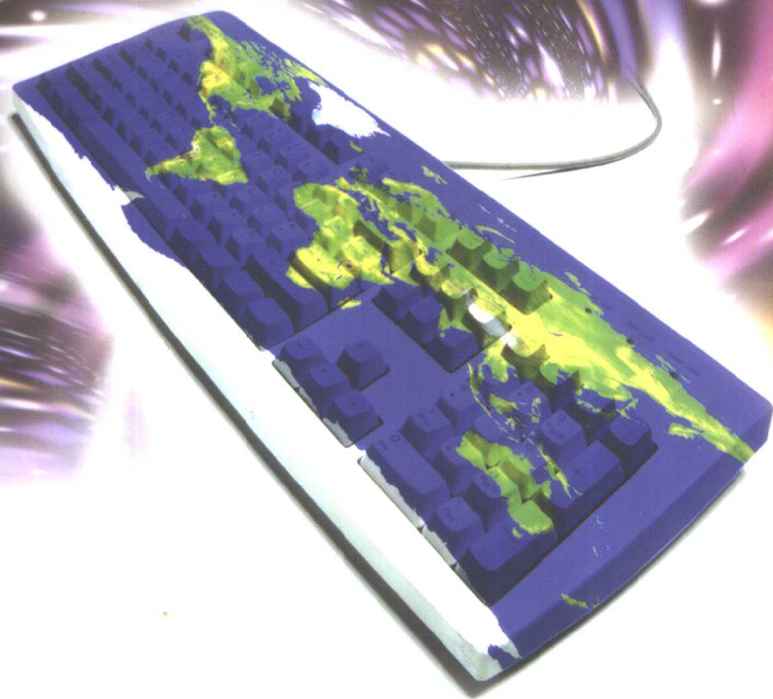


Java技术教程系列丛书



许 斌 主编
王克宏 审核

JXTA —

Java P2P 网络编程技术



清华大学出版社
<http://www.tup.tsinghua.edu.cn>

Java 技术教程系列丛书

JXTA——Java P2P

网络编程技术

许 斌 主编

王克宏 审核

清华大学出版社
北 京

内 容 简 介

JXTA 是美国 Sun 公司在 Java 领域里推出的新技术, 目的是为 P2P(Peer-to-Peer) 的网络应用开发提供一个统一的平台。而且为了鼓励和支持该技术的发展, JXTA 项目采用了开放源代码的方式进行, 吸引了大量业界人士参与到 JXTA 技术的研究与应用当中。

本书较详细地介绍了 JXTA 技术的基本概念、P2P 技术的来龙去脉以及因特网的发展情况, 阐述了 P2P 技术面临的挑战和机遇, 并介绍了 JXTA 整个项目的情况, 包括标准的 JXTA 应用、JXTA Shell 以及 JXTA 的编程方法, 详细地阐述了 JXTA 的核心协议。

通过本书的学习, 可以帮助广大读者掌握利用 Java 语言进行 P2P 网络编程的方法。读者对象包括大专院校师生以及信息产业界的技术人员。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

JXTA—Java P2P 网络编程技术/许斌主编. —北京: 清华大学出版社, 2003. 6
ISBN 7-302-06667-1

I. J... II. 许... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 038921 号

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

<http://www.tup.com.cn>

责任编辑: 林庆嘉

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 16 字数: 378 千字

版 次: 2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

书 号: ISBN 7-302-06667-1/TP·4989

印 数: 0001~3000

定 价: 25.00 元

目 录

第一章 P2P 介绍	1
1.1 因特网的发展和 P2P 的演变	1
1.1.1 早期的因特网就是 P2P(1969—1995)	1
1.1.2 因特网大爆炸时期的网络模型(1995—1999)	2
1.1.3 P2P 应用重新出现在因特网上(2000—)	3
1.2 常见的 P2P 应用程序	4
1.2.1 聊天程序	4
1.2.2 协同工作和白板	5
1.2.3 游戏	5
1.2.4 文件共享	6
1.3 P2P 面临的问题	7
1.3.1 对称带宽问题	7
1.3.2 Web 标准操作过程的影响	7
1.3.3 名字空间问题	8
1.3.4 知识产权问题	8
1.3.5 用户问题	9
1.3.6 创建 P2P 网络	10
1.4 P2P 相关技术	11
1.4.1 JINI	11
1.4.2 软件代理(Agent)	12
1.4.3 其他 P2P 平台	13
1.4.4 Web Service	15
1.5 本章小结	16
第二章 JXTA 概述	18
2.1 P2P 网络	18
2.2 JXTA 是什么	20
2.3 JXTA 的概念	22
2.3.1 Peer(对等机)	22
2.3.2 Peer Group(对等组)	22
2.3.3 Endpoint(端点)	22
2.3.4 Pipe(管道)	23

