

Java 中间件 开发技术

Java Middleware
Development

李华飏 郭英奎 等编著



中国水利水电出版社
www.waterpub.com.cn

Java 中间件开发技术

李华飏 郭英奎 等编著



中国水利水电出版社

内 容 提 要

本书使用丰富的案例介绍了使用 Java 技术进行中间件编程的方法及技巧,包括 JSP、Java Servlet、JDBC 数据库开发、使用 JavaMail 开发邮件应用程序、RMI(远程方法调用)、Enterprise JavaBeans 开发、使用 CORBA 以及 Java IDL 进行开发、JNDI 编程、JMS 应用开发、Java XML 编程以及 Java 开发 Web 服务等。

本书的内容均为目前的热点和读者所关注的问题,也包括对很多人来说悬而未决的难题。书中的许多案例甚至可以不加修改就应用到开发实践中。

本书适合专业的 Java 程序员阅读,也可以作为正在进行 Java 开发的各类程序员的必备参考书。即使是不擅长这一领域的开发人员,通过详细的实例讲述,也可以循序渐进地掌握本书的内容。

图书在版编目(CIP)数据

Java 中间件开发技术 / 李华飏等编著. —北京: 中国水利水电出版社, 2005
ISBN 7-5084-3135-9

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

中国版本图书馆 CIP 数据核字 (2005) 第 089215 号

书 名	Java 中间件开发技术
作 者	李华飏 郭英奎 等编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 25 印张 604 千字
版 次	2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷
印 数	0001—5000 册
定 价	38.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

序

互联网技术的蓬勃发展使得越来越多的企业应用从传统的 C/S（客户/服务器）结构向三层架构转化。中间件的最早定义为：介于应用程序和操作系统之间的软件，它代表一类产品。IT 环境的复杂性使得企业在信息化建设的过程中，需要构建一个能够集门户、集成、运行和管理、安全、开发、部署等众多功能于一体的应用基础平台，中间件在完成这一任务方面发挥了巨大的作用。而从用户角度看，企业所构建的平台越复杂，就越需要一个具有统一性、简便性和扩展性的方案。

一般说来，中间件产品主要应用了以下几类技术：

- **数据访问技术** 提供直接访问不同数据源的手段，而不必改变应用层的程序，比如大家很熟悉的 ODBC、JDBC、JDO、ADO 这些数据标准接口。
- **基于消息的中间件技术（MOM）** MOM 提供一个异步消息传送机制。MOM 中间件在所有需要集成的应用系统内，都要安装 MOM 的 API 调用程序，以把消息递交给中间件进行处理。显然，MOM 的优点是：发送方和接收方不必在线等待（松耦合）。缺点是：会发生过载情况。当然随着产品的不断改进，在负载平衡方面会得到进一步改善。提供 MOM 产品的主要有 IBM MQSeries、微软的 MSMQ。随着 JMS（Java Message Service）的兴起，MOM 产品的应用也将变得更为广泛。相对于下面讲到的应用服务器，它比较“单纯”——专为整合服务。
- **远程过程调用（Remote Procedure Call, RPC）** 远程过程调用也同 MOM 一样利用客户/服务器模式，不同点是，RPC 是同步在线的，即要等待接收方的回复才继续自己的工作。这减少了出错概率，当然由于等待使速度变慢。
- **交易处理监控器（Transaction Processing Monitors）** TP 监控器作为一个中间件，是现代应用服务器的雏形。它主要用于监控和管理客户端及多个后台应用之间的交易状况，考虑负载平衡，将客户的请求映射到各个应用。
- **对象请求代理（Object Request Brokers, ORB）** ORB 是专门替应用转交功能组件的“中介”，是业务功能整合层次的整合中间件，基于 COM、CORBA 及 Java RMI 等标准组件。
- **应用服务器** 应用服务器是一个软件开发平台，除可以开发独立的应用系统外，服务器也集成了各种整合技术。此外，很多厂商还专门开发了许多特制的适配器或连接器，因此，利用应用服务器做整合平台，也是目前很多企业考虑的选项之一。
- **Portal Server** Portal Server 往往是提供一种集成的门户服务。通过可扩展的门户框架集成了对企业信息标准访问途径，可以将内部和合作伙伴与他们所需要的信息连接起来。

