

全分布式 P2P 即时通信

杨晓军, 尚振宏, 郭琳

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650021)

摘要:在 P2P 领域中,文件共享和即时通信的应用程序使用最为广泛,最流行的包括 Napster, KaZaA, MSN Message, ICQ 和 QQ。基于 JXTA 协议,实现全分布式的 P2P 即时通信系统。为了使系统更加便于扩展和减少各功能模块的耦合度,将系统划分为 JXTA 网络层、信息服务层和应用层。JXTA 网络层构建一个全分布式的 P2P 网络,实现对等体的搜索和 XML 消息的发送和接收;信息服务层实现 XML 消息的解封装和封装;应用层实现信息的输入和显示。各个层向下调用服务并向上隐藏具体实现的细节。

关键词:P2P; JXTA; 即时通信**中图分类号:** TP393.2**文献标识码:** A**文章编号:** 1673-629X(2008)03-0096-03

P2P Instant Message Based on Decentralized Structured Model

YANG Xiao-jun, SHANG Zhen-hong, GUO Lin

(School of Information Engineering and Automation, Kunming University of
Science and Technology, Kunming 650021, China)

Abstract: In P2P field, file shared and instant message applications are used widely, including Napster, KaZaA, MSN Message, ICQ and QQ. Use JXTA protocol to implement P2P instant message system based on decentralized structured model. To make the system more easy to spread the system and reduce the coupling among all function modules, here divided it to three layers which are JXTA net layer, information service layer and application layer. JXTA net layer construct a P2P net, which implements the searching of peers and sending and receiving XML information. The information service layer implements the encapsulation and releases of XML information. The last layer is application layer which implements inputting and displaying information. Each layer will call service to its down-layer and hide the details of implementation to its up-layer.

Key words: P2P; JXTA; instant message

0 引言

目前,最流行的商业即时通信解决方案仍使用集中式实现,例如国外的 MSN Message、ICQ 和国内的 QQ,所有交互都要通过一个或多个中心服务器^[1]。中心服务器提供连接成员的准确目录并路由全部信息。这种实现不具有很好的伸缩性,需要拥有庞大的大型数据中心,以便支持大量用户。全分布式结构消除了中央服务器带来的问题。没有中央控制点,不会因为一点故障导致全部瘫痪,这种模式具有自组织行为,降低了拥有者的成本,提供可扩展性,特别适合在自组

织网上的应用,如即时通信。JXTA 技术非常适合于全分布式应用的开发,同时也很好地克服了全分布式结构下发现的准确率和性能较低的问题。文中将基于 JXTA 技术提出一个三层模型来分析实现全分布式 P2P 即时通信系统。

1 集中式与全分布式

在 P2P 的 4 种研究形式中,全分布式和集中式是其中的两种形式。集中式是最早出现的 P2P 网络采用的结构模式,其拓扑结构如图 1 所示。集中式结构模型的特点就是服务器以索引方式存储各对等体的信息,如地址、共享的资源位置等。每个对等体在获取资源或其他对等体通信前,都需要在服务器上检索,获取相关的信息,然后才能连接到其他对等体。最经典案例就是著名的 MP3 共享软件 Napster,它通过一个中央服务器保存所有 Napster 用户上传的音乐文件索引和存放位置的信息。当某个用户需要某个音乐文件

收稿日期: 2007-06-28**基金项目:** 云南省自然科学基金(2005F0016Q); 云南省教育厅科学研究基金(5Y0674D); 昆明理工大学基金项目(校引 2005-6)**作者简介:** 杨晓军(1981-),男(哈尼族),云南思茅人,硕士研究生,研究方向为多媒体技术、分布式计算;尚振宏,副教授,研究方向为图像处理、分布式计算。

时,首先连接到 Napster 服务器,在服务器进行检索,并由服务器返回存有该文件的用户信息;再由请求者直接连到文件的所有者进行文件传输。

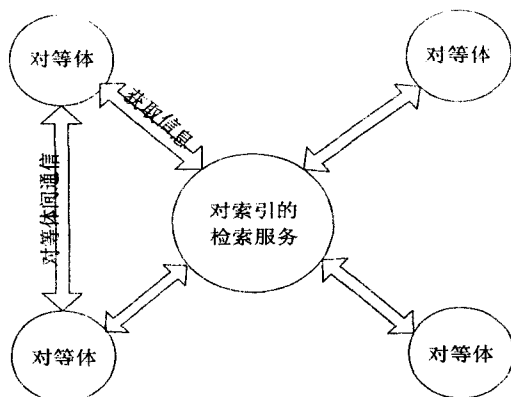


图1 集中式

这种集中式的模型最大的优点是维护简单、发现效率高^[2]。最大的问题与传统客户机/服务器结构类似,中央服务器的瘫痪容易导致整个网络的崩溃,可靠性和安全性较低。

全分布式的模型如图2所示。全分布式没有中央索引服务器,所有的对等体都是平等的,每个对等体都提供资源服务和搜索服务,搜索请求遍历整个 P2P 网络,拥有相关资源的对等体返回搜索请求。全分布式完全避免了因中央服务器的瘫痪而导致整个网络的崩溃,部分节点受到攻击也不会影响整个网络的性能。

