



Sun 公司核心技术丛书

精通 RMI

——Java与EJB企业级应用开发

Mastering RMI

*Developing Enterprise Applications
in Java and EJB*



(瑞典) Rickard Öberg 著

王海卫 等译



机械工业出版社
China Machine Press



中信出版社
CITIC PUBLISHING HOUSE

精通RMI

—Java与EJB企业级应用开发

本书是关于远程方法调用 (Remote Method Invocation, RMI) 的权威级编程指导书。作者有丰富的Java和EJB编程经验, 书中清晰地解释了为何需要RMI来创建企业级分布式应用, 以及如何使用RMI的各种优秀功能建立容错并且灵活的程序。

本书从最基础的RMI架构教程开始, 接着描述创建RMI应用的整个过程。书中提供了完整、可运行的代码, 读者可以根据自己的需求进行裁剪, 同时书中还示范了如何将RMI的强大功能与Jini和EJB结合共同发挥作用。

主要内容如下:

- 创建一个基本的RMI应用, 并采用动态类载入和Java命名及目录接口 (Java Naming and Directory Interface, JNDI) 来扩充其功能
- 使用自定义连接管理和激活技术来增强程序功能以及客户/服务器间通信
- 定义和创建高级聊天室以及移动代理应用
- 应用Jini来增强网络服务和设备之间的协作
- 开发EJB组件来用于类似于事务处理、数据库和安全等应用

随书光盘包括:

- 本书实例的所有源代码
- 运行EJB实例的JBoss应用服务器

相关Web站点 (www.wiley.com/compbooks/oberg/) 内容:

- RMI资源的链接
- 源代码例子的更新

作者介绍:

Rickard Öberg 是Java和EJB方面的世界顶尖专家, 已经实现了三个EJB服务器, 他是开源应用服务器JBoss项目的主架构师。Öberg是瑞典Java编程竞赛“Software Reuse Café”的两度折桂者, 这使得他成为瑞典最好的Java程序员之一。

适用水平: 中、高级



ISBN 7-111-12152-X



华章图书

网上购书: www.china-pub.com

北京市西城区百万庄南街1号 100037
读者服务热线: (010)68995259, 68995264
读者服务信箱: hzedu@hzbook.com
<http://www.hzbook.com>

ISBN 7-111-12152-X/TP · 2671

定价: 35.00 元 (附光盘)

Sun公司核心技术丛书

精通 RMI

Java与EJB企业级应用开发

(瑞典) Rickard Öberg 著

王海卫 等译

本书是讲述RMI技术的经典著作，清晰地展示了设计和创建RMI应用中涉及到的思想，包括最常用的设计模式。详细讲述了如何利用RMI的所有高级特性，如何创建RMI系统，实例丰富，还介绍了常用的客户/服务器结构、远程调用的本质以及与RMI相关的技术。作者把既简单又很有说服力的应用与理论相结合，全书由浅入深，概念叙述和应用分析并举，使读者不仅“知其然”，而且“知其所以然”。对于所有关注Java技术、从事企业应用实现的读者来说，是一本内容详尽、技巧高超的参考书籍。

Rickard Öberg: Mastering RMI: Developing Enterprise Applications in Java and EJB
(ISBN:0-471-38940-4)

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

Copyright © 2001 by Rickard Öberg.

All rights reserved.

本书中文简体字版由约翰·威利父子公司授权机械工业出版社与中信出版社合作出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-2003-1020

图书在版编目（CIP）数据

精通RMI: Java与EJB企业级应用开发 / (瑞典)奥博格 (Öberg, R.) 著; 王海卫等译.
-北京: 机械工业出版社, 2003.7

(Sun公司核心技术丛书)

书名原文: Mastering RMI: Developing Enterprise Applications in Java and EJB

ISBN 7-111-12152-X

I. 精… II. ①奥… ②王… III. Java语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2003)第036050号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 于杰琼 武恩玉

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2003年7月第1版第1次印刷

787mm×1092mm 1/16·14.5印张

印数: 0 001-4 000册

定价: 35.00元(附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

这是一本介绍远程方法调用（Remote Method Invocation, RMI）的书籍。书中展示了设计和创建RMI中用到的基本思想，阐述了如何利用RMI的高级特性开发企业级网络应用。20世纪90年代，Java技术开始飞速发展，由于Java必须能支持已经存在的应用和数据，因而对企业应用提出了新的问题。同时，信息技术和网络技术的飞速发展，也为企业对内和对外优化业务流程管理，实现数据流和业务运作的自动化，以及在互联网上开展电子商务提供了更强有力的工具。RMI正是在这种环境下应运而生的。

