

Distributed Programming with Java

JAVA

分布式程序设计



- Sockets
- RMI
- CORBA
- 移动代理

[美] Qusay H. Mahmoud 著

欧阳光 安锦 译

 MANNING

国防工业出版社

中国人民解放军总装备部专项资金资助出版

编程经典译丛

Java 分布式程序设计

Distributed Programming with Java

Qusay H. Mahmoud 著

欧阳光 安锦 译

国防工业出版社

·北京·

著作权合同登记号 图字:军-2001-009号

图书在版编目(CIP)数据

Java 分布式程序设计/(美)马穆夫(Mahmoud, Q. H.)著;欧阳光, 安锦译. —北京:国防工业出版社, 2002.1

书名原文:Distributed Programming with Java

ISBN 7-118-02687-5

I .J... II .①马...②欧...③安... III .JAVA 语言-程序设计
IV .TP312

中国版本图书馆 CIP 数据核字(2001)第 077053 号

©2000 by Manning Publications Co. All rights reserved

Published by arrangement with Manning Publications Co., USA.

本书中文简体版由 Manning Publications Co. 授予国防工业出版社
(独家)出版。版权所有,侵权必究。

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

三河腾飞印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 17³/₄ 393 千字

2002 年 1 月第 1 版 2002 年 1 月北京第 1 次印刷

印数:1~4000 册 定价:33.00 元

(本书如有印装错误,我社负责调换)

前 言

“Java is the language of network computer.”

Lawrence J. Ellison, Chairman and CEO, Oracle Corporation

对计算机领域而言,Java 程序设计语言的出现,是计算机工业的一个转折点。它是一种清晰的、丰富的、强有力的程序设计语言。更重要的是,它的简洁性使编程成为一种乐趣,尤其是 Java 虚拟机(JVM),更增强了不同网络之间的软件的可移植性。

作为一种真正独立于平台的语言,使用 Java 开发的应用程序可以运行在任何有 Java 解释器的系统之上。这是一个重要特征,因为基于网络的应用程序应该能运行在所有 Internet 平台之上。由于 Java 提供了对网络的高级支持,所以被认为是分布式语言。

Java 提供了大量机制以适应网络和分布式计算机,包括对 Socket 编程的高级 API 接口,以及 Java 的 RMI 系统。Java 一出现,许多厂家就使用它构建系统和工具,以使分布式应用程序的开发变得更容易。例如,Inprise 公司推出的 CORBA 2.0。为众人所知的 VisiBroker,即完全是用 Java 开发的。另外一些厂家也开始打起了移动代理的主意,因为 Java 是开发它们的理想语言。特别值得一提的是对象空间,如他们开发的 Voyager ORB,提供了开发分布式和移动代理应用程序的更容易、更便捷的方法。

本书总结了作者使用 Java 开发分布式应用程序方面的经验。在 Java 第一次发布时作者即开始使用它。然而,与其他开发者不同的是,作者不是开始于 applet,而是使用 java.net 包,构建基于网络的应用程序。本书覆盖了使用 Java 开发分布式应用程序的所有方面,目的就在于为你开发这类应用程序提供实际的建议。你可以把本书看做使用 Java 开发分布式应用程序的指南。本书中包括了许多不同的范例,如 Socket、RMI、CORBA(使用 VisiBroker)以及移动代理程序(使用 Voyager),这些范例表明,本书将给你一个使用 Java 开发分布式应用程序的良好开端。

我希望本书为你提供使用 Java 开发分布式应用程序的好的起点。如果本书能帮助你快速上手开发分布式应用程序,那也就达到了本书的编写目的。

本书导读

本书共分四部分:

第一部分:Socket 编程 第 1~6 章。本部分详细介绍了分布式程序设计,以及如何使用 Java 进行 Socket 编程。本部分还给出了许多有趣的网络应用程序,在此你将看到实际应用中的 Socket 程序。我们还探讨了线程、安全问题以及对象序列化。

第二部分:RMI 第 7~10 章。在这一部分中,我将介绍一个新的开发分布式应用程序的非常容易的模式。在“了解 RMI”中,我们将剖析一个基于 RMI 的应用程序,探讨 RMI 高级编程技术(如动态服务器和 Factories,签名的报文以及回调(Callback))。介绍每一技术时均附有实例。

