的瓶颈。DDoS(分布式拒绝服务) 攻击的出现, 加剧了这个矛盾, 同时服务器系统也导致了因特网中的负载不均衡。研究者希望在对 P2P 技术、算法的研究中找到解决上述问题的方案。

1 相关工作

覆盖网络是建立在已存在的一个或多个网络之上的一个间接的或者是可视化的抽象。利用覆盖网络, 可以不需修改已存在的软件协议和网络的底层结构而快速地添加新的网络功能。研究者们提出了很多种 P2P 覆盖网络的组织结构。比如著名的音乐交换软件 Napster 的中心索引服务器结构; 文件共享软件 Gnutella 的松散的纯分布式结构; Fasttrack 项目的超节点结构(纯分布式结构和服务器结构的混合体); Freenet 项目[1]的非结构化的 DHT; 以及现在的各种结构化的 DHT。P2P 网络的组织结构发生了质的变化。结构化的 DHT 具有纯分布式的特性, 可以避免由服务器结构带来的问题; 同时由于其结构化的特征, 可以避免非结构化的纯分布式系统的不确定性(Freenet) 以及对网络带宽的急剧消耗(Gnutella) 等问题, 从而扩展了 P2P 技术的应用范围。结构化 DHT 的代表性系统有加州大学伯克利分校的 CAN 项目和 Tapestry 项目; 麻省理工的 Chord 项目; 微软研究院的 Pastry 项目等。最近开始的研究有麻省理工的 IRIS 项目。目前设计的 DHT 均假设节点具有相同的能力。对于规模较小的系统来说, 这样的假设是非常合适的。但是, 因特网上的大规模 P2P 系统的特点是节点在权利上的平等性、目标系统规模巨大、高度动态性、高度多样性(节点的带宽、有效性、延迟存在 3 ~ 5 个数量级的差异[2])、以及节点自治和不可信任性。各个节点能力相等的假设并不适合这样的一个系统。本文提出了一种新的基于蝶形网络的能够捕捉因特网节点多样性的网络结构。Viceroy[3]项目中, 已经成功地将蝶形网络用于组

收稿日期: 2003-06-10; 修订日期: 2003-07-21
基金项目: 国家教育部博士点基金资助项目(20020699011)、国家自然科学基金(60073055) 资助项目
文章网址: http://www.hkxb.net.cn/hkxb/2003/05/0447/

建动态的 P2P 网络结构,实现了使用常数级的连接度达到 $O(\log_2 N)$ 的查询路径长度。Chord 的一个变种 Koorde[4] 则更进一步,通过在全局环中嵌入一个 de Bruijn 图,只使用 2 条边即可达到 $O(\log_2 N)$ 的查询路径长度,使用 k 条边可以达到 $O(\log_k N)$ 的查询路径长度。同时该文还证明了如果要网络能够具有常数级可能性的连接性,则必须有一些节点有 $O(\log_2 N)$ 条边。这意味着对于连接度的优化可以到此为止,在此基础上应进一步提高系统的性能和可靠性。SBN 在查询路径、连接度等方面具有优于 Viceory 的性能,不假设网络中的节点能力相等,更贴近因特网的实际情况,能体现因特网的多样性,且能利用这种多样性保障数据高可靠性和进行负载均衡。

2 设计目标和基本假设

目标是通过提出一种新的 P2P 虚拟网络构造协议来解决如下问题:

分布化:传统的客户/服务器系统在小型的分布式系统(如集群系统)中工作得很好,但仍然有很多缺陷。本文提出了一种新的纯分布式的 P2P 网络组织协议。

多样性:根据 Saroiu 等的报告[2],多样性广泛地存在于现实的 P2P 网络中。目前提出的协议相比之下仅仅抓住了当前因特网环境的一小部分特征。本文提出的强蝶形网络协议加入了对多样性这一重要特征的考虑,希望能够提高系统的性能和可靠性。