全书共分为四部分：第一部分介绍了客户/服务器结构。第二部分引导读者完成基本RMI应用的创建过程。第三部分讲述了RMI的一些更高级的特性。第四部分讨论了建立在RMI顶层的两种技术：Jini和Enterprise JavaBeans。书中在介绍RMI基本思想和设计模式的同时，注意与实际应用相结合。本书也使用了介绍开发应用书籍中惯用的HelloWorld典型示例，并随着内容的深入，不断对该例子进行调整，即便在讲述高级特性时，也能使读者接受。

作为Java和EJB方面世界顶尖的专家，作者使用的语言简单生动，容易与读者产生共鸣。在书中常以自问自答的方式，讲述RMI技术中需要注意的一些问题。受益于作者是一名极富经验的Java和EJB开发专家，本书清晰地解释了为何需要RMI来创建企业级分布式应用，以及如何使用RMI的各种优秀功能，使用Java和EJB建立容错并且灵活的程序。从最基础的RMI架构教程开始，书中描述创建RMI应用的整个过程。书中提供了完整、可运行的代码，读者可以根据自己的需求进行裁剪，同时书中还示范了如何将RMI的强大功能与Jini和EJB结合共同发挥作用。

随书光盘还提供了本书实例的所有源代码，附有完整的build结构和make文件以及运行EJB实例的JBoss应用服务器，供读者进行RMI内容的实战演练。同时在John Wiley出版社的Web站点上还提供了相应的RMI资源的链接以及书中源代码例子的任何更新。

全书由王海卫、王宏伟、刘新喜、王海峰、初元芳、李位星、李新鹏、邵月慧、莫伟锋、王慧、李兴华、丁敏、杜庆平、刘淑云、董家国、张毅华、贾兴权、郭卫萍、王晓瑞、张玉峰、刘浦、韩军海等翻译。前导工作室全体工作人员共同完成了本书的录排、校对等工作。由于时间仓促，且译者的水平有限，在翻译过程中难免会出现一些错误，请读者批评指正。

宋 涛

2003年2月

前 言

本书是一本有关RMI（Remote Method Invocation，远程方法调用）的指南。本书将向读者展示设计和创建RMI应用中涉及到的思想，包括最常用的设计模式。它将教读者如何利用RMI的所有高级特性，并在如何创建RMI系统方面举出了很多例子。它还将展示RMI与建立在其上的其他技术之间的联系。读完这本书后，就会知道如何在自己的工作中更好地使用RMI。更为重要的是，读者将不仅知道要做什么，还会知道为什么应该那样做（知道为什么应该用某种方法做某些事是很重要的，其中包括RMI）。

总的来说，分布式计算相当复杂，尤其是RMI。它们涉及到很多东西，使得它们比其他的Java编程更加困难。虽然本书的目标是使RMI编程不再像以前那么复杂，但应该认识到其内部的复杂性。本书将从一开始就声明这一点，因此除了Java的一些知识外，在阅读本书时，不需要知道其他的任何东西。我们推荐读者应该首先了解一些网络及其相关原理，但这不是必须的。

虽然RMI本身就是非常有用的，但当用其他技术实现时，其功能会更加强大。本书中用到了相当多这样的技术，在阅读本书的时候读者很可能会想要下载和读取这些技术的说明。在本书最后的附录中提供了这些说明的出处。相关的Web站点上也包含了这些说明和技术的链接。

本书内容

本书共分为四部分：

第一部分介绍客户/服务器（Client/Server）结构。第1章将给客户和服务器下定义并看一下n层系统的思想。此后，将讨论在一般情况下创建远程方法调用机制所需要的准备工作，以及它要处理的问题。随后，将看几个使用目前这种客户/服务器结构的例子，并分析一下它们是如何解决定义的问题的。第2章是RMI结构的速成班教程。该章将遍历RMI的整个设计过程，并看一下它是如何实现必须注意的任务的。还将讨论RMI实现的几个内部因素，比如线程、序列化和动态类加载（classloading），从而可以使得在使用RMI的时候，对它内部发生的情况有一个清楚的认识。这两章为读者应用其他章节的内容提供了坚实的基础。经验丰富的读者可能想跳过第1章。然而，建议所有读者至少应该浏览一下第2章的提示和警告部分，因为它们含有很多有趣的提示和技巧。

