

# 一种用于 VoIP 的负载平衡对等网络架构

张建东<sup>1,2</sup>,王劲林<sup>2</sup>

ZHANG Jian-dong<sup>1,2</sup>,WANG Jin-lin<sup>2</sup>

1.中国科学院 研究生院,北京 100080

2.中国科学院 声学研究所,北京 100080

1.Graduate School of the Chinese Academy of Sciences,Beijing 100080,China

2.Institute of Acoustic,Chinese Academy of Sciences,Beijing 100080,China

E-mail:zhangjd@dsp.ac.cn

**ZHANG Jian-dong,WANG Jin-lin.Load balancing Peer-to-Peer network architecture for VoIP applications.Computer Engineering and Applications,2007,43(12):17-19.**

**Abstract:** This paper presents a load balancing peer-to-peer network architecture for VoIP applications,which is based on Chord protocol,and balances the user registrations among nodes by means of changing node identification dynamically.According to two thresholds of user registration loads,nodes check and balance user registration loads with two different methods,therefore network bandwidth overhead are reduced by depressing the frequency of loads transferring between nodes. Simulation result shows that,our peer-to-peer network architecture has less maximum load of single node and load mean square deviation of overall nodes than Chord.Peer-to-peer network implemented with the architecture in this paper,employ less resources in single node,and is suitable for deployment on embedded VoIP terminals with limited resource.

**Key words:** Peer-to-Peer network;load balancing;VoIP

**摘要:**针对 VoIP 应用,提出了一种负载平衡的对等网络架构。该架构基于 Chord 对等网络,采用动态更改节点标识的方法平衡各个节点上的注册用户负载。根据两个阈值,监测节点的注册用户负载量,根据负载量使用两种方法实现节点间负载的平衡,以减少节点间负载的转移对网络带宽的额外占用。仿真实验结果表明,单个节点的最大注册用户负载量和节点负载量均方差均小于 Chord 对等网络。基于该架构实现的对等网络,单个节点资源占用少,适于在资源受限的嵌入式 VoIP 终端上实现。

**关键词:**对等网络;负载平衡;VoIP

文章编号:1002-8331(2007)12-0017-03 文献标识码:A 中图分类号:TP393

## 1 引言

近年来,VoIP 服务迅速发展并得到了广泛的应用。传统的 VoIP 服务采用服务器/客户端的架构实现,用户使用客户端(基于 PC 的软件电话或专用的 VoIP 终端),使用服务提供商的 VoIP 服务器实现注册、呼叫和通话等业务。但是,由于采用了服务器/客户端架构,传统的 VoIP 服务扩展性和鲁棒性都比较差,在某些场景下无法应用,例如没有因特网连接的情况,或是注重安全却无法承担搭建自有 VoIP 服务费用的小型组织。

对等网络在扩展性和鲁棒性方面有着很大的优势。使用对等网络承载 VoIP 服务,利用构成对等网络的各个节点的资源完成 VoIP 服务器的功能,可以提高 VoIP 服务的扩展性和鲁棒性,并且能够降低应用 VoIP 服务的成本。

在嵌入式 VoIP 终端间构建对等网络时,由于资源受限,因此,需要保证分布在各个嵌入式 VoIP 终端中注册用户负载的平衡。结构化的对等网络,一般使用分布式哈希表(Distributed Hash Table,DHT)实现构成对等网络的节点中资源的分布式利用<sup>[1-4]</sup>。分布式哈希表使用下面两种手段保证负载平衡:(1)

使用足够好的哈希函数随机化关联目标资源的地址;(2)利用每个节点负责分布式哈希表地址空间中的均分部分。但是,在对等网络节点稀疏时,由于地址空间上的各个节点所处的位置并非完全平衡的,因此,某些节点可能会负责一大部分地址空间,从而造成资源目标负载的不平衡。为了解决节点稀疏带来的负载不平衡问题,许多 DHT 算法采用虚拟节点的方法,在一台终端上虚拟多个节点,终端的负载就是所有虚拟节点负载的总和,达到近似于平均负载的水平。在一个具有  $N$  个节点的 Chord 对等网络中,采用一致哈希算法,使用虚拟节点的方法,每个终端虚拟  $O(\log N)$  个节点,可以使得单个节点的负载以很大概率趋近于平均水平。终端中虚拟多个节点,需要维护的信息则成倍增加,如 finger 表、前驱后继节点等,因此需要更多的存储资源,占用更大的网络带宽。D. Karger 等采用仅激活多个虚拟节点中的一个的方法,以减少维护信息的开销,但其局部优化算法仅能提高节点在地址空间中分布的平衡程度,并不能保证目标资源负载在各个节点上的平衡;另外,维护、检测最优的虚拟节点与虚拟节点切换等操作,将带来系统资源和网络带

