

开发专家

之 Sun ONE

EJB

应用开发详解



飞思科技产品研发中心 编著

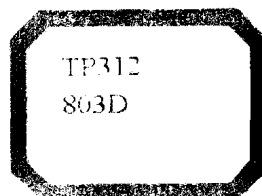


本书附赠
光盘包括
全部实例
代码



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



开发专家之 Sun ONE

EJB 应用开发详解

飞思科技产品研发中心 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书详细介绍了 J2EE 框架的重要组成部分——EJB 及其相关技术。全书以剖析 EJB 2.0 的技术核心以及开发与部署的过程为中心,共分 3 大部分 20 章,包括无状态会话 Bean、有状态会话 Bean、CMP 实体 Bean、EJB QL 语言、集成 RMI-IIOP 与 JNDI、应用客户端开发、EJB 部署、持久化 EJB 与 CORBA 的集成等内容,均是目前的技术热点和读者所关注的问题。书中的例程对实际的应用开发有非常强的借鉴意义。

本书适合于专业和准专业的 Java 程序员阅读,也可作为正在进行 Java 开发的各类程序员的必备 Java 参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

EJB 应用开发详解 / 飞思科技产品研发中心编著. —北京: 电子工业出版社, 2002.1

(开发专家之 Sun ONE)

ISBN 7-5053-7370-6

I.E... II.飞... III.Java 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2001)第 093129 号

责任编辑: 郭 晶 韩素华

印 刷: 北京兴华印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 32 字数: 819.2 千字 附光盘 1 张

版 次: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

印 数: 5000 册 定价: 49.00 元 (含光盘)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话: (010) 68279077

出版说明

“开发专家”是电子工业出版社计算机图书研发部长期以来精心培育的计算机科学技术类本版品牌。这个品牌是由多个专题系列组成的横向大系列，涵盖了计算机技术的各个方面，特别是一直受到极大关注的程序开发类系列，例如《开发专家之数据库》、《开发专家之网络编程》、《开发专家之 Delphi》以及《开发专家之 Sun ONE》等。这些专题系列基于各自的角度，从纵向上包含了该专题的所有内容。因此，整个“开发专家”的品牌架构纵横交错，囊括了所有的计算机技术和所有的技术层面，海纳百川而又极具可扩展性。

“开发专家”的作者队伍主要依托于“飞思科技产品研发中心”。“飞思科技产品研发中心”是由专业的策划人员、权威的技术专家和资深的作者队伍共同构成。在图书的出版上，形成了以研发为基础、以出版为中心、以服务为支持的专业化出版框架和流程。通过深入的市场调查和技术跟踪，在综合了技术需求和读者焦点等因素的基础上，形成各系列丛书的写作重点和大纲，然后聘请业界的最前沿学者进行写作。同时，策划工作全程介入写作进程，严格控制写作质量，用最专业的技术背景、最深刻的理论基础、最具代表性的案例、最能为专业读者接受的形式，为读者提供品质最佳的图书产品，体现了出版者和著作者的完美结合。

多年来，计算机图书研发部始终把创造社会效益摆在首位，秉承一切为国内计算机技术专业读者服务的精神，为推动国内 IT 技术发展、为体现国内技术的原创水平，穷尽所有的创意与努力，将出版者的命运与读者的支持紧紧地连在了一起。

在此，我们临出版之残酷竞争而不惧，旌旗猎猎而异军突起，这与广大读者的支持是分不开的。为使我们的脚步更坚实、使我们的队伍永葆活力和创造力，我们期待着您能为我们的前进贡献出您的意见和建议。同时，我们也在等待着您的加入。

我们的联系方式：

电 话： （010）68134545

E-mail： support@fecit.com.cn

网 址： <http://www.fecit.com.cn> <http://www.fecit.net>

电子工业出版社计算机图书研发部

前 言

关于本套丛书

从来没有任何事物像互联网那样，对人类的活动产生如此深刻的影响，无论是政府、企业，以及个人，莫不如此。与此同时，IT 工业也正面临着一场变革——传统应用向基于 Internet/Web 的服务模式转化。