第二部分将引导读者完成基本RMI应用的创建过程。在第3章一开始就创建一个简单的HelloWorld示例，用来展示RMI应用所需要的最少步骤。接下来讲述如何设置应用的结构以便于打包和配置。还介绍一个从客户到服务器再从服务器到客户的调用。这将使读者对RMI调用流有一个清楚的印象，这是理解一般RMI的一个很重要的因素。第4章继续第2章的HelloWorld示例，并对它进行了扩充，把原来的客户变为一个applet，使用Web服务器的动态类加载功能，并借助Java命名及目录接口（Java Naming and Directory Interface，JNDI），使用标准化的命名API把命名服务隐藏起来。具有RMI经验的读者可以跳过第3章直接进入第4章，但是对RMI不熟悉的读者

应该阅读这两章的内容。

第三部分讲述的是RMI的一些更高级的特性。仍将继续使用本书的前面章节所举的HelloWorld示例，然后由每一章展示一些特性来增强这一示例。所描述的特性是自定义连接管理（又名自定义套接字工厂）和激活。自定义套接字工厂允许控制客户与服务器之间的网络连接的处理方式。激活还可以使得在需要的时候激活自己的服务器，并提供远程对象分支，可以在服务器出现故障时继续存活。自定义连接管理是提高服务器安全性的有效方法，激活是在编写Jini时的一种非常有用的技术。因此如果想要使用这些特性，就会发现这些章节是很有用的。

第四部分讨论了建立在RMI顶层的两种技术：Jini和Enterprise JavaBeans（EJB）。Jini在最初推向市场时是作为一种技术推广的，使用这种技术可构建以一种灵活容错的方式进行合作的网络化设备。结果发现Jini可以应用的领域远不止于此。所有想要以一种容错方式与其他服务合作的网络服务（也就是几乎所有的网络服务）都可以使用Jini。我们将会看到Jini提供的神奇的特性使得它的应用如此之广，还将对HelloWorld示例进行扩展，以展示Jini是如何改善这种情况的。最后一章讲述EJB。EJB的目的是为创建可移植的Java组件提供一个模型，这些组件有助于读者实现包含事务处理、数据库和安全等功能的复杂业务逻辑。它还可以帮助读者将这样的组件以一种灵活的方式组装到应用中，从而使这些组件尽可能地可复用并尽可能松散地连接在一起。还将看一下EJB服务器提供的服务以及开发EJB组件需要进行的准备工作。最后，我们创建一个简单的EJB组件演示一下涉及的步骤。

文中的插图

本书所有插图（除了屏幕界面）使用的都是统一建模语言（Unified Modeling Language, UML）。UML是创建软件图表的一种标准方法。如果读者不熟悉UML，推荐看一下Grady Booch等编写的*Unified Modeling Language User Guide*（Addison-Wesley, 1999），书中讲述了如何高效地使用UML。UML是任何严谨的开发人员的工具箱中的一个非常重要的工具，我们强烈建议开发人员使用这种工具把思想和设计传达给其他开发人员。本书中的图表是用TogetherSoft公司的Together/J UML工具创建的。

使用的例子

本书使用经典的HelloWorld示例作为基础。这使得我们可以把精力集中在需要使用RMI的内容上，而不必为细节分神。虽然这可能不是非常生动的例子，但它的确为读者提供了源代码，读者可以用它作为自己应用的基础。

本书还选择了包括两个完整的样例应用。其中一个移动代理系统，突出讲述了RMI的动态类加载特性的广泛用途。另一个样例是一个聊天系统，显示了很多重要的设计模式，在设计真正的RMI应用时很有用。这两个例子应该给读者一个更加完整的印象，使读者可以知道能够设计和实现出多大的RMI应用。

我们最后展示一个Jini示例和EJB示例，还是采用HelloWorld的样例代码，这对于向读者展示这两种技术能够完成的内容已经足够了。

随书光盘

在本书所附光盘中，可以找到书中例子的所有源代码。它们都有完整的build结构和make文件。这里选择使用非常好的Ant编译工具，Ant是一个Apache工程。可以从<http://jakarta.apache.org>中找到有关该工具的更多信息。

