

Server

应用服务器 原理与实现

王千祥
飞思科技产品研发中心

编著
审校

Loading 100%



电子工业出版社

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

TP368.15
W 238
TP392.07

应用服务器 原理与实现

王千祥
飞思科技产品研发中心

编著
审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

应用服务器是网络环境中应用程序的高层运行平台,使得应用系统的代码更为简洁、开发更为方便。应用服务器被认为是继操作系统、数据库管理系统之后,随着计算机网络环境的发展而出现的里程碑式的基础软件。本书系统地介绍了应用服务器涉及的概念、原理、规范及实例等内容。特别地,本书结合人们经常接触到的 CORBA、J2EE、Web 服务等规范,详细介绍了软件互操作、软件构件、软件中间件(公共服务)等当前软件的热点技术,它们也是应用服务器的核心内容。本书的背景知识包括:计算机网络、操作系统、数据库管理系统、Java 语言等。

本书适合于应用服务器的使用、研究与开发人员,包括在校的研究生、高年级本科生,以及 IT 公司的技术人员等。

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,侵权必究。

图书在版编目(CIP)数据

应用服务器原理与实现/王千祥编著. —北京:电子工业出版社, 2003.6

ISBN 7-5053-8607-7

I.应... II.①王... III.网络服务器—基本知识 IV.TP368.5

中国版本图书馆 CIP 数据核字(2003)第 020530 号

责任编辑:赵红梅

印 刷:北京市增富印刷有限责任公司

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:787×980 1/16 印张:21.25 字数:544 千字

版 次:2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

印 数:5000 册 定价:29.00 元

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

前言

软件运行平台正在从单机环境发展为网络环境，这导致了软件在支持对象、交付形式、关注内容等方面发生了变化。

软件系统正在由个体的计算工具向群体的合作工具发展，近年来出现的电子商务、电子政务等模式是这一发展的典型表现；软件的交付形式正在从以产品为中心向以服务为中心发展，近年来出现的应用服务提供商（ASP）、Web 服务等是这一发展的典型表现；软件关注的内容正在从以正面功能为主要考虑对象向兼顾侧面约束发展，近年来出现的面向侧面的编程与早期的划分关注相呼应，是这一发展的典型表现。

值得关注的是，在软件发生上述转变过程的同时，几乎每年都有一系列的新术语、新技术出现，例如：RPC、DCE、OMA、CORBA、COM、DCOM、EJB、J2EE、DNA、.NET、Web 服务、CA、PKI 等。这些英文缩略语中的每一个都对应着一系列的概念与规范，概念如互操作、接口、构件、服务、命名、安全、代理等，规范如 JRMP、IIOP、SOAP、IDL、MIDL、WSDL、UDDI 等，软件研究与开发人员对此应接不暇。不仅如此，对于大多数人员，可供阅读的往往是一些零散的介绍或枯燥的规范。而这些规范又多数只讲结果，不讲原因，容易造成人们知其然而不知其所以然。理解起来比较困难，深入地掌握不同规范之间的复杂关系更为不易。

正是在这样一个背景下，我们决定撰写一本书，希望帮助研究人员较快地把握上面提到的种种概念与规范。但用一本书讲述大量的概念是一件危险的事情，因为这样的书容易空洞抽象，缺乏主题，缺乏条理性。幸运的是，在酝酿过程中，与应用服务器相关的一些概念与产品逐步为广大的软件开发与研究人员所接受。应用服务器被认为是网络环境中的“操作系统”，是继操作系统、数据库管理系统之后的第三类里程碑式的基础软件。不仅如此，应用服务器几乎涉及了上面提到的所有概念与规范。这给我们提供了一个极佳的主题，使我们可以围绕应用服务器所涉及的互操作、构件与公共服务等内容设计全书的组织结构。

与此同时，我们开始动手研制自主版权的应用服务器原型——PKUAS。在分析了业界的多种应用服务器之后，我们发现：由于关注重点的不同，不同应用服务器对上述概念的支持差别很大。例如：有的较少考虑互操作问题，有的对构

件的支持不足等。上述种种因素都促使我们做出了自行开发应用服务器的决定。通过这样一个开发过程的磨练,促进我们更加深入地去了解规范,去思考概念与概念、规范与实现等相互之间的关系。到后来,这反而成了本书顺利完成预期目标的关键因素。