基金项目:国家发改委下一代互联网示范工程 2005 项目(No.CNGI-04-15-2A)。

作者简介:张建东(1978-),男,博士研究生,主要研究方向为多媒体通信;王劲林,研究员,博导。

宽的额外开销<sup>[5]</sup>。D. Karger 等也提出动态更换节点标识的方法来平衡 P2P 网络中的负载,但其节点标识动态切换的方式会造成对等网络结构的变化,并且没有提到负载不均衡的具体发现机制。

本文针对 VoIP(Voice over Internet Protocol)应用的特点,提出了一种用于 VoIP 的负载均衡对等网络架构。该架构基于 Chord 对等网络架构,使用动态更换节点标识的方式平衡负载,采用两种负载量阈值探测节点负载平衡程度,并控制负载均衡的频度,降低网络带宽的额外开销。以本文提出的架构实现的,需要维护的信息与不使用虚拟节点的 Chord 对等网络相同,特别适于在资源受限的嵌入式 VoIP 终端上实现。

## 2 负载均衡对等网络架构

在 VoIP 应用中,一般情况下,一个终端对应一个电话号码;使用电话号码或相应的用户名,注册到注册服务器上;根据电话号码,通过注册服务器向其他终端发起呼叫或实现其他业务。用于 VoIP 的对等网络,注册服务器的功能可以分布到各个终端节点中实现,每个终端节点向部分节点提供注册服务,实现呼叫、通话等其他业务。由于每个终端节点上,同一时间,很大概率都只有一个用户注册到整个系统,因此,用于 VoIP 的对等网络,每个终端节点上面的负载,即注册到该节点的用户,平均值在 1 个用户左右。

本文提出的对等网络架构,基于 Chord 对等网络,也采用一致哈希算法,每个终端节点上负责向映射到其节点标识的用户提供注册与呼叫等功能;并采用动态更改终端节点标识的方法平衡负载,针对 Chord 对等网络做了如下的改进:(1)节点加入时,根据当前对等网络部分节点的注册用户负载情况,选择有利于平衡负载的位置加入,并以该位置作为节点标识;(2)当节点探测到自身的注册用户负载过大时,使用负载均衡算法,将过多的注册用户转移到临近节点上,并相应的更改自身的节点标识。本文提出的对等网络架构,设计为容纳  $2^m$  个节点,  $2^{mk}$  个用户;其中,  $k$  可以为大于 1 的整数,本文实验中选择为 2,提高  $k$  值可以减少哈希算法产生冲突的概率。

### 2.1 节点加入的改进

每个终端节点加入前,需要通过某种方式找到对等网络上某一终端节点,例如通过预设地址、缓存地址或是固定地址等;这一节点一般称为自举点。在找到了自举点之后,获取该节点的以下信息:

- (1)所有 finger 表中节点信息;
- (2)最佳加入位置;
- (3)加入后用户负载转移量。

而后,向自举点 finger 表中各节点发送请求加入信息,获取以下信息:

- (1)最佳加入位置;
- (2)加入后用户负载转移量。

每个节点提供的最佳加入位置与加入后用户注册负载的转移量,根据自身情况计算得出。如果注册到某一节点用户负载数目为  $n$ ,按照环序由小到大排列,每个用户负载映射的节点标识为  $ID(i)$ ,则最佳加入位置通过以下方式计算:

(1)如果前驱节点标识为本节点标识减 1,则该节点与前驱节点之间无法加入其他节点,因此无最佳加入位置,用户负载移动量为 0;

(2)选取候选加入位置  $p$  满足(环序):

$$ID(\lfloor n/2 \rfloor) \leq p < ID(\lfloor n/2 \rfloor + 1)$$

(3)如果  $p$  不等于本节点标识,则最佳加入位置为  $p$ ;否则,最佳加入位置为本节点标识减 1;

(4)根据最佳加入位置,查找本节点用户负载列表中映射于该标识的用户负载,并记录负载数目为用户负载转移量。

待加入节点获取到以上信息后,选取加入后用户负载转移量最大的那个节点作为加入点;如果有两个或多个节点同时具有最大的加入后用户负载转移量,那么将随机选取其中的一个作为加入点;如果不存在具有最佳加入位置的节点,那么,则选取对等网络中另外一个节点作为自举点,重新进行加入点的选取。

