面向对象程序设计系列教材

PowerBuilder

面向对象开发教程

崔巍 编著

高等教育出版社

内 容 简 介

本书以 PowerBuilder 8.0 为工具和环境(同样适用于 PowerBuilder 9.0),介绍了如何利用可视化技术、面向对象技术及组件技术等开发数据库应用系统,使读者了解当前计算机应用和计算机应用开发工具的最新发展,了解可视化、面向对象技术的开发方法和特点。

本书可作为高等学校计算机专业、部分非计算机专业学习数据库开发及其工具的教材和学习 PowerBuilder 的培训教材,也可以供广大数据库应用系统开发人员阅读、参考。

图书在版编目(CIP)数据

PowerBuilder 面向对象开发教程/崔巍编著 . —北京:
高等教育出版社,2002.12
ISBN 7-04-011555-7

Ⅰ. P... Ⅱ. 崔 ... Ⅲ. 数据库系统—软件工具,
PowerBuilder —高等学校—教材 Ⅳ. TP311.56

中国版本图书馆 CIP 数据核字(2002)第 096716 号

出版发行	高等教育出版社	购书热线	010-64054588
社　　址	北京市东城区沙滩后街 55 号	免费咨询	800-810-0598
邮政编码	100009	网　　址	http://www.hep.edu.cn
传　　真	010-64014048		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	北京铭成印刷有限公司		
开　　本	787×1092　1/16	版　　次	2002 年 12 月第 1 版
印　　张	22.75	印　　次	2002 年 12 月第 1 次印刷
字　　数	550 000	定　　价	28.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

前　言

国内的数据库应用广泛始于20世纪80年代，大部分中、小型应用都经历过FoxBase/FoxPro时代。但是随着计算机技术和应用需求的发展，数据库应用的模式已经由原来的单机/主机模式、网络文件服务器模式发展到现在的客户/服务器模式、分布式计算模式和浏览器/服务器模式等，相应的运行环境和开发工具都有了很大变化。

现在可以用于开发数据库应用系统的工具的确很多，例如PowerBuilder、Delphi、Visual Basic、Visual FoxPro等等，它们各有所长、各具千秋，但是PowerBuilder的数据窗口技术却使PowerBuilder应用程序与数据库交互更加方便、更加如鱼得水。

现在的PowerBuilder已经不仅仅是传统意义上的数据库应用系统开发工具，它还支持多层组件开发技术、分布式应用技术和因特网上的Web应用开发。

作为一个当代大学生（不管是计算机专业、还是非计算机专业）在学习了程序设计课程之后，不能停留在“学过某种程序语言”或“编过一点程序”的水平上，应该实实在在地掌握一些开发工具，在走出校门前就具备开发应用系统的基本能力。

本书以PowerBuilder为工具和环境，介绍了利用可视化技术、面向对象技术及组件技术等开发应用系统，使读者了解当前计算机应用和计算机应用开发工具的最新发展，了解可视化、面向对象技术的开发方法和特点。

本书共有11章。第1章基础知识主要内容包括数据库应用模式的发展和变化，数据库应用系统的开发方法，特别强调了面向对象的方法和使用PowerBuilder开发应用系统的基本思路。第2章PowerBuilder简介全面介绍了PowerBuilder的特点、功能、开发环境和PowerBuilder 8.0的安装。第3章介绍了PowerBuilder的编程语言PowerScript。第4章介绍了关系数据库标准语言SQL，其中包括标准SQL、SQL的嵌入使用方式、动态SQL语句等内容。第5章介绍了PowerBuilder的数据库操作。第6章通过一个比较完整的实例使读者初步体验PowerBuilder的开发方法。第7章简单介绍了PowerBuilder程序的调试与发行。第8章全面介绍了数据窗口技术，其中包括数据存储对象和数据窗口的动态应用等较深入的内容。第9章全面介绍了PowerBuilder的应用程序界面设计，其中包括一些比较复杂的控件的应用和界面的设计，如选项卡界面、列表窗口界面、树状结构窗口界面等。第10章介绍了面向对象的可重用组件的开发。第11章介绍了一些常用的编程技术，例如，注册表的使用、数据管道技术、动态链接库的调用和RichText应用等。

本教材的教学目标是使同学们了解最新的数据库应用系统开发方法和工具，初步掌握面向对象、可视化、组件技术等代表目前先进开发思想的编程技术，为同学们今后参加信息系统和数据库应用项目的开发打下一个良好的基础。