EJB的例子可以使用JBoss应用服务器运行，在随书光盘中有JBoss。JBoss对所有用途都是免费的，因此可以放心地在JBoss上建立应用，而无需担心许可问题。EJB的一个最重要的目标是允许组件在不同的EJB服务器之间移植，从而使得例子中不包含任何会把它约束在本JBoss服务器上的成分。有关JBoss的更多信息，可参见它的主页www.jboss.org。

至于Jini的例子，则需要从Jini的www.sun.com/jini/主页下载并安装Jini 1.1（或者更新版本）工具箱。

书中所有的例子均是在Sun的Java Development Kit 1.3版本上开发和测试的。在编译和运行这些例子之前需要下载和安装此JDK版本（或者更新的版本）。有关例子的更多信息，请参阅随书光盘的文档。

相关Web站点

本书有一个相应的Web站点，用它来通知读者本书的更新，包括：

- 文字的错误改正
- RMI资源的链接
- 源代码例子的更新

建议读者在继续阅读前先检查一下该Web站点，因为自从本书出版后，网站上可能已经有重大的变动。该站点是www.wiley.com/compbooks/oberg。

反馈信息

非常乐于听取任何读者的RMI工程，以及使用RMI、Jini或EJB的经验。欢迎你借助E-mail告诉我这样的例子、案例研究、失败教训或有帮助的技巧。得到你的同意后，我将把其中的一些放到Web站点上与其他人共享。你可以通过rickard@dreambean.com和我联系。

目 录

译者序

前言

第一部分 基本原理和结构

第1章 客户/服务器结构	3
1.1 客户/服务器基础	3
1.1.1 客户和服务器的定义	3
1.1.2 为什么使用客户/服务器结构	4
1.1.3 n层系统	6
1.2 网络化计算基础	7
1.2.1 网络通信	7
1.2.2 编组	9
1.2.3 代理	11
1.2.4 命名	14
1.3 其他类型的客户/服务器系统	15
1.3.1 万维网	16
1.3.2 远程过程调用	17
1.3.3 CORBA	18
1.4 小结	21
第2章 远程方法调用的本质	23
2.1 什么是RMI	23
2.1.1 RMI不能进行的工作	23
2.1.2 RMI原理	24
2.1.3 RMI与普通Java的区别	25
2.2 RMI/JRMP结构	26
2.2.1 存根	26
2.2.2 编组	33
2.2.3 RMI线程和网络连接管理	45
2.2.4 分布式垃圾收集	48
2.2.5 命名	49
2.3 小结	50

第二部分 建立RMI应用

第3章 HelloWorld应用	55
3.1 应用综述	55
3.2 远程接口	57
3.3 服务器	58
3.3.1 远程对象	58
3.3.2 管理器	60
3.4 客户	61
3.5 远程调用	63
3.6 应用	65
3.6.1 封装	65
3.6.2 启用命名服务	66
3.6.3 启动服务器	66
3.6.4 启动客户	67
3.7 小结	67
第4章 改进HelloWorld示例	69
4.1 改进后的应用综述	69
4.2 服务器	71
4.2.1 远程对象	71
4.2.2 管理器	72
4.3 客户	76
4.4 应用	78
4.4.1 封装	78
4.4.2 启动服务器	79
4.4.3 启动客户	79
4.5 小结	80
第5章 自定义连接管理	81
5.1 自定义连接管理综述	81
5.2 RMIServerSocketFactory接口	82
5.3 RMIClientSocketFactory接口	83
5.4 一个简单的连接工厂	84

5.4.1 IdentityServerSocketFactory类	84
5.4.2 IdentityClientSocketFactory类	85
5.4.3 IdentityServerSocket类	86
5.4.4 IdentitySocket类	87
5.5 使用自定义连接工厂	89
5.6 小结	90
第6章 激活	91
6.1 激活综述	92
6.1.1 激活组	93
6.1.2 可激活	94
6.2 激活示例	95
6.2.1 Setup类	95
6.2.2 服务器类	99
6.2.3 客户类	100
6.2.4 运行示例	103
6.3 小结	105

第三部分 创建真正的应用

第7章 开发一个聊天应用	109
7.1 聊天设计综述	110
7.1.1 TopicServer远程接口	112
7.1.2 Topic远程接口	113
7.1.3 MessageListener远程接口	113
7.1.4 ListenerInfo类	115
7.1.5 TopicInfo类	116
7.1.6 Message类	118
7.1.7 Listener类	121
7.1.8 TopicServerImpl类	121
7.1.9 MessageQueue类	129
7.1.10 Worker类	131
7.1.11 Main类	135
7.1.12 ChatClient类	138
7.1.13 ChatGUI类	147
7.2 小结	149
第8章 开发移动代理应用	151
8.1 移动代理设计综述	152

