

Java 技术丛书

Java P2P 程序设计

Daniel Brookshier 等 著
常晓波 李静 译

SAMS



中国电力出版社

www.infopower.com.cn

Java 技术丛书

Java P2P

程序设计

Daniel Brookshier 等 著
常晓波 李静 译

SAMS



中国电力出版社
www.infopower.com.cn

内 容 提 要

点对点 (P2P) 网络无疑是当前最热门的话题之一。所谓P2P, 其实是一个网络模型, 它的基本概念就是任何的节点都可以作为服务器或者客户端。而JXTA是SUN推出的一个用于P2P应用开发的网络计算平台。本书由JXTA社团成员编写, 具有很强的权威性。本书旨在使读者理解P2P概念, 全面介绍了JXTA的Java J2SE参考平台实现, 另外还介绍了JXTA API和精心挑选的几个应用程序和服务实例。

本书适合希望了解P2P和希望学习JXTA的开发人员阅读。

图书在版编目 (CIP) 数据

Java P2P 程序设计 / (美) 布鲁克斯尔等著; 常晓波, 李静译.
—北京: 中国电力出版社, 2002.10

ISBN 7-5083-1310-0

I . J... II. ①布...②常...③李... III. JAVA 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 084648 号

著作权合同登记号 图字: 01-2002-4734 号

Authorized translation from the English language edition, entitled JXTA: JAVA P2P PROGRAMMING, 1st Edition, Published by Pearson Education, Inc, publishing as Sams, Copyright © 2002 by BROOKSHIER, DANIEL; BORWANKAR, NITIN; GOVONI, DARREN; KRISHNAN, NAVANEETH; SOTO, JUANCARLOS.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

本书由培生集团授权出版

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

2003 年 1 月第一版, 2003 年 1 月北京第一次印刷
787 毫米×1092 毫米, 16 开本 18.5 印张 414 千字
定价 39.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题, 我社发行部负责退换)

前 言

起初，创立 Project JXTA 的目的是把它作为一种开放源代码方面的尝试。这是一种新型而有争议的方式，其基础是 Sun Microsystems 中该工程的创立者所抱有的信念。信念之一是，最好的创新常常发生在自己的团队之外。为此，需要一种方式来鼓励 Sun 和非 Sun 的开发者。其次，要在较短的时间内获得成功，JXTA 必须成为开放式的标准。这是 Sun 的一贯立场；该公司积极在开放标准的基础上与其他公司进行合作，并开发了可以与其他公司的解决方案交互的实现。最后，任何一家单独的公司都无法取得 JXTA 这样不同寻常的成就，特别是，它基于开放式的标准。这意味着需要一种环境，包括 Sun 在内的任何实体都不具备这种环境要求的全部力量。

Project JXTA 开放源代码团体已经在短时间内取得长足的发展。在前 6 个月中，大约有 8000 名开发者参与了这项工作，这种技术和文档已经被下载 30 多万次。jxta.org 网站已经成为许多智者的聚会场所，虽然他们具有不同的背景，怀有不同的动机，但是却拥有相同的目标，那就是开发一种能够让各种对等实体相互合作的解决方案。jxta.org 团体的初步成功验证了开放源代码这一决策的正确性。本书反映了这种方式，是 JXTA 团体中的作者们相互合作的结果。

本书与 Project JXTA 之间的关系可以扩展到开放源代码工程的许多方面。例如，本书的许多作者都在遥远的地方工作，多数情况下，他们相互没有见过面。与 jxta.org 中的所有工程类似，各位参与者所完成的工作是不尽相同的。一般来讲，每位作者都是各尽所能，而主编或工程的管理者则从总体上维护版本的连贯性。本书的写作过程验证了开放源代码的“精英管理”理念“完成的工作越多，可以做的工作就越多”。这就是有关 JXTA 软件开发过程的详细介绍。