2.3.5	Advertisement(广告)	24
2.3.6	Message(消息)	24
2.3.7	Rendezvous Peer(集合点)	25
2.3.8	Router Peer(路由 Peer)	26
2.3.9	Gateway Peer(网关 Peer)	27
2.4	用于通信的 Peer	27
2.4.1	防火墙	28
2.4.2	NAT(网络地址转换)	28
2.4.3	代理服务器	29
2.4.4	DHCP(动态 IP 分配)	29
2.4.5	网络的不稳定	29
2.4.6	网关问题	29
2.5	Peer 和对等组	30
2.5.1	Peer 和用户的关系	30
2.5.2	对等组的必要性	30
2.5.3	JXTA 应用程序与对等组	32
2.5.4	对等组的成员资格	32
2.5.5	对等组的服务	33
2.6	广告(Advertisement)	34
2.6.1	广告的详细类型	35
2.6.2	模块(Module)广告	37
2.6.3	管道广告	43
2.6.4	端点路由消息	44
2.6.5	消息(Message)	44
2.7	JXTA 的协议	46
2.7.1	核心协议	46
2.7.2	管道绑定协议(PBP)	47
2.7.3	Peer 解析协议(PRP)	48
2.7.4	集合点协议(RVP)	49
2.8	JXTA 的标志	51
2.8.1	Codat ID	51
2.8.2	Peer ID	51
2.8.3	Goup ID	51
2.8.4	Service/Module ID	52
2.8.5	Pipe ID	52
2.9	本章小结	52

第三章 JXTA 标准应用	53
3.1 JXTA 社区	53
3.1.1 下载并安装 JXTA	53
3.1.2 JXTA 的配置工具	55
3.2 Hello JXTA 的简单编程	58
3.2.1 系统要求	58
3.2.2 程序源代码	58
3.2.3 编译和运行	59
3.2.4 程序分析	61
3.3 即时通信和文件共享(myJXTA)	62
3.3.1 项目介绍	62
3.3.2 安装与配置	62
3.3.3 功能介绍	64
3.4 网络游戏(Chess)	67
3.4.1 项目介绍	67
3.4.2 安装与配置	67
3.4.3 功能介绍	69
3.5 发现对方 JXTA View	72
3.5.1 项目介绍	72
3.5.2 安装与配置	72
3.5.3 功能介绍	73
3.6 计算机辅助设计(JXTA-CAD)	74
3.6.1 项目介绍	74
3.6.2 安装与配置	74
3.6.3 功能介绍	75
3.7 本章小结	77
第四章 JXTA Shell	78
4.1 JXTA Shell 的用途和价值	78
4.2 JXTA Shell 的命令介绍	78
4.2.1 数据管理命令	79
4.2.2 系统管理命令	82
4.2.3 文件存取命令	84
4.2.4 Advertisement 的管理命令	85
4.2.5 Peer Group 的管理命令	87
4.2.6 Peer 的管理命令	88
4.2.7 Pipe 的管理命令	89
4.2.8 message 的操作命令	90