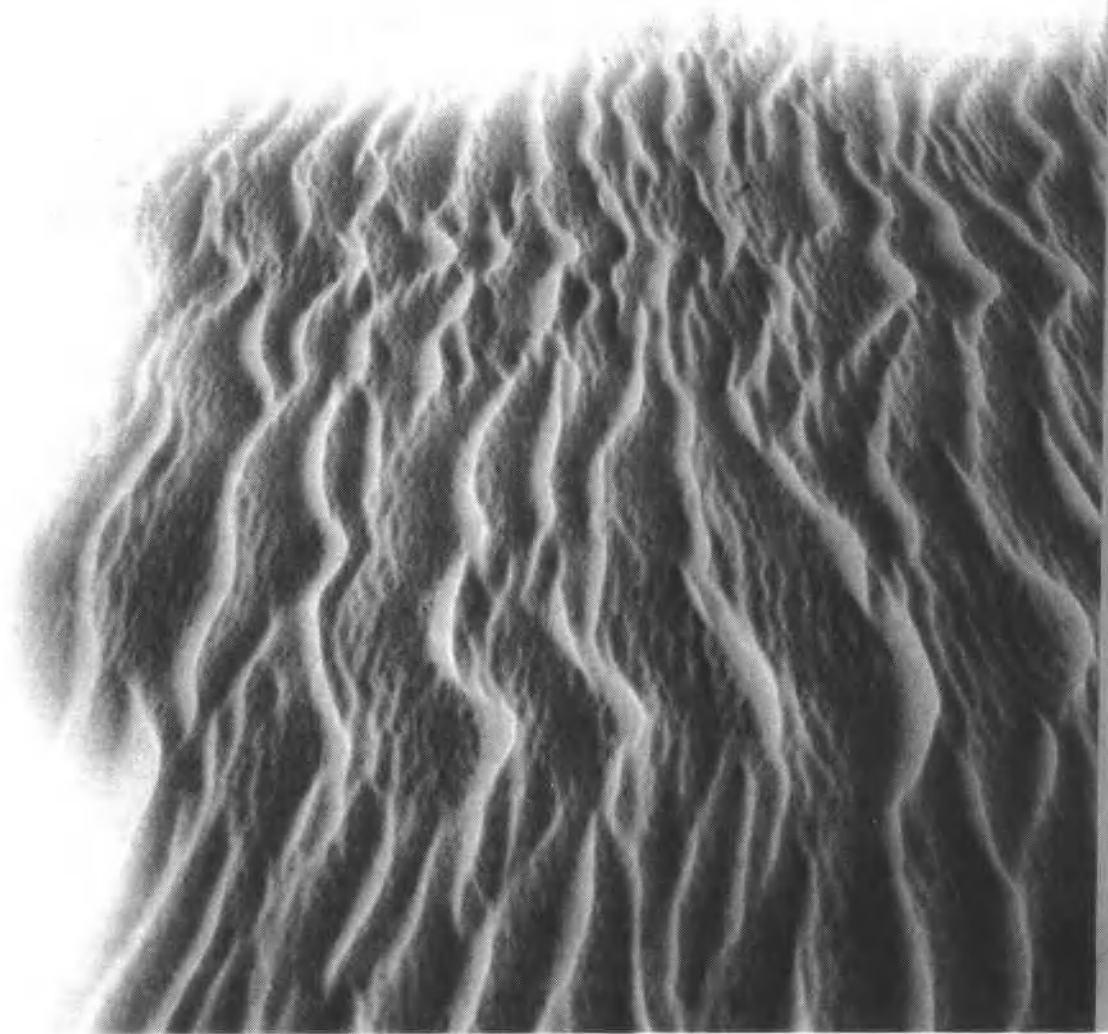
8.2 代理接口类	154
8.2.1 AgentHost接口	154
8.2.2 NoSuchAgentException异常	155
8.2.3 Agent接口	155
8.3 代理主机实现	156
8.3.1 AgentHostImpl类	156
8.3.2 Main类	158
8.4 移动代理实现	160
8.4.1 HelloAgent接口	161
8.4.2 HelloAgentImpl类	161
8.5 代理客户实现	167
8.6 代理管理器实现	170
8.7 小结	176

