

文章编号:1001-9081(2006)11-2558-03

即时通信中一种新的群拓扑结构——三层 K 叉树

韦健鹏, 吕光宏

(四川大学 计算机学院, 四川 成都 610065)

(wei_jianpeng@163.com)

摘要:目前一些主流即时通信软件已经嵌入了聊天群,但都过分地依赖中心服务器,使得聊天群的应用受到限制。给出了聊天群的一种新的拓扑结构——三层 K 叉树,将中心服务器的负载分担到一些选定的即时通信客户端,以减轻中心服务器的负载,增强聊天群的扩展性。分析和仿真结果表明三层 K 叉树在不降低聊天群性能的情况下确能达到以上目的。

关键词:三层 K 叉树;聊天群;负载;即时通信

中图分类号: TP393.04 **文献标识码:** A

Three-tier K-ary tree: A new group topology architecture in instant messaging

WEI Jian-peng, Lü Guang-hong

(School of Computer Science, Sichuan University, Chengdu Sichuan 610065, China)

Abstract: As a new instant messaging (IM) technology, chat group has been developed to resolve multi-communication problems. Some popular IM applications, such as MSN and QQ, have been integrated with the latest products. However, they are so dependent on central server that their applied domain is difficult to enlarge. A new chat group topology architecture, 3-tier K-ary tree, was proposed. It distributed the payload of central server among some selected clients with some specific rules so as to alleviate central server payload and enhance the chat group scalability. The analysis and simulation show that 3-tier K-ary can achieve the expected aims above and maintain the performance of the chat group at the same time.

Key words: 3-tier K-ary tree; chat group; payload; Instant Messaging(IM)

0 引言

即时通信(Instant Messaging, IM)是一种能使两个异地、联网的用户实时地发送短信息的应用技术^[1]。它是 Internet 用户进行交互的主要工具,目前比较受欢迎的即时通信软件有 ICQ, MSN, QQ 等。

群聊天是 IM 的一项重要功能,IM 领域至今还没有聊天群的明确定义,在表现形式上,聊天群(本文将简称为群)是一种具有名称、讨论主题及加入规则的虚拟社团,它为多人交互提供了平台。群的应用非常广泛,如网络授课、网络讨论、网络会议、网络拍卖等。普通聊天是单点对单点的交互,而群是单点对多点的交互。当前很多 IM 协议都支持群,较为著名的有 MSN 协议和 QQ 协议。

1 相关研究

MSN 协议^[2]是微软公司开发的一种即时通信协议。MSN 服务器架构如图 1 所示。MSN 主要依靠两种服务器来提供服务,一种是 NS(Notification Server),负责管理用户的状态信息。另一种是 SB(Switchboard),负责管理交互过程的“会话(Session)”。用户之间的交互通过“会话”机制实现,一个“会话”可以接纳两个或更多的用户,这使得普通聊天和群能有机融合于 MSN 架构。当进行普通聊天时,“会话”只允许添加两个用户,而当进行群交互时,“会话”允许添加所有的群用户。

QQ 协议是腾讯公司开发的一种商业性的通信协议。从

已被译的 QQ 协议^[3]上分析,其服务器架构如图 2。QQ 协议中的群一共使用了两种服务器:一种是群服务器,负责管理群的信息和维护消息分发过程;另一种是中转服务器,负责群消息的转发,维护群成员的状态。群服务器使用了集群技术。集群^[4]是一组计算机,它们作为一个整体向用户提供一组网络资源。集群技术能有效地平衡服务器的负载,增加服务器的可靠性。

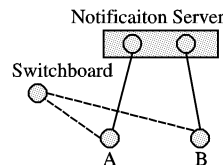


图 1 MSN 的网络拓扑

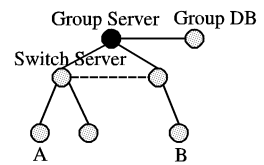


图 2 QQ 的网络拓扑

2 三层 K 叉树的拓扑结构

群分为临时群和永久群。MSN 的群是一种临时群,群的生命周期取决于在线成员的在线时间,如果群中所有成员下线,群的生命也随之结束。QQ 的群则是一种永久群,群的信息被服务器永久保存,不受群成员状态变化的影响。永久群拥有强大的功能,是未来发展的方向,但永久群的信息维护却给服务器带来了巨大压力。本文中的群特指永久群。