另外由于PowerBuilder的内容非常丰富，不可能在本书中面面俱到，也不可能很详尽，如果读者通过本书的学习能够基本掌握现代数据库应用的开发方法和开发工具，掌握PowerBuilder的开发思路和开发技术，能够为以后的数据库应用开发工作奠定一个良好的基础，那么笔者也就甚感欣慰了。

本书以 PowerBuilder 8.0 为基础，虽然 PowerBuilder 9.0 可能将于不久发布，但是 9.0 版的开发界面、基本构成和使用方法与 8.0 版是完全一样的(9.0 版主要增加了一些底层的开发技术和手段)，所以本书也将完全适用于 9.0 版。

由于时间仓促和作者水平有限，书中疏漏之处在所难免，欢迎广大读者批评指正。

作 者

2002 年 10 月

目　录

第1章　基础知识

PowerBuilder是一个支持面向对象技术，支持客户/服务器机制，支持分布式组件开发，支持Web应用的数据库应用系统开发工具。本章首先介绍数据库应用的模式及其演变过程，数据库开发方法的变化，面向对象的方法以及使用PowerBuilder开发的基本思路等。

1.1　数据库应用模式的发展

计算机的应用结构经历了集中式结构、文件服务器的网络结构到目前的客户/服务器网络结构，分布式客户/服务器网络结构和Web网络结构的发展阶段。以下通过概述这几种结构的特点，来充分理解客户/服务器网络结构、分布式客户/服务器网络结构和Web网络结构，以及为什么使用这些结构等。

1.1.1　主机应用模式

在20世纪60～70年代，真正需要使用计算机的企业都会考虑使用大型机，它代表一种集中式的系统结构。它由两个关键硬件组成：主机和客户终端，如图1－1所示，这里的主机相当于现在的服务器。

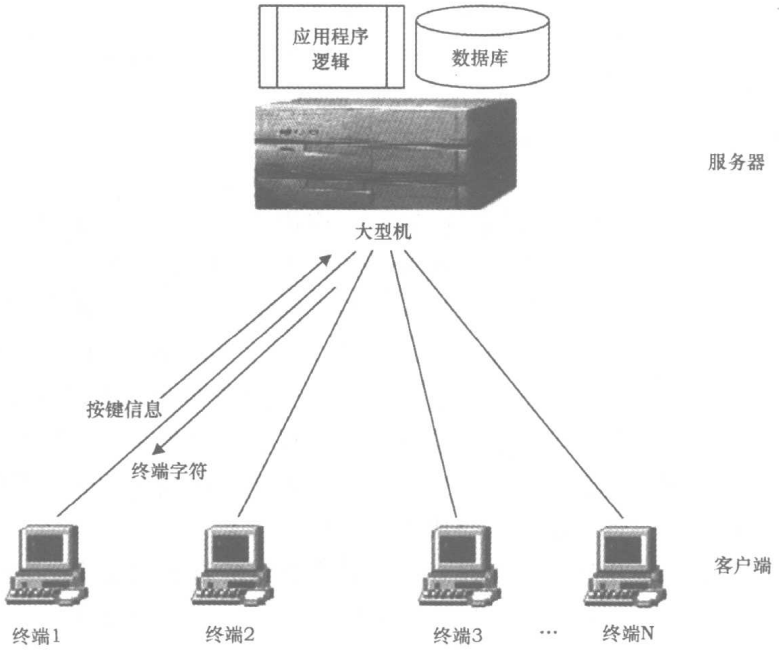

图1－1　集中式主机结构

在集中式结构中，客户终端和主机之间传递数据的方式非常简单，一是用户从客户终端键盘输入信息到主机，二是由主机返回到终端上的字符。这个时期的计算机的所有资源（数据）都在主机上，所有处理（程序）也在主机上完成。这种结构的优点是可以实现集中管理，安全性很好。这种计算机的费用非常昂贵，并且应用程序和数据库都存放在主机中，没有办法真正划分应用程序的逻辑。

1.1.2　文件服务器应用模式

到 20 世纪 80 年代，个人计算机进入了商业舞台，同时计算机应用的范围和领域也日趋广泛。这对那些没有能力实现大型机方案的企业来说，个人计算机无疑就有了用武之地。在个人计算机进入商用领域不久，局域网也问世了，同时也诞生了文件服务器技术，文件服务器结构如图 1－2 所示。