第四部分 高级应用

第9章 RMI和Jini	179
9.1 综述	179
9.1.1 发现	179
9.1.2 查找	180
9.1.3 租借	181
9.1.4 远程事件	181
9.1.5 事务处理	182
9.2 HelloJini示例综述	184
9.3 HelloJini服务	185
9.4 HelloJini客户	191
9.5 运行HelloJini示例	196
9.6 小结	198
第10章 RMI和Enterprise JavaBeans	199
10.1 综述	200
10.1.1 EJB容器	201
10.1.2 EJB组件	203
10.1.3 无状态会话bean	204
10.1.4 有状态会话bean	204
10.1.5 实体bean	205
10.2 HelloEJB示例综述	205
10.3 HelloEJB组件	205

10.3.1 HelloEJB Home接口	206	10.4 HelloEJB客户	211
10.3.2 HelloEJB 远程接口	206	10.5 小结	214
10.3.3 HelloEJB Bean实现	207	附录 更多阅读材料	217
10.3.4 HelloEJB 配置描述文件	210	光盘内容简介	221

第一部分 基本原理和结构

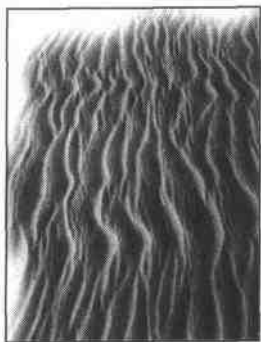


本书的第一部分解释了客户/服务器结构的基本思想和原理。理解了这样的结构所需的组件后，接下来介绍远程方法调用（Remote Method Invocation, RMI）的架构。随后介绍RMI的不同组成部分。这会使读者能很好地理解RMI的设计和引用实现，从而更加容易地开始开发各种应用。

读者可能会问为什么一本写RMI的书会有一整章是专门介绍基本的客户/服务器结构的。我已经使用RMI技术有相当长的时间了，我经常看到出现很多问题，其原因只是因为开发者不理解底层原理。在学习的过程中，知道“如何去做”是不够的，还需要理解为什么这样做。这就是第1章要讨论的内容。只要看一下为了创建一个RMI应该解决的问题，就会发现RMI的每个部分都是为了解决这些基本结构问题其中的一个。

第1章开始研究所有客户/服务器结构的原理以及使用这些结构的原因。大多数客户/服务器系统的关键思想是相同的，还介绍了客户/服务器结构的历史。理解了这些基本原理之后，才能更好地理解RMI的各种细节。

第2章详细解释了RMI，它的设计、标准Java开发工具箱（Java Development Kit, JDK）提供的RMI实现以及Java远程方法协议（Java Remote Method Protocol, JRMP）。还研究了RMI/JRMP的一般设计以及需要用来实现远程对象的一些RMI基本理念。该章还展示了RMI的各个部分是如何与第1章出现的基本概念相联系的。



第1章 客户/服务器结构

第1章首先介绍客户/服务器结构的特性：使用的组件是什么，通用思想是什么以及为什么要使用它们。最后举出几个客户/服务器结构最常用的一些例子。

首先介绍客户/服务器结构的基本定义是什么。很可能读者已经对客户/服务器的结构有了一定的了解，让我们从温习这个基本定义开始，以确保掌握正确定义。

随后介绍在创建分布式结构（如RMI）中涉及到的最通用的思想。前面已经指出这是每个分布式结构的核心部分，当研究RMI的具体细节时，就会看到RMI设计的各个部分是如何与这些基本的构建模块联系到一起的。

本章最后将展示几个不同的客户/服务器结构，并简单地介绍它们是如何实现在本章描述的思想的，这样能更加容易看出RMI与其他结构的相同与不同之处。

如果读者想直接跳到第2章，应该首先简单看一下1.2节的内容，因为这里会提到在描述RMI结构时需要用到的思想。

1.1 客户/服务器基础

首先看一下什么是客户/服务器结构，并给出客户和服务器的定义。多数人对于它们是什么都有一个直觉的概念，但随着Internet的引入，客户和服务器的两个词经常用于不同的环境中。它们的含义也随着用途的不同稍微有些差别。

