

21世纪重点大学规划教材

李文军 万海 周晓聪 李师贤 编著

分布式计算



机械工业出版社
CHINA MACHINE PRESS



配电子教案

九江学院图书馆



1825076

1509651

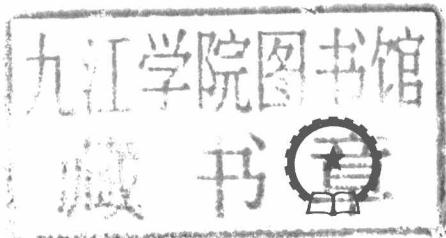
21 世纪重点大学规划教材

分布式计算

李文军 万海 周晓聪 李师贤 编著

不外借

TP 338.8 / 21350



机械工业出版社

分布式计算是软件设计与开发领域重要的发展方向。本书以分布式计算范型的发展演变为线索,由简入繁、由浅入深地介绍 Socket 通信程序、分布式对象技术、组件开发技术、面向服务体系结构等内容;以 Socket 应用层程序、Sun 公司的 RMI (Remote Method Invocation)、OMG 组织的 CORBA (Common Object Request Broker Architecture)、Sun 公司的 EJB (Enterprise JavaBeans) 规范、Web 服务 (Web Services) 和 SOA (Service-Oriented Architecture) 具体开发技术为例,向读者全面介绍分布式计算技术的基本概念、发展演变、开发过程、规范标准、部署应用等内容。

本书可作为计算机科学与技术专业高年级本科生和研究生学习分布式计算技术的教材,也可作为分布式计算领域的研究与开发人员的参考书。

图书在版编目(CIP)数据

分布式计算/李文军等编著. —北京:机械工业出版社,2011.10
21世纪重点大学规划教材
ISBN 978-7-111-36039-1

I. ①分… II. ①李… III. ①分布式算法-高等学校-教材
IV. ①TP301.6

中国版本图书馆 CIP 数据核字(2011)第 202061 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:唐德凯

责任印制:李 研

中国农业出版社印刷厂印刷

2012 年 1 月第 1 版·第 1 次印刷

184mm×260mm·31.75 印张·782 千字

0001-3000 册

标准书号:ISBN 978-7-111-36039-1

定价:59.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者购书热线:(010)88379203

封面无防伪标均为盗版

出版说明

“211工程”是“重点大学和重点学科建设项目”的简称，是国家“九五”期间唯一的教育重点项目。

进入“211工程”的100所学校拥有全国32%的在校本科生、69%的硕士、84%的博士生，以及87%的有博士学位的教师；覆盖了全国96%的国家重点实验室和85%的国家重点学科。相对而言，这批学校中的教授、教师有着深厚的专业知识和丰富的教学经验，其中不少教师对我国高等院校的教材建设做过很多重要的工作。为了有效地利用“211工程”这一丰富资源，实现以重点建设推动整体发展的战略构想，机械工业出版社推出了“21世纪重点大学规划教材”。

本套教材以重点大学、重点学科的精品教材建设为主要任务，组织知名教授、教师进行编写，教材适用于高等院校计算机及其相关专业，选题涉及公共基础课、硬件、软件、网络技术，内容紧密贴合高等院校相关学科的课程设置和培养目标，注重教材的科学性、实用性、通用性，在同类教材中具有一定的先进性和权威性。

为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习指导、习题解答、课程设计、毕业设计指导等内容。

机械工业出版社

前 言

分布式计算是软件设计与开发领域重要的发展方向。著名的计算机科学家 C. A. R. Hoare 指出：“分布式计算是一个具有重大理论与实践意义的迷人课题，其迷人之处在于理论与实践的同步发展，一方面实践推动了理论，另一方面理论又指导着实践。”本书为读者介绍分布式计算领域的基本概念、开发过程、规范标准、高级课题等内容。