SUN 公司的 Sun ONE（Open Net Environment，开放网络环境）是主流的 Web 技术体系之一。SUN ONE 以 Java 技术为核心，包括 J2SE/J2EE/J2ME，并基于一系列开放和流行标准、技术及协议。SUN ONE 的开放性、可移植性、跨平台和扩展性在进行中间件编程方面得天独

厚，具备无可比拟的优势。

本书介绍了基于 Java 技术的中间件编程，包括如下方面：JSP、Java Servlet、JDBC 数据库开发、使用 JavaMail 开发邮件应用程序、RMI（远程方法调用）、Enterprise JavaBeans 开发、使用 CORBA 以及 Java IDL 进行开发、JNDI 编程、Java XML 编程以及 Java 开发 Web 服务等。

相对于《精通 Java 中间件编程》来说，本书加深了对中间件理论体系的介绍，并且对增加了消息中间件技术的介绍，使得读者可以更深刻地了解中间件技术原理。

本书详细介绍了如何搭建开发和运行环境，在讲解理论的同时提供了精彩案例，读者按照本书的实例，可以举一反三，融会贯通，理解中间件编程的精髓。

本书的内容均是目前的热点和读者所关注的问题，也包括对很多人来说悬而未决的难题。书中的许多案例甚至可以不加修改就应用到开发实践中。

本书适合专业的 Java 程序员阅读，也可以作为正在进行 Java 开发的各类程序员的必备参考书。即使是不擅长这一领域的开发人员，透过详细的实例讲述，也可以循序渐进地掌握本书的内容。有关本书的技术讨论，可以发邮件到 washing_lhb@sina.com。

本书由李华飏和郭英奎编写。在编写过程中，李旭伟、徐志强、柳振良、唐洪福、彭锦艳、彭传明、史进才、李艳丽、王恒、顾有松、毕宗睿等参与了部分工作。IBM 中国有限公司的褚洪峰提供了许多技术上的支持。顾有松教授和中国科学院周剑平博士对本书的结构提供了许多宝贵的意见。最后，特别感谢北京万水电子信息有限公司孙春亮总编的大力鼓励和支持。

编 者

2005 年 4 月

目 录

序

第 1 章 中间件技术导论	1
1.1 中间件的定义	1
1.2 中间件的分类	2
1.2.1 终端仿真/屏幕转换	2
1.2.2 数据访问中间件	2
1.2.3 远程过程调用中间件	3
1.2.4 交易中间件	3
1.2.5 消息中间件	3
1.2.6 对象中间件	4
1.2.7 应用服务器	4
1.2.8 企业应用集成中间件	5
1.2.9 工作流中间件	5
1.2.10 门户中间件	5
1.2.11 安全中间件	6
1.3 中间件、基础件和平台	6
1.3.1 中间件和基础件	6
1.3.2 平台和中间件	7
1.4 中间件特点及优势	7
1.5 小结	9
第 2 章 应用服务器概述	10
2.1 传统的应用体系结构	10
2.1.1 C/S（客户端/服务器）模式	10
2.1.2 B/S（浏览器/服务器）模式	11
2.2 多层应用体系结构	12
2.3 J2EE 应用体系结构	13
2.4 应用服务器	14
2.5 小结	15
第 3 章 准备上手	16
3.1 开发环境的搭建	16
3.2 运行环境的搭建	17
3.2.1 Java 运行环境的搭建	17
3.2.2 Tomcat 的安装	20
3.2.3 WebLogic 的安装	22

3.3	小结	24
第 4 章	JSP 编程范例	25
4.1	简单的 JSP 范例：显示一句话	25
4.1.1	实例说明	25
4.1.2	代码分析	26
4.1.3	运行结果	27
4.2	使用表单实例	28
4.2.1	实例说明	28
4.2.2	代码分析	28
4.2.3	运行结果	29
4.3	JSP 处理 cookie	30
4.3.1	实例说明	30
4.3.2	代码分析	30
4.3.3	运行结果	31
4.4	Session 的管理范例：购物车	31
4.4.1	实例说明	31
4.4.2	编程思路	33
4.4.3	代码分析	33
4.4.4	运行结果	34
4.5	连接数据库并分页显示记录范例	35
4.5.1	实例说明	35
4.5.2	准备工作	35
4.5.3	编程思路	38
4.5.4	代码分析	39
4.5.5	运行结果	42
4.6	JSP 实现文件上传范例	42
4.6.1	实例说明	42
4.6.2	编程思路	43
4.6.3	代码分析	44
4.6.4	运行结果	49
4.7	小结	49
第 5 章	Java Servlet 编程范例	50
5.1	简单的 Servlet 例子	50
5.1.1	实例说明	50
5.1.2	代码分析	51
5.1.3	运行结果	53
5.2	Servlet 与 JSP 之间的通信	53
5.2.1	实例说明	53
5.2.2	编程思路	53