开放源代码开发也存在风险。例如，就 Project JXTA 而言，针对这种技术的发展方向存在多种观点，现在还无法确定到底应该朝什么方向发展。JXTA 是应该使用自己的协议，还是使用现有 Web 服务使用的协议（例如 SOAP）？当前的协议合适吗？当然，还存在非技术方面的风险。如果 JXTA 以开放源代码方式开发，那么是否存在足够的机会来赚取利润？是否应该现在开始？或者说，是否应该等到前途更为明朗一点的时候？更不清楚的是，如果以开放的方式完成工程，那么其他人可以更为清楚地了解自己当前所从事的工作以及所处的状态。值得冒被别人在市场上击败的风险吗？出于对这些风险考虑，我们编写了这样一本有关 JXTA 的书。减轻这些风险需要为新的机会做好准备，做出明智的判断，当新的信息被收集起来时加以采用，以便形成及时的对策。

新的技术机遇通常难以发现和利用，只有那些走在创新前列的人才能看出它们。随着连接设备的迅速发展，每种设备不仅是一种消费，而且还向网络提供了有价值的资源。分布式计算的时代已经到来，点对点（peer-to-peer）技术正是分布式计算的创新前沿。JXTA

提供了一种基础结构，可以用来开发创造性的能够与其他解决方案互用的崭新解决方案。jxta.org 提供了一个论坛，以便接纳其他思维超前的开发者、研究者、科学家、企业家和属于 JXTA 团体的业务人员所做出的努力。

大家随后可以看到，本书非常有助于理解 Project JXTA 的目标、技术和实现状况。充满激情的描述将深入剖析如何使用这种技术来创建非凡的点对点应用程序。本书是由大量专注的作者通过大量努力完成的，正如 Project JXTA 一样，尽管存在许多挑战和不确定性，但是这些作者依然坚定不移。JXTA 的早期版本连续发生了无数变化，而作者们相互合作，共同学习了新的技术领域。他们相互依赖，与 jxta.org 团体的许多成员进行的交流确保了本书内容的正确，并且能够以开发者的角度来介绍 JXTA。本书提供了一个通向 Project JXTA 的全新而实用的窗口。

请你加入 JXTA 团体（包括作者创立的团体和评论家创立的团体，他们编著了最好的书，并扩展了 jxta.org 团体）。欢迎加入 JXTA！

jxta.org Sun Microsystems 公司 Project JXTA 的团体管理者和集团市场经理

Juan Carlos Soto

于 January 2002, San Francisco, California

目 录

前 言

绪 论	1
第 1 章 什么是 P2P?	7
1.1 定义点对点	7
1.2 定义 P2P	7
1.3 P2P 概念	10
1.4 P2P 应用程序	14
1.5 P2P 的核心问题	16
1.6 P2P 相关技术	22
1.7 小结	28
第 2 章 JXTA 总览	31
2.1 JXTA 的定义	31
2.2 JXTA 的概念	33
2.3 JXTA 协议	44
2.4 点和组	45
2.5 通告	51
2.6 管道绑定协议	61
2.7 解析器	62
2.8 集合点协议	63
2.9 JXTA 标识符	64
2.10 小结	65
第 3 章 JXTA 协议	67
3.1 协议和 API	67
3.2 JXTA 的目标	68
3.3 JXTA 点和 Java	68
3.4 JXTA 协议 JAVA API 概述	70
3.5 JXTA 协议的 Java API 综述	71
3.6 JXTA 应用程序从何处开始	73

3.7	点	73
3.8	启动 JXTA	75
3.9	点发现协议 API	75
3.10	点解析器协议 API	84
3.11	点信息协议	90
3.12	点成员资格协议	92
3.13	管道绑定协议 API	96
3.14	对等端点协议	101
3.15	小结	107
第 4 章	JXTA Shell	109
4.1	为什么要使用 Shell?	109
4.2	安装和排错	109
4.3	Shell 命令	115
4.4	使用 Shell	118
4.5	添加 Shell 命令	121
4.6	小结	124
第 5 章	JXTA 的 ping 命令	125
5.1	ping	125
5.2	命令总览	126
5.3	工程总览	126
5.4	需求	127
5.5	设计方法	127
5.6	实现	128
5.7	安装	152
5.8	测试	152
5.9	对 ping 的进一步改进	153
5.10	小结	153
第 6 章	在组中工作	155
6.1	点组的重要性	155
6.2	点组管理的重要性	156
6.3	点组管理的类型	156
6.4	点成员关系管理	157
6.5	小结	174