如果对客户/服务器结构到底是什么有了清楚的认识，这里就简单地看一下为什么希望使用这种结构，或者是否有其他的替代方案。

1.1.1 客户和服务器的定义

如果要看客户和服务器的主要特性，可以使用下面的定义，该定义在大多数场合都是适用的：

- 服务器是一个提供若干种服务的软件程序，可以有零个或多个客户使用这些服务。
- 客户是某种使用至少由一个服务器提供的最少一种服务的对象。

这些定义如图1-1所示。

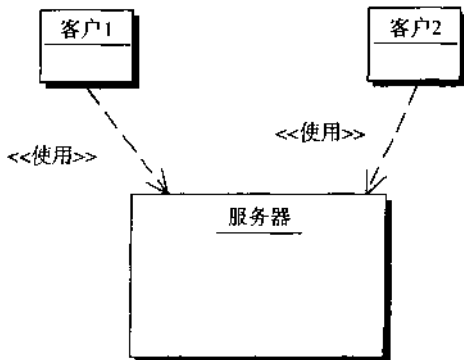


图1-1 客户和服务

这就是客户和服务涉及到的所有东西。正如读者可以看到的，这种定义的范围相当广泛，因而可以适用于很多不同的场合。其中需要注意的最重要的一点是，这种定义允许一台服务器成为另一个服务器的客户。实际上这是很常见的。例如，一个提供文档的Web服务器可由客户浏览器通过HTTP（HyperText Transfer Protocol，超文本传输协议）访问，该服务器可以使用一个登录服务器存储所有访问的列表。它也可以使用磁盘服务器，用来提供需要传输的文档的物理存储。

在前面的定义中，没有提及客户/服务器结构是建立在不同物理机器之间的网络通信的基础上的。然而，因为本书主要讲述RMI结构，所以假设这种条件是存在的。虽然服务器和客户可以在同一个物理主机上执行，但网络化的环境中通常的情况不是这样的。

1.1.2 为什么使用客户/服务器结构

客户/服务器结构现在已经变得越来越流行了。传统情况下是桌面应用程序，比如办公自动化套件，已经越来越多地接受客户和服务思想。使用客户/服务器结构的一些动机如下：

- **资源共享** 使用客户/服务器结构的最普通的原因是因为位于一台计算机上的资源可以在若干个客户之间共享，此时在客户和物理用户之间通常有一个一对一的映射。这样的例子是World Wide Web（WWW，万维网），WWW中的文档可以由几个冲浪者通过特定的Web服务器几乎同时访问。
- **访问远程资源** 这一条与前面的一条相关，但也有些不同之处。在一些情况下，特定的资源只能由特定的物理主机访问，这意味着访问该资源的惟一途径是使用允许访问该资源的客户/服务器结构。例如，现在流行的消息系统ICQ在每个用户的计算机上创建一个服务器，它可以为位于该主机上的用户接收消息。这种服务器——实际上是ICQ客户，反过来使用位于中心的ICQ服务器以便将信息发送给其他用户。因此在这种情况下，用户主机上的