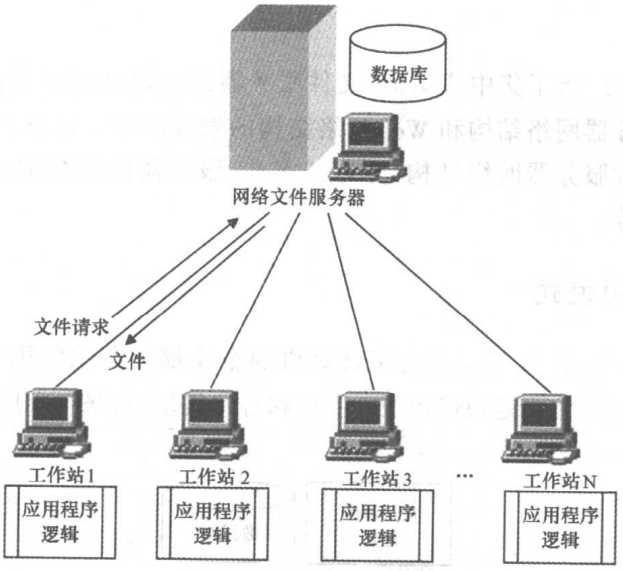

图 1－2　文件服务器结构

从图 1－2 可以看出，在文件服务器系统结构中，应用程序是在客户工作站上运行的，而不是在服务器上运行的，文件服务器只提供了资源（数据）的集中管理和访问途径。这种结构的特点是将共享数据资源集中管理，而将应用程序分散安排在各个客户工作站上。文件服务器结构的优点在于实现的费用比较低廉，而且配置非常灵活，在一个局域网中可以方便地增减客户端工作站。但是，文件服务器结构的缺点也非常明显，由于文件服务器只提供文件服务，所有的应用处理都要在客户端完成，这就意味着客户端的个人计算机必须要有足够的能力，以便执行需要的任何程序，或能完成任何必要的任务。这可能经常需要客户端的计算机升级，否则改进应用程序的功能，提高应用程序的性能等都会成为一句空话。

特别要提出的是，虽然应用程序可以存放在网络文件服务器的硬盘上，但它每次都要传送到客户端的个人计算机的内存中执行。另外，所有的处理都是在客户端完成的，网络上就要经常传送大量无用的数据。

1.1.3 客户/服务器应用模式

文件服务器结构的费用虽然低廉,但是和大型机的集中式相比,它缺乏足够的计算和处理能力。为了解决费用和性能的矛盾,客户/服务器结构就应运而生了,这种结构允许应用程序分别在客户工作站和服务器(注意,不再是文件服务器)上执行,可以合理划分应用逻辑,充分发挥客户工作站和服务器两方面的性能。客户/服务器的结构如图 1-3 所示。

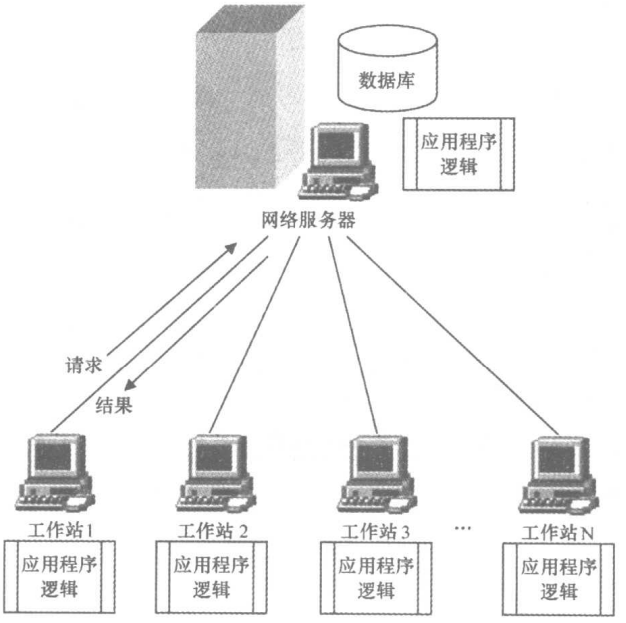

图 1-3 客户/服务器结构

在客户/服务器结构中,应用程序或应用逻辑可以根据需要划分在服务器和客户工作站中。这样,为了完成一个特定的任务,客户工作站上的程序和服务器上的程序可以协同工作。从图 1-3 可以看出客户/服务器结构和文件服务器结构的区别,客户/服务器结构的客户工作站向服务器发送的是处理请求,而不是文件请求;服务器返回的是处理的结果,而不是整个文件。