4.2.9	Pipe 的通信命令	90
4.2.10	配置管理与信息查看	91
4.2.11	talk 命令	92
4.3	JXTA Pipe 的重要角色	95
4.3.1	两个独立的 JXTA Shell	95
4.3.2	创建输入管道	96
4.3.3	发布 Advertisement 用于查询使用	96
4.3.4	从输入管道接收 Pipe Message	97
4.3.5	绑定到管道的输出端	97
4.3.6	通过输出管道发送消息	98
4.4	理解 Rendezvous 和 Routers	98
4.4.1	用 shell 命令验证配置	100
4.4.2	Discovery 的范围	102
4.4.3	Peer 间使用 Pipe 进行虚拟连接	103
4.5	扩展 JXTA Shell 的功能	106
4.5.1	Shell 扩展的基本机制	106
4.5.2	练习编写一个 Shell 扩展命令	107
4.5.3	创建一个复杂的 Shell 功能扩展	110
4.6	本章小结	116
第五章	JXTA 深入编程	118
5.1	基于 Pipe 的编程	118
5.1.1	Pipe 的类型	118
5.1.2	JXTA Pipe API 简介	119
5.1.3	Pipe 通信的过程	119
5.2	简单实用的程序开发包 P2PSocket	121
5.2.1	P2PSocket 开发包的简介	121
5.2.2	P2PSocket 的 API	122
5.2.3	P2PSocket.java 源代码	125
5.2.4	OutputListener.java 源代码	129
5.3	即时消息 XChat	129
5.3.1	XChat 的通信过程	129
5.3.2	XChat 用到的类	130
5.3.3	XChat 代码分析	131
5.3.4	XChat.java 的源代码	131
5.4	文件传送	133
5.4.1	XFile 的通信过程	133
5.4.2	XFile 用到的类	133

5.4.3	XFile 代码分析	135
5.4.4	XFile. java 的源代码	137
5.5	P2P 发布个人主页	140
5.5.1	XClient 的通信过程	141
5.5.2	XWebServer 和 XWebClient 用到的类	142
5.5.3	XWebServer 代码分析	142
5.5.4	XWebClient 代码分析	143
5.5.5	XWebServer. java 的源代码	145
5.5.6	XwebClient. java 的源代码	146
5.6	数据同步	149
5.6.1	XChatp 的通信过程	149
5.6.2	XChatp 用到的类	149
5.6.3	XChatp 代码分析	150
5.6.4	XChatp. java 的源代码	151
5.7	本章小结	153
第六章	JXTA Content Manage Service (CMS)	154
6.1	CMS 概述	154
6.1.1	CMS 的产生	154
6.1.2	什么是 CMS	154
6.2	CMS 核心内容	154
6.2.1	CMS 体系结构	154
6.2.2	CMS 协议细节	156
6.2.3	数字权限管理	158
6.3	CMS 编程	158
6.3.1	CMS 编程基础	158
6.3.2	CMS 的初始化	159
6.3.3	文件的简单共享	160
6.3.4	文件的简单搜索	161
6.3.5	获取共享内容	163
6.3.6	基于 Metadata 的共享	166
6.3.7	基于 Metadata 的搜索	169
6.4	CMS 的发展目标	171
6.4.1	分布式搜索	171
6.4.2	支持动态内容	171
6.4.3	内容广告的单独共享	172
6.4.4	与 JXTA Search 结合	172
6.5	本章小结	172

第七章 JXTA 核心协议(一)	173
7.1 对等机发现协议(PDP)	173
7.1.1 PDP 的消息格式	173
7.1.2 Discovery Query Message	174
7.1.3 Discovery Response Message	175
7.1.4 Discovery Service	177
7.1.5 DiscoveryListener 接口	178
7.1.6 DiscoveryEvent 类	179
7.1.7 发现远程的 Advertisements	179
7.1.8 发现缓存中的 Advertisement	180
7.1.9 清除缓存中的 Advertisement	180
7.1.10 使用 Advertisement	180
7.1.11 实例化一个 Advertisement	181
7.1.12 发布 Advertisement	182
7.1.13 小结	182
7.2 管道绑定协议(PBP)	183
7.2.1 Pipe 概念	183
7.2.2 Pipe Advertisement	184
7.2.3 PBP 消息格式	184
7.2.4 Pipe Binding Query Message	185
7.2.5 Pipe Binding Answer Message	186
7.2.6 Pipe Service	187
7.2.7 使用 Pipe Service 来发送和接收 Message	188
7.2.8 Secure Pipe	190
7.2.9 Propagation Pipe	190
7.2.10 Bidirectional Pipe	190
7.2.11 小结	192
7.3 端点路由协议(ERP)	192
7.3.1 Endpoint 简介	193
7.3.2 Endpoint Service	193
7.3.3 Endpoint Transport Implementation	195
7.3.4 Endpoint 地址	196
7.3.5 Message 的格式化过程	198
7.3.6 接收到达的 Message	198
7.3.7 用 Endpoint Service 广播 Message	199
7.3.8 用 EndpointMessenger 直接发送 Message	199
7.3.9 Endpoint Filter Listener	199
7.3.10 Endpoint Routing Protocol(ERP) 介绍	200