第 7 章 JXTA 内容管理器服务 (CMS)	175
7.1 总览	175
7.2 CMS 的工作方式	177
7.3 一个简单的 CMS 示例	178
7.4 总体设计	180
7.5 初始化内容管理器	180
7.6 搜索内容	183
7.7 小结	188
第 8 章 JXTA 和安全	189
8.1 安全的重要性	189
8.2 安全是多方面的	189
8.3 P2P 网络中的安全攻击	190
8.4 JXTA 平台安全性	191
8.5 JXTA 的安全需求	191
8.6 加密工具包	192
8.7 安全问题和解决方案	193
8.8 P2P 系统中的信任关系	208
8.9 P2P 安全模型	209
8.10 小结	212
第 9 章 在点间同步数据	213
9.1 为 JXTA 设计一个 PDA 管理器	213
9.2 设计考虑事项	215
9.3 组安全与联合	216
9.4 类的设计	218
9.5 小结	233
第 10 章 JXTA Chess: 游戏编程	235
10.1 运行 JXTA Chess	236
10.2 类的总览	242
10.3 小结	258
第 11 章 JXTA Explorer	259
11.1 设计一个通告浏览器	259
11.2 设计总览	260
11.3 Explorer 类	260

11.4	PeerDiscoveryThread 类	265
11.5	AdvertisementTreeCellRenderer 类	270
11.6	运行 Explorer	272
11.7	小结	274
附录 A	JXTA 社团项目	275
A.1	参加一个项目	275
A.2	提交一个新项目	275
A.3	JXTA.ORG 开发环境	276
A.4	jxta.org 项目	276
A.5	选择项目的细节	278
A.6	商业项目	280
附录 B	XML 初步	283
B.1	XML 基础	283
B.2	自描述	284
B.3	强类型和语法验证	284
B.4	XML 中的问题	285

绪 论

在考虑如何介绍本书时，我想可以从欢迎你了解新的软件概念入手。我开发过多种软件，自从 1995 年出现 Java 以来，我一直在从事 Java 编程。我已经感受到自己对这种将改变世界的全新概念和思想的参与。Java 对我的一生影响最大，我也坚信，各种迹象表明它已经改变了计算机世界。何为 JXTA？为什么要使用这样一种全新而违反 Web 服务和客户端服务器技术的技术？

在我开始观察它的时候，我发现与作为一场革命相比，JXTA 并不是非常陌生的概念。就创造力而言，它不是一场革命（像法国大革命那样）。

与大多数发生问题的情形类似，大家憎恶那些当权者。法国人对他们的政府怀有极大的不满。在路易十四和路易十五统治时期，奢侈浪费之风盛行，与外国之间进行着不受欢迎的战争，高额的税收主要影响着普通人。法国的革命者认为必须推翻君主制和特权者。正如他们所主张的那样，要改变当政者。

点对点是对基于服务器的专制进行的一种反抗。客户端/服务器、多层以及 Web 服务器技术都是类似的概念。服务器集中了功能和资源，限制了访问，并且限制了个人访问和控制自己所拥有数据的能力。虽然这不是对公民权利的公开冒犯，但是它意味着公司的服务器占有了我自己的数据。

前进的道路上还存在障碍。富人和贵族是天生的，法国的特权者控制着资源，只有大型组织才能够购买和维护大型服务器的资源。虽然随着 Linux 的流行，现在可以创建一家小公司，但是仍然需要为带宽和其他资源投资。服务器保存着我们使用的应用程序和数据，而我们却无法控制它们。作为个人，甚至是大型组织，除非像贵族那样天生就富有，或者拥有像公司那样的资源，否则都无法掌握创建自己的服务器需要的资源。

赶走国王，然后会突然发现希望由其他人来统治。在设计自己的革命时，法国人和今天的 JXTA 开发者拥有相同的想法。1789 年 8 月 26 日，National Assembly of France 通过一份文件，通过了 *Declaration of the Rights of Man and of the Citizen*。他们的基础是美国的独立宣言。法国人的文件更多的是关于社会中的个人行为，因此它更像是点对点系统。下面介绍一些有关该宣言的条款，从而探讨一下法国的革命者和 JXTA 在哪些方面存在一致性。