可扩展性:无需进行任何配置即可从少量的节点扩展到大量的节点。

对因特网的应用模型有基本假设:所有节点可以和其他任意节点建立连接。

节点有不同的能力,仅仅考虑其中最重要的带宽和节点在 P2P 网络中生存期这 2 个指标的多样性。

3 强蝶形网络协议的提出与分析

实际上,Viceroy 项目已经建立了动态的蝶形网络。本文方法与之相比有很多不同之处。希望能够表达因特网的多样性特征,因此使用了两维的空间来表达:一维表示节点的综合能力,称为节点的层级;另一维是键值空间,称为标识符。而 Viceroy 的协议与其他的协议一样使用了一维的平坦的键值空间。最重要的是,实际的系统能够利用在不同层级的节点共同维护相同的键值空间来构成一个小的机器簇,以提供极高的数据可靠性并且保持网络的稳定性。这是因为将节点按其生存概率分级之后,可以极大地减少高度动态性对网络稳定性的影响。如果采用为地址接近的节点分配接近的标识符的策略,本文算法就可以获得本地性,这种特性可极大提高系统的性能。

3.1 定义

定义 **1**(层级),节点的层级是以节点的网络带宽 B、节点在网络中生存期 T、节点的 CPU 能力 C、节点的内存容量 M、节点的存储能力 S 为自变量的函数 L。表示为 Level = $L(B, T, C, M, S)$。其中节点的网络带宽和节点在网络中存活期在因特网环境中是 2 个最重要的指标。因此在本文中做了适当的简化,只考虑 B, T 两个指标,即 Level = $L(B, T)$。节点的层级将会根据节点的行为变化。

定义 **2**(标识符),节点的标识符是节点 IP 地址用 SHA256 算法生成的 256 位的二进制数。

定义 **3**(PeerID),PeerID 是网络中节点的 < 层级,标识符> 对。

定义 **4**(全局键值空间),是 0~ $(2^{264}-1)$ 之间的所有整数构成的数值空间。

定义 **5**(节点负责的键值空间),是从节点在全局环中的前驱 PeerID 到节点自身 PeerID 的这个数值空间中所有的键值都由该节点负责。

定义 **6**(路径长度),用跳跃数目来表示节点之间的路径长度。

3.2 蝶形网络结构

蝶形网络最初用于交换网络和无阻塞网络。如果假设每个层级有 2^k 个节点,则一个蝶形网络中存在 $k+1$ 个层级。图 1 是一个 $k=3$ 的蝶形网络的例子。蝶形网络中的边由低级的层级指向高一级的层级。用符号 $< l, e_0 e_1 \cdots e_k >$ 表示每一个节点,其中 l 表示节点的层级,$e_0 e_1 \cdots e_k$ 表示节点的标识符。每个节点有 2 个蝶形边(层级 k 的节点除外)。如果 $l < k$,一条边指向 $< l+1, e_0 e_1 \cdots \underline{e_l} \cdots e_k >$,另一条边指向 $< l+1, e_0 e_1 \cdots \overline{e_l} \cdots e_k >$。从层级 0 节点到任何一个层级 k 的路径长度是 k。整个蝶形网络可以容纳 $N = (k+1)2^k$ 个节点。

定理 1 蝶形网络的直径小于等于 $5k/2$。

证明:首先从任何节点出发到达层级 k(或者层级 0)最多需要 $k/2$ 次跳跃,而从层级 k 的节点

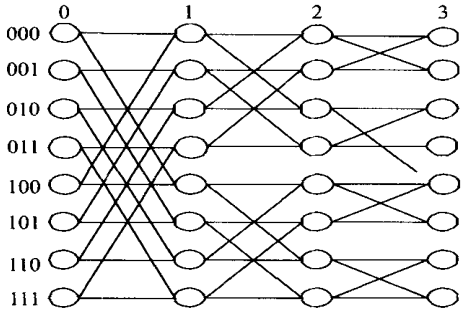

图 1　蝶形网络

Fig. 1　BN

到层级 0 的任何节点需要 k 次跳跃(从层级 0 的节点到层级 k 的任何节点也需要 k 次跳跃), 那么到达一个与目标节点的标识符相同的层级 0(或层级 k) 节点需要 k 次跳跃。最后从这个节点到达目标节点最多需要在不同的层级之间跳跃 k 次, 因此任何 2 个节点之间的最短距离小于等于 $5k/2$。