5.2.3	代码分析.....	54
5.2.4	运行结果.....	55
5.3	控制输出流	56
5.3.1	实例说明.....	56
5.3.2	编程思路.....	56
5.3.3	代码分析.....	56
5.3.4	运行结果.....	57
5.4	用 Servlet 管理广告条显示	58
5.4.1	实例说明.....	58
5.4.2	编程思路.....	58
5.4.3	准备工作.....	58
5.4.4	代码分析.....	59
5.4.5	运行结果.....	68
5.5	小结	69
第 6 章	JDBC 数据库编程范例	70
6.1	JDBC 简介.....	70
6.1.1	获得连接.....	70
6.1.2	发送 SQL 语句.....	70
6.1.3	返回结果.....	71
6.2	Java 数据库连接范例	71
6.2.1	实例说明.....	71
6.2.2	准备工作.....	71
6.2.3	代码分析.....	73
6.2.4	运行结果.....	75
6.3	JavaBean 封装数据库操作范例	76
6.3.1	实例说明.....	76
6.3.2	代码分析.....	76
6.3.3	运行结果.....	78
6.4	数据库连接池	78
6.4.1	实例说明.....	78
6.4.2	编程思路.....	79
6.4.3	代码分析.....	79
6.4.4	运行结果.....	82
6.5	Java 开发存储过程	83
6.5.1	实例说明.....	83
6.5.2	代码分析.....	84
6.5.3	生成调用.....	85
6.6	使用 Java 开发触发器	87
6.6.1	编写代码.....	87

6.6.2	生成调用.....	88
6.7	使用 SQLJ	89
6.7.1	什么是 SQLJ	89
6.7.2	准备工作.....	91
6.7.3	代码分析.....	91
6.7.4	运行结果.....	93
6.7.5	SQLJ 的要点	93
6.8	小结	94
第 7 章	使用 Java 进行 XML 编程.....	95
7.1	XML 简介.....	95
7.1.1	XML 与 HTML 的比较	95
7.1.2	XML 的优缺点.....	95
7.1.3	XML 的注释.....	98
7.2	DTD 和 Schema.....	98
7.2.1	DTD 简介	98
7.2.2	Schema 简介	101
7.2.3	Schema 的文件结构.....	102
7.2.4	名域和 Schema 的结合.....	105
7.2.5	使用 Java 解析 XML 文件	105
7.3	XML 的样式表.....	106
7.3.1	使用 CSS 样式表	106
7.3.2	使用 XSL 样式表	109
7.3.3	使用 XSL 的例子	110
7.4	DOM 和 SAX	115
7.4.1	使用 DOM	116
7.4.2	使用 SAX.....	121
7.5	在 JSP 中使用 XML	123
7.5.1	实例说明.....	123
7.5.2	编程思路.....	124
7.5.3	代码分析.....	124
7.5.4	运行结果.....	128
7.6	小结	129
第 8 章	分布式对象概述.....	130
8.1	分布式计算介绍	130
8.1.1	分布式计算的概念.....	130
8.1.2	构成分布式计算的一些基本要素	131
8.2	分布式对象	131
8.2.1	CORBA 体系结构.....	132
8.2.2	COM/DCOM	133