第三部分:CORBA 第 11~19 章。本部分在概述了 CORBA 的同时,详细剖析了一个基于 CORBA 的应用程序。另外,本部分还将介绍 IDL-to-Java 的映射,并通过实例介绍高级 CORBA 技术(如激活、继承、约束机制以及接口库),最后介绍了 Caffeine。

第四部分:移动代理和 Voyager 第 20~24 章。本部分在介绍了移动代理和 Voyager 的同时,给出了大量实例,以及如何利用这些实例开发分布式和移动代理应用程序。另外,本部分还将探讨高级 Voyager 特征,如发布/订阅(Publish/subscribe)、激活(Activation)、计时器(Timer)。最后,本部分中还给出了使用 CORBA 集成 Voyager 的实例。

本书的每一部分都独立于其他部分,采用这种写作方式,是为了使你在最短的时间内掌握相关的内容。如果你刚开始使用 Java 编写分布式应用程序,建议你从第 1 页读起,一直到本书的结束。因为作者是从分布式程序设计的基础(Socket)开始,逐渐深入到最高级(使用 Voyager)。而如果你已有了 Internet 编程方面的经验,且只是对移动代理和 Voyager 感兴趣,那么你可以略过前面几部分,直接阅读第四部分。总之,你可以根据你的经验和喜好,以任意顺序浏览本书。

读者对象

本书可供下列人员阅读：

- 想要或正在使用 Java 从事分布式应用程序开发的人员。通过实例,本书介绍了使用 Java 开发分布式应用程序的几个相关技术。
- 想要掌握不同开发方法,或者想为一个特殊项目选择一个合适范例的,客户/服务器体系结构设计人员。本书能够帮助他们决定对于一个特殊的应用程序,应选择哪一种开发方式。
- 那些只熟悉自己常用的分布式应用程序开发方法的开发人员。例如,你可能只熟悉 CORBA,但通过本书,你将掌握 RMI 以及 Voyager。
- 教师,如果他们希望自己的学生有一本分布式或网络计算方面的易读、易掌握的书。本书将告诉这些同学,如何使用不同技术去解决同样的问题,并且比较这些技术的优缺点。
- 计算机相关专业的本科生或研究生,如果他们想了解分布式对象程序设计,掌握开发分布式应用程序的不同技术。

作者在线

本书的购买者可以免费访问个人 Internet 论坛,在此,你可以发表有关本书的评论,询问技术问题,获得作者或其它用户的帮助。论坛的 Web 地址为: [Http://www.manning.com/Mahmoud](http://www.manning.com/Mahmoud)。在该网址,你可以提交到论坛。一旦你注册完毕,该站点将告诉你如何访问论坛,能获取何种帮助,以及论坛的行为准则是什么。

本书中所有示例的源代码均可以从 Manning 的 Web 站点购得。URL <http://www.manning.com/Mahmoud> 中包含有到源代码文件的链接。

目 录

第一部分 Socket 编程

第1章 了解分布式程序设计	3
1.1 概述	3
1.2 Internetwork 概览	3
1.2.1 网络体系结构	3
1.2.2 协议	4
1.2.3 OSI 参考模型	4
1.2.4 命名与路由	5
1.2.5 Internet 地址	5
1.3 分布式程序设计简介	6
1.3.1 客户/服务器模式	6
1.3.2 基于对象模式	7
1.4 分布式程序设计技术	7
1.4.1 Socket	7
1.4.2 远程过程调用	7
1.4.3 分布式程序为什么好	8
1.4.4 分布式程序为什么是困难的	8
1.5 Java 支持的分布式程序设计	9
1.6 移动网络	10
1.7 Java 安全性概览	10
1.8 小结	12
第2章 了解 Socket 编程	13
2.1 内部通信方式	13
2.2 什么是 Socket?	14
2.2.1 Socket 类型	15
2.3 TCP/IP 和 UDP/IP 通信	15
2.3.1 数据报通信协议	15
2.3.2 流通信协议	15
2.3.3 UDP 与 TCP	15
2.4 客户/服务器通信	16
2.5 使用 Java 进行 Socket 编程	16
2.5.1 数据流	17
2.5.2 TCP Socket	17