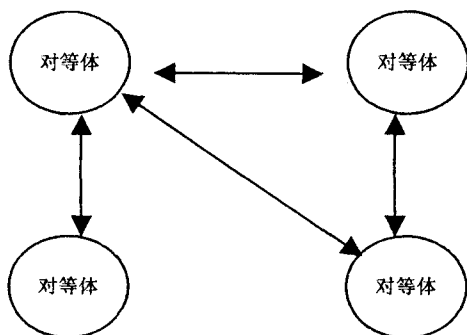


图2 全分布式

由于这种模型的搜索请求是遍历整个 P2P 网络,其缺点是随着对等体的不断增多,网络规模不断扩大,通过洪泛方式定位对等体的方法将造成网络流量急剧增加,从而导致网络中部分低带宽节点因网络资源过载而失效^[2]。解决这个问题的办法是把搜索请求约束在一个小的范围内,而不是在整个点对点网络中搜索,利用 JXTA 可以很好地做到这一点。

2 JXTA 简介

选择 JXTA 的理由主要来自两个方面。第一个理由是,对许多应用而言,P2P 体系结构所具有的优势:

可伸缩、健壮、动态、容错、自发和自组织。第二个理由来自 JXTA 所给予我们的:即构建一个 P2P 应用的公共平台^[3]。今天有很多种流行的 P2P 应用得以广泛使用,但这些应用程序都是按传统的方法从底层开始开发,并互不兼容。不同 P2P 社区之间的用户不能互相访问。JXTA 正是试图提供能构建 P2P 应用的公共平台。

对等组是 JXTA 的一个重要机制,每个对等体只有加入对等组才能获取该组提供的资源^[4]。对等组是由具有共同关注点的对等体自发组织的,因此,一个对等组定义的抽象区域为对等组的操作提供了一个隐式的范围机制,当对等体查找资源时,对等组的边界划分了查找范围,该查找信息仅能传播到该对等组的其他对等体,很好地控制了网络流量,避免了在整个网络中洪泛式的查找,克服了因网络流量急增而造成的网络瘫痪。

3 全分布式即时通信系统的设计与实现

本系统以分层的思想来构建一个点对点即时通讯的模型,如图3所示。各个层实现各自的功能并向上提供服务接口,上层通过接口调用下层的的服务。当某层的技术实现发生变化时,分层的模型可以让原有的功能更加容易被替换,而这种替换其他层无需了解^[5]。

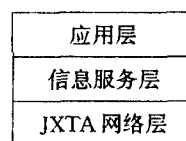


图3 即时通讯模型

3.1 JXTA 网络层

该层是系统的核心层,封装了 P2P 网络下一些核心的功能,具体如下:

(1)加入网络对等组(NetPeerGroup)。在 JXTA 网络中,所有对等体首先都需要加入到网络对等组中,只有加入到该组中才能获取其他服务^[6],比如管道服务和发现服务等。

(2)管道通告(PipeAdvertisement)及输入管道的创建和发布。本系统通过管道来进行信息的传输,管道为输入管道和输出管道之间提供了一种单向的虚拟连接^[6]。每个对等体要进行通信都需要创建并在网络中发布管道通告,同时利用该管道通告创建输入管道监听发送来的消息。

(3)管道通告的发现和输出管道的创建。对等体加入到网络对等组后,对等体就可获得 JXTA 提供的发现服务,利用发现服务可在 JXTA 网络层搜索其他对等体发布的资源通告,例如对等体通告、管道通告

等。当发现其他对等体发布的管道通告时,便可利用发现的管道通告创建输出管道,两个对等体间就建立了一条虚拟的连接并可利用管道进行单向通信。为了实现两对等体间的全双工通信,只需各自对等体各创建一个管道通告就可以实现。

(4)信息的传输。实现上层提交下来的 XML 格式信息的传输。在 JXTA 网络层,利用管道发送的信息和接受的信息都是以 XML 格式封装了的。信息的封装和解封装由上层信息服务层实现,JXTA 网络层不用再对信息进行处理,通过管道直接发送上层传送下来的信息封装体。