7.3.11	Route Query Message	201
7.3.12	Route Response Message	202
7.3.13	Endpoint Router Message	203
7.3.14	Endpoint Router Transport Protocol	204
7.3.15	小结	205
第八章	JXTA 核心协议(二)	206
8.1	对等机解析协议(PRP)	206
8.1.1	PRP 消息格式	206
8.1.2	解析请求消息	207
8.1.3	解析响应消息	208
8.1.4	Resolver Service	208
8.1.5	小结	209
8.2	集合点协议(RVP)	210
8.2.1	RVP 消息格式	210
8.2.2	租约请求消息	211
8.2.3	租约授权消息	212
8.2.4	租约取消消息	212
8.2.5	控制消息传播	213
8.2.6	集合点服务	214
8.2.7	传播消息	214
8.2.8	与集合点建立连接和断开与集合点的连接	215
8.2.9	RendezvousListener 和 RendezvousEvent 类	215
8.2.10	Rendezvous 服务使用的支持类	216
8.2.11	维护 Rendezvous Connections	217
8.2.12	小结	217
8.3	对等机信息协议(PIP)	218
8.3.1	PIP 消息格式	218
8.3.2	Peer Info Query Message	219
8.3.3	Peer Info Response Message	219
8.3.4	Peer Info Service	221
8.3.5	PeerInfoListener 接口	223
8.3.6	小结	223
8.4	模块 Class 与 PeerGroup 解析	223
8.4.1	模块、服务、应用	224
8.4.2	模块类广告	224
8.4.3	模块规范广告	225
8.4.4	模块实现广告	226

8.4.5	创建 World Peer Group	228
8.4.6	创建 Net Peer Group	230
8.4.7	创建 Peer Group	231
8.4.8	加入 Peer Group	231
8.4.9	离开 Peer Group	232
8.4.10	销毁 Peer Group	232
8.4.11	创建服务	232
8.4.12	小结	236
第九章	JXTA 发展方向	237
9.1	JXTA 的发展	237
9.1.1	各种语言的实现	237
9.1.2	各种平台的实现	237
9.1.3	各种应用	237
9.1.4	Service	237
9.2	参与 JXTA 的发展	238
9.2.1	参加各种项目	238
9.2.2	JXTA 邮件列表	238
9.2.3	提出一个新项目	239
	JXTA 术语表	240

第一章 P2P 介绍

最近两三年, P2P(Peer-to-Peer)又成为了因特网上的一个热点。P2P 是因特网的一种应用模式, 其意思是指网络上的任何设备(包括大型机、PC 机、PDA、手机、机顶盒等等)都可以平等地直接进行连接并进行协作。相比当前因特网上主流的应用模式 Client/Server 或者 Client/Service 而言, P2P 具有自己鲜明的特点和优势。

1.1 因特网的发展和 P2P 的演变

1.1.1 早期的因特网就是 P2P(1969—1995)

因特网的发展本身就是一个“螺旋式上升, 波浪式前进”的过程。P2P 并不是一个新的概念, 早在 1969 年因特网的前身 ARPANET 刚出现的时候, 网络的应用模式就是 P2P, ARPANET 的最初目的是在全美国范围内共享计算资源, 其所面临的挑战是如何集成当时各种不同的网络, 使之成为一个通用的网络, 并且使得各个主机成为网络上平等的成员。ARPANET 上最早的主机包括 UCLA、SRI、UCSB 和 Utah 大学的计算机系统, 这些计算机系统都是独立的而且是平等的, ARPANET 是以一种平等的计算 Peer 的方式把这些计算机系统连接起来, 而不是用 Master/Slave 或者是 Client/Server 的方式连接。

早期的因特网比现在的因特网更加开放和自由, 例如在 20 世纪 80 年代之前, 人们从未听说过防火墙的概念。就通常意义而言, 当时的网络上任意两台计算机都可以给对方发送网络包, 网络就是人们进行协同研究工作的场所, 不需要提防任何东西。而且当时的协议和系统都设计得足够专业化, 很少出现危及安全的入侵情况, 一般说来也没有任何伤害。而现在的因特网有了更多的划分。

后来因特网上出现了 FTP 和 Telnet 这样的比较受欢迎的应用程序。虽然上述单个的应用程序都是 Client/Server 方式的, 但是整体上的使用模式是对称的。网络上的每台主机都可以 FTP 或者 Telnet(远程登录)到其他主机上, 同时自己也可充当服务器。