2.5.3	UDP Socket(数据报)	20
2.5.4	多点传送 Socket	21
2.6	Greetings 服务器实例	21
2.6.1	服务器程序	22
2.6.2	客户程序	24
2.6.3	错误?	25
2.7	解析 Internet 地址	25
2.7.1	例:GetName	26
2.7.2	例:GetIP	26
2.7.3	例:NsLookup	27
2.7.4	例:IPtoName	28
2.8	小结	29
第3章	客户和服务程序设计	30
3.1	对现有服务编写客户程序	30
3.1.1	SMTP 客户程序	30
3.1.2	finger 客户程序	33
3.1.3	ping 客户程序	35
3.2	使用线程进行编程	37
3.2.1	什么是线程?	37
3.2.2	线程的优点	38
3.2.3	示例:创建并运行线程	38
3.2.4	创建和启动线程	39
3.2.5	使线程睡眠	41
3.2.6	控制线程	41
3.2.7	改变线程优先级	41
3.2.8	同步	42
3.3	编写新的服务器和客户程序	43
3.3.1	示例:算术客户/服务器应用程序	44
3.4	小结	49
第4章	应用程序中的 Socket	50
4.1	开发一个多线程 HTTP 服务器	50
4.2	HTTP 的安全	54
4.3	代理服务器	55
4.3.1	代理服务器如何工作	56
4.3.2	代理服务器的优点	56
4.4	通过 Socket 访问数据库	56
4.5	解释 JDBC	57
4.5.1	JDBC 安全模式	58
4.5.2	JDBC DriverManager	58

4.5.3	剖析 JDBC 应用程序	59
4.5.4	Java 对象的持久性存储器	61
4.6	小结	64
第 5 章	Socket 高级编程	65
5.1	对象序列化	65
5.1.1	对象序列化与持久性	66
5.1.2	对象序列化的安全	68
5.1.3	控制序列化	68
5.1.4	序列化对象的版本	69
5.2	Socket 对象	70
5.2.1	在线发送对象	70
5.3	数字签名报文	74
5.3.1	java.security 包	74
5.3.2	数字签名	74
5.3.3	示例:通过 Socket 签名文件	74
5.4	小结	81
第 6 章	实例分析:全局计算引擎	82
6.1	简介	82
6.2	基于 Web 的全局计算	82
6.2.1	为什么需要全局计算	83
6.2.2	为什么需要基于 Web 的计算	83
6.2.3	需考虑的问题	83
6.3	为什么不用 CGI	84
6.3.1	优点	84
6.3.2	缺点	84
6.4	客户/服务器模式	85
6.5	安全问题	85
6.6	实现细节	86
6.6.1	类装载器	86
6.6.2	安全策略	90
6.6.3	计算引擎	93
6.6.4	命令行客户程序	96
6.7	其它问题	97
6.7.1	搜索计算资源	97
6.7.2	代理程序	98
6.8	小结	98

第二部分 RMI

第 7 章	了解 RMI	101
--------------	---------------------	------------

7.1 RMI 简介	101
7.2 什么是 RMI?	101
7.3 RMI 的目标	102
7.4 RMI 体系结构	103
7.4.1 桩/构架层	103
7.4.2 远程引用层	103
7.4.3 传输层	104
7.5 RMI 如何工作	104
7.6 分布垃圾回收	104
7.7 RMI 和 OSI 参考模型	105
7.8 安全	105
7.9 小结	106
第 8 章 从 RMI 开始	107
8.1 基于 RMI 应用程序的剖析	107
8.1.1 定义远程接口	108
8.1.2 实现远程接口	108
8.1.3 使用远程接口开发客户程序	111
8.1.4 产生桩和构架	112
8.1.5 启动 RMI 注册表	113
8.1.6 运行服务程序和客户程序	113
8.2 使用 RMI 注册表工作	114
8.2.1 从注册表中删除一个对象	114
8.2.2 重新绑定一个对象到注册表	115
8.3 RMI 与 Socket 的比较	115
8.4 小结	116
第 9 章 高级 RMI 编程	117
9.1 “城市信息”服务器	117
9.1.1 定义一个远程接口	117
9.1.2 实现远程接口	118
9.1.3 开发一个客户程序	119
9.1.4 产生桩和构架	120
9.1.5 启动 RMI 注册表	120
9.1.6 运行服务器和客户程序	120
9.2 实现“工厂”	120
9.2.1 定义工厂接口	120
9.2.2 实现 City2 接口	121
9.2.3 实现 CityFactory 接口	122
9.2.4 调用工厂	123
9.2.5 运行工厂	124