2.1 当前 IM 技术中群的优缺点

“会话”具有临时性,只有在用户有交互需求时才建立,当用户全部退出或在一段时间内未进行交互,服务器将销毁“会话”。“会话”机制使服务器对资源的利用非常高效,但却

收稿日期:2006-05-29;修订日期:2006-07-20

作者简介:韦健鹏(1981-),男,广西贵港人,硕士研究生,主要研究方向:计算机网络、信息系统;吕光宏(1963-),男,四川成都人,教授,主要研究方向:计算机网络、信息系统。

使得永久群的实现变得很困难。SB 的分散性和可选性使它不适合保存群的信息,“会话”的独立性和灵活性使得服务器很难对群状态进行维护。

QQ 协议中的群信息和群的状态只由一个群服务器管理,可轻易地解决永久信息的保存和状态的维护问题。群服务器是 QQ 群架构的中枢,群的所有信息必须首先发送到群服务器,由群服务器进行处理。群服务器的硬件和网络性能将决定群的规模,群的扩展性将是其面对的主要问题。QQ 协议目前采取限制群的数量和群的人数来缓解群服务器的压力。

XMPP 是 IETF 开发的一种协议^[5],试图对即时通信协议进行标准化,其服务器架构^[6]关注于兼容各种 IM 协议而未考虑群,没有一个中心服务器,从而使得实现永久群十分困难。

本文尝试通过使用一种新的群拓扑结构——三层 K 叉树,来解决这些问题。

2.2 三层 K 叉树中四种类型的节点

三层 K 叉树架构有四种类型的网络节点:中心服务器(Central Server, CS)、群服务器(Group Server, GS)、备份群服务器(Backup Group Server, BGS)和群成员(Group Member, GM)。中心服务器的主要功能为登录验证,任命群服务器,保存群信息,系统消息广播等。群成员由群成员的客户端及其连接的转发服务器构成,群服务器和备份群服务器从群成员中选取。群服务器的主要功能为群信息转发,群成员的状态维护,备份群服务器的选取与管理,向中心服务器提交群的永久信息等。备份群服务器作为群服务器的候选者,一旦群服务器失败或放弃服务,将立即“升职”为群服务器,承担群服务器的任务。图 3 是三层 K 叉树的拓扑图,一个中心服务器可管理 K 个群服务器,而群服务器管理 K 个群成员,其中有一个特殊的群成员——备份群服务器。

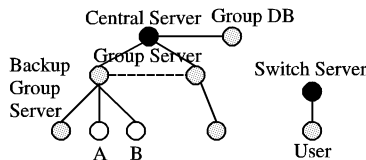


图 3 三阶 K 叉树的拓扑结构

每个客户端绑定一个转发服务器,使三层 K 叉树可进行普通的点对点交互。如果把群成员看成是一个客户端与转发服务器对,MSN 协议、QQ 协议等目前主流 IM 协议中的服务器架构则是一种二层 K 叉树架构。

2.3 三层 K 叉树的结构和特性

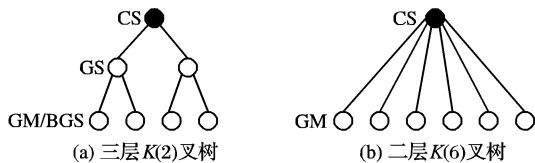


图 4 IM 中的树

图 4(a) 是三层 K 叉树的示意图。在三层 K 叉树中,第 i 层的节点管理 K 个第 $i+1$ 层节点,在此模型下,中心服务器能管理的节点数为: $N = K^2$ 。传统的群模型如图 4(b) 图所示,是一个二层 K 叉树,根节点是中心群服务器,功能相当于三层 K 叉树的中心服务器和群服务器,此时一个中心服务器能管理的群节点数为: $N' = K$ 。因此,三层 K 叉树下一个中心服务器管理的用户量最大可达到传统架构下的 K 倍。