3.3　蝶形网络的变种

蝶形网络有多个变种, 如 Benes 网络、Multi-Butterfly 网络等。这 2 种网络本身都不适用于动态的、能够处理大量节点频繁的加入和退出的网络环境。Viceroy 项目构造了一个蝶形的变种, 它可以通过常数级的连接度达到接近 $O(\log_2 N)$ 跳跃数的目标。但是它和其他的协议一样, 不能表达因特网的多样性。本文提出了一种新的能够表现因特网多样性的动态蝶形网络结构, 它具有比 Viceroy 更好的网络特性。

假定每个层级的节点数为 2^k, 网络中总共有 k 个层级, 将层级 $k-1$ 的节点的蝶形边指向层级 0, 同时将每个层级的节点都连接起来构成一个全局环。一个 3×2^3 个节点构成的网络如图 2 所示, 图中省略了全局环的边。

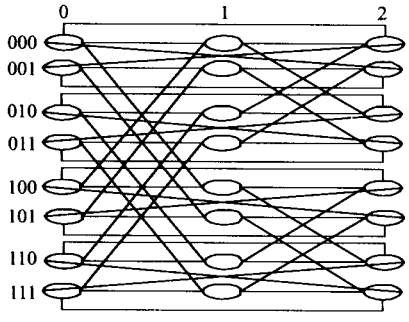

图 2　强蝶型网络

Fig. 2　SBN

定理 2　构造的蝶形变种网络的直径小于 $2k$。

证明: 显然, 从任何一个节点到达层级 0 最多需要 $k/2$ 次跳跃。从层级 0 出发到达任何一个与目的节点有相同的标识符的节点最多需要 $k-1$ 次跳跃, 从这个节点到达最终的目的节点最多需要 $k/2$ 次跳跃。所以任何 2 个节点之间的最短距离小于 $2k$。

3.4　强蝶形网络

这个蝶形网络的变种结构是可以扩展的。有 3 种方法对其进行扩展。

(1) 从层级 0 开始添加更多的维。从另一个角度看来, 网络就像一个星型结构, 当只添加一个维时, 网络便是 Benes 网络的扩展。这种方法成倍地增加了层级 0 节点的连接度, 但是构造的网络直径固定不变。其缺点是, 如果能够移除层级 0 的节点, 则可以成功地将网络分割开来。

(2) 对每个节点添加 $k-l-1$ 条边, 指向层级较高, 且具有相同的标识符的节点。这样, 使得任何 2 个标识符相同节点之间的跳跃数为 1。所以网络的直径小于等于 $k+2$。但是这样节点的连接度较大。

(3) 使用 $2^{m+1}-2$ 的出度达到小于等于 $2k/m$ 的网络直径。这种扩展方式是让节点 $<l, e_0e_1\cdots e_k>$ 建立 $2^{m+1}-2$ 条边到节点 $<(l+i)\bmod k, e_0e_1\cdots\overline{e_l}\cdots e_{l+i-1}\cdots e_k>$ …, 其中 $e_l\cdots e_{l+i-1}$ 这几位是 l 到 $l+m-1$ 共 m 位所表示的全部可能的数字。这样, 从任意节点可以一次匹配 m 位, 从而可以将跳跃点数减少到原来的 $1/m$。将这种扩展的蝶形网络称为 Strong Butterfly 网络, 记作 SBN(k, m), 其中 k 表示有 k 个层级, 每个层级有 2^k 个节点; m 表示每次可以匹配的位数。图 3 给出了一个 Strong Butterfly 网络中 $m=2$, 层级 l 的一个节点的连接情况。