本书不可避免地涉及到了许多规范,但我们努力避免简单地罗列各种规范,而是以探讨规范出现的动机为引子,阐述相关的概念,解释其中的原理,并介绍相关的解决思路、解决方案。书中对规范的介绍甚至可以看成是对原理的一种注解。实际上,如果想更深入地掌握某种规范的细节,仍然需要下大力气研读规范原文。

值得高兴的是,国家高技术研究发展计划(863计划)十分重视应用服务器的研究与开发,在“863计划”中支持了多个与应用服务器密切相关的重点项目,我们所开发的PKUAS也得到了“863计划”的资助。这不仅有力地推动了国内应用服务器的研究与产品化工作,增强了我们专注于应用服务器的信心,也促使我们反复修改本书,以尽量提高讲解的效果。

本书附带的一个期望值是为一些概念、术语确定一个合理的中文名字。在软件领域,由于技术发展与文化背景的不对称,许多术语首先是以英语形式出现的。不仅如此,这些术语几乎都是借用其他领域中的概念,例如Proxy、Stub、Contract、Signature、Marshal等,在借用的过程中掺杂了许多文化背景,带有很强的隐喻色彩。再加上不同语言在构造方式与演化特性等方面的不同,给许多术语的中文表述带来了很大的困难,直接翻译过来的中文十分晦涩。实际上,与英语更多地将概念与发音进行关联不同,作为一种以书写为核心的语言,汉语更多地将概念与书写符号进行关联,而且汉语在术语表达上的自由性是举世瞩目的。美国前总统肯尼迪就曾这样借用中文来解释英语单词Crisis:“在汉语中,‘危机’表达了两层含义,一是危险,一是机会”。如果通过本书的出版,能够在这方面有所促进的话,也将使作者感到十分欣慰。与国内一些专家的讨论也使我们感觉到:对于当前软件技术由于快速发展而导致不断出现新概念的状况,汉语表达上的自由性具有更强的适应能力。

另外,对于术语的中英文对照问题,一个通用的做法是在中文术语第一次出现时给出其对应的英文术语。而本书由于涉及的术语较多,采用这种对照方式将使部分章节充斥着各种英文单词,十分不利于阅读。因此,除了必需的缩写及程序代码外,本书采用正文中尽量不包含英文单词的策略,所有可以进行中英文对照的中文术语尽量以字体为楷体的形式书写,具体的中英文对照关系则放在术语索引中,其他一些重要的术语用加粗宋体的形式书写。

本书由王千祥执笔完成,但它同时也是集体工作的成果。首先,本书与杨芙

清院士的鼓励与支持是分不开的。笔者从 1997 年开始在北京大学从事博士后研究工作，后留校任教。期间，有幸得到杨老师多方面的直接指导，后来又协助杨老师主持“软件前沿技术与问题”系列学术讨论等活动。杨老师细致入微的认知态度、严谨缜密的思维方式、清晰条理的表述方法深深影响着我的学术道路。不仅如此，本书的许多思想与观点直接来自杨老师的一些公开发表的论文或未公开发表的讲稿。其次，本书也是在邵维忠教授、王立福教授、梅宏教授等老师们的关注与支持下完成的。笔者自进入北京大学以来长期与梅宏老师密切合作，共同对构件组装技术、中间件技术等进行研究，所取得的研究成果也部分反映在本书的内容之中。在本书即将出版之际，笔者向他们表示最诚挚的感谢。另外，笔者所在单位的许多老师：孟庆余教授、孙家肃教授、陈向群教授、谢冰副研究员、张世琨副教授、孙艳春副教授等，都曾与笔者通过多种途径进行广泛交流，对本书的内容产生了不同程度的影响。最后，参与 PKUAS 开发的众多研究生也对本书的出版做出了贡献。他们中的许多人在阅读了本书的初稿后，提出了有益的反馈意见。其中，博士生黄罡、博士生刘天成、硕士生陈锋的工作最为突出。其他参与应用服务器开发的同学还有：博士生曹东刚、硕士生王晓鹏、硕士生滕腾、硕士生向俊莲、硕士生王栋等。本科生申峻嵘阅读了初稿后也提出了许多很好的建议。选修北京大学计算机系硕士研究生课程《高级软件工程》、工程硕士课程《软件开发工具》的一些同学也对本书的出版提出了许多有益的建议，在此向他们一并表示感谢。