假设每个群成员占用的网络带宽为 $Conn(gm_i)$,中心服务器管理 m 个群,每个群的成员数为 n 。当采用二层 K 叉树时,

中心服务器所需的网络带宽为 $Conn(CS) = m * n * Conn(gm_i)$ 。当采用三层 K 叉树时,中心服务器的网络负担被平分到群服务器上,此时一个群与中心服务器的通信量约相当于一个群成员与群服务器通信量的 γ 倍($\gamma > 1, \gamma \ll n$),中心服务器所需的网络带宽为 $Conn'(CS) = m * \gamma * Conn(gm_i)$,是二层 K 叉树架构的 γ/n 。

3 三层 K 叉树群结构中的关键技术

3.1 备份群服务器的选取策略

群信息的转发,群状态的维护都由群服务器来完成,因此群服务器必须拥有较优的硬件条件、较轻的负载和将会有较长的在线时间,在选取备份群服务器必须考虑这些因素。假设群成员 gm_i 的资源信息为:CPU 运行能力为 $CPU(gm_i)$,存储能力为 $Mem(gm_i)$,网络带宽为 $Conn(gm_i)$;已在线时间为 $OnlineT(gm_i)$,群成员的平均在线时间 $AvgOnlineT(gm_i)$,在线时间阈值为 $ThreshOnlineT(gm_i)$ 。群成员的在线时间因子 $OnlineTfactor(gm_i)$ 由公式(1)可求出。

$$OnlineTfactor(x) = \begin{cases} 0, & OnlineT(x) \geq AvgOnlineT(x) * ThreshOnlineT(x) \\ 1 - \frac{OnlineT(x)}{AvgOnlineT(x)}, & OnlineT(x) < AvgOnlineT(x) * ThreshOnlineT(x) \end{cases} \quad (1)$$

式中当群成员的已在线时间超过在线时间阈值时,极有可能随时离线,不选取作备份群服务器,以减少备份群服务器的频繁更替。而对于那些已在线的时间远小于在线时间门限的 gm_i ,表明其将在线时间越长,被选作备份群服务器的可能性越大。

$$Weight(gm_i) = w1 * f(CPU(gm_i)) + w2 * f(Mem(gm_i)) + w3 * f(Conn(gm_i)) + w4 * f(OnlineTfactor(gm_i)) \quad (2)$$

其中 $f(x) = 1 - (x + 1 - X_{min})^{-d}$, X_{min} 表示 x 能取的最小值, d 表示变化强度。

公式(2)改进了传统的权值计算公式, $w1, w2, w3, w4$ 是程序依据一定的策略确定的系数, $f(x)$ 可使权值更全面地反映 gm_i 的能力。当 gm_i 中的某个性能指标低于群服务器所要求的最小值时,通过 $f(x)$ 计算后其对应权值为负数,降低了整体的权值。而如果此性能指标在 gm_i 超过群服务器所要求的最大值后,通过 $f(x)$ 计算后其对应权值与最大值对应权值相近,对整体权值的影响被逐渐减弱。

3.2 群服务器的选取及失败后的恢复

群成员每隔一段很短的时间重新计算权值 $Weight$,如果权值发生了变化,将立即向群服务器报告,群服务器即时更新群成员状态信息表里的性能信息。权值最大的将被选取为备份群服务器,群服务器把群的状态信息复制到新的备份群服务器上,并在备份群服务器变更前每隔一段时间对其进行更新。

群服务器的选取策略为:1) 当群的第一个成员登录中心服务器时,中心服务器指定其为当前的群服务器。2) 当群中有多个成员登录后,群服务器将根据群成员权值的大小,选取出一个备份群服务器;如果当前群服务器仍可以胜任工作,将继续履行其职责,以减少群服务器的交换带来的额外开销。3) 如果群服务器检测到自身资源消耗超过最大限度,则“让权”给备份服务器,如果没有备份服务器则限定转发速率来降低资源消耗。