人们天生就拥有自由，并且在权利上是平等的。社会分工只能建立在全体受益的基础之上（第 1 条）。

通过它们贡献的信息或独特的处理，点同样获得了社会状态。在服务器世界中，服务器几乎拥有所有资源，而客户端拥有很少资源，甚至没有资源。

所有权利法则本质上都属于国家。任何个人都不能行使任何不是直接来自国家的权力（第 3 条）。

JXTA 创立了一个团体，这里的任何一台单独的计算机都无法影响整个网络，除非其他成员点允许这么做。在某种程度上，这有点像民主政治，不过是在一种更为个人化的意义上，这是因为可以通过参与群组或应用程序进行投票。各台计算机的权利同样由点对点网络中所有计算机必须遵守的协议来授予。

在服务器环境中，客户端必须遵守服务器软件和服务器所有者的规则。当存在许多服务器时，同样会存在许多主人，致使客户端必须遵守太多不同并有冲突的规则。

自由的基础是不伤害他人；因此，除了要确保社会的其他成员享有相同权利之外，每个人可以行使的权利是不受限制的。这些限制只能由法律来确定（第 4 条）。

与现实中的自由类似，JXTA 中的自由同样难以界定。不过与客户端/服务器技术之间的关键区别在于成为应用程序一部分的能力。虽然其优点有些理想化，但是可以想像一下真正控制自己所拥有数据的能力。另外，还可以在任意时间处理自己的数据。这是自由吗？难说，但这只是一个起点。

JXTA 提倡自由，同时提倡对滥用自由者进行惩罚。即使自由的社会也有法律。网络要取得成功，需要有一些方法来确定是否伤害到了他人并对伤害他人的责任予以追究。在 P2P 网络中，每个成员造成伤害的能力被加以限制。网络冗余减轻了个体对社会造成的影响，但是与任何社会类似，网络中同样存在罪犯（或者说至少被认为是罪犯）。JXTA 引入了证书概念。如果个体不能成为守法公民，那么它的权力可能被剥夺，其证书也将失效。

服务器环境有些不同。除了拒绝服务攻击之外，好坏客户端之间没有明确的区分，主要是因为应用程序对普通用户提出了很多限制。服务器通常扮演着坏人的角色，它们是数据的保存者，甚至通过允许销售客户端的数据来破坏客户端的信任。

所有公民都有权亲自或通过代表来决定公共捐款的必要性、自由地授予这项权力、了解捐款的用途、确定税收的评估以及征收比例和模式，以及税收的持续时间等（第 14 条）。

税收应该与服务费或创建服务的成本相对应。JXTA 个体确定参与网络的程度，进而确定其硬件和其他资源的成本。与消费税类似，存在这样一种倾向：支付得越多，对网络的使用就越多。根据冗余和共享处理，所有用户都受益，而不是由于落后的硬件而受害。

用户根据如何配置和使用自己的 P2P 软件来做出自己的决策。由服务器所有者或选择的软件所带来的不当控制和苛刻要求都被消除了。

另一方面，第 14 条还表现了服务器与 P2P 技术的区别。对于服务器，必须维护基础结构。由于其成本，服务器类似于政府，它要求税收来运转，除了基于参与之外，这种做法通常很单调。在个体网络中，个体共享资源，每个个体都根据它的存在和参与级别来支付自己分摊的费用。

如果在社会中法律的遵守得不到保证，并且没有说明权力的分离，那么这种社会根本就是无章可循（第 16 条）。

这是 JXTA 的明确主张。如果不使用 JXTA 协议（我们的宪法和基本法），那么不能成为团体的成员。如果使用 JXTA 但却滥用它的团体，那么通常只会伤害到自己。

由于财产是神圣不可侵犯的权利，所以社会中的任何人都不应被剥夺这项权利，除非法律确定的公共需要明确声明这一点，此时所有者应该已经得到公正的补偿（第 17 条）。

P2P 随着 Napster 的引入而流行。遗憾的是,与此有关的是 P2P 与盗版相关联。尽管 Napster 最初的思想是只有音乐的有效所有者才能访问数字版本,但是盗版超过了合法的使用。因此, Napster 在法庭上遭到了处罚,同时用户的数量急剧减少。