基础的对称性就是因特网的基本特征, 在其之上也出现了若干经典的 P2P 复杂系统, 例如 Usenet 和 DNS。Usenet 通过利用非集中式的控制, 在计算机之间进行文件复制, 所采用的是 UUCP 协议(UNIX-to-UNIX copy protocol), 该协议使得任意一台 UNIX 计算机都可以拨号到另外一台 UNIX 计算机上, 并在交换完文件后自动断线。DNS 是另一个仍在使用的 P2P 的经典例子, DNS 的作用就是把因特网的域名和 IP 地址进行映射, 最早的时候这个映射是保存在一个大文件中的, 并在因特网上进行复制。随着因特网规

模的增大,必须采用一种方式来进行域名数据的共享。DNS 提供了一种实现文件共享的解决方法,采用了层次性的信息归属办法,即我们通常所说的域名层次。DNS 之间可以相互交换域名信息,这个时候,每个 DNS 也可能是服务器(接收其他 DNS 的请求),也可以是客户端(向其他 DNS 发出请求)。

1.1.2 因特网大爆炸时期的网络模型(1995—1999)

从 1995 年开始,随着 PC 机的广泛使用并接入因特网,成千上万的人挤到因特网上,他们使用因特网的方式就是发送电子邮件、浏览网页并在网上购物,这些现象已经长远地影响到了网络架构的发展,也直接影响了 P2P 应用程序的发展。这种变化改变了人们使用网络的方式,出现了许多新的情况,例如网上协作的崩溃、防火墙的大量使用,产生了许多非对称的网络连接方式,比如说 ADSL 和 Cable Modem。

一个非常显著的变化就是网络的主流使用方式变成了 Client/Server 模式,所改变的是计算机之间相互寻址和通信的方式。随着因特网的商业化和大量家庭用户的出现,像 SLIP 和 PPP 这样的通过调制解调器进行连接的网络协议变得非常普遍,但是面向的都是低速的模拟调制解调器;而许多公司开始通过防火墙和网络地址转换(NAT)来管理公司的网络。这些变化都是基于一种使用模式,那就是计算机只是下载数据,并不需要发布或者上载信息。Web 浏览器和其他应用程序的运行都是基于一种简单的 Client/Server 模式:客户机建立一条连接到某一众所周知的服务器上,下载一些数据,然后中断连接。运行 Web Client 的计算机不需要拥有一个永久或者是众所周知的地址,它也没有必要一直连接到因特网上,它惟一需要做的事情是知道如何提问并接收响应。

与此同时,因特网的平等协作的本质受到威胁,早期的因特网中所有的主机都是平等的参与者,网络是对称的,也就是说如果一台计算机能够访问网络的话,那它也可以被其他主机所访问,每台计算机都是可以是客户机或服务器。但是这些都随着防火墙的发展、动态 IP 的增长和网络地址转换(NAT)的流行而发生改变。

防火墙是随着网络安全的需要而诞生的。防火墙处在内部网和因特网之间,起到网关的作用,但是防火墙要过滤网络包,选择部分流量可以通过网关;典型的防火墙的作用就是允许内部网的计算机可以与因特网上的计算机建立连接,但是却阻止因特网中的任意一台计算机与内部网的计算机建立连接。这种防火墙就像是一种单向的网关:可以连接出去,但是不可以连接进来。被这种方式保护起来的主机不太容易充当服务器,往往只是一个客户机。而且,向外的连接也可以被限制到某些特定的应用程序上,譬如 FTP 和 Web 就可以被限定到防火墙上的某些端口上。虽然防火墙是一种非常有用的安全工具,但是它为 P2P 计算模式制造了非常严重的障碍。

动态 IP 的产生是随着家庭网络用户的出现而出现的。随着越来越多的用户通过调制解调器拨号接入网络,给每台计算机分配一个固定 IP 的做法已经不可行了,而代之以动态 IP,因为 IP 资源也是有限的。这样造成的后果是某些计算机的 IP 地址每天都在变化,这些计算机在因特网上并不容易被访问到。诸如即时消息和文件共享之类的 P2P 应用程序不得不建立动态主机目录来帮助计算机之间建立连接,而在因特网的早期,所