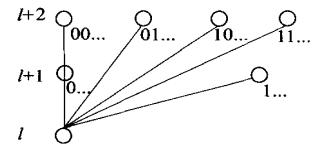

图 3　SBN 的扩展方法

Fig. 3　The method to extend SBN

定理 3　Strong Butterfly 网络的 SBN(k, m) 的直径小于等于 $2k/m$。

由前面的描述和定理 2, 该定理显然成立。

4　Strong butterfly 协议的基本算法

目标系统使用了 256 位标识符和 8 位层级。它将有足够的空间容纳当前和可预见的将来的全

部物理节点和可能存在的对象。数据键值是在数据上执行 SHA256 算法得到的一个 256 位的数值。发布数据时,在数据上执行 SHA256 算法,得到数据键值,将发布节点所处的层级作为其高位,并将其发布到键值空间包含这个键值的节点上。

4.1 节点加入操作

加入操作是构造 SBN 的第一步。如果是第一个加入网络中的节点,它将自动负责整个键值空间。假设已经存在一个已知的 SBN,那么一个新的节点加入到网络中来需要如图 4 的操作。

(1) 新节点 X 通过函数 $L(B, T)$ 计算初始的层级 l,其中参数 T 为 0,这意味着节点将会从高层级移动到合适的低层级;
(2) X 以自身 IP 地址为输入,通过 SHA256 算法生成标识符;
(3) 开始加入操作。X 首先将 JOIN 消息发送给任意一个已知的在网络中的节点 Y,称之为 Entry 节点。
(4) Y 节点将 JOIN 消息转发到适合 X 的位置 Z。这个位置就是当前负责的键值空间包含 X 节点的标识符,同时与 X 节点在同一个层级的节点。称之为 JOIN 节点。
(5) Z 节点将自己的 IP: Port 发回给 Y 节点,再由 Y 节点转给 X 节点。
(6) X 节点与 Z 节点建立直接连接,将自己的后继置为 Z 节点,前驱置为 Z 节点的前驱 P。Z 节点将自己的前驱置为 X 节点,让自己原来的前驱 P 将后继改为 X 节点。
(7) Z 节点将自己的键值空间按照 X 节点的标识符拆分成两个部分,并且将属于 X 节点键值空间中的数据转移给 X 节点。这样,X 节点就被加入到了全局环中。
(8) X 节点按照 SBN 协议建立其指向高层级节点的 Butterfly 边。
(9) X 节点查找一个层级比自己低,标识符大于自己的节点作为父(层级为 0 节点查找一个层级为 255 的节点为父)

图 4 节点加入操作

Fig. 4 Node joining operation

4.2 路由算法

提出两种路由算法: ①简单的贪婪算法,此算法只简单地选择离目标节点最近的边进行路由; ②Butterfly 算法,利用蝶形网的特性进行路由。

假设当前节点为 n,要查询的键值为 key,具体的贪婪算法 Greedy _ Lookup(k) 的伪代码如图 5 所示。贪婪算法在当前节点与目标节点的层级不同时每次能够逼近一个层级,所以这个过程最多进行 k 步;在当前节点与目标节点在同一个层级时,会通过前驱或者后继到达目标节点,这将需要 $O(\log_2 N)$ 步,最坏情况 $O(\log_2^2 N)$ 步。由于 $k < O(\log_2 N)$,所以贪婪算法的复杂度为 $O(\log_2 N)$,最坏情况 $O(\log_2^2 N)$。

```
Greedy _ Lookup(key)
        If key is contained in n
                retrun n;
        else
                for each node n_i linked with n
                        find the nearest n_i
                Greedy _Lookup(key);
```