分布式计算有两种典型的应用途径。第一种应用途径是将分布式软件系统看做直接对现实世界或问题域中分布性的建模。例如，当今许多业务处理流程通常呈现一种分布式运作方式，负责制造或加工的工厂可能位于珠江三角洲一带，而负责市场营销的部门则可能分别位于北京、上海和广州，此时处理业务流程的软件系统也应采用分布式处理方式。分布式对象 (Distributed Objects) 技术是这一应用途径的主要技术。

第二种应用途径是改进某些应用程序的运行性能，使它们比单进程的集中式实现具有更高的运行效率，此时软件系统的分布性并不是现实世界或问题域中分布性的直接映射，而是为充分利用额外的计算资源而人为引入的。例如，探索外星文明的 SETI@home 项目自 2000 年起已利用全世界 50 多万台 PC 的计算能力分析一台射电望远镜收集的海量数据；美国 Entropia 公司自 2000 年开始的 FightAIDS@Home 项目利用 3 万多台家庭 PC 评价艾滋病的候选药物；2003 年，美国 D²OL 实验室也采用类似的计算技术探索防治非典型肺炎 (SARS) 的候选药物；2001 年，美国与意大利数学家开展的 MetaNEOS 项目汇集两国近千台计算机，用一周时间求解一个名为 Nug30 的优化问题。近年来，全球广泛开展的网格计算 (Grid Computing) 研究是这一应用途径的未来发展方向。

在计算机硬件技术与网络通信技术的支持下，应用需求驱使计算机软件的规模与复杂性不断增长。面对这种情况，对整个软件系统的体系结构进行分析与设计，就远远重要于对算法与数据结构的选择。软件体系结构 (Software Architecture) 所关心的是整个软件系统的结构，它决定了一个软件系统由什么样的组件组成，以及这些组件之间的交互关系如何。典型的软件体系结构风格有设计图形用户界面常用的事件驱动风格、设计操作系统常用的层次化设计风格、设计编译程序常用的管道与过滤器风格，以及当前应用程序普遍采用的面向对象风格等。一个实用软件系统的体系结构通常是若干典型风格的有机结合。

分布式软件系统通常基于客户机/服务器风格，其中客户程序提出对信息或服务的请求，由服务程序提供这些信息或服务。客户机/服务器计算模型发展至今，大约经历了 3 个里程碑：局域网文件服务器、数据库服务器以及分布式对象技术。由于当前面向对象技术几乎已渗透到软件开发的每一个角落，先进的分布式软件开发方法当然离不开与面向对象技术的结合，因而分布式软件体系结构通常是客户机/服务器风格与面向对象风格的有机结合，典型的例子有 OMG 组织的公共对象请求代理体系结构 (CORBA)、Sun 公司的企业级 JavaBeans (EJB)、Microsoft 公司的分布式组件对象模型 (DCOM) 等。

预测不同技术平台的未来发展趋势是一件困难的事情，但倘若单纯从学习与研究的角度出发，CORBA 以其特有的规范严谨性、技术开放性、平台无关性和其他许多先进的分布式计算特性，理应成为学习分布式对象计算技术与分布式软件体系结构的首选。在理论方面，

可参考 OMG 发布的一系列规范和关于 CORBA 的丰富读物；在实践方面，既可下载使用 IO-NA Orbix、Borland/Inprise VisiBroker 等商品化 CORBA 产品的 30 天或 60 天试用版，也可使用 OmniORB、TAO 等免费 CORBA 产品。相对于其他分布式对象计算模型而言，CORBA 在理论上更为完整和严谨，即使读者所采用的开发与运行平台未必是 CORBA 兼容的，CORBA 模型中提出的许多问题也应加以慎重考虑，并可借鉴 CORBA 提出的问题解决方案。

本书以分布式计算范型的发展演变过程为线索，由简入繁、由浅入深地介绍基于 Socket 的分布式程序、分布式对象技术、组件开发技术、面向服务体系结构，分别以 TCP/IP 协议应用层程序、Sun 公司的 RMI、OMG 组织的 CORBA、Sun 公司的 EJB 规范、Web Services 和 SOA 相关开发技术为例，向读者介绍分布式计算技术的基本概念、发展演变、开发过程、规范标准、部署应用及高级应用课题等内容。