目前客户/服务器最流行的领域就是数据库应用领域,比较著名的数据库厂商都提供了支持客户/服务器结构的数据库管理系统,例如,Microsoft 的 SQL Server,Sybase 的 Adaptive Server 和 Oracle 等。

从以上可以看出,大型机集中式结构的所有程序都在主机内执行,而文件服务器局域网结构的所有程序都在客户端执行,这两种结构都不能提供真正的可伸缩应用系统框架。而客户/服务器结构则可以将应用逻辑分布在客户工作站和服务器之间,可以提供更快、更有效的应用程序性能。

在客户/服务器结构中,常把客户端称做前台或前端客户,把服务器称做后台或后端服务器。

1.1.4 文件服务器与客户/服务器的数据库操作

Microsoft 的 Access 和 FoxPro 都是非常流行的数据库管理系统,它们的数据库都可以建立在

服务器上，也允许多用户访问，但它们绝对不是客户/服务器方案，它们是文件服务器方案。为此，需要搞清楚文件服务器方案和客户/服务器方案的区别，如果对它们之间的差异没有清醒的和正确的认识，即便有支持客户/服务器方案的环境，也不可能充分利用客户/服务器结构提供的强大功能。这两种结构之间的差异如图 1－4 和图 1－5 所示，假设在数据库上都有一个含有 30 000 条记录的历届学生的数据库表格 student(并且进一步假设是 FoxPro 的 DBF 格式文件)，并且要求查询学号为 0121116 的学生的信息。

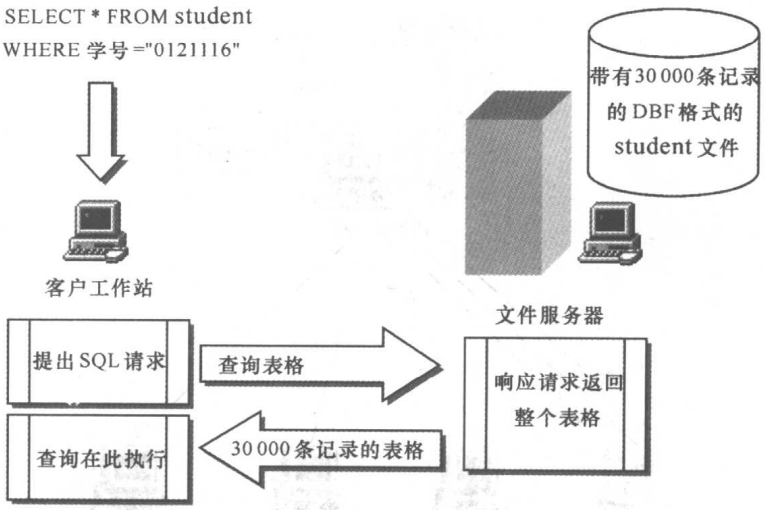

图 1－4　访问文件服务器的数据库

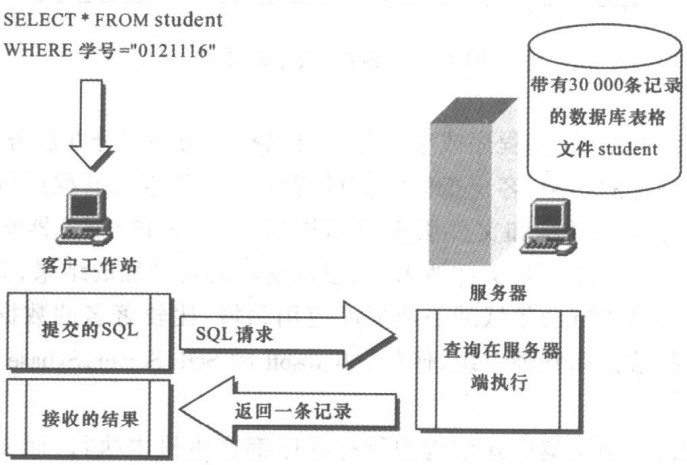

图 1－5　访问客户/服务器的数据库

图 1－4 是文件服务器结构，查询根本不会传送给服务器，查询是在客户端完成的。客户端的系统根据访问数据库的要求，它“意识”到自己需要一张数据表，以便进行查询和得出结果。因此，在正式执行 SELECT 语句之前，查询逻辑会请求通过网络将 30 000 行的表格传送到本地的客户工作站上，然后在客户端进行处理、完成查询，如果需要查找的记录不存在，则网络传输实际是在做无用功。由此看来，文件服务器只是负责文件的集中管理，并根据客户端的请求向