图 5 贪婪算法

Fig. 5 Greedy algorithm

Butterfly 查询算法 Butterfly _ Lookup(key, t, status) 的伪代码如图 6 所示,其中: key 是要查询的键值; t 是一个初值等于 key 的中间变量,表明还剩多少位需要匹配; status 指明了当前是处在向层级 $k - l - 1$ 移动的过程(MOVE _ TO _ LEVEL),还是在向标识符逼近的过程(TRAVERS _ TREE),还是在已找到正确的标识符向正确的层级移动的过程(GET_TO_LEVEL), n 表示当前节点, SBN 构造参数为 $(k, 1)$。初始状态下, l 为满足目的节点的标识符与当前节点的标识符的距离 $\leqslant 2^l$ 的最小值, status = MOVE _TO _ LEVEL。

```
Butterfly _Lookup(key, t, status)
If k is contained in n
    return n;
else
    switch(staus)
    case MOVE _TO _LEVEL:
      if n.level! = k - l - 1
        find the lowest level node n_i in the deges
        n_i. Butterfly _Lookup(key, t, status);
      else
        staus = TRAVERS _TREE
        n. Butterfly _Lookup(key, t, status);
      break;
    case TRAVERS _TREE:
      if(n. id! = key. id)
        t = t ≪ k - l - 1;
        find the Butterfly edge goNext, which match the top l bits of t
        goNext. Butterfly _Lookup (key, t ≪ l, status);
      else
        staus = GET _TO _LEVEL
        n. Butterfly _Lookup(key, t, status);
      break;
    case GET _TO _LEVEL;
      Find the butterfly edge goNext, which is responsible for the key. ib
    goNext. Butterfly _Lookup (key, t, status);
      break;
```

图 6 Butterfly 算法

Fig. 6 Butterfly algorithm

对于一个理想状况下的网络,即 $k2^k$ 个节点全部存在的网络, Butterfly 路由算法必能在 $2k$ 的路径长度内找到目标节点。但实际环境只存在少量的节点,是一个稀疏的网络。在移动到层级 $k - l - 1$ 的过程中,最多需要 $k/2$ 步;在寻找匹配的标识符的过程中,每次能保证匹配 1 位,所以能在 k 步内达到找到匹配的标识符。而第 3 步移动到正确的标识符和正确的层级的过程中,可以证明其复

杂度为 $O(\log_2 N)$, 最坏情况 $O(\log_2^2 N)$。

5　性能评价

构造了一个验证 SBN 算法的原型系统, 它可以构造动态 SBN 网络, 支持多种路由算法进行键值查询。获取数据的方法是, 系统随机生成指定数目的节点, 组建 SBN, 随机地挑选查询的键值进行查询操作, 并进行统计。图 7 的数据为贪婪算法当 $k = 27$, $m = 1, 2, 3$ 时, 在 500 个随机生成的节点的网络中进行 500 次随机查询的结果。