感谢电子工业出版社飞思科技产品研发中心的几位编辑，她们为本书的尽快面市做了许多细致的工作，并在我的反复修订过程中表现出了极大的耐心。

感谢我的妻子和女儿，她们对我的爱和支持从来没有间断过。

尽管付出了上述种种努力，由于软件技术仍然处于快速发展时期，许多技术、规范不一定都能像 TCP/IP 那样被长期、广泛地使用。这就给全书内容的协调性提出了挑战：部分内容可能深度不足，部分内容可能又过于琐碎，恳请读者对书中的不足之处批评指正，按下面的电子邮件地址给我发邮件，我本人对这些不足之处负全部责任。

王千祥

wqx@cs.pku.edu.cn

飞思科技产品研发中心的联系方式:

咨询电话: (010) 68134545 68131648

答疑邮件: support@fecit.com.cn

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

答 疑: <http://www.fecit.com.cn> 的“问题解答”专区

通用网址: 计算机图书、FECIT、飞思教育、飞思科技、飞思

王千祥： 山东莱州人，
获博士学位，北京大学信息
科学技术学院软件研究所副
教授，中国计算机学会软件
工程专业委员会秘书长。
1991年于国防科技大学计算
机系获得学士学位，1994
年、1997年分别于西北工业
大学获得硕士、博士学位，
1999年从北京大学计算机科
学技术系博士后流动站出站
并留校任教。目前主要从事
软件工程、网络化软件等技
术的研究与开发工作，已发
表学术论文30余篇。



作者简介

本书提纲挈领地介绍了应用服务器中的软件互操作、软件构件架构、构件容器、软件公共服务、命名目录服务、事务服务技术和安全架构等理论和实践。作为有幸使用过MTS、Orbix和几种J2EE应用服务器的老程序员，看到这本书真有一种相见恨晚的感觉。

美国花旗银行高级软件工程师、博士
——阎宏

这是一本不可多得的好书，出现在合适的时间和地点。本书的讲解透彻深刻，鞭辟入理，涵盖了基于应用服务器的企业级软件的方方面面，是一本优秀的参考书。初级开发人员可阅读本书，为更好的发展打下坚实的基础。高级开发者和架构设计人员可借由本书从整体上把握应用服务器软件的架构，从而设计出优秀的应用程序。

《高质量Java程序设计》作者
——顾小刚

应用服务器、分布式计算及其倡导的组件服务思想在现代电信网管、综合网管以及电子商务等基于Internet的应用软件开发中正发挥着重要作用。但是，目前国内还缺乏关于应用服务器的系统研究和相关著述，本书无疑会填补这一空白。本书对应用服务器的原理、体系结构、开发以及实施过程中所涉及的概念、理论和环节进行了翔实的讲解，并通过一个具体实例演示了这些过程，无论应用服务器开发人员，还是像我们这些电信网管开发人员，都会从中吸收富有价值的营养。

上海大唐移动通信设备有限公司
软件工程师
——韩永泉

目录

第 1 章 概述1	2.1.3 软件互操作.....31
1.1 为什么需要应用服务器...2	2.2 远程过程调用.....32
1.1.1 面临的问题.....2	2.2.1 工作原理.....32
1.1.2 操作系统与数据库 管理系统.....3	2.2.2 编排/还原/分派.....35
1.2 诞生基础.....4	2.2.3 基于 RPC 的开发 过程.....37
1.2.1 中间件.....4	2.3 对象请求代理.....44
1.2.2 网络软件总体结构.....5	2.3.1 公共对象请求 代理结构.....45
1.3 相关概念.....9	2.3.2 基于 ORB 的 互操作过程.....46
1.3.1 应用与应用系统.....9	2.4 互操作体系结构.....47
1.3.2 构件与构件模型.....10	2.5 应用服务器对互操作的 支持.....50
1.3.3 互操作与数据交换.....11	2.6 小结.....51
1.3.4 服务与公共服务.....12	阅读资源推荐.....51
1.3.5 软件体系结构.....13	第 3 章 互操作协议53
1.4 应用服务器.....14	3.1 互操作协议要素.....54
1.4.1 位置.....14	3.2 通用对象请求代理间 协议.....56
1.4.2 功能.....15	3.2.1 公共数据表示.....57
1.4.3 结构.....16	3.2.2 消息格式.....57
1.5 应用服务器的现状 与未来.....17	3.2.3 协议映射.....63
1.6 关于本书.....18	3.2.4 引用表示.....63
1.7 小结.....22	3.3 简单对象访问协议.....64
阅读资源推荐.....22	3.3.1 数据表示.....65
第 2 章 软件互操作25	3.3.2 消息格式.....65
2.1 背景.....26	3.3.3 协议映射.....67
2.1.1 TCP/IP.....26	
2.1.2 基于 TCP/IP 的通信 过程.....27	