有计算机都是静态 IP，相对而言建立连接就简单多了。

网络地址转换(NAT)不仅没给计算机提供一个合法的公开地址，而且还把主机隐藏在防火墙后面，因此它综合了防火墙和动态 IP 的所有问题：主机地址不仅是不固定的，而且是不可访问的。所造成的后果是损失了应用程序之间通信的灵活性。例如有许多因特网游戏要求玩家之间的计算机能够相互连接，但是 NAT 的存在阻碍了计算机的直接连接，直接的后果是在应用程序层面上提供一个中央服务器来充当消息路由器，仿效 TCP/IP 所应该发挥的功能。

20 世纪 90 年代末期，因特网发展的另一个趋势也对 P2P 应用程序带来新的挑战，那就是非对称网络连接的发展。网络带宽提供者为了得到更高的效率而决定提供非对称的带宽，往往是下行带宽非常大而上行带宽非常小，ADSL 和 Cable Modem 的下行/上行带宽往往相差 3~8 倍。这种用法的根源在于 Web 是因特网上的主要应用，绝大部分用户仅仅是 Web 的客户机，而不是服务器。即使有部分用户想发布自己的网页也不会通过家庭宽带连接的方式来实现，而是通过第三方提供的专用的服务器来发布自己的网页。

现在的问题在于 P2P 应用程序正在改变原有的假设：网络用户仅仅想从因特网下载东西，而不会上传信息。Napster 和 Gnutella 等文件共享应用程序颠倒了带宽的使用方式，使得计算机提供的文件比它下载的文件还要多，要求的上行带宽要比下行带宽大。但是现有的网络架构并不能很好地满足要求，而且更糟糕的是，由于 TCP 协议对速度的控制，一旦上行通路被堵塞，下行通路也会受到影响，所以如果某台计算机正在通过慢速的上行通路给其他计算机提供文件服务的话，那它就不可能同时在快速的下行通路上下载文件。因此 P2P 应用程序在非对称带宽的网络并不能够很好地运行，一旦 P2P 程序广泛流传，网络基础架构就应该变得易于处理这种新的网络流量模式。

1.1.3 P2P 应用重新出现在因特网上(2000—)

从 2000 年开始，一个能够在网上进行音乐文件共享的名为 Napster 的 P2P 程序在网上广为流行，吸引了广大网上乐迷的注意力，短时间内就吸引了成千上万的用户，原因在于用户可以下载 MP3 音乐文件。Napster 应用模式与通过 FTP 下载文件不同，FTP 是把所有的文件都集中到服务器上，所有的用户都到该服务器上下载文件，其缺点是当用户连线众多时会造成服务器工作繁忙、带宽拥挤、下载速度变慢，而且所共享的文件仅仅限于服务器上的文件；而 Napster 上所有的共享 MP3 文件都是由用户提供的，种类繁多，而且所有的文件都保存在用户的计算机上，Napster 提供了一个服务器来保存所有用户提供的 MP3 文件的目录以及其计算机的地址，当某个用户需要下载 MP3 文件时，他通过服务器查询到所需要的 MP3 文件所在的计算机地址的列表，由于同一个 MP3 文件在许多不同的计算机上都有副本，该用户只需要从任意一台计算机上下载就可以了。这种用户之间通过 P2P 的访问来进行文件共享的方式有许多优点，首先是有不会有带宽问题，因为大量的文件数据都是通过用户之间的计算机来进行交换的，不存在集中的服务器的带宽拥挤问题；其次是共享的文件内容是无穷无尽的，因为每个用户都可以在自

己的计算机上共享个人 MP3 音乐文件,非常方便,不需要把文件上载到 FTP 服务器上;再次是不需要一个庞大的服务器,充分利用了资源。因为一般的 FTP 服务器都需要有很大的硬盘来保存共享的文件,而 Napster 利用的是用户计算机上闲散的存储资源。Napster 也有服务器,但是其巧妙之处在于服务器上只保存共享文件的地址信息,所占用的存储资源相对很少,而把占据大量存储资源的 MP3 文件保存在用户的计算机上;另一个巧妙之处在于文件的交换是 P2P 方式的,不需要通过服务器进行,解决了带宽拥挤的问题。但是,非常不幸的是,由于 Napster 共享的是 MP3 音乐文件,虽然其初衷是让音乐发烧友们交换合法的 MP3 音乐文件,可是这种行为最终触及了音乐唱片公司的利益,世界各大音乐唱片公司联合把 Napster 告上了法庭,而且由于 Napster 提供的服务器上共享了 MP3 音乐文件的目录信息,此种情况作为重要论据使 Napster 在法庭上处于不利地位,导致最后的败诉。终于,风靡一时的 Napster 就此退出历史舞台,但是其巧妙的 P2P 应用模式和短时间内急剧增长的用户数量还是给人们留下了深刻的印象,为后续的 P2P 文件共享应用奠定了基础。