本书共分 5 个部分：第一部分“基本概念”介绍从传统的集中式计算过渡到面向对象分布式计算的基本概念与原理，着重论述分布式计算的基本范型，最基本的汇编级分布式计算程序，即基于 Socket 通信程序的开发方法。

第二部分“分布式对象技术”分别以 Sun 公司的 RMI、OMG 组织的 CORBA 为例，介绍分布式对象技术的基本原理与开发过程；CORBA 应用程序是学习的重点，以一个 CORBA 应用程序的典型开发过程为线索，详细讨论如何利用 OMG IDL 编写对象接口，如何利用可移植对象适配器（POA）编写服务端程序，如何采用不同调用类型和通信方式编写客户端程序，以及如何在运行环境中部署应用程序。为避免习惯于集中式软件开发的读者第一次就接触太多的分布式对象技术新概念，我们需要一个过渡环节以达到循序渐进的学习目标，Java RMI 以其简单性与实用性自然进入我们的视野。

第三部分“分布式对象高级课题”探讨分布式应用程序中的一些高级课题，提出可能产生的问题以及这些问题的可能解决途径，包括分布式环境下对象的查找方法、对象之间的异步与多目通信、分布式事务处理、如何提高服务端程序的性能与可伸缩性等课题。这些高级课题对于 CORBA 平台与非 CORBA 平台的分布式应用开发均有重要的指导意义。

第四部分“分布式组件模型与 Web 应用开发”介绍了 EJB 组件模型的基本原理以及各类 EJB 组件的开发与部署过程，帮助读者理解基于组件的软件开发技术和基于开源框架的 Web 应用程序开发技术。

第五部分“面向服务的体系结构及展望”着眼于目前业界的研究热点，即面向服务的体系结构（SOA），着重介绍 SOA 架构的 3 项关键技术：Web 服务技术架构、业务流程描述 BPEL、企业服务总线技术 ESB，SOA 设计方法论及 SOA Governance 原理，最后还探讨了分布式计算的发展方向。

本书例程运行在 JDK 1.6 版本，读者可从 Sun 公司的网站免费下载不同版本的 JDK。喜爱集成化环境的读者也可利用支持该版本 JDK 的集成化开发环境调试或运行该例程，例如，可扩展的开放源代码 IDE Eclipse，WebGain 公司的 VisualCafé 4.5 专家版或企业版、Borland 公司的 JBuilder 6 或 JBuilder 7 个人版或标准版、IBM 公司的 VisualAge for Java 4.0 专业版等。

本书第 4~11 章中的所有 CORBA 应用例程均运行在 Borland/Inprise 公司的 VisiBroker for Java 4.5.1 版。VisiBroker 目前已被集成到 Borland 企业服务器（BES）产品中，读者可从网上免费下载 Borland 企业服务器 VisiBroker 版本的 60 天试用版。

本书第 12 章的所有 EJB 例程均运行在开源 EJB 应用服务器 JBoss，读者可从 JBoss 的下

载中心免费下载该软件。同样，读者也可在一个集成化开发环境中开发与部署这些例程，例如，IDE Eclipse 支持在集成化环境中快速开发与部署基于 WebLogic 应用服务器的 J2EE 应用程序。

本书在正文中还使用特别的字体穿插了一些“注意”、“提示”和“思考”。“注意”用于给出程序设计中的一些技巧、常见错误和注意事项等；“提示”是对正文内容的补充和提高，可供基础好的读者作进一步的学习和思考，初学者可以忽略这些内容；“思考”用于给出一些与正文密切相关的练习，通常是对例程的修改和完善等，可用做课堂练习。

读者还可从中山大学计算机科学系软件工程实验室网站 <http://www.cs.sysu.edu.cn/selab> 下载书中所有例程的源代码清单。这些例程稍加修改即可移植到其他开发平台上。