群服务器失败处理:1)当群服务器失败时,备份服务器将自动上任;2)当群服务器和备份服务器均失败时,中心服务器将从群中随机指定一个成员作为临时群服务器,临时群服务器通过选取机制尽快选出合理继任者。

3.3 对 XMPP 的兼容

XMPP 是未来统一的即时通信标准协议,三层 K 叉树架构可以与 XMPP 无缝地结合。在 XMPP 环境下,若 Switch Server 加入 XMPP 协议解释中间件,便可作为 XMPP Server (如图 5),Central Server 将作为一个 User 和 XMPP Server 同为一台主机的节点加入到 XMPP 环境中。这样三层 K 叉树可通过 XMPP 协议来建立群,进行群活动。

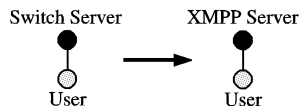


图 5 在三层 K 叉树上使用 XMPP

4 性能分析

实际的中心服务器管理下的群及每个群中的成员不一定是严格意义上的 K 叉树,但为了理论上分析的方便,可近似地假设是 K 叉树。

4.1 群的可靠性分析

假设一个简单的模型:一个 CS(中心服务器),一个 GS(群服务器)和一个 BGS(备份群服务器)。CS 和 BGS 从正常运行到出现运行失败的时间服从参数 λ 的指数分布,一个群从 GS 失败到恢复正常的的时间服从参数 μ 的指数分布,此时 $\rho = \lambda/\mu$ 。GS 和 BGS 都失败到恢复正常的的时间服从参数 $\alpha\mu$ ($\alpha < 1$) 的指数分布。CS 有一个群成员数的阈值 N_{thresh} , 1) 用户量低于等于 N_{thresh} 的概率为 τ , 此时 CS 正常工作, CS 从正常运行到出现运行失败的时间服从参数 $\psi\lambda$ ($\psi \ll 1$) 的指数分布。2) 用户量高于 N_{thresh} 的概率为 $1 - \tau$, 此时 CS 工作不稳定, CS 从正常运行到出现运行失败的时间服从参数 $\varphi\lambda$ ($\psi \ll \varphi < 1$) 的指数分布。CS 从失败到恢复正常的的时间服从参数 $\beta\mu$ ($\beta \ll 1$) 的指数分布。求一个群处于失败状态的概率。GS 失败到恢复正常的的时间可以理解为:若备份 GS 运行失败,则当前 GS 重新选择新的备份 GS 的时间是 $1/\mu$; 或者当前 GS 运行失败,则备份 GS 继任为新的 GS, 并通知上级 CS 的时间是 $1/\mu$ 。如果备份 GS 和当前 GS 都停止服务,恢复所需的时间是 $1/\alpha\mu$ 。GS 和 BGS 都失败或 CS 失败代表群服务失败。在三层 K 叉树架构中,CS 能容纳的用户量接近原来的 K 倍 ($\gamma \approx 1$), N_{thresh} 相应变为原来的 K 倍,为减少分析复杂度,可近似地认为此时 N_{thresh} 很大,群成员数不可能达到 N_{thresh} , 即 CS 从正常运行到运行失败的时间服从参数 $\psi\lambda$ 的指数分布。

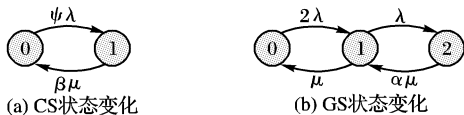


图 6 状态变化

CS 的状态转移关系如图 6(a) 所示,图中状态表示 CS 失败的数目,其平衡方程为: $\psi\lambda P_{CS0} = \beta\mu P_{CS1}$, $P_{CS0} + P_{CS1} = 1$, 得: $P_{CS0} = \frac{\beta\mu}{\psi\lambda + \beta\mu}$; $P_{CS1} = \frac{\psi\lambda}{\psi\lambda + \beta\mu}$ 。