与此同时, OICQ 和 QQ 等国内外即时聊天应用也迅速在国内流行起来,个人拥有一个 QQ 账号就相当于拥有一个手机号码一样,只要用户在自己的计算机上安装了 QQ 的客户端软件并进行登录,就可以与因特网上任何一位安装有 QQ 客户端软件的用户进行文字聊天。实际上 QQ 的实现机制与 Napster 系统非常类似,用户通过客户端软件把自己的账号和计算机地址登记到服务器上,并通过服务器查找其他在线的用户账号和地址,然后直接进行聊天。可以说,即时聊天是一种非常典型的 P2P 应用,而且在现实中也取得很大的成功,受到全世界许多用户的欢迎。

由于文件共享和即时聊天等 P2P 应用程序的成功,越来越多的程序员在因特网上开发他们的 P2P 程序,越来越多的用户在使用 P2P 应用程序。

1.2 常见的 P2P 应用程序

现在许多常见的 P2P 应用程序往往夹杂着某种程度的集中特性,包含了完全集中到完全非集中两个极端, Gnutella 和 AOL Instant Messenger 就是典型的例子。但是这些 P2P 应用都具有分布式计算的特点,分布式计算是一种通过使用一组对等机(Peer)的计算资源来完成的应用模式,其目标往往是通过更多的计算机在更短的时间内完成任务。虽然许多应用程序也可以采用 Client/Server 的方式来实现,但是 P2P 自身的特点使这些传统的应用有了很大的提高。

1.2.1 聊天程序

聊天程序在更大范围被认为是 P2P 程序的原因是因为每个节点具有与其他节点进行直接联系的特点。聊天程序的实现办法非常广泛,往往和商业模式有关系。最典型的聊天程序包括 AOL Instant Messenger, Yahoo! Messenger, MSN Messenger 和 ICQ,无论是

在家里还是在办公室中，一般的消费者和专业人员都使用它们，主要是因为聊天程序给人们提供了比备忘录、电子邮件、电话和会议更加合适的实时通信媒介。

最流行的商业化的聊天程序往往采用的是集中式的实现方法。简而言之，就是通过一个或者多个服务器来保存上线用户的准确地址信息，并转发所有的聊天信息。这种实现方法的扩展性不是很强，而且要求服务器要有更多的存储空间来保存数据，以便支持更多的用户。但是另一方面这种集中式的解决方法可以在运营过程中准确地给一部分用户提供服务，聊天程序运营商可以利用各种服务来吸引新的用户，例如可以在聊天程序的界面上销售广告，等等。

JXTA 非常适合编写聊天程序，而且可以有多种实现方法，包括完全的集中式、代理方式到完全非集中式。而且一个非常有意思的情况是，用 JXTA 来实现一个集中式的聊天程序非常困难，而实现完全非集中式的聊天程序反而简单，因为 JXTA 平台本身已经完全解决了底层 Peer 之间的发现和通信问题。在现有的 JXTA 项目中已经有许多聊天程序的例子，包括 myJXTA 应用程序（原先称为 InstantP2P）、JXTA shell 中的 Talk 命令、HotWire 应用程序和其他几个编程指南。

1.2.2 协同工作和白板

协同工作包括一系列 P2P 程序，其目的是使得分布在不同地点的人能够通过计算机进行实时的工作协作。例如在一个工作组中，协同工作包括想法和资源的共享，而这种共享是通过交互式的通信进行的。为了实现在一个项目中的合作，项目组成员彼此间可能需要交换信息与文档资料。信息的交换可以通过 E-mail、电话或是聊天软件来实现。在一次会议中，使用一个双方可共用的虚拟白板与远方的合作者进行交流是一种非常有效、非常及时的方法。白板就相当于在通信双方面前的信息“黑板”，任何一方在上面进行文字输入或者是图形绘制，都可以被双方同时看到。在此过程中，文档可以由某一方单独起草并共享，或者是由多方联合起草，但无论采用哪种起草方式，P2P 技术都可以使得文档可以实时地进行共享，并且不依赖中央服务器的存在。