本书可供从事基于 Internet/Intranet 的分布式软件开发人员作为参考书，也可作为计算机科学与技术专业高年级本科生与研究生学习分布式计算的教材。本书假设读者已掌握面向对象程序设计方法与 Java 程序设计语言，并具备面向对象软件工程的基本知识。

在 ACM/IEEE - CS (ACM: Association for Computing Machinery 美国计算机协会/IEEE - CS: Institute of Electrical and Electronics Engineers-Computer Society 电气电子工程师协会计算机学会) 最新修订并发布的《2001 计算教学计划》(简称 CC2001) 中，根据计算技术的发展现状与趋势新增了“以网络为中心的计算 (NC)”作为 14 个知识领域之一，其中包含客户机/服务器计算、开发 Web 应用、分布式对象系统、无线与移动计算等专题。本书内容覆盖了该知识领域的许多专题。

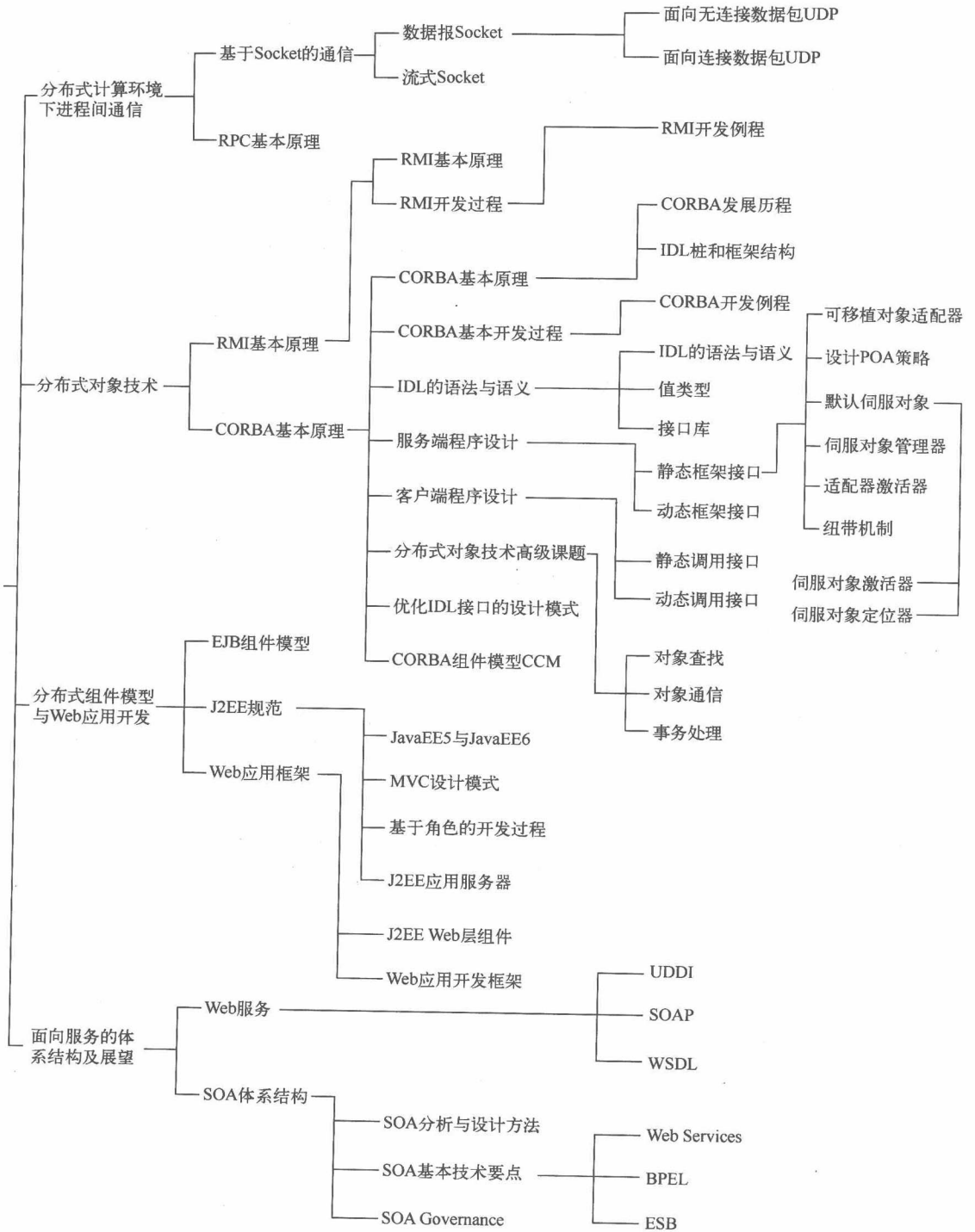
本书每一章都提供了思考与练习，其中规模较大或复杂性较高的题目以“*”、“**”、“***”标出，这些题目也适合作为课程设计的选题。本书配套的电子教案可在机械工业出版社网站上下载，地址是 www.cmpedu.com。