图 6(b) 中的状态表示 GS(包括 BGS)失败的数目,其平衡方程为: $2\lambda P_{GS0} = \mu P_{GS1}$, $(\mu + \lambda) P_{GS1} = 2\lambda P_{GS0} + \alpha\mu P_{GS2}$; $P_{GS0} + P_{GS1} + P_{GS2} = 1$ 。得: $P_{GS0} = \frac{\alpha\mu^2}{2\lambda^2 + 2\lambda\alpha\mu + \alpha\mu^2}$; $P_{GS1} =$

$\frac{2\lambda\alpha\mu}{2\lambda^2 + 2\lambda\alpha\mu + \alpha\mu^2}$; $P_{GS2} = \frac{2\lambda^2}{2\lambda^2 + 2\lambda\alpha\mu + \alpha\mu^2}$ 。则群服务失败的概率为: $P_{fail} = P_{CS1} + P_{CS0} \times P_{GS2} = \frac{(2\lambda^2 + 2\lambda\alpha\mu + \alpha\mu^2) \times \psi\lambda + 2\beta\mu\lambda^2}{(2\lambda^2 + 2\lambda\alpha\mu + \alpha\mu^2)(\psi\lambda + \beta\mu)}$; 假设群中只有一个 CS,

GS 在 CS 上运行,则必须分两种情况进行考虑:1) 当用户量小于阈值时,由以上可知其失败概率为 $P_{lowerfail} = \frac{\psi\lambda}{\psi\lambda + \beta\mu}$;

2) 当用户量大于阈值时,其失败的概率为: $P_{upperfail} = \frac{\varphi\lambda}{\varphi\lambda + \beta\mu}$, 则群服务失败的概率为: $P_{failold} = \tau \times P_{lowerfail} + (1 - \tau) \times P_{upperfail}$, 因此 $P_{failold} = \tau \times \frac{\psi\lambda}{\psi\lambda + \beta\mu} + (1 - \tau) \times \frac{\varphi\lambda}{\varphi\lambda + \beta\mu}$ 。

假设失败概率比例系数 ζ 为 $P_{failold}$ 和 P_{fail} 的比值,则: $\zeta = \frac{P_{fail}}{P_{failold}} = \frac{[(2\lambda^2 + 2\lambda\alpha\mu + \alpha\mu^2) \times \psi\lambda + 2\beta\mu\lambda^2]}{(2\lambda^2 + 2\lambda\alpha\mu + \alpha\mu^2)(\psi\lambda + \beta\mu)} \div [\tau \times \frac{\psi\lambda}{\psi\lambda + \beta\mu} + (1 - \tau) \times \frac{\varphi\lambda}{\varphi\lambda + \beta\mu}]$, 因此 $\zeta = \frac{[(2\rho^2 + 2\alpha\rho + \alpha)\psi + 2\beta\rho] \times (\varphi\rho + \beta)}{[(\varphi\rho + \beta)\tau\psi + (1 - \tau)\varphi(\psi\rho + \beta)] \times (2\rho^2 + 2\alpha\rho + \alpha)}$ 。在正常情况

下 $\rho, \beta, \psi \ll \alpha, \varphi < 1$, 则 $\zeta \approx \frac{(2\rho + 1)\psi(\varphi\rho + \beta)}{[(\varphi\rho + \beta)\tau\psi + (1 - \tau)\varphi(\psi\rho + \beta)] \times (2\rho + 1)} = \frac{\psi(\varphi\rho + \beta)}{\psi(\varphi\rho + \beta) + (1 - \tau)\beta(\varphi - \psi)} < 1$, 因此,三层 K 叉树架构,一个中心服务器可接纳群成员数接近于普通架构的 K 倍,而群服务失败的概率要小于普通架构下的群。

4.2 仿真结果分析

假设群中心服务器能同时处理的最大连接数为 4000, 中心服务器管理的群成员数为 4500。对通常的群架构而言,4000 个连接代表中心服务器在用户超过 4000 个后,群失败的概率将大大提高。对于三层 K 叉树而言,4000 个连接代表中心服务器可接纳接近 4000 个群,用户不限。除去人为因素和物理损坏因素产生的影响,我们对两种架构进行仿真,其结果如图 7、图 8 所示。