翻开历史，我们可以看到互联网的形成和发展就是以分布性、开放性和平台无关性为基础，这是 Internet 与生俱有的属性。随着互联网应用的发展，又引入了诸如 RPC/COM/CORBA 等技术，但这些的技术在实际应用中，又存在着很多不足和局限。它们的特定协议也难以通过防火墙，因而不适于 Web 上的应用开发。为了进一步开发基于 Web 的应用，相继出现了 Sun 公司的 Sun ONE (Open Net Environment 开放网络环境) 和 Microsoft 公司的 .NET 两大 Web 服务技术体系。其中，Sun ONE 以 Java 技术为核心，更接近或者满足于互联网在智能化 Web 服务上对分布性、开放性和平台无关性的要求，同时其在健壮性、安全性、组件化等方面也更为成熟稳定，获得了众多 IT 厂商和产品的支持，是目前惟一在市场上得到了广泛应用的技术体系。

Sun ONE 体系结构以 Java 语言为核心，包括 J2SE/J2EE/J2ME，并基于一系列开放和流行标准、技术及协议。要特别指出的是，Sun ONE 体系结构本身作为开放式体系结构，在得到 IBM/BEA/Oracle/Sybase 等这些 IT 巨擎支持的同时，更得到了互联网上 Open Source 社区的青睐。我们很容易地从网上免费获得和使用包括 Java 集成开发环境、Java 数据库，甚至是中间件 (Application Server) 服务器等产品，以及它们的源代码。这对于加速国内中小企业的信息化建设和自有知识产权产品开发、提高企业应用和软件行业的整体水平，无疑是一次难得的机会。

综观国内的技术发展，广大的 Java 程序开发人员以及正在转向 Java 体系进行开发的技术人员虽然已面临这一令人激动和鼓舞的转型期，却苦于没有足够的相关资料和文献，尤其对国内的最新 Java 技术动态和技术现状知之甚少，而图书市场上 Java 的书籍尽管汗牛充栋，但精品罕见，能反映出 J2EE 以及 Sun ONE 的框架全貌的书籍更是奇缺。

电子工业出版社计算机图书研发部为进一步推动国内 Java 技术的应用与发展，不失时机地推出了《开发专家之 Sun ONE》系列丛书。

本套丛书以 Sun ONE 整体架构为基础，全面体现了 Sun ONE 的技术核心——Java 的应用开发。丛书从各个角度深入 Java 应用开发的各个层面，涵盖了 Java 技术的所有重要思想和实践，体现了最新的 Java 技术进展和动态，大幅度提升读者的理论和应用水平。同时，丛书重点突出实用性。书中引入了大量的行业应用范例，使读者不仅能快速掌握开发技能，而且对于开发者进行综合系统分析也有所裨益。

关于本书

本书描述并展示了 EJB 的体系结构。尽管 EJB 使分布式组件开发得到简化，但它仍是一种需要大量开发实践才能掌握的复杂技术，因为它涉及到许多重要领域，包括分布式计算、安全控制、数据库、组件等。

本书以 J2EE 框架为基础，详细介绍了 J2EE 中的核心新技术——EJB 及其相关技术。全书以剖析 EJB 的技术核心以及开发与部署的过程为中心，阐述了 EJB 的概念、方法和开发，描绘了如何使用 EJB 组件体系结构开发与部署企业应用。全书共分 3 大部分 20 章，包括无状态会话 Bean、有状态会话 Bean、CMP 实体 Bean、EJB QL 语言、集成 RMI-IIOP 与 JNDI、应用客户端开发、EJB 部署、持久化 EJB 与 CORBA 的集成等内容，均是目前的技术热点和读者焦点。书中的例程对实际的应用开发有非常强的借鉴意义。

本书将指导你掌握入门级 EJB 技术直到精通，既着眼于基础知识，又探究了大量的高级特性。在阅读本书前，当然要熟悉 Java 语言，还要有一些开发商务解决方案的实际经验，同时最好有分布式对象的开发经验，并对数据库、UML 模型图等概念有所了解。

本书适合于专业和准专业的 Java 程序员阅读，也可作为正在转向进行 Java 开发的各类程序员的必备 Java 参考书。

本书由飞思科技产品研发中心策划并组织编写，李昕骞、沈英俊、吴艳琳、陈丽琼、孙承顺、孙鹰、何峰、柴春林、王秀云、曹彬等参加了本书的写作工作。同时，在本书的写作过程中得到了董毅先生的大力支持和协助，他提出了大量的参考性意见使本书增色不少，在此表示衷心的感谢。