9.3 实现回调	124
9.3.1 Java 中的回调	125
9.3.2 在 RMI 中实现回调	125
9.4 通过 RMI 签名对象	129
9.4.1 java.security 包	129
9.4.2 示例	129
9.5 小结	132
第 10 章 使用 Java 2 进行 RMI 编程	134
10.1 创建定制的 Socket 类型	134
10.1.1 创建定制的 Socket 类型的步骤	134
10.2 创建一个定制的 RMISocketFactory	135
10.2.1 减少一个 Socket 类型	135
10.2.2 减少更多的 Socket 类型	135
10.2.3 使用定制的 RMISocketFactory	136
10.3 SSL Socket	137
10.4 远程对象激活	138
10.4.1 简介	138
10.4.2 激活协议	138
10.4.3 创建一个可激活的远程对象	139
10.5 版本控制	141
10.5.1 包的版本	141
10.5.2 产品版本与用户	143
10.6 小结	144

第三部分 CORBA

第 11 章 CORBA 概述	147
11.1 了解 CORBA	147
11.2 CORBA 结构	147
11.2.1 对象请求代理	148
11.2.2 不同厂商和不同 ORB	149
11.2.3 界面定义语言 (IDL)	149
11.2.4 动态调用接口 (DII)	149
11.2.5 动态构架接口 (DSI)	150
11.2.6 接口仓库 (IR)	150
11.2.7 对象适配器 (OA)	150
11.3 客户和对象实现	150
11.4 对象服务	150
11.5 CORBA 3.0 的新特点	151
11.5.1 可移植对象适配器	151

11.5.2	CORBA 通信 (Messaging)	152
11.5.3	通过值传递对象	153
11.6	小结	153
第 12 章	开始使用 CORBA	154
12.1	CORBA 实现	154
12.2	基于 CORBA 应用程序的剖析	154
12.2.1	定义 IDL 接口	155
12.2.2	实现 CORBA 类	157
12.2.3	开发服务器程序	158
12.2.4	开发客户程序	159
12.2.5	启动 smart agent, 服务器和客户机	160
12.3	比较 CORBA 和 RMI	160
12.4	小结	161
第 13 章	CORBA IDL: 接口定义语言	162
13.1	IDL 定义	162
13.1.1	注释	162
13.1.2	标识符	162
13.1.3	字面量	163
13.2	IDL 接口	163
13.3	模块声明	164
13.4	属性声明	164
13.5	类型声明	165
13.5.1	基本类型	165
13.5.2	构造类型	165
13.5.3	模板类型	166
13.6	数组	166
13.6.1	数列和序列	167
13.7	typedef 声明	167
13.8	常量声明	167
13.9	异常声明	167
13.10	操作声明	168
13.10.1	参数声明	168
13.10.2	非等待操作	168
13.10.3	raises 表达	168
13.11	预处理	169
13.12	关键字	169
13.13	小结	169
第 14 章	IDL 到 Java 的映射	171
14.1	名称	171

14.2	模块	171
14.3	Holder 类	172
14.4	常量	174
14.4.1	在接口内的常量	174
14.4.2	不在接口内的常量	174
14.5	基本类型	174
14.5.1	布尔型	175
14.5.2	字符型	175
14.5.3	八进制型	175
14.5.4	字符串	176
14.5.5	整型	176
14.5.6	浮点型	177
14.6	构造类型	177
14.6.1	enum	177
14.6.2	结构	178
14.6.3	联合	179
14.6.4	序列	180
14.6.5	数列	181
14.7	异常	181
14.7.1	用户定义异常	182
14.7.2	系统异常	182
14.8	接口	182
14.8.1	参数传送模式	183
14.9	typedef 映射	184
14.9.1	简单 IDL 类型	184
14.9.2	复杂 IDL 类型	184
14.10	小结	185
第 15 章	命名和绑定	186
15.1	一个银行应用程序	186
15.1.1	在 IDL 中定义接口	186
15.1.2	实现 CORBA 类	187
15.1.3	开发服务器程序	189
15.1.4	开发客户程序	190
15.1.5	运行应用程序	191
15.2	命名对象	191
15.3	绑定对象	192
15.3.1	绑定功能	192
15.3.2	指定绑定选项	193
15.3.3	本地对远程对象定位	193