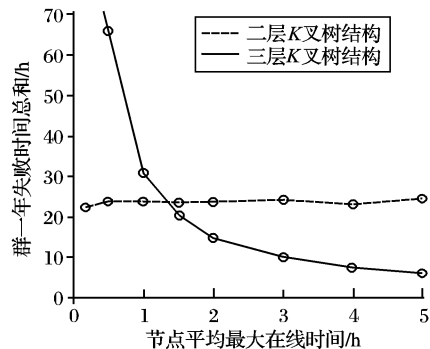


图 7 群一年失败时间总和

节点最大在线时间 (MT) 表示在一年内节点连续在线最久的一次的的时间,而节点平均最大在线时间 (AMT) 则表示群内所有节点一年内节点最大在线时间总和的均值。《2005 年中国 5 城市互联网使用现状及影响调查报告》显示中国网民平均每天上网的时间是 2.73h。从图 7 可知,在正常情况下 ($AMT < 2.73h$),三层 K 叉树结构的性能稍好于二层 K 叉树结构。图 8 表明用户在线时间越久,中心服务器的负载越轻,群的扩展性越好,在正常情况下 ($AMT < 2.73h$),在三层 K 叉

目标函数最小为 0。因此设 $\gamma = c_{ij} + \lambda_1 - \lambda_3 K$, 则有

$$P2(\lambda) = \begin{cases} \gamma, & \gamma > 0, y_{ij} = 1 \\ 0, & \gamma \leq 0, y_{ij} = 0 \end{cases}$$

对于问题 P3, 也是一个无约束的求最小值问题, 根据问题 P2 可得, 设 $\varphi = h_i - \lambda_1 u_i$, 则有 $P3(\lambda) = \begin{cases} \varphi, & \varphi > 0, z_i = 1 \\ 0, & \varphi \leq 0, z_i = 0 \end{cases}$

根据拉格朗日松弛理论可知, 对利用拉格朗日松弛方法分解问题求得的解不能够保证是原问题的可行解, 因此要对分解问题求的解进行可行性验证。对于本文算法就是在求得分解后问题 P1P2P3 的解后, 把解代入原问题目标函数及约束条件中, 如果满足就是可行解, 如果不满足条件就要重新计算求解。

对于 λ 值的选取, 初始条件下根据经验设定初始值, 然后每次迭代对 λ 值进行修改, 当满足迭代次数或者目标函数值 $Z(\lambda)$ 在规定的步数内变化不超过一个给定的值 ϵ , 这时就认为目标函数值不可能再变化, 可以停止运算。

3 仿真实验及结果分析

本文采用 Matlab 7.0 作为仿真工具, 分别对具有 40 节点和 80 个节点的应用层组播网络进行仿真实验。在仿真环境中, 节点间的链路采用随机生成的方法, 并保证仿真网络为全连通网络。网络链路的时延代价为一个范围的随机整数, 并且每个节点的度约束指定为 $[1, n/2]$ 之间的一个随机整数。

表 1 不同约束条件下目标函数值

节点数	时延度	目标函数	时延度	目标函数	时延度	目标函数
40	10, 3	126	10, 8	116	20, 5	109
	10, 5	121	20, 3	112	20, 8	106
80	15, 5	206	15, 15	185	25, 10	171
	15, 10	194	25, 5	174	25, 15	168

为了评价算法性能, 分别对不同规模的网络在不同时延和度约束条件下求的目标函数值即最小代价进行比较, 如表

(上接第 2560 页)

树结构下, 随着群用户的的增长, 中心服务器负载的增长速度远远慢于二层 K 叉树的增长速度。

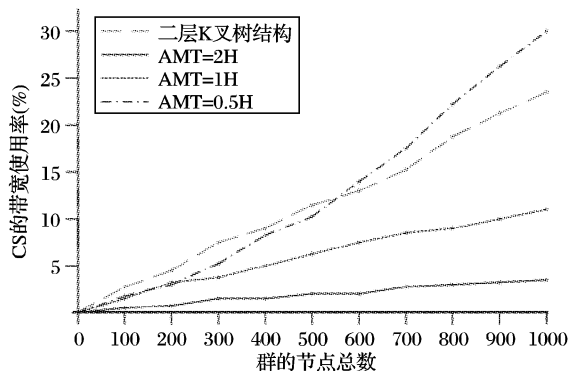


图 8 CS 带宽使用率

5 结语

三层 K 叉树架构有效地将群的转发处理和状态维护分担至群成员节点, 使用双层服务器共同来实现群服务。中心服务器主要维护群的永久信息, 转发功能和状态管理由从群成员节点中选取的群服务器来担当, 大大减轻了中心服务器的