当然，限于作者水平，加之时间仓促，书中不足之处难免，敬请读者批评指正。

我们的联系方式：

电 话： (010) 68134545 68134811

E-mail: support@fecit.com.cn

网 址: <http://www.fecit.com.cn> <http://www.fecit.net>

飞思科技产品研发中心

目 录

第一篇 J2EE 框架下的 EJB

第 1 章 J2EE 概述	3
1.1 企业开发体系结构	3
1.2 J2EE 平台基础	5
1.3 J2EE 1.3 核心技术	8
1.3.1 J2EE 标准服务	8
1.3.2 J2EE 的未来	11
1.4 对比 .NET 与 Sun ONE	13
1.4.1 Microsoft .NET 开发框架	13
1.4.2 Sun ONE 开发框架	15
1.4.3 Sun ONE 与 .NET 的战略对比	19
1.5 服务器端组件结构	22
第 2 章 EJB 基础	27
2.1 组件概述	27
2.2 EJB 组件框架	28
2.3 EJB 组件	32
2.4 EJB 项目案例	34
2.5 EJB 开发步骤	35
2.5.1 定义远程接口	36
2.5.2 定义本地接口	36
2.5.3 定义 Bean 类	37
2.5.4 部署 J2EE 应用	37
2.5.5 编写客户端	39
2.5.6 编译与运行客户端	40
2.6 会话 Bean	40
2.6.1 状态管理	41
2.6.2 何时使用会话 Bean	41
2.7 实体 Bean	42
2.7.1 会话 Bean 与实体 Bean 区别	42
2.7.2 何时使用实体 Bean	44
2.8 消息驱动 Bean	44
2.9 EJB 容器与服务器	45
2.10 EJB 编程限制	47

2.11 从 EJB 1.1 到 EJB 2.0	49
--------------------------------	----

第二篇 开发与部署

第 3 章 无状态会话 Bean	65
3.1 如何开发 EJB 组件	65
3.2 如何调用 Bean	75
3.3 无状态会话 Bean 生命周期	79
第 4 章 有状态会话 Bean	81
4.1 有状态会话 Bean 特点	81
4.2 购物车示例	82
4.3 有状态会话 Bean 的生命周期	87
第 5 章 BMP 实体 Bean	89
5.1 实体 Bean 编程基础	89
5.2 银行账户示例	93
5.3 运行客户端程序	117
5.4 BMP 实体 Bean 的生命周期	118
第 6 章 CMP 实体 Bean	121
6.1 CMP 实体 Bean 的特点	121
6.2 容器管理持久化的实现	127
6.3 产品线示例	130
6.4 运行客户端程序	147
6.5 CMP 实体 Bean 的生命周期	147
第 7 章 EJB QL 语言	149
7.1 EJB QL 术语	149
7.2 为什么需要 EJB QL 语言	150
7.3 EJB QL 语法	150
7.4 EJB QL 查询示例	158
7.5 BNF 图解	161
7.6 EJB QL 优势和局限	165
第 8 章 消息驱动 Bean	167
8.1 消息驱动 Bean 动机	167
8.2 Java 消息服务 (JMS)	169
8.3 集成 JMS 和 EJB	174
8.4 开发消息 Bean	176
8.5 消息 Bean 探究	182
第 9 章 集成 RMI-IIOP 与 JNDI	191
9.1 理解 RMI	191

9.2	Java RMI-IIOP	193
9.3	JNDI 接口	203
9.4	集成 RMI-IIOP 和 JNDI	211
第 10 章	应用客户端开发	215
10.1	EJB 应用客户端概述	215
10.2	会话 Bean 客户视图	216
10.3	实体 Bean 客户视图	217
10.4	消息 Bean 客户视图	219
10.5	客户事务管理	220
10.6	EJB 客户端开发示例	221
10.6.1	独立的 Java 应用	221
10.6.2	J2EE 应用客户端	221
10.6.3	Servlet 客户端	224
10.6.4	JavaServer Pages 组件	227
10.6.5	其他企业 Bean	234
第 11 章	容器与 Bean	239
11.1	Bean 的调用	239
11.2	资源工厂 (Resource Factories)	241
11.3	环境属性	243
11.4	EJB 安全	245
11.5	对象句柄	254
第 12 章	EJB 应用服务器	257
12.1	应用服务器概述	257
12.2	应用服务器功能	258
12.3	几种应用服务器产品介绍	259
12.4	如何选择 EJB 应用服务器	263
12.5	EJB 服务器评测	269
第 13 章	部署 EJB	279
13.1	XML 简介	279
13.1.1	XML 的定义	279
13.1.2	XML 语法	279
13.1.3	DTD 声明	282
13.2	部署描述符	285
13.3	EJB-JAR 文件	289
13.4	应用服务器部署	291
13.4.1	HelloWorldLdap: LDAP 无状态会话 Bean	291