JXTA 技术已经展示它在协同工作应用程序中的能力。用 JXTA 开发出来的集成开发环境可以允许一个以上的开发者同时开发一个软件程序；另外一个协作式的 JXTA 应用程序允许几个人同时编辑同一个电子数据表。其他 JXTA 项目中开发的合作开发工具包则利用了 JXTA 进行会话、共享浏览和文件共享。

由于具有通话功能，JXTA 技术在合作开发中拥有这样一个优点：一个应用程序的开发不需依赖于服务器或其他中央管理系统。JXTA 处理新发现的节点，并保证路线通畅，在用户和交互式应用程序间交换消息。

1.2.3 游戏

游戏是 P2P 可以大有作为的领域，尤其是网络游戏。现在单机版的游戏已经不如网络版的游戏吸引人了，网络游戏使得游戏可以由许许多多的人来参与，而且是许多不

认识的人的参与，大大增加了游戏的刺激性。而在网络游戏的通信过程中，经常是玩家之间的计算机在进行直接通信，这就是 P2P 技术非常适用的原因。在游戏中使用 P2P 技术有下列优势：

- 在游戏中产生的消息传递通常在 P2P 网络中是易于处理和实现的。
- P2P 可实现更好的群体控制，而这一点正是一个中心服务器对所有的游戏者必须提供的支持。
- 游戏设计者不希望为了容纳大量的游戏玩家而购买很多资源，希望服务器的规模能够保持稳定。而 P2P 技术无疑是达到这一目的的有力手段。

JXTA 技术已在许多游戏应用方面得到了展示。一个互动式的国际象棋比赛的应用程序使用 JXTA 来传递两个棋手间的对奕过程，并将它展现给观众。国际象棋游戏和“tic-tac-toe”游戏都可以用来展示 JXTA 在小型设备上的应用。一个游戏者可以创建一个新的 JXTAPeer 组并邀请全世界其他游戏者加入他的游戏，当然也可以限制他人进入他的游戏组。

1.2.4 文件共享

文件共享这类应用程序保留了大部分 P2P 技术中有争议的应用。像 Napster 和现在的 Morpheus 与 KaZaA 等应用程序，都广泛使用于共享媒体文件，这在某些情况下带来了侵犯著作权的问题。不论一个人对当今的著作权法持何种观点，都不能忽视 P2P 在文件共享上产生的影响。这种影响会为企业带来直接的利益，这也包括那些利益已被充分发掘的企业。

P2P 的一个特点是信息倾向于被有组织地分发出去，从而使这些信息难以控制。在涉及著作权的情况下，由于文件易于快速复制而使情况变得糟糕。但是从另一方面看，这一现象也有它的好处，文件内容变得更易获得，也更容易恢复。一些项目，比如 OceanStore(USB)和 LOCKSS(Lots of Copies Keeps Stuff Safe-Stanford)精确利用了这些特点，建立了一个具有高可靠性的永久数据仓库的网络。

对某些用户文件共享实现的争论，比如 Napster，有可能会抹杀了它真正的贡献。可以考虑采用其他方式来取代 P2P 在内容出版(文件共享)方面的应用，使得文件内容易于被用户和 Peer 设备获得。有两种替代方法，一种是在一个组服务器保存信息，另一种是在 Web 站点上保存信息。在服务器中保存文档具有的优点是：这些在一个已知的地点的文档可以得到专家恰当的维护和备份。但是，实际上，这些服务器往往没有被严格地用于使那些文档可靠地为组中其他节点所访问。另外，服务器经常由于防火墙而不易被访问或根本不能被访问。结果，用户转而求助于电子邮件把文档作为附件来进行发送，但是也带来了版本问题。

另一个替代方法是建立自己的 Web 服务器。这是大部分用户遇到的主要困难。通常要求用户获得一个静态 IP 地址，保存一个域名，建立一个 DNS 入口(假设你有权使用 DNS 服务器)，安装并维护一个 Web 服务器，管理对系统的访问。相比之下，Napster 要求你所做的全部就是复制一个文件到共享目录下。而且一些 Napster 用户常对一些他