8.2.3	Java RMI.....	134
8.2.4	CORBA、DCOM 和 RMI 的比较.....	136
8.3	基于面向对象技术的应用软件体系结构	136
8.4	Web Service 和 SOA	140
8.4.1	介绍 SOA.....	140
8.4.2	Web Service 与 SOA	141
8.5	小结	141
第 9 章	RMI 编程范例	142
9.1	介绍 RMI.....	142
9.2	一个 RMI 会话系统.....	143
9.2.1	构建服务器程序.....	143
9.2.2	构建客户程序.....	145
9.2.3	注册对象并启动服务器和客户程序	146
9.3	带有回调的 RMI 会话.....	146
9.3.1	服务器程序.....	147
9.3.2	客户程序.....	149
9.4	远程对象激活	149
9.4.1	创建远程接口.....	150
9.4.2	实现远程接口.....	151
9.4.3	注册激活对象程序.....	151
9.4.4	客户程序.....	152
9.4.5	运行程序.....	153
9.5	在 IIOP 上运行 RMI.....	153
9.5.1	服务器端程序.....	154
9.5.2	客户端程序.....	155
9.5.3	运行程序.....	156
9.6	小结	156
第 10 章	EJB 编程范例	157
10.1	了解 EJB.....	157
10.1.1	Enterprise Bean 是什么.....	157
10.1.2	Enterprise Bean 的优点.....	157
10.1.3	使用 Enterprise Bean 的时机.....	158
10.1.4	Enterprise Bean 的类型.....	158
10.1.5	EJB 对象.....	158
10.1.6	RMI 和 EJB 对象	159
10.1.7	远程接口.....	159
10.1.8	Home 接口.....	160
10.1.9	Home 对象.....	160
10.2	Session Bean 概述	160

10.2.1	Session Bean 是什么	160
10.2.2	有状态 Session Bean	161
10.2.3	无状态 Session Bean	161
10.2.4	Session Bean 接口	162
10.3	无状态 Session Bean 开发示例	163
10.3.1	主接口 (StatelessHelloWorldHome)	164
10.3.2	远程接口 (StatelessHelloWorld)	165
10.3.3	Bean 实现类 (StatelessHelloWorldImp)	165
10.3.4	部署 EJB	166
10.3.5	生成访问无状态 Session Bean 的客户端程序	170
10.4	有状态 Session Bean 开发示例	171
10.4.1	StatefulFundManagerEJB 的主接口	172
10.4.2	StatefulFundManagerEJB 的远程接口	172
10.4.3	StatefulFundManagerEJB 的实现类	173
10.4.4	部署 EJB	174
10.4.5	生成访问有状态 Session Bean 的客户端程序	174
10.5	Entity Bean 开发示例	178
10.5.1	Entity Bean 和 Session Bean 的比较	178
10.5.2	容器管理的持久性	178
10.5.3	本地接口	179
10.5.4	远程接口	180
10.5.5	Bean 实现类	181
10.5.6	回调方法	185
10.5.7	Bean 管理的持久性	188
10.6	Java 消息服务和消息驱动 Bean	194
10.6.1	消息概述	194
10.6.2	消息驱动 Bean	195
10.6.3	EJB 和 JMS	195
10.6.4	MDB 体系结构	197
10.6.5	JMS 消息接口	198
10.6.6	MDB 客户程序接口	199
10.6.7	点对点消息队列模式	200
10.6.8	发布—订阅消息模式	201
10.6.9	消息驱动 Bean 应用实例	201
10.7	基于 Web 的 EJB 应用程序示例	207
10.7.1	收集需求	207
10.7.2	层的设计	208
10.7.3	生成实体 Bean	209
10.7.4	生成会话 Bean	221

10.7.5	生成 Web 接口	234
10.7.6	部署应用程序	241
10.8	小结	242
第 11 章	CORBA 以及 Java IDL 编程范例	243
11.1	介绍 CORBA	243
11.1.1	对象管理组织 (OMG) 简介	243
11.1.2	CORBA 主要版本的发展历程	243
11.1.3	CORBA 体系结构概述	244
11.1.4	CORBA 的主要应用方向及中间件产品介绍	245
11.2	介绍 IDL	245
11.2.1	OMG IDL 文件概述	245
11.2.2	Java IDL 介绍	246
11.3	开发 CORBA 应用的服务器程序	246
11.4	开发 CORBA 应用的客户机程序	248
11.5	编写 CORBA 客户机 applet	249
11.6	使用 CORBA 范例: Java 和 C++混合编程	251
11.6.1	编写 SysProp.idl	251
11.6.2	编写 Java 的服务器程序	251
11.6.3	编写 Java 的客户机	253
11.6.4	编写 C++的 IOR 客户机	254
11.6.5	运行程序	256
11.7	利用动态调用方式实现分布式应用	256
11.7.1	客户端动态调用接口 (DII)	257
11.7.2	服务对象动态骨架接口 (DSI)	258
11.7.3	程序举例	259
11.8	小结	260
第 12 章	JNDI 编程范例	262
12.1	介绍 JNDI	262
12.2	JNDI 架构	265
12.3	利用 JNDI 在网络上搜索资源	265
12.4	用 JNDI 查找实例	267
12.5	小结	271
第 13 章	Java 开发 Web Service	272
13.1	什么是 Web Service	272
13.2	一个简单的 SOAP 程序	273
13.2.1	实例说明	273
13.2.2	准备工作	273
13.2.3	编写代码	274
13.2.4	部署服务	275