13.4.2 ProfileTest: CMP 实体 Bean.....	296
--------------------------------------	-----

第三篇 深入 EJB 开发

第 14 章 事务.....	301
14.1 ACID 特征	301
14.2 容器管理事务	302
14.3 Bean 管理的事务	307
14.4 隔离与数据库锁定	311
14.5 分布式事务	312
14.6 设计 EJB 事务会话.....	315
第 15 章 BMP 和 CMP 数据关系	319
15.1 CMP 和 BMP 的关系区别	319
15.2 集的势 (Cardinality)	320
15.3 方向性 (Directionality)	333
15.4 聚合与合成、级联删除.....	338
15.5 递归与循环关系	340
15.6 引用完整性	341
第 16 章 持久化 EJB	347
16.1 关于持久化的基本构架.....	347
16.1.1 数据还是对象	347
16.1.2 会话 Bean 管理的持久化.....	349
16.1.3 实体 Bean 持久化	371
16.2 在关系型数据库中存储对象	378
16.2.1 并发控制的基础	380
16.2.2 映射 EJB 到关系数据库	381
16.3 版本化 EJB	387
16.4 与遗留数据共存的设计	388
第 17 章 集成 CORBA.....	393
17.1 理解 CORBA	393
17.1.1 理解 CORBA 如何工作	394
17.1.2 RMI-CORBA 互用	398
17.1.3 使 RMI 和 CORBA 一起工作	403
17.2 结合 CORBA 与 EJB	404
第 18 章 可伸缩性设计	407
18.1 大型 EJB 系统概念.....	407
18.1.1 大型系统是什么	407
18.1.2 大型 J2EE 系统结构	409

18.2	EJB 集群	416
18.2.1	怎样把 EJB 集群起来	416
18.2.2	集群无状态会话 Bean	420
18.2.3	集群有状态会话 Bean	422
18.2.4	集群实体 Bean	424
18.2.5	集群消息驱动 Bean	426
18.2.6	在一个集群中部署 EJB	427
18.2.7	其他集群 EJB 问题	428
18.2.8	BEA WebLogic Server 6.0 集群示例	429
第 19 章	性能优化	431
19.1	有状态还是无状态	431
19.2	消息传递还是 RMI-IIOP	433
19.3	保证响应时间	434
19.4	获得 EJB 单体 (singleton)	435
19.5	使用会话和实体 Bean	436
19.6	实体 Bean 的性能调整	437
19.7	本地接口还是远程接口	438
19.8	调试 EJB	438
19.9	资源分类	439
19.10	装配组件	440
19.11	在 EJB 中使用 XML	441
19.12	与遗留系统集成	441
第 20 章	EJB 项目管理	445
20.1	项目基础	445
20.2	构建与发布	461
附录 A	EJB API	467
A.1	javax.ejb 包	467
A.2	javax.ejb.spi 包	473
A.3	javax.ejb.deployment 包	474
附录 B	J2EE 兼容应用服务器	475
B.1	J2EE SDK 测试配置	476
B.2	ATG Dynamo 应用服务器配置	478
B.3	BEA WebLogic 服务器 6.1 配置	479
B.4	Borland AppServer 4.5 配置	481
B.5	Fujitsu INTERSTAGE 配置	481
B.6	Hitachi Cosminexus 服务器标准版配置	483

B.7	HP Bluestone Total-e-Server 配置	483
B.8	IBM WebSphere 应用服务器配置	484
B.9	IONA iPortal 应用服务器 1.3 配置	487
B.10	iPlanet 应用服务器 6.0 配置	488
B.11	Macromedia Jrun 服务器配置	488
B.12	Oracle 9I 应用服务器配置	489
B.13	Persistence PowerTier 7.0 配置	489
B.14	SilverStream 应用服务器配置	490
B.15	Sybase EAServer 配置	492
B.16	TogetherSoft ControlCenter 配置	493
B.17	Trifork 企业应用服务器配置	494
B.18	J2EE SDK v1.2 配置	494
B.19	COOL:Joe 配置 - 1.3 测试	495
B.20	J2EE SDK v1.3 配置	496

开发专家之

Sun ONE

第一篇 J2EE 框架下的 EJB

本篇主要介绍服务器端开发平台——Java 2 企业版（J2EE 1.3）及其核心技术——企业 JavaBeans（EJB 2.0）组件体系结构。目的是让读者理解，在企业级开发中，为什么要选择 J2EE 平台，其核心技术 EJB 组件体系有哪些独特的优势以促使组织和开发者选择 EJB 技术进行商业建模和应用开发。