3.4 互操作协议之间的关系	69	5.2.5 构件运行环境	105
3.4.1 协议之间的不同	69	5.2.6 构件种类	106
3.4.2 协议之间的映射	70	5.2.7 基于构件的软件 开发	107
3.5 小结	70	5.3 COM	108
阅读资源推荐	70	5.3.1 COM 构件	109
第 4 章 互操作接口定义语言	71	5.3.2 构件规约	110
4.1 概念	72	5.3.3 构件实现	114
4.1.1 接口	72	5.3.4 COM 构件种类	116
4.1.2 接口定义	73	5.3.5 COM 构件运行 环境	116
4.1.3 接口定义语言	75	5.4 EJB	117
4.2 OMG 的接口定义 语言	75	5.4.1 EJB 构件	117
4.2.1 IDL 规范	76	5.4.2 构件规约	118
4.2.2 IDL 文件示例	82	5.4.3 构件实现	119
4.2.3 IDL 向具体语言 的映射	83	5.4.4 EJB 构件种类	120
4.2.4 接口与 IIOP 消息	84	5.4.5 EJB 构件运行 环境	122
4.3 Web 服务描述语言	87	5.5 不同构件模型的 比较	123
4.3.1 WSDL 元素	87	5.6 应用服务器对构件的 支持	124
4.3.2 WSDL 文件示例	88	5.7 小结	125
4.3.3 WSDL 与 SOAP 消息	91	阅读资源推荐	125
4.4 不同语言的比较	92	第 6 章 构件合约	127
4.5 小结	93	6.1 构件合约	128
阅读资源推荐	93	6.1.1 构件提供的接口	128
第 5 章 软件构件	95	6.1.2 构件依赖的资源	130
5.1 背景	96	6.1.3 构件的可定制 属性	130
5.2 概念	97	6.2 EJB 构件的合约	131
5.2.1 构件	98	6.2.1 EJB 构件的类型 接口	131
5.2.2 构件规约	101		
5.2.3 构件实现	102		
5.2.4 构件模型	104		

6.2.2 EJB 构件的实例	
接口	135
6.2.3 EJB 构件的声明	139
6.3 构件访问	143
6.4 小结	148
阅读资源推荐	148
第 7 章 构件运行环境	149
7.1 概述	150
7.1.1 构件上下文	151
7.1.2 构件容器	151
7.1.3 构件约束	152
7.2 构件容器的功能	153
7.2.1 实例池	153
7.2.2 冻化/活化	153
7.2.3 生命周期管理	154
7.3 容器与实现体的合作	155
7.3.1 构件实现体向容器	
提供的方法	155
7.3.2 容器向构件实现体	
提供的方法	158
7.3.3 构件实现体与构件	
接口	161
7.4 会话构件实例的	
管理	163
7.4.1 无态构件实例的	
管理	164
7.4.2 有态构件实例的	
管理	166
7.5 实体构件实例的管理	169
7.5.1 BMP 实体构件	
实例的管理	170
7.5.2 CMP 实体构件	
实例的管理	173