JXTA 网络层的主要方法描述用 JAVA 实现如下:

```
Class JxtaNet implements PipeMsgListener, OutputPipeListener {
public JxtaNet() //加入 JXTA 网络对等组, 获取网络对等组提供的发现服务和管道服务
}
public static void createMyPipeAdv() //创建自己的管道通告并发布
}
public static Vector researchOtherPipeAdv() //搜索其他对等体发布的管道通告
}
public static void createInPipe (PipeAdvertisement myPipeAdv) //利用自己的管道通告创建输入管道
}
Public static void createOutPipe (PipeAdvertisement otherPeerPipeAdv) //利用搜索发现的其他对等体的管道通告创建输出管道
}
Public static void toFirstLayerMessage (StringMessageElement sme) //接受信息服务层传送下来的信息封装体
}
public void pipeMsgEvent (PipeMsgEvent pipeMsgEvent) //实现 PipeMsgListener 接口的方法, 该接口绑定于输入管道, 监听收到的信息
}
public void outputPipeEvent (OutputPipeEvent outputPipeEvent) //实现 OutputPipeListener 接口的方法, 该接口绑定于输出管道, 用于发送信息
}
.....
}
```

3.2 信息服务层

信息服务层主要处理信息的封装和解析,并向上提供服务。具体如下:

(1)信息封装。把上层传送下来的信息封装为 XML 格式,并将封装体传送给 JXTA 网络层,由 JXTA

网络层发送。

(2)信息解析并向上层提交。JXTA 网络层所提交上来的信息是以 XML 格式封装的,因此需要从封装体中解析出实际发送的信息并提交给应用层。

信息服务层的主要方法描述用 JAVA 实现如下:

```
class MessageService {
public static StringMessageElement messagePacked (String message) //封装应用层传递下来的信息
}
public static String resolveMessage (StringMessageElement sme) //解析从 JXTA 网络层提交上来的信息封装体
}
public static toApplicationLayer () //将解析出的信息提交给应用层
}
.....
}
```

3.3 应用层

由于 JXTA 网络层和信息服务层已经实现了 P2P 通信的功能,应用层主要实现与用户交互的界面,主要包括将用户输入的信息传递给信息服务层处理,将信息服务层提交上来的信息显示给用户,其他对等体的发现和信息的传输都由下面两层实现。

应用层的主要方法描述用 JAVA 实现如下:

```
public class Application {
public void showMessage (String message) //显示从信息服务层提交上来的消息
}
Public void sendToMessageServiceLayer (String message) //将要发送的信息传递给信息服务层
}
.....
}
```

4 结束语

文中分析了全分布式结构下的 P2P 即时通信的优点,并提出三层模型实现基于 JXTA 的即时通信,将各个功能分别封装在不同的层,使各层只需使用下面一层所提供的服务,而不用关心服务如何实现,只要提供服务的接口一致,即使底层实现技术发生改变也不会影响到上层,提高了本系统的健壮性和可扩展性。

参考文献:

- [1] 常晓波,李 静. java P2P 程序设计[M]. 北京:中国电力出版社,2002:13-15.
- [2] 罗杰文. Peer To Peer (P2P) 综述[EB/OL]. 2007-04-04. <http://www.p2psky.com/tech/article2740-3.html>.

(下转第 101 页)

前,即 $iR - p$ (i 表示进行第 i 次时间同步化, p 为参考时间的信息可确实送达的时间),即先从其节点获取参考时间值 t_m ,再经过估算其余时间同步化周期到达的时间估计值 $\tilde{X}_j = t_m + t_s - t_r + e_{mean}$,其中 j 表示从节点 j 获取的参考时间值, t_s 表示下次时间同步化时间(即 $t_s = iR$), t_r 表示收到该参考时间值的时间, e_{mean} 表示平均误差^[5]。其示意图如图 1 所示。

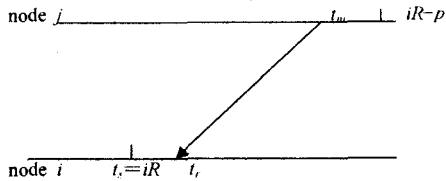


图 1 时间估计值示意图

在计算完各个参考时间估计值后,接着就要采用一个收敛函数计算其合理的同步时间,也就是使用滑动窗口演算法来获取所需要的同步化时间结果。该演算法步骤如下:

- 1) 执行滑动窗口程序,输入 n 和 γ (n 为参考时间估计值个数, γ 为视窗长度)。
- 2) 计算标准视窗样本 M_i (包含时间估计值个数最多的窗口样本作为标准视窗样本 M_i)。
- 3) 计算 m_{min} (若包含最多时间估计值个数的窗口样本非唯一,则选取当中标准差最小的一组窗口样本为标准窗口样本 M_{min})。
- 4) 求出 M_{admin} (若包含最多时间估计值个数且标准差最小的窗口样本非唯一,则选取第一个窗口样本为标准窗口样本 M_{admin})。

滑动窗口演算法的概念如图 2 所示,其横轴代表时间估计值的时间轴。

3 综合分析

滑动窗口演算法中的视窗长度 $\gamma = \Delta_{max} + \epsilon_{max}$,根据分布式容错时间同步化系统精度公式,其所获得的同步化时间结果的精确度为:

$$II = \epsilon_{max} + \frac{k(\Delta_{max} + \epsilon_{max})}{n - k} + \frac{k(2\Delta_{max} + \epsilon_{max})}{n - k + 1}$$

其中 n 表示时间估计值个数, k 表示发生错误的时间估计值个数, Δ_{max} 表示时间估计值间隔中最大者, ϵ_{max} 表示时间信息延迟最大值。

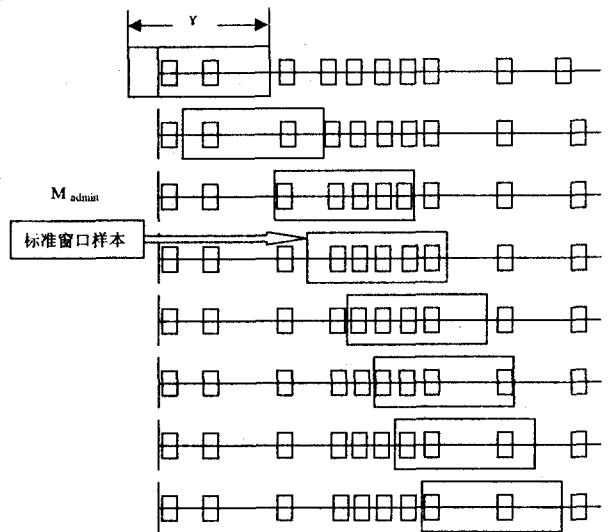


图 2 滑动窗口演算法示意图

影响系统精度的主要原因有二:其一是时间信息延迟最大值 Δ_{max} ,在整个系统启动或新客户端加入时,可能会因为 Δ_{max} 值较大影响精确度,但是经过数次时间同步周期后,应可收敛至一定的数值,因此 Δ_{max} 的影响会趋于固定,变动性不大;另一个影响时间同步化精确度的因素就是网络传输延迟 Δ_{max} ,它产生的原因主要有:TCP/IP 网络通信协议对网络传输延迟有影响;网络负载与该项机器处理程序的忙碌状态对网络传输延迟也有影响等。对于影响上述精度的因素可以通过对当时的通信协议延迟进行测量,但无法代表在交换信息时所有信息通信协议延迟,可以用最小参考值进行估算,尽量接近当时的通信协议延迟。

参考文献:

- [1] Pankaj J. Fault Tolerance in Distributed Systems[M]. [s. l.]: Prentice - Hall, 1994.
- [2] Cristian F. Probabilistic Clock Synchronization[J]. Distributed Computing, 1989, 3: 146 - 158.
- [3] Peluegl M J, Blough D M. A New and Improved for Fault - Tolerant Clock synchronization[J]. Journal of Parallel and Distributed Computing, 1995, 27: 1 - 14.
- [4] 欧阳询,李 榕. 分布式容错系统的同步化策略[J]. 计算机系统应用, 2000(2): 23 - 35.
- [5] 贺 鹏,李 菁,吴海涛. 网络时间同步算法研究与实现[J]. 计算机应用, 2003, 23(2): 15 - 17.

(上接第 98 页)

- [3] 技 桥. JXTA 技术手册[M]. 北京:清华大学出版社, 2004: 9 - 15.
- [4] 孙 鹏,武 波,张玉清. JXAT 对等组机制的应用研究[J]. 计算机工程, 2006, 32(2): 110 - 112.

- [5] Tanenbaum A S. 计算机网络[M]. 第 4 版. 潘爱明,译. 北京:清华大学出版社, 2004: 31 - 40.
- [6] Project JXTA2. 3: Java Programmers Guide[M/OL]. 2005 - 04 - 07. http://www.jxta.org/docs/JxtaProgGuide-v2.3.pdf.