第 1 章 J2EE 概述

本章通过企业开发体系结构的发展，阐述了为什么需要 J2EE 框架，其核心技术有哪些。然后从战略角度上介绍了 Sun ONE 和 .NET 这两大开发体系结构的各自特点，并进行对比。最后，介绍了几种主流的服务器端开发的组件结构。

1.1 企业开发体系结构

长期以来，“客户端/服务器”两层结构广泛应用，即客户端提供用户界面、运行逻辑处理应用，而中央服务器接受客户端 SQL 语句并对数据库进行查询，然后返回查询结果。

两层结构系统的确给人们带来了相当的灵活性，但随着业务处理对系统提出更高的要求后，它也逐渐暴露出其客户端渐渐庞大和服务器端负担过重的现象。比如，如何提供灵活的可扩展的工作流定制；如何保证数据在网络传输的稳定性和准确性；如何应对峰值数据的高负荷处理和平衡负载等诸如此类的要求，两层结构系统往往难以满足。

于是，多层（三层或以上）结构应运而生了。所谓“多层”，就是在原有的“两层结构”（客户端和服务端）之间增加了多种服务，包括事务处理逻辑应用服务、数据库查询代理和连接适配器等。在多层部署中，表现层、商业逻辑层和数据层各自独立。对于四层或者四层以上，可以进一步的分解以便系统不同的部分能够独立地伸缩。

多层体系结构有如下特点：

- 低的部署费用

数据库驱动程序只在服务器端安装和配置而无须关心客户机。在一个服务器环境内部署和配置软件，要比在数以千计的个人电脑或用户终端上部署廉价得多。

- 低的数据库转变费用

客户端不再直接访问数据库，而是要通过中间层访问。这样不必重新部署客户端就能移植数据库模式，更换数据库驱动程序，甚至改变持久化存储类型。

- 低的商务逻辑移植费用

改变商务逻辑层不必重编译和发布商务逻辑层。

- 协同防火墙部署安全部分

许多商务应用包含敏感的受保护数据，但仍然需要部署。比如，一个基于 Web 的部署，商务应用不愿直接向外部用户直接暴露业务层。而部署还必须将表现层呈现给外部用户以便他们能够单击 Web 页面。一个解决方案是在表现层和业务逻辑层之间放置一个防火墙，如图 1-1 所示。

- 资源有效池化和重用

对于多层体系结构，外部资源连接能够很有效的被管理。客户端通常除了使用资源，

还做别的事情，像渲染图形用户接口。资源池化就是利用了这一事实。一方面业务组件获取和释放资源（比如数据库）连接，另一方面，资源能够被池化并为不同的客户请求所重用。所需的数据库连接结果集一般远少于部署在系统内组件的数目。因为数据库连接代价非常昂贵，所以这种方案增加了部署的整体可伸缩性。而且，资源连接不需要反复重建从而改善了应用性能。资源池化也能够被应用于别的资源，比如，套接字连接和线程。事实上，对于多层结构，业务组件本身也能够被多个客户端池化和重用。组件池化意味着不需为每个客户端准备一个专门的组件，对于两层瘦客户端也是这样。