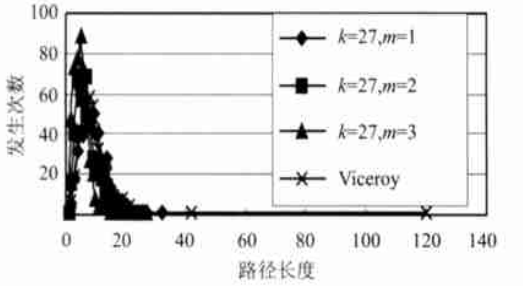

图 7　SBN 与 Viceroy 的路径长度

Fig. 7　Hop counts of SBN and Viceroy

可以看出随着 m 的增加, 查询路径的长度明显减小。在 $m = 1$ 时, 路径长度平均值为 9.424; $m = 2$ 时为 7.352; $m = 3$ 时为 5.63。而 Viceroy 在贪婪算法下的平均路径长度是 9.006。从图中还可以看出贪婪算法在 SBN 的性能, 在 $m = 1$ 时尽管平均值大于 Viceroy, 但是查询路径长度的上边界明显; 在 $m = 2$ 和 $m = 3$ 时则明显优于 Viceroy。总的来说 SBN 的性能优于 Viceroy。图 8 表明SBN 在随着网络规模大小的增加, 贪婪算法查询路径长度的变化状况, 其中 $m = 1$。图中表示了不同的节点数量的情况下, 路径长度的上下界和平均路径长度。图 9 表示了 Viceroy 在相同条件下路径长度变化的情况。可以看到, Viceroy 在贪婪算法下查询路径长度的上界不稳定, 可能出现非常坏的情况。而 SBN 的上界稳定地落在其理论边界以内。

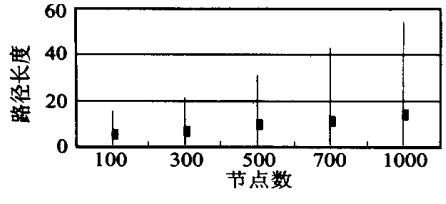

图 8　网络规模增加对 SBN 路径长度的影响

Fig. 8　Hop counts of SBN as nodes number increase

6　结论和工作展望

从前面的论述和初步试验结果可以看到,

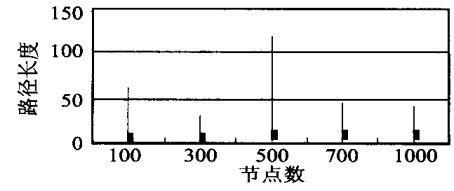

图 9　网络规模增加对 Viceory 路径长度的影响

Fig. 9　Hop counts of Viceroy as nodes number increase

SBN 具有良好的网络特性: SBN 是一个纯分布式的网络组织结构, 具有良好的可伸缩性; 从查询路径来衡量, SBN 的性能要明显优于 Viceroy。

在下一步的工作中, 将进一步完善原型系统, 取得 Butterfly _ Lookup 算法模拟结果; 讨论节点层级的自适应调整算法; 应用概率模式进行更接近因特网实际环境的模拟, 并与 Koorde 和 Viceroy 进行进一步的比较; 同时讨论 SBN 的其他特性: 负载均衡能力、数据可靠性、网络的容错能力等等; 以及进一步加强假设条件, 引入对防火墙、动态 IP 地址等条件进行分析。

参　考　文　献

[1] Clarke I, Sandberg O, Wiley Brandon, *et al*. Freenet: a distributed anonymous information storage and retrieval system [A]. In: Proceedings of Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, LNCS2009 [C]. Heidelberg: Springer Verlag, 2001: 46- 66.

[2] Saroiu S, Gummadi P K, Gribble S D. A measurement study of Peer to Peer file sharing systems [R]. Technical Report UW-CSE-01-06-02, Seattle: Department of Computer Science and Engineering, University of Washington, 2001.

[3] Malkhi D, Naor M, Ratajczak D. Viceroy: a scalable and dynamic emulation of the butterfly [A]. In: Proceedings of the 21st ACM Symposium on Principles of Distributed Computing [C]. New York: ACM Press, 2002: 183- 192.

[4] Kaashoek F, Karger D R. Koorde: a simple degree-optimal hash table [A]. In: Kaashoek F. ed. Proceedings of IPTPS03, Peer-to-Peer Systems Ⅱ, LNCS2735 [C]. Heidelberg: Springer Verlag, 2003: 43- 48.

作者简介:

唐　焱(1976-)　男, 四川攀枝花人, 在读博士, 1998 年毕业于西北工业大学计算机软件专业, 2000 年取得该专业硕士学位。主要研究方向为分布式系统, P2P 计算环境, 中间件技术。Email: tangyan@cotthink.com。

胡正国(1939-)　男, 西北工业大学计算机科学与工程系教授, 博士生导师, 1961 年毕业于西北工业大学计算数学专业。主要研究方向为软件工程及数据库技术。

(责任编辑: 俞　敏)