本书作者长年从事中山大学软件学院、信息科学与技术学院高年级本科生和研究生“分布式计算”、“服务计算”、“面向服务的软件工程”等课程的教学工作，本书是由教学所编写的讲义和课程的实验案例等积累而来的成果。全书由李文军统筹编撰，由李文军、万海执笔完成。书中所有实验都由万海配置验证完成。周晓聪对全书进行了仔细地审阅与校对。李师贤对全书的编写提出了许多宝贵的意见。

分布式计算是软件开发技术的一个新兴领域，并且各种分布式计算技术还在持续、迅猛地发展。作者水平有限，书中谬误之处在所难免，恳请广大读者批评指正。

作者

本书知识点结构



目 录

前言

本书知识点结构

第一部分 基本概念

第 1 章 绪论	1
1.1 引言	1
1.1.1 分布式计算的产生背景	1
1.1.2 分布式计算的发展历程	2
1.1.3 分布式计算技术的特点	3
1.1.4 中间件技术	4
1.2 软件设计的基本思想	4
1.2.1 隐式地与显式地	4
1.2.2 逻辑的与物理的	5
1.2.3 面向对象技术与 UML	6
1.2.4 软件体系结构与 MDA	10
1.3 分布式计算基本范型	11
1.3.1 分布式计算范型的概念	11
1.3.2 消息传递范型	12
1.3.3 远程过程调用范型	13
1.3.4 分布式对象范型	13
1.3.5 分布式组件模型	14
1.3.6 面向服务的体系结构	15
1.4 进程间通信	17
1.4.1 进程间通信的基本原理	17
1.4.2 接口与接口定义语言	20
1.4.3 数据表示与编码	20
1.4.4 事件同步机制	21
思考与练习	22
进一步阅读	22
第 2 章 基于 Socket 的通信	24
2.1 基本原理	24
2.1.1 Socket API 的基本概念	24
2.1.2 Java 对网络通信的支持	25
2.1.3 TCP、UDP 与端口	25

2.2 数据报 Socket	27
2.2.1 基本编程原理	27
2.2.2 面向无连接数据报 UDP	29
2.2.3 面向连接数据报 UDP	32
2.3 流式 Socket	34
2.3.1 基本编程原理	34
2.3.2 单线程服务程序与客户程序	36
2.3.3 多线程服务程序	41
2.4 应用层协议开发	42
2.4.1 理解应用层协议	42
2.4.2 Daytime 协议开发	43
2.4.3 FTP 协议开发	51
2.4.4 HTTP 协议开发	55
思考与练习	57
进一步阅读	58