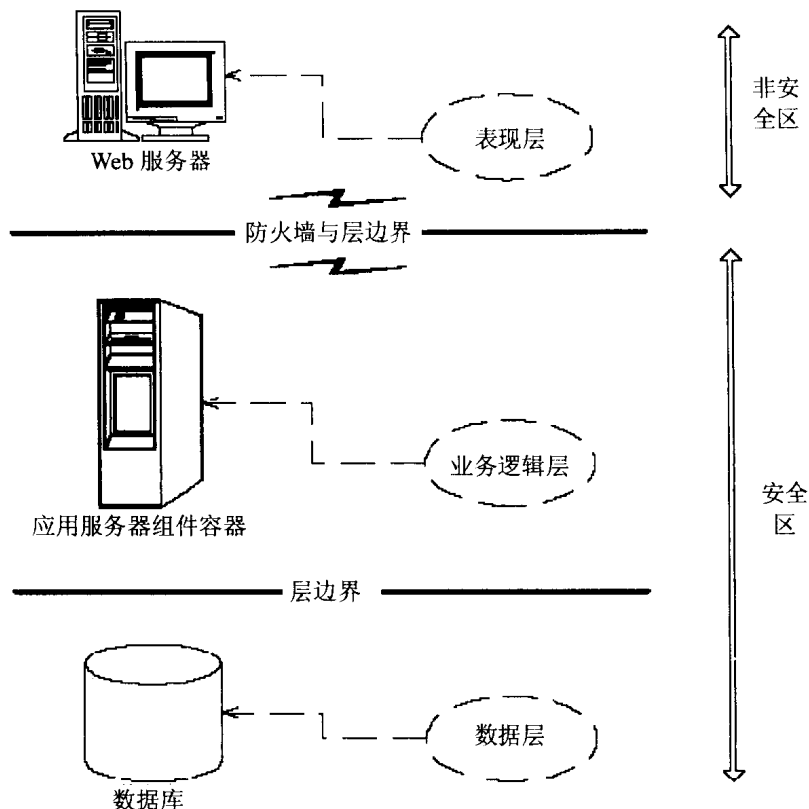


图 1-1 协同防火墙部署安全

- 每层独立变化

比如，能添加数据图像而将别的层的改变和重编译减至最小。

- 性能降低的局部化

当某一层超载时，别的层仍然能够正常工作。在 Web 部署中，即使应用服务器负载过重，用户还是能够浏览网站的首页。

- 错误局部化

当发生严重错误时，错误只停留在一个单独的层内而不扩散。而其他层能够顺畅地处理这种情形，仍然能够正常工作。例如，如果应用服务器崩溃，Web 服务器依旧能向客户浏览器报告友好的“网站暂时不可用”页面。

- 通信性能恶化

既然这些层物理上是分离的，那么他们就必须跨越进程界线、跨越机器界线，或跨越企业域界线进行通信。这导致相当高的通信负载。只有正确地设计分布式对象应用，才能获得有效的部署。不幸的是，这往往要求非常清楚各层界线并减少应用代码的位置透明度。

- 维护费用高

比如，部署三个或更多的物理分离的层，软件安装费用，软件升级费用，重部署费用和管理费用都有显著的增加。

- 多层结构实现

没有免费的午餐，多层部署的优势也是这样。在实现多层结构时，下述功能代码需要自行编写：代理方法请求；执行资源池化；处理组件生命周期；在层间处理负载平衡通信逻辑；处理多个客户端并发地访问同一个组件的结果；故障时转送客户请求到别的机器；提供安全环境使未授权客户端无法破坏系统状态；灾难发生时提供监控软件呼叫系统管理员；授权用户执行可靠的操作或其他的部分。

对于重要的服务器端计算，上述任何一个专题都是一个需要关注的单独服务。每一个服务都需要多方考虑和中间件支持。经常地，公司内部私有框架只给出这些服务的一些。那些公司正把自己引向失败，因为高端应用服务非常复杂，创建和维护异常困难，他们要求专家级的中间件知识，有业界标准的工具支持，并且跟绝大多数公司的核心业务完全无关。为什么要建造而不是购买呢？应用服务器的诞生就是为了让用户购买这些中间件服务而不是亲自建造他们。应用服务器为服务器端组件提供运行时刻环境，为组件开发者提供公共的中间件服务，像资源池化，网络等。应用服务器使得用户专注于应用本身，而不是去考虑如何实现健壮的服务器端部署中间件。

1.2 J2EE 平台基础

为了降低成本，并加快企业应用程序的设计和开发，J2EE 平台提供了一个基于组件的方法，用来设计、开发、装配及部署企业应用程序。J2EE 平台提供了多层的分布式的应用模型、组件再用、一致化的安全模型以及灵活的事务控制。

J2EE 1.3 规范定义了以下种类的组件：

- 应用的客户组件（Application client components）
- 企业 JavaBeans 组件（Enterprise JavaBeans components）
- Web 组件，包括 Servlets 及 JSP 组件（Servlets and JavaServer Pages (JSP) components）
- GUI 组件 Applets

一个多层的分布式的应用模型意味着应用逻辑根据功能而被划分成组件，并且可以在同一个服务器或不同的服务器上安装组成 J2EE 应用的这些不同的组件。一个应用组件应被安装在什么地方，取决于该应用组件属于该多层的 J2EE 环境中的哪一层。这些层是客户层、Web 层、业务层及企业信息系统层（EIS）。图 1-2 所示为多层的分布式的应用模型。

- 客户层（Client Tier）

J2EE 应用可以是基于 Web 的，也可以是不基于 Web 的。在一个基于 Web 的 J2EE 应