7.5.3 BMP 与 CMP 实体构件	
代码比较	175
7.6 小结	179
阅读资源推荐	180
第 8 章 软件公共服务	181
8.1 背景	182
8.2 概念	185
8.2.1 公共服务	185
8.2.2 公共服务与构件	186
8.2.3 公共服务与构件	
容器	186
8.2.4 公共服务的接口	186
8.2.5 公共服务的	
使用方式	187
8.2.6 公共服务的实现	188
8.3 OMA 中的公共服务	188
8.3.1 公共服务设计原则	189
8.3.2 公共服务规范	189
8.3.3 OMA 的调用式	
使用方式	192
8.4 J2EE 中的公共服务	192
8.4.1 J2EE 中的公共服务	
接口	192
8.4.2 J2EE 的声明式	
使用方式	193
8.5 应用系统组装	193
8.5.1 什么是组装	193
8.5.2 基于 EJB 构件的	
组装	195
8.6 小结	201
阅读资源推荐	201
第 9 章 查找服务	203
9.1 应用系统的分布性	204

9.2 命名服务.....206	10.5 小结.....251
9.2.1 实体的名字.....206	阅读资源推荐.....251
9.2.2 命名服务的基本 实现.....210	第 11 章 安全服务.....253
9.2.3 命名上下文.....210	11.1 应用系统的安全性.....254
9.2.4 解析名字.....211	11.2 安全技术与安全 体系.....256
9.2.5 OMA 的命名服务.....211	11.2.1 加密.....256
9.3 目录服务.....213	11.2.2 认证.....260
9.3.1 基本实现.....213	11.2.3 访问控制.....262
9.3.2 LDAP.....214	11.2.4 审计.....264
9.4 合约服务.....215	11.2.5 安全体系.....264
9.5 JNDI.....218	11.3 安全基础设施.....265
9.5.1 JNDI 体系结构.....218	11.3.1 公钥基础设施.....265
9.5.2 查找的声明.....221	11.3.2 IP 层安全.....265
9.6 小结.....223	11.3.3 传输层安全.....267
阅读资源推荐.....223	11.3.4 Java 安全体系.....268
第 10 章 事务服务.....225	11.4 Java 认证与授权 服务.....273
10.1 应用系统的可靠性.....226	11.4.1 认证.....274
10.1.1 什么是可靠性.....226	11.4.2 授权.....276
10.1.2 状态错误.....226	11.5 J2EE 的安全服务.....277
10.1.3 事务.....229	11.5.1 安全服务模型.....278
10.2 事务技术.....232	11.5.2 安全服务接口.....278
10.2.1 单机环境中的 事务.....232	11.5.3 安全性声明.....282
10.2.2 网络环境中的 事务.....234	11.6 小结.....288
10.2.3 X/Open 的 DTP 模型.....236	阅读资源推荐.....288
10.3 OMA 的事务服务.....237	第 12 章 应用服务器实例.....289
10.4 J2EE 的事务服务.....240	12.1 设计目标.....290
10.4.1 事务服务模型.....240	12.2 系统结构.....291
10.4.2 事务服务接口.....243	12.2.1 总体结构.....291
10.4.3 事务性声明.....246	12.2.2 应用.....292
	12.2.3 构件容器.....293
	12.2.4 命名服务.....294

12.2.5	事务服务	295
12.2.6	安全服务	295
12.2.7	通信服务	296
12.2.8	系统监控	297

12.3	PKUAS 的启动过程 ..	298
	参考文献	301
附录 A	缩略语索引	305
附录 B	术语索引	309

1

CHAPTER 概述

软件运行平台正在从单机环境发展为网络环境,应用服务器是网络环境中应用系统的高层运行平台。正确地使用应用服务器,将使应用系统的代码更为简洁,使开发人员的精力可以更加集中于系统的逻辑部分。因此,目前运行在网络环境中的应用软件正在越来越多地以应用服务器为基础进行开发。

问题:

1. 为什么需要应用服务器?
2. 什么是应用服务器?
3. 应用服务器有哪些诞生基础?
4. 应用服务器具有什么功能?
5. 应用服务器的现状与未来如何?
6. 本书是怎样讲解应用服务器的?