P2P 网络(例如 Gnutella)同样缺乏权限管理。这些系统之所以不会像 Napster 那样被送上法庭,是因为他们确实是分布式的。然而,由于它们缺乏控制的天性,公司和 ISP 限制了他们的传播,个人用户要对犯罪和失去服务权利负责。这些系统很有可能被禁止,或者至少带来麻烦。

JXTA 的最终目标是成为合法公民并遵守版权和财产法。原因是简单的,如果不遵守法律,那么 JXTA 会成为另一个 Napster 或 Gnutella,进而被 ISP 和公司过滤掉。尊重他人的财产权利,这样才可能成为同道公民,从而才能使用 Internet 和公司的基础结构。

我们中的大多数人都生活在一种商业化社会中,要面对商业实体。只要存在不平等交易或犯罪行为,政府机构或受到影响的人就将清除那些违犯制度的人。尽管可以认为唱片公司这样的实体进行着不平等交易,但当前的法律对他们提供了保护(不包括那些不喜欢法律以及通过钻空子违犯法律的公司)。Napster 及其新的变形没有在他们公开反对法律和主动违犯法律的过程中改变任何法律。为了取得成功,我们仍然需要遵守法律制定的规则。

JXTA 的伸缩

JXTA 的另一种革命性思想就是它所提供的功能。由于不需要中心服务器以及它的成本和限制,创建更多的功能成为可能。这并不一定意味着的新的应用程序类型,只不过是比服务器环境中的范围更大而已。

电子商务目录就是说明 JXTA 应用程序可伸缩性的一个例子。通常,不需要使用大量的群集服务器来处理大量的事务。使用 JXTA,目录及其软件自动分布在每个点计算机上。不同于必须向成百万个用户显示相同目录的服务器,只需要一台 PC 来分发首份拷贝和每次更新即可。需要集中的只有定单验收和信用卡事务,甚至它们也被分布在某种范围之内。

P2P 目录存在许多优点,如节省成本、用户可以离线访问目录等等。由于用户不受连接速度的限制,不用在其他用户组成的队列中等待,所以应用程序运行得更快。向尽可能多的用户提供这种可用性,从而确保 Internet 或服务器群的事实不会再成为风险因素。

另一种伸缩特征是原始计算能力。在服务器环境中,每个客户端只能访问由所有用户共享的有限资源。使用 JXTA,每个点都拥有自己所在计算机的全部运算能力,以及与之合作的所有其他点的共享计算能力。

JXTA 是新概念吗?

迄今为止,我们已经介绍了一些大家非常熟悉的概念。在上面介绍伸缩的例子中,很容易把某些目标与分布式计算关联起来。本书所用的 P2P 示例都可能使用其他方法实现。然而, JXTA 的概念不一定要取代这些方法。JXTA 是一种平台,使用特定的协议与点对点网络中的其他 JXTA 平台通信。它不是应用程序或创建特定应用程序使用的库。使用 JXTA 是为了让 P2P 环境中的许多不同应用程序能够相互协作。与目录示例类似,其思想是从集

中的基础结构转向利用分布式系统的优点。

RMI、CORBA 和 Web Services 与 JXTA 的区别很大。它们面向客户端/服务器或有限的点对点通信。JXTA 似乎提供类似的服务，但是其内部架构存在很大不同。例如，可以实现远程方法调用。JXTA 与其他技术之间的关键区别在于要执行命令的发送可以越过防火墙之类的边界。可以把远程命令发送给计算机组或一台计算机，这要根据任务的类型来定。

JXTA 的风险

可以肯定 JXTA 与其他技术不同。JXTA 是在拿您的时间和命运打赌吗？的确存在风险。有些是新出现的，有些则是已知的。有些在本书中得到了解决，还有一些完全需要在当前的 JXTA 平台上解决。

现在最大的风险是 JXTA 将在随后的几年内不断发生变化。好的一面是开发者团体将设法保持网络的稳定，以便使他们的产品发挥作用。达到某种程度之后，开发者开始厌恶变化，即使工程以开放源代码方式进行。