15.3.4	对于对象引用的操作	193
15.3.5	缩小对象引用	194
15.4	小结	194
第 16 章	注册和激活对象	195
16.1	基本对象适配器	195
16.2	对象激活模式	195
16.3	对象引用	196
16.3.1	暂态对象引用	196
16.3.2	永久对象引用	196
16.4	注册对象	197
16.5	实现仓库	197
16.6	小结	198
第 17 章	继承和约束机理	200
17.1	接口继承	200
17.1.1	示例	200
17.2	约束机理	201
17.2.1	示例	201
17.2.2	修改 Server 类	201
17.2.3	修改 Accountimpl 类	202
17.2.4	修改 ManagerImpl 类	202
17.3	小结	203
第 18 章	动态调用/构架接口	204
18.1	动态调用接口	204
18.1.1	使用 DI1	205
18.1.2	请求参数	206
18.2	动态构架接口	206
18.2.1	使用 DI1	206
18.3	接口仓库	207
18.3.1	创建一个接口仓库	208
18.4	小结	208
第 19 章	Caffeine	209
19.1	Caffeine 概述	209
19.2	使用 java2iop 编译器	209
19.3	剖析一个 Caffeine 基的应用程序	210
19.3.1	定义接口	210
19.3.2	编写一个服务器对象	211
19.3.3	编译服务器	212
19.3.4	编写客户机代码	212
19.3.5	编译客户机代码	213

19.3.6 运行服务器	213
19.3.7 运行客户机	213
19.4 数据类型映射	213
19.5 小结	214

第四部分 移动代理和 Voyager

第 20 章 移动软件代理概述	217
20.1 软件代理介绍	217
20.2 移动软件代理	218
20.3 一个新的分布计算范例	218
20.4 移动代理应用程序	218
20.5 移动代理的安全	219
20.6 移动代理实现	220
20.6.1 General Magic's Odysseys	220
20.6.2 IBM Japan 的 Aglet	220
20.6.3 ObjectSpace 公司的 Voyager ORB	221
20.7 小结	221
第 21 章 开始使用 Voyager	222
21.1 概述	222
21.2 Voyager 的使用	223
21.2.1 启动和终止 Voyager 程序	223
21.2.2 记录控制台信息	224
21.2.3 远程接口	224
21.2.4 创建远程对象	225
21.2.5 剖析一个 Voyager 应用程序	226
21.2.6 命名服务和代理	228
21.3 小结	231
第 22 章 移动代理编程	232
22.1 动态集合	232
22.1.1 使用 facet	233
22.1.2 Facet 创建和远程访问	233
22.2 移动性	236
22.2.1 移动对象	237
22.2.2 移动性和消息	238
22.2.3 得到通知	239
22.3 移动代理	241
22.3.1 使用移动代理	241
22.3.2 开发移动代理	242
22.4 小结	246

第 23 章 高级 Voyager 编程	247
23.1 安全	247
23.1.1 安装安全管理器	247
23.2 高级消息	248
23.3 发布和订阅	249
23.4 激活	250
23.4.1 编写一个激活器	250
23.5 计时器	251
23.6 小结	251
第 24 章 Voyager 和 CORBA 集成	253
24.1 Voyager CORBA 编程	253
24.1.1 导入和导出 CORBA 对象	253
24.2 剖析 Voyager CORBA 应用程序	253
24.2.1 开发接口	254
24.2.2 实现接口	254
24.2.3 开发服务器程序	254
24.2.4 开发客户机程序	255
24.2.5 编译和运行应用程序	256
24.3 Voyager CORBA 集成	257
24.3.1 Voyager 客户机和 CORBA 服务器	257
24.3.2 Voyager 服务器和 CORBA 客户机	260
24.4 小结	262
参考文献	263
在线参考	264