负担, 从而使群有很强的扩展性。性能分析和仿真表明, 三层 K 叉树下的群具有很强的可靠性与扩展性。

1 所示。从表中可以看出, 约束条件放宽求得的最小代价越小, 而且目标函数值对时延约束要比度约束更敏感, 网络规模增大, 代价也相应增加。为了发现分裂分支节点规模与所需新组播服务节点数目的关系, 对不同分裂分支节点规模网络进行了仿真, 仿真结果如图 2 所示。从图中可以看出, 随着分裂节点的增多, 所需新组播服务节点数目也在增多, 而应用层组播网络规模越大需要的新组播服务节点数目也就越多。

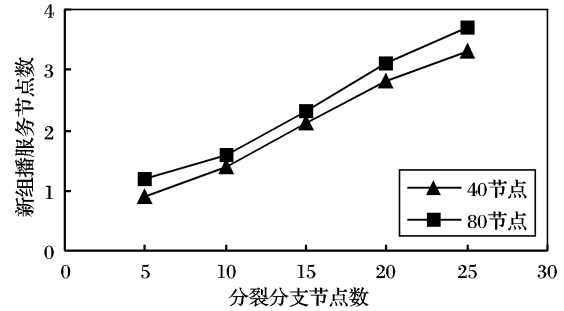


图 2 分裂分支节点与组播服务节点关系

参考文献:

- [1] LU J, SHAN X, REN Y. Overlay networking: applications and research challenges[J]. Journal of Institute of Communication, 2004, 25(12): 46-51.
- [2] ZHU Y, WU M, SHU W. Comparison study and evaluation of overlay multicast networks[A]. Multimedia and Expo 2003[C]. 2003, 3. 493-496.
- [3] YEO CK, LEE BS, ER MH. A survey of application level multicast techniques[J]. Computer Communications, 2004, 27: 1547-1568.
- [4] MATHY L, CANONICO R, HUTEHISION D. An overlay tree building control protocol[A]. Proceedings of the Third International Workshop on Networked Group Communication (NGC)[C], 2001, 7. 76-87.
- [5] 潘耘, 余镇危, 王励成. 求解应用层组播路由问题的遗传算法[J]. 小型微型计算机系统, 2005, 26(1): 55-58.
- [6] 王珩, 王华, 孙亚民. 一种基于拉格朗日松弛的时延约束多播路由算法[J]. 通信学报, 2004, 25(5): 83-93.

参考文献:

- [1] CHATTERJEE S, ABHICHANDANI T, LI H, et al. Instant Messaging and Presence Technologies for College Campuses[J]. IEEE Network, 2005, 19(3): 4-13.
- [2] MINTZ M, SAYERS A. MSN Messenger Protocol[EB/OL]. http://www.hypothetic.org/docs/msn/MSN_Messenger_Protocol-Home_Page.html, 2006-02.
- [3] Jeff_ycen. QQ 协议分析之 TCP/IP 包数据分析[EB/OL]. http://www.cnpanet.com/Class/OtherAnalysis/056912583393970113.htm, 2006-03.
- [4] ABAWAJY JH, DANDAMUDI SP. Parallel Job Scheduling on Multicenter Computing Systems[A]. IEEE CNF CLUSTER 2003[C]. 2003. 11-18.
- [5] SAINT-ANDRE P. IETF RFC3921, Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence[S], 2004.
- [6] CHU H-T, CHEN W-S, HUANG Y-H, et al. A Novel Design of Instant Messaging Service Extended From Short Message Service With XMPP[A]. Fifth IEE International Conference on 3G Mobile Communication Technologies[C], 2004. 504-508.