第二部分 分布式对象技术

第3章 RMI 基本原理与开发过程	59
3.1 分布式对象技术	59
3.1.1 分布式对象基本技术架构	59
3.1.2 分布式对象技术特点	60
3.1.3 分布式对象技术分类	61
3.2 RMI 基本原理	63
3.2.1 RMI 技术结构	63
3.2.2 RMI API 介绍	64
3.2.3 RMI 与 RPC 的区别	66
3.3 简单 RMI 例子开发过程	66
3.4 复杂 RMI 例子开发过程	70
3.4.1 数据库设计	71
3.4.2 对象接口定义	72
3.4.3 服务端程序	73
3.4.4 客户端程序	76
3.4.5 部署并运行应用程序	79
3.5 分布式应用程序设计决策	82
3.5.1 网络传输	82
3.5.2 通信方式	82
3.5.3 运行平台	82
3.5.4 资源优化	82

3.5.5 其他决策问题	83
思考与练习	83
进一步阅读	84
第4章 CORBA 基本原理与开发过程	85
4.1 对象管理体系结构	85
4.1.1 对象请求代理	86
4.1.2 对象服务	87
4.1.3 公共设施	87
4.1.4 领域接口	88
4.1.5 应用接口	88
4.2 CORBA 体系结构	88
4.2.1 OMG 的对象模型	88
4.2.2 对象请求代理的体系结构	89
4.2.3 对象接口定义	90
4.2.4 客户端机制	91
4.2.5 服务端机制	92
4.3 CORBA 应用程序开发过程	92
4.3.1 编写对象接口	93
4.3.2 编译 IDL 文件	94
4.3.3 编写对象实现和服务程序	94
4.3.4 编写客户程序	94
4.3.5 创建并部署应用程序	95
4.3.6 运行应用程序	95
4.4 一个银行账户管理程序	95
4.4.1 对象接口定义	96
4.4.2 生成的接口和类	97
4.4.3 编写对象实现和服务程序	103
4.4.4 编写客户程序	106
4.4.5 创建应用程序	107
4.4.6 运行应用程序	108
4.5 深入 IDL 桩和框架	109
4.5.1 IDL 桩和框架的类层次	109
4.5.2 IDL 桩与客户端对象引用	110
4.5.3 IDL 框架的代码	113
4.5.4 用于纽带机制的 IDL 框架	115
4.6 分布式对象的可互操作性	116
4.6.1 可互操作性	116
4.6.2 ORB 域和桥接	116

4.6.3	GIOP、IIOP 与 ESIOP	117
4.6.4	更高的可互操作性	118
4.7	CORBA 规范与 CORBA 产品	118
4.7.1	CORBA 规范	118
4.7.2	CORBA 产品概述	119
4.7.3	商品化 ORB 产品	120
4.7.4	免费与开源 ORB 产品	121
4.7.5	产品与规范的一致性	122
4.8	内容小结	123
4.8.1	CORBA 带来了什么	123
4.8.2	未解决的问题	124
	思考与练习	124
	进一步阅读	125
第 5 章	编写对象接口	126
5.1	对象接口与对象实现	126
5.1.1	接口与实现分离	126
5.1.2	接口定义语言	126
5.2	OMG IDL 的语法与语义	127
5.2.1	词法规则	127
5.2.2	模块的声明	128
5.2.3	类型的声明	128
5.2.4	常量的声明	132
5.2.5	异常的声明	133
5.2.6	接口的声明	133
5.2.7	值类型的声明	136
5.3	使用值类型	140
5.3.1	IDL 定义	140
5.3.2	编译 IDL 文件	141
5.3.3	实现 IDL 接口	142
5.3.4	实现值类型	143
5.3.5	服务程序与客户程序	145
5.4	使用接口库	147
5.4.1	接口库的定义	147
5.4.2	接口库的结构	148
5.4.3	接口库管理工具	149
5.4.4	编写接口库客户程序	150
5.5	设计对象接口的准则	154
5.5.1	编写对象接口	154