在其他方面不全是好消息。P2P 系统在几方面可能存在问题。在目录示例中，向全部用户传播目录需要时间。更新和交易同样存在延时。

JXTA 革命刚刚开始。现在应该考虑创新概念。客户端，服务器的统治要跨台了。请接着往下读并加入这场革命。革命万岁！

JXTA 团体成员，Java 顾问

Daniel Brookshier

turbogeek@cluck.com

于 2002 年 1 月，德克萨斯，达拉斯

本书内容

本书仅介绍 JXTA 的 Java J2SE 参考平台实现，不介绍 C++、J2ME、Perl 或其他用来创建 JXTA 平台的语言。J2SE 版本是参考平台，最适合实验和说明 JXTA 协议。Java 也是当前用于 JXTA 开发的最流行语言。

本书要向新的开发者介绍 JXTA API 和经过挑选的应用程序和服务。我们的目标是让读者理解 P2P 概念，并使用 JXTA 创建有用的应用程序。

本书不介绍 JXTA 平台的实现细节，除非有助于说明使用它的方法。

本书面向的读者对象

本书面向的读者是那些希望了解 P2P 和希望学习 JXTA 的人。读者应该已经熟悉 Java，但不需要了解 JXTA 或点对点编程。

读完本书以后，大家应该能够使用 JXTA 和 J2SE JXTA 参考平台创建简单的 P2P 应用程序。

本书的结构

本书是根据两个目标进行组织的。第一个目标是对 P2P 和 JXTA 进行总体介绍。第二

个目标是使用 JXTA 创建应用程序。最后，本书介绍了一些具体的应用程序，它们的目标是加深读者对 JXTA 的理解。同时，本书也说明了如何编写更加完整的应用程序。这样组织是为了让读者概括了解 JXTA，然后理解如何使用它的各个部分。

Web 资源和示例代码

可以从 www.sampublishing.com 上下载本书示例的源代码。打开该页面之后，在搜索框中输入本书的 ISBN 编号（0672323664），以便访问有关本书的信息和源代码链接。

本书的许多例子都使用了 NetBeans IDE。可以在 www.netbeans.org 上得到 NetBeans IDE。由于 Sun Microsystems 的 Forte 派生于 NetBeans，所以也可以使用 Forte。也可以使用自己喜欢的 IDE 编辑器，但是 ANT 脚本都是使用 NetBeans 和 Forte 编写的。

该站点上还有一些文件，可以和 NoMagic（www.MagicDraw）的 MagicDraw 一起使用。MagicDraw 是使用 Java 编写的，可以运行在大多数与 Java 兼容的平台上。演示版本允许浏览和打印本书使用的 JXTA 图解，但是不允许保存更改。由于 MagicDraw 文件遵守 XML 中 UML 表示法使用的标准，所以也可以使用其他支持该标准的 UML 工具（绘制结果可能看起来有所不同）。

Daniel Brookshier

1

什么是 P2P?

本章内容

- 1.1 定义点对点
- 1.2 定义 P2P
- 1.3 P2P 概念
- 1.4 P2P 应用程序
- 1.5 P2P 的核心问题
- 1.6 P2P 相关技术
- 1.7 小结

Daniel Brookshier 和 Juan Carlos Soto 著

JXTA 是一种点对点 (peer-to-peer) 网络平台, 不过到底什么是点对点呢? 它是如何定义的? 有什么问题? 这种技术有什么应用? 本章将回答这些问题, 并介绍其他相关内容。点对点网络与基于服务器的技术存在很大区别, 理解这种区别是创建应用程序之前需要掌握的基本知识。

1.1 定义点对点

点对点 (peer-to-peer, P2P) 计算不是新概念。可以从最初连接两台计算机时说明这一点, 两台相连的计算机就形成了 P2P 网络。邮件服务器、网络新闻服务器 (NNTP) 和域名服务器 (DNS) 都运行于点对点网络中。例如, 为了发送、路由和接收电子邮件, 电子邮件服务器之间要直接相互作用, 可以被看成是 P2P 网络。