选取了加入点后,待加入节点作为加入点的前驱节点加入对等网络中,对应的最佳加入位置作为自身的节点标识。加入对等网络的步骤与 Chord 对等网络的加入操作类似,与 Chord 网络不同的是,加入了通知前驱节点更新后继节点列表的步骤,以保证对等网络的环结构上,节点前后关系的完整性。

### 2.2 负载发现与转移算法

本文提出的对等网络架构,使用两个阈值,  $T_1$  和  $T_2$ ,度量单个节点的注册用户负载量状态。 $T_1$  表示当前负载处于较低水平及以下,  $T_2$  表示当前负载尚处于可接收范围,超过  $T_2$  的负载值则表明系统负载严重不平衡。这两个阈值用于发现负载不平衡,并控制相应的负载转移方式,一般情况下,  $T_1$  应设置为  $2^k$ ,而  $T_2$  则可根据节点的存储能力决定。转移负载时将带来网络带宽的额外开销,因此,需要选取适当的阈值,以减少负载转移操作的频率。

每当节点加入时,整个对等网络的用户负载将会增加。当用户负载不平衡时,通过动态更改节点标识、转移用户负载至前驱或后继节点,完成负载均衡操作。节点负载的平衡有如下两种方式:

(1)局部平均:根据节点自身用户负载情况,综合前驱节点和后继节点的情况,将用户负载平均分布在这三个节点上,可能需要更改自身和前驱节点的节点标识;

(2)范围传递:在前驱节点和后继节点用户负载量也比较大的情况下,局部转移将不能解决这三个节点用户负载的不平衡问题,因此需要从向前和向后两个方向传递用户负载,直到不再探测到存在高用户负载的节点后停止。为了避免重复转移,除了发起范围传递操作的节点外,其他节点应仅进行单向的负载传递。

两种负载均衡的方法将带来网络带宽的额外开销,尤其是执行范围传递操作时。为了减少由于平衡负载带来的额外开销,需要降低平衡操作执行的频度。因此,节点根据自身负载量与前面提到的两个阈值之间的关系,决定是否进行负载转移以及转移的方式。节点的负载均衡算法如下:

- (1)当自身负载超过  $T_1$  时,尝试局部平均负载转移操作;
- (2)当自身负载超过  $T_2$  时,执行范围传递负载转移操作。

两种负载均衡操作涉及节点标识的更改,因此,对等网络中部分节点的 finger 表也相应的出现了改变。某一节点如果由于平衡负载更改了自身节点标识,在其接收到查询请求时,除做相应的回应外,还需要将自身节点标识捎带回发出查询请求的节点,使查询节点更正原有 finger 表对应表项。而节点对 finger 表的定期维护,将解决各表项与预想条目不符的问题。转

移了部分用户负载后,这些用户所属的节点并不知道其用户对应的注册节点已经改变,但此时无需通知用户所属的节点更改注册节点信息,待其下次注册时,由原注册节点反馈重新注册的信息,通知用户所属的节点初始化注册操作即可。

2.2.1 节点负载局部平均算法

局部转移算法如下:

- (1)获取前驱节点和后继节点的用户负载情况;
- (2)如果前驱节点和后继节点的用户负载均少于  $T_1$ ,则计算三者用户负载的平均值;根据平均值计算前驱节点和后继节点应该接收的用户负载数目  $L_1$  和  $L_2$ ;向前驱节点转移  $L_1$  个用户负载,向后继节点转移  $L_2$  个用户负载;根据负载转移情况,调整自身节点标识;前驱节点根据负载转移情况,调整其节点标识;
- (3)如果前驱节点的负载大于等于  $T_1$ ,则仅向后继节点转移用户负载;计算本节点与后继节点用户负载的平均值;根据平均值计算后继节点应该接收的用户负载数目  $L_2$ ; 向后继节点转移  $L_2$  个用户负载;根据负载转移情况,调整自身的节点标识;
- (4)如果后继节点的负载大于等于  $T_1$ ,则仅向前驱节点转移用户负载;计算本节点与前驱节点用户负载的平均值;根据平均值计算前驱节点应该接收的用户负载数目  $L_1$ ; 向前驱节点转移  $L_1$  个用户负载;前驱节点根据负载转移情况,调整其节点标识。

2.2.2 节点负载范围传递算法

范围传递算法如下:

- (1)如果由自身发起,则:
  - (1.1)获取前驱节点和后继节点的用户负载情况;
  - (1.2)计算自身节点负载数目减去  $T_1$  与前驱和后继节点负载数目的总和的 1/2; 根据该值计算使得当前节点负载量降低到  $T_1$  时前驱节点和后继节点需要接收的用户负载量  $L_1$  和  $L_2$ ;
  - (1.3)向前驱节点转移  $L_1$  个用户负载,向后继节点转移  $L_2$  个用户负载;根据负载转移情况,调整自身节点标识;前驱节点根据负载转移情况,调整其节点标识;
  - (1.4)前驱节点执行前向范围传递,后继节点执行后向范围传递。
- (2)如果由后继节点发起的前向大范围传递,则:
  - (2.1)获取前驱节点的用户负载情况;
  - (2.2)计算使得当前节点负载量降低到  $T_1$  时前驱节点需要接收的负载量  $L_1$ ;
  - (2.3)向前驱节点转移  $L_1$  个用户负载;前驱节点根据负载转移情况,调整其节点标识;
  - (2.4)前驱节点执行前向范围传递。
- (3)如果由前驱节点发起的后向范围传递,则:
  - (3.1)获取后继节点的用户负载情况;
  - (3.2)计算使得当前节点负载量降低到  $T_1$  时后继节点需要接收的负载量  $L_2$ ;
  - (3.3)向后继节点转移  $L_2$  个用户负载;根据负载转移情况,调整自身节点标识;
  - (3.4)后继节点执行后向范围传递。

3 仿真实验

本文就提出的对等网络架构,与 Chord 对等网络做了对比

仿真实验。在实验中,最大容纳的节点数目为 16 384 个,注册用户最大数目为 65 536 个。采用 SHA-1 哈希算法计算用户名对应的值,用户名采用电子邮件地址格式,随机生成。基于 Chord 对等网络的仿真实验中,节点标识根据自身 IP 地址和端口地址也采用 SHA-1 算法计算,而节点的 IP 地址与端口地址使用随机方式产生。基于本文实现的仿真实验中, $T_1$  选择为 4, $T_2$  选择为 8。在两个对等网络的仿真实验中,节点加入的时间随机生成;节点的持续时间按照 Pareto 分布随机产生。实验分别仿真 Chord 对等网络和本文提出的对等网络架构在一段时间内的节点加入、离开过程,按照固定时间间隔,查看对等网络系统中节点的最大注册用户负载量,以及系统各节点负载量的均方差,作为评估负载平衡的参数。

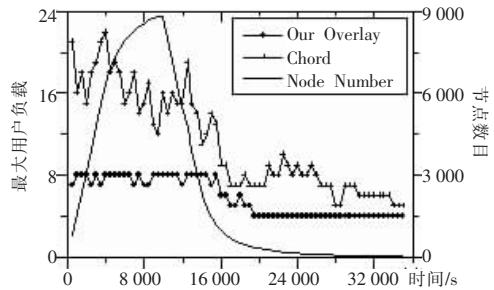


图1 最大负载对比图

图 1 为 Chord 对等网络和本文提出的对等网络架构的单个节点的最大注册用户负载量对比图。从时间的 0 时刻开始,随着节点的加入,组成对等网络节点的总数目开始增加;加入的节点在其在线时间结束后,离开对等网络。由图 1 可以看出,节点数目达到接近 9 000 时,节点总数目开始减少;Chord 对等网络的单个节点最大用户负载量在节点总数目未达到一定数量时,单个节点的最大负载量处于较高的水平,最大到达 22 个;随着节点数目增加,单个节点的最大负载量有所下降;而当节点总数目减少时,单个节点的最大负载量又有所增加;在节点数目非常稀疏时,单个节点的负载量降低到 10 以内。出现这种情况的原因,首先是哈希算法能够保证一定程度的负载平衡,因此,当节点数目增加时,整个网络中单个节点的最大负载量相应降低;而当网络中节点数目少到一定程度时,由于应用的是 VoIP 服务,用户注册负载与网络中在线的节点数目相当,在这种情况下,哈希算法也能够保证节点间一定程度的负载平衡。而本文提出的对等网络架构,随着节点数目的增加和减少,单个节点的最大负载量都控制在阈值  $T_2$  及以下,即少于或等于 8;当节点数目非常稀疏时,单个节点的最大负载量则降低到阈值  $T_1$  左右,即 4 个左右。在同一时刻,Chord 对等网络单个节点的最大负载量,均大于本文提出的对等网络架构。

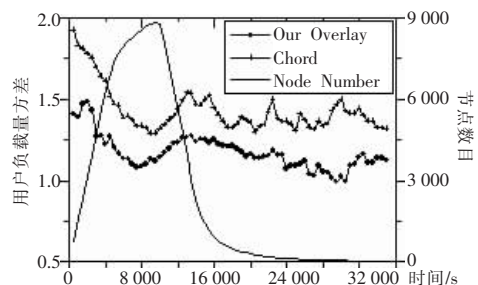


图2 负载量均方差对比图