5.5.2 典型的对象接口	154
思考与练习	155
进一步阅读	156
第6章 服务端程序设计	158
6.1 可移植对象适配器	158
6.1.1 CORBA 对象与伺服对象	158
6.1.2 CORBA 对象的基本语义	159
6.1.3 对象适配器	160
6.1.4 可移植对象适配器	160
6.1.5 POA 及其组件的 IDL 定义	162
6.2 设计 POA 策略	166
6.2.1 POA 策略	166
6.2.2 选择 POA 策略	166
6.2.3 组合使用 POA 策略	168
6.2.4 请求的处理过程	169
6.3 使用 POA	169
6.3.1 获取根 POA 的对象引用	170
6.3.2 创建自定义策略的 POA	170
6.3.3 使用 POA 管理器	171
6.3.4 激活与冻结对象	173
6.3.5 使用默认伺服对象	175
6.4 伺服对象管理器	177
6.4.1 伺服对象激活器	177
6.4.2 伺服对象定位器	181
6.5 适配器激活器	183
6.5.1 按需激活 POA	183
6.5.2 使用适配器激活器的例程	184
6.6 纽带机制	187
6.6.1 纽带机制的工作原理	187
6.6.2 使用纽带机制的例程	188
6.7 POA vs BOA	191
6.7.1 基本对象适配器	191
6.7.2 POA 对 BOA 的改进	191
思考与练习	192
进一步阅读	192
第7章 客户端程序设计	194
7.1 请求的基本概念	194
7.1.1 请求是一个对象	194

7.1.2	ORB 伪对象	195
7.1.3	创建请求对象	196
7.1.4	Any 类型	198
7.1.5	实际参数与返回结果	200
7.1.6	服务端的请求对象	201
7.2	选择调用类型	202
7.2.1	静态调用接口	202
7.2.2	动态调用接口	202
7.2.3	选择合适的调用类型	203
7.3	选择通信方式	204
7.3.1	同步通信	204
7.3.2	延迟同步通信	204
7.3.3	单向通信	204
7.3.4	使用不同的通信方式	204
7.4	使用动态调用接口	205
7.4.1	基于 DII 的 IDL 桩	205
7.4.2	使用 DII 的开发步骤	207
7.4.3	完整的例程	208
7.5	使用动态框架接口	211
7.5.1	动态框架接口	211
7.5.2	基于 DSI 的 IDL 框架	211
7.5.3	使用 DSI 编写对象实现	213
7.6	请求的上下文	217
7.6.1	上下文对象的概念	217
7.6.2	Context 伪对象	218
7.6.3	使用附带上下文的请求	219
7.7	消息传递的服务质量	220
7.7.1	消息传递的 QoS 策略	220
7.7.2	为消息传递指定 QoS 策略	224
7.7.3	一个完整的例程	225
	思考与练习	226
	进一步阅读	227
第 8 章	部署 CORBA 应用程序	229
8.1	部署阶段的总体决策	229
8.1.1	选择运行平台	229
8.1.2	决定包装形式	229
8.1.3	考虑系统容错性	229
8.1.4	开放系统 vs 封闭系统	230

8.1.5 安装过程启示	230
8.2 部署服务端程序	231
8.2.1 创建和发布对象引用	231
8.2.2 部署对象实现	231
8.2.3 部署持久服务程序	231
8.3 部署客户端程序	231
8.3.1 获取对象引用	232
8.3.2 部署接口定义	232
8.3.3 部署 IDL 客户桩	232
8.4 部署 VisiBroker 应用程序	232
8.4.1 安装运行环境	233
8.4.2 安装支持服务	233
8.4.3 定制运行环境	234
8.4.4 启动应用程序	234
8.4.5 使用管理工具	234
8.5 设置 VisiBroker 属性	235
8.5.1 VisiBroker 属性	235
8.5.2 设置 VisiBroker 属性	236
8.6 配置智能代理	237
8.6.1 部署智能代理	237
8.6.2 使用 ORB 域	238
8.6.3 提高对象容错性	240
8.7 使用实现库	240
8.7.1 实现库服务	240
8.7.2 实现库管理工具	241
8.7.3 编写实现库客户程序	243
8.7.4 按需激活对象	246
思考与练习	246
进一步阅读	247