13.2.5	程序调用.....	276
13.3	SOAP 的信息结构	279
13.3.1	SOAP 封套 (Envelope)	279
13.3.2	SOAP 信息头 (Header)	280
13.3.3	SOAP 信息体	283
13.3.4	SOAP 错误	283
13.4	WSDL 语言和 UDDI	284
13.4.1	WSDL 服务接口实例	289
13.4.2	根据 WSDL 服务接口创建的 UDDI tModel.....	290
13.4.3	WSDL 服务实现示例	292
13.4.4	根据 WSDL 服务实现创建的 UDDI 商业服务	292
13.4.5	查找 WSDL 服务接口	293
13.4.6	查找 WSDL 服务实现描述	293
13.4.7	查找 UDDI bindingTemplate	294
13.4.8	UDDI bindingTemplate 示例	294
13.5	JSP 调用 Web Service 范例	294
13.5.1	实例说明.....	295
13.5.2	代码分析.....	296
13.5.3	运行结果.....	303
13.6	小结	303
第 14 章	消息中间件概述	304
14.1	消息传递概述	304
14.1.1	消息传递服务实现模型.....	304
14.1.2	点到点的消息传递.....	305
14.1.3	发布—订阅消息传递.....	305
14.1.4	“推”消息传递模型和“拉”消息传递模型	305
14.1.5	消息过滤、同步和质量.....	306
14.1.6	电子邮件消息传递.....	306
14.2	消息中间件	306
14.3	WebSphere MQ 概述.....	306
14.3.1	MQ 的基本概念	307
14.3.2	MQ 的工作原理	308
14.3.3	MQ 的基本配置举例	309
14.3.4	MQ 的通信模式	309
14.3.5	MQ Server 和 MQ Client	310
14.3.6	MQ 的 API.....	310
14.4	小结	311
第 15 章	JMS 应用开发	312
15.1	JMS 概述	312

15.1.1	什么是消息.....	312
15.1.2	什么是 JMS API.....	312
15.1.3	什么时候应用 JMS API.....	312
15.1.4	JMS API 如何同 J2EE 平台工作	313
15.1.5	JMS Provider (JMS 提供者)	314
15.1.6	JMS Messages (JMS 消息)	314
15.1.7	Portability (移植性)	314
15.1.8	JMS 不提供的功能	314
15.1.9	与其他 Java API 的关系	314
15.2	JMS 体系结构	315
15.2.1	JMS 应用的组成	315
15.2.2	管理	315
15.2.3	两种消息模式.....	315
15.2.4	JMS 接口	316
15.2.5	开发 JMS 应用	317
15.2.6	安全性.....	317
15.2.7	多线程.....	317
15.2.8	客户端触发机制.....	317
15.2.9	请求/答复 (Request/Reply)	318
15.3	JMS 消息模型	318
15.3.1	目标.....	318
15.3.2	JMS 消息组成	318
15.3.3	消息头 (header fields)	318
15.3.4	消息属性 (Message Properties)	320
15.3.5	消息选择.....	320
15.3.6	JMS 消息体	320
15.4	JMS 消息工具	321
15.4.1	管理对象 (Administered Objects)	321
15.4.2	Connection	321
15.4.3	会话 (Session)	322
15.4.4	MessageConsumer	325
15.4.5	MessageProducer	325
15.4.6	消息发送模式.....	325
15.4.7	消息生存时间.....	325
15.4.8	异常.....	326
15.5	JMS Point-to-Point 模型	326
15.6	JMS Publish/Subscribe 模型	327
15.7	JMS 异常	329
15.8	JMS 应用程序服务器工具	331