ICQ程序和ICQ主服务器都既是服务器，也是客户。这种结构如图1-2所示。

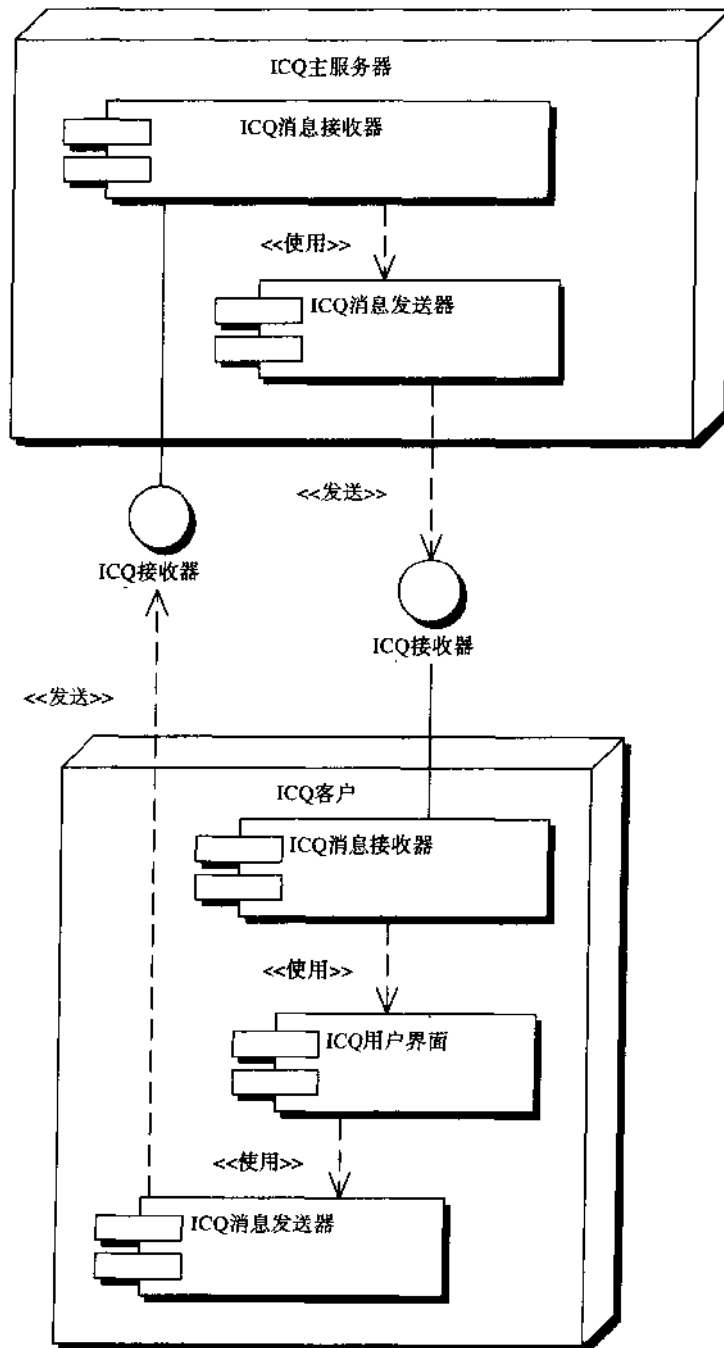


图1-2 ICQ结构

- **代理责任** 可以把很多客户经常使用的一些任务放在所有客户都可以访问的服务器上。这样做可以使这些常用的任务中心化，从而使得它们更加便于维护和控制。
- **性能分布（带宽、CPU和内存）** 有时某个任务执行起来可能会非常昂贵。如果每台计算机都有足够的硬件来执行这一特定任务，这可能会是一个相当昂贵的解决方案。相反，投资建立一个高性能的服务器而保持客户的硬件开销最小，从而降低所有开销通常是一种更好的做法。在这种方案下增加客户也变得更加便宜。
- **安全** 一些应用可能会需要敏感数据。在这些情况下，把数据放在每台可以访问它的计算机上不是一种好的做法。想要读取该数据的人能够轻易盗用这台计算机，并且可以简单地检查该机的存储以获得这些数据。可以把该数据放到一个高度安全的服务器上，并在它的使用上加上严格的安全限制。想要访问该数据的用户必须使用某种方式确定其身份，还可以让不同的用户访问不同种类的数据。

正如读者可以看到的，最基本的规则是如果需要使用网络以便实现一个应用，就需要使用客户/服务器结构。

客户/服务器结构的使用以及它们的设计和实现是不断变化的。最先出现的客户/服务器结构中的一种结构是大型计算机——一个拥有大量计算资源的巨型服务器，这些资源可供没有或只有很少计算能力的客户终端使用。当然，现在的设计和实现会复杂得多。

当听到客户/服务器作为一种典型编程模式是否还存有活力的讨论时，我觉得很有趣。有些人认为个人电脑的引入会使客户/服务器编程模式过时。然而，正如大家都知道的，Internet改变了所有这一切。现在客户/服务器结构非常庞大，并且它的新的分布式标准和产品正在以一种非常快的速度发展。

只要哪里有网络，哪里就会有客户/服务器结构。它们无疑将与网络模型早期使用的那些结构存在着很大的差别，但其基本思想（及其存在的问题）基本上仍然相同。

1.1.3 n层系统

可以把客户/服务器结构描述成一个两层系统。一层是客户，另一层是服务器。每一层都有严格定义的责任，它们之间的接口非常严谨。也可以把这些层扩展为任意深度，称之为n层。为什么会想这么做呢？一个非常普通的原因是为了减小每一层的职责范围，并且这样做，可以最大限度地复用其中的软件设计。

下面看一个典型的例子。假设你有一个SQL服务器，已经把它高度优化来存储结构化的数据。这台服务器的客户可以是终端用户使用的计算机，但是允许该终端用户访问SQL服务器可能不是个好主意。不允许终端用户访问SQL服务器有两个原因，一是安全方面的考虑，二是可能会