第三部分 分布式对象高级课题

第9章 对象查找与对象通信	249
9.1 对象查找	249
9.1.1 查找对象的基本机制	249
9.1.2 CORBA 命名服务	250
9.1.3 命名服务的类结构	255
9.1.4 CORBA 交易对象服务	256
9.1.5 选择合适的对象查找机制	260

9.2 对象通信	260
9.2.1 分布式对象的通信	260
9.2.2 CORBA 事件服务	261
9.2.3 CORBA 通知服务	274
9.2.4 CORBA 消息服务	275
思考与练习	276
进一步阅读	277
第 10 章 事务处理及其他对象服务	278
10.1 对象事务处理	278
10.1.1 事务	278
10.1.2 基于 DBMS 的事务处理	278
10.1.3 分布式事务处理	279
10.1.4 CORBA 对象事务服务	282
10.1.5 OTS 应用编程模型	288
10.1.6 OTS 与 TP 监控程序	291
10.2 并发控制	291
10.2.1 CORBA 并发控制服务	291
10.2.2 不同的锁模式	292
10.2.3 并发控制服务提供的接口	293
10.3 对象安全性	294
10.3.1 信息系统的安全性	294
10.3.2 CORBA 安全服务	294
10.3.3 安全策略及其实施	295
10.3.4 安全服务的体系结构	296
10.4 对象持久性	297
10.4.1 CORBA 持久对象服务	297
10.4.2 CORBA 持久状态服务	299
10.4.3 CORBA 外表化服务	299
10.5 对象集管理	300
10.5.1 CORBA 关系服务	300
10.5.2 CORBA 生存期服务	301
10.5.3 CORBA 对象类集服务	301
10.5.4 CORBA 查询服务	302
10.6 其他服务	302
10.6.1 CORBA 属性服务	302
10.6.2 CORBA 时间服务	303
10.6.3 CORBA 特许服务	303
思考与练习	303

进一步阅读·····	304
第 11 章 分布式应用程序性能与可伸缩性 ·····	306
11.1 基本概念·····	306
11.1.1 分布式应用程序的性能和可伸缩性 ·····	306
11.1.2 POA 与可伸缩性 ·····	307
11.2 优化 IDL 接口的设计模式·····	307
11.2.1 优化分布式应用程序性能的设计原则 ·····	307
11.2.2 一个忽略性能的 IDL 接口设计例子 ·····	308
11.2.3 值对象设计模式 ·····	310
11.2.4 迭代器设计模式 ·····	311
11.3 资源池·····	313
11.3.1 资源池的概念 ·····	313
11.3.2 资源池的体系结构 ·····	313
11.3.3 资源池的 QoS 策略 ·····	314
11.4 多线程·····	316
11.4.1 多线程体系结构 ·····	316
11.4.2 服务程序的多线程模型·····	317
11.4.3 VisiBroker 的线程管理 ·····	318
11.5 连接管理·····	319
11.5.1 连接与可伸缩性 ·····	319
11.5.2 可伸缩的连接管理 ·····	320
11.5.3 VisiBroker 的连接管理 ·····	321
11.6 内存管理·····	322
11.6.1 杜绝内存泄漏 ·····	322
11.6.2 伺服对象实例池 ·····	325
11.7 负载均衡·····	328
11.7.1 实现负载均衡的体系结构 ·····	328
11.7.2 基于网络的负载均衡 ·····	329
11.7.3 基于操作系统的负载均衡·····	329
11.7.4 基于中间件的负载均衡·····	330
11.7.5 基于应用程序的负载均衡·····	330
思考与练习·····	331
进一步阅读·····	332

第四部分 分布式组件模型与 Web 应用开发

第 12 章 EJB 组件模型 ·····	334
12.1 EJB 的基本概念 ·····	334
12.1.1 EJB 组件的分类 ·····	336