15.8.1	订阅消息的并发处理.....	331
15.8.2	XAConnectionFactory	334
15.8.3	XAConnection	334
15.8.4	XASession.....	334
15.8.5	JMS 应用程序服务器接口	334
15.9	JMS 队列生产者/消费者范例	334
15.9.1	发送消息.....	335
15.9.2	接收消息.....	336
15.9.3	配置服务器.....	337
15.9.4	运行实例代码.....	337
15.10	小结	338
第 16 章	JavaMail 应用开发	339
16.1	用 JavaMail 发送简单邮件.....	339
16.1.1	实例说明.....	339
16.1.2	准备工作.....	340
16.1.3	编程思路.....	341
16.1.4	代码分析.....	341
16.1.5	运行结果.....	343
16.2	用 JavaMail 发送 HTML 邮件	345
16.2.1	编程思路.....	345
16.2.2	代码分析.....	346
16.2.3	运行结果.....	349
16.3	用 JavaMail 发送需要 SMTP 认证的邮件	350
16.3.1	实例说明.....	350
16.3.2	编程思路.....	351
16.3.3	代码分析.....	351
16.3.4	运行结果.....	355
16.4	用 JavaMail 发送带附件的邮件.....	355
16.4.1	实例说明.....	355
16.4.2	编程思路.....	356
16.4.3	代码分析.....	356
16.4.4	运行结果.....	359
16.5	发送电子邮件综合演练	360
16.5.1	实例说明.....	360
16.5.2	编程思路.....	360
16.5.3	代码分析.....	361
16.5.4	运行结果.....	373
16.6	用 JavaMail 接收邮件列表.....	374
16.6.1	实例说明.....	374

16.6.2	编程思路.....	375
16.6.3	代码分析.....	375
16.6.4	运行结果.....	379
16.7	用 JavaMail 接收单封邮件.....	379
16.7.1	实例说明.....	379
16.7.2	编程思路.....	380
16.7.3	代码分析.....	380
16.7.4	运行结果.....	383
16.8	小结	383

第 1 章 中间件技术导论

计算机技术迅速发展，从硬件技术看，CPU 速度越来越快，处理能力越来越强；从软件技术看，应用程序的规模不断扩大，特别是 Internet 及 WWW 的出现，使计算机的应用范围更为广阔，许多应用程序需在网络环境的异构平台上运行。这一切都对新一代的软件开发提出了新的需求。在这种分布异构环境中，通常存在多种硬件系统平台（如 PC、工作站、小型机等），在这些硬件平台上又存在各种各样的系统软件（如不同的操作系统、数据库、语言编译器等），以及多种风格各异的用户界面，这些硬件系统平台还可能采用不同的网络协议和网络体系结构连接。如何把这些系统集成起来并开发新的应用是一个非常现实而困难的问题。

为解决分布异构问题，人们提出了中间件（middleware）的概念。

1.1 中间件的定义

中间件（middleware）是基础软件的一大类，属于可复用软件的范畴。顾名思义，中间件处于操作系统软件与用户应用软件的中间。中间件在操作系统、网络和数据库之上，应用软件之下，总的作用是为处于自己上层的应用软件提供运行与开发的环境，帮助用户灵活、高效地开发和集成复杂的应用软件。

在众多关于中间件的定义中，人们比较普遍接受的是 IDC 的定义：中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信。

IDC 对中间件的定义表明，中间件是一类软件，而非一种软件；中间件不仅仅实现互连，还要实现应用之间的互操作；中间件是基于分布式处理的软件，最突出的特点是其网络通信功能。

有人根据这种定义提出一个简单的公式：

$$\text{中间件} = \text{平台} + \text{通信}$$

也可以认为，中间件是位于平台（硬件和操作系统）和应用之间的通用服务，如图 1-1 所示，这些服务具有标准的程序接口和协议。针对不同的操作系统和硬件平台，它们可以有符合接口和协议规范的多种实现。

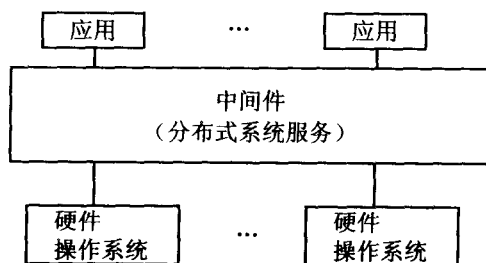


图 1-1 中间件