计算机网络的发展，特别是因特网的广泛使用，极大地丰富了软件的内涵。通常，人们将软件划分为系统软件、支撑软件与应用软件三类。其中，系统软件主要是指操作系统等与系统资源密切相关的软件，支撑软件是指软件开发工具等支持开发过程的软件（系统软件与支撑软件又可以合称为基础软件），而应用软件则是指特定于应用领域的专用软件。应用服务器是随着计算机网络的发展，为方便应用软件的开发与维护而逐步形成的、具有里程碑意义的基础软件。简单地讲，应用服务器是网络环境中应用系统的高层运行平台，使应用系统的代码更为简洁，使开发人员的精力可以更加集中于系统的逻辑部分。

1.1 为什么需要应用服务器

分布在网络环境中的大型软件系统完全不同于单机环境中的软件。在本书中，网络环境是指一群通过网络互相连接的处理系统，每个处理节点（大型机、台式机、笔记本电脑，或者其他嵌入式设备等）由处理机硬件、操作系统及基本通信软件等组成。网络软件则是指运行在网络环境中的软件系统。

1.1.1 面临的问题

随着计算机网络向社会各个方面的持续渗透，软件正变得无处不在。与此同时，软件所面临的挑战也正在日益加剧：软件需要实现的功能与需要满足的约束越来越多，而且这些需求呈现出越来越强的变化性。软件在规模不断增长的同时，复杂性表现得更加突出。

对于网络软件，分布在不同节点上的应用程序不仅需要能够互相通信，还需要能够互相协作，以共同完成目标任务。网络环境具有明显的分布性、开放性、演化性、异构性、并发性等特点[George 2001]。运行在这样一种环境下的软件系统必须充分适应、利用这些特点。因此，网络软件必须解决互操作、数据交换，以及相关的分布性、可靠性、安全性等问题，这导致了开发网络环境中的大型软件系统是一项艰巨的任务。

在软件发展史上，人们曾多次面临与上述困境相似的情形：尽管人们控制复杂性的能力相对稳定，但面临的问题却越来越多。针对这种困境，人们通常采取的主要策略之一是：提取软件的共性成分，屏蔽系统低层的复杂度，从而在高层保持复杂度的相对稳定。这样的一个过程往往导致新型系统软件的产生。操作系统与数据库管理系统都是经历了这样的过程而诞生的。

1.1.2 操作系统与数据库管理系统

在计算机发展的初期，计算机系统基本上仅是由硬件组成的。用户用纸带编写的应用程序直接控制硬件的各种运行细节，大量的代码被用于管理各种物理器件。以访问数据为例，程序必须控制怎样连接磁盘，如何读取其中的数据，如何处理各种例外情况等。可以想像，编写这样的程序需要掌握大量的关于器件接口的知识，这不仅使代码十分庞大，而且代码的正确性难以保证。必须寻找一种方法，将程序员从复杂的硬件控制中解脱出来。人们最终找到的办法是在硬件的基础上，从各种应用程序中提取出一层共性软件，用来专门管理各种硬件资源。这层软件就是今天与计算机密不可分的操作系统（OS），如图 1-1 所示。

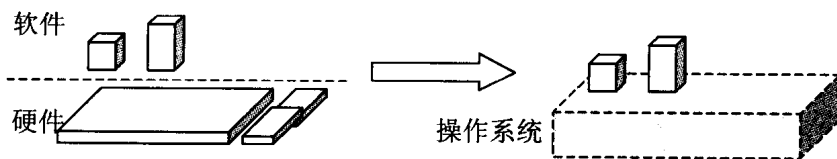


图 1-1 操作系统

初期的操作系统被称为管理程序，或监督程序，它提供了大量与硬件相关的代码（系统调用）来完成上层应用程序的各种请求，这些系统调用隐藏了与硬件相关的程序执行过程的复杂性，将硬件裸机改造成抽象的虚拟机。操作系统管理处理器、存储器、显示器、打印机、时钟等系统资源，向用户提供良好的界面，使计算机系统易学、易用，提高了应用系统的开发效率。

在操作系统出现之后，随着计算机应用范围的扩大，需要处理的数据迅速膨胀。最初，数据与程序一样，以简单文件作为主要的存储形式。以这种方式组织的数据逻辑简单，但可扩展性差，访问这种数据的程序需要了解数据的具体组织格式。特别地，当系统数据量大或者用户访问量大时，应用程序还需要解决数据的完整性、一致性及安全性等一系列问题。因此，必须开发出一种系统软件，它应该能够像操作系统屏蔽了硬件访问复杂性那样，屏蔽数据访问的复杂性。由此产生了数据库管理系统（DBMS），如图 1-2 所示。