尽管 DNS 依赖于名称登记员和权威名称服务器, 但也可以将 DNS 服务器连接成 P2P 网络。最终由一定数量相互协作的计算机以分布式方式来共同完成名称解析, 进而提高了可靠性和性能。

尽管在 Internet 之前已经出现了这方面的例子, 但是术语 “P2P” 出现得相对较晚。有些人把 P2P 当成 Gene Kan 和其他早期的 Gnutella 先驱。随着 20 世纪 90 年代点对点计算的宣传以及媒体的关注, 以一种更为简洁的方式向出版社和分析家们解释这种现象就变得非常容易了。

然而, 为 P2P 提供一种简洁的定义却非易事。因为不仅要说明 P2P 应用程序的组成部分, 而且还有许多相互竞争的 P2P 协议和实现, 它们以截然不同的方式运转。现在给出 P2P 的定义, 以便大家掌握基本概念。

1.2 定义 P2P

说明 P2P 的最好方式是指出它不是什么, 而不是局限于它是什么。P2P 不是要消除服

务器。它不是一种单独的技术、应用程序或商业模型。最有争议的地方可能是它不应该由集中与分散的程度来严格界定。

P2P 网络中的集中制由中心目录（如 Napster）组成。当用户查找音乐时，Napster 相当于传统的客户端服务器；当用户传输文件时，它相当于 P2P 网络。这意味着系统利用了这种事实：可以方便地创建集中的音乐文件数据库以及它们的定位，但是托管音乐却是非常困难和耗资的。

Gnutella 是完全分散的 P2P 网络。除了共享的内容之外，点之间没有区别。Napster 中的目录服务由 Gnutella 中的点共享。

JXTA 混合了集中和分散，同时寻求一种适当的折衷。JXTA 的立场是 P2P 网络中的一些服务最好由一组有限的点完成。尽管不是完全类似，但是电子邮件服务器和电子邮件客户端是可比的概念。电子邮件服务器负责电子邮件的临时存储，并在其他电子邮件服务器之间路由邮件，从而使它们到达自己的目的地。JXTA 具有相同的功能，所不同的是电子邮件不能使用分散和集中的点所组成的集合来路由到点对点之间的所有通道。

一般来讲，P2P 更是一种计算形式，它使得网络交互更加均衡。即使可能存在集中服务，最终的用户点也是应用程序的重要焦点。如果集中服务被分散，例如电子邮件，那么系统受网络问题影响的可能性就会减少。Napster 是单纯集中的终极示例，如果主服务器失效或者被断开连接，那么所有点对点功能都会丧失。Gnutella 之所以不同，是因为即使单独的点被删除，也不会显著影响网络的质量。

虽然 P2P 不是新概念，但是现在有许多因素使 P2P 应用于许多应用程序。这些因素包括连接设备的迅速发展、可供带宽的迅速增加、计算能力的飞速发展、更大的存储能力以及信息在网络边缘的增值。

1.2.1 个人计算机

现在的个人计算机（Personal computer，PC）通常配置 800MHz 以上的 CPU、128MB 以上的 DRAM 和 20GB 以上的硬盘。计算机不断变得更快，拥有更多的磁盘空间，并且变得更廉价。计算机不再受到资源的限制（只受到技术的限制）。即使达到晶体管的最小尺寸进而最终否定摩尔定律（每过两年，CPU 的速度就翻一倍），公司也只需制造更大的芯片和增加并行性。

当今的 PC 有时甚至比几年前的服务器计算机还要强大。PC 和服务器之间的惟一区别就是服务器可以有效地处理更大的 I/O 带宽。虽然其他性质也可区分 PC 和服务器，但这些性质主要与管理、可靠性和伸缩能力有关。在 CPU 和通常所说的存储及内存方面，它们几乎是相同的。

应用程序及其对更快速度的需要推动了对这些更快 PC 的需要。大多数应用程序没有使用 PC 的全部资源，所以存在许多富余能力。P2P 应用程序和协议使用这些富余能力和存储空间来创建系统，这些系统中的全部计算都由点完成，而不是服务器。

1.2.2 服务器

真正的问题在于综合体（equation）中的服务器一端。即使经济低迷，涉足 Internet 的