客户端发送文件,除此之外文件服务器不会执行其他任何程序逻辑。显然,文件服务器的处理方式会增加网络线路的传输负荷,降低传输的效率和响应时间,很容易造成网络阻塞。

图 1-5 是客户/服务器结构,实际的 SQL 查询语句将在服务器中执行,服务器发送给客户端工作站的只是查询的结果。也就是说,客户端的应用程序向数据库服务器(即运行在服务器端的数据库管理系统)发出的请求是"传送数据库表 student 中学号是 0121116 的记录",数据库服务器响应该请求后,在服务器上对数据库表 student 进行查询,并通过网络把查询到的一条记录(而不是 30 000 条)返回给客户端的用户程序。所以,客户/服务器结构降低了网络线路的高负荷,提高了传输的效率,较好地避免了造成网络阻塞的问题,并且在多数情况下都能明显缩短响应时间。

客户/服务器结构的核心是当前端用户需要后台服务器的服务时,仅仅发出请求,而服务器接受该请求后,执行相应的功能,并把满足条件的那部分数据反馈给前台客户端。

图 1-4 和图 1-5 之间的区别清楚地说明了客户/服务器方案的显著优点。由于查询和操作可以在数据库服务器上完成,所以完全有理由配置一台功能强大的服务器,所有客户工作站都可以从中得到好处。

客户/服务器结构的另一个主要特点是,数据库服务器的平台与客户端无关(无论是软件平台还是硬件平台)。数据库服务器上的数据库管理系统集中负责管理数据库服务器上的数据和资源,它向客户端提供一个开放的使用环境,客户端的用户通过数据库接口和 SQL 语言访问数据库。也就是说,不管客户端采用的是什么样的硬件平台和软件环境,它只要能够通过网络协议和数据库接口程序连接到服务器就可以对数据库进行访问。

1.1.5 分布式计算应用模式

早期的客户/服务器结构均是基于双层结构的,即一层是客户层,一层是服务器层。这种结构存在着以下一些缺点。

1. 缺乏中心控制

传统的客户/服务器结构存在的一个主要问题是:在网络计算机系统的中心位置不可能插接应用程序组件;而在很多企业应用中,由于业务逻辑的要求,经常需要为每个客户机插接一个应用程序,这将给程序的维护带来很大的困难。在这种环境下,即使是对应用程序产品做一些小的修改,也会有很多困难。

2. 缺乏安全性

在两层模式的客户/服务器结构中,客户机要经常对处理敏感业务数据的算法进行访问;而非中心化的计算环境,无法对客户的信息存取进行控制,容易产生安全漏洞。

3. 沉重的客户端负载

由于应用程序的业务逻辑存储在每台客户机上,所以要求客户机有处理大量数据的能力,这会给桌面计算机资源带来过分沉重的负担。如果应用程序非常复杂,桌面计算机就必须升级去适应程序的要求;有些时候这种要求可能超过了标准桌面计算机的能力。例如,标准桌面计算机可能就不支持一些新的、先进的操作系统特征(如多线程、对称多处理)的应用程序。如果不能发挥多任务服务器的计算能力,客户端应用程序就不能利用这些先进的操作系统特性。

为了适应企业应用和迎接各种挑战,又出现了一种三层(或称为多层)客户/服务器模型。

所谓三层，实际就是在客户层和服务器层之间又添加了一个中间层，这个中间层一般用于实现商业或企业规则等。现在人们说的中间件常常也是这个意思。Sybase 已经将自己的数据库产品命名为 Adaptive Server，这里 Adaptive 的意思是“适应”或“自适应”，它通过引入中间件使系统配置更加灵活。

双层模型的物理实现方式为：一台桌面个人计算机做客户工作站使用，而另一台网络服务器则用于后台的数据库服务器。在双层模型中，程序逻辑分担在客户机和服务器两个位置，而其中的商业或企业规则或者放在应用程序中驻留在客户工作站，或者安排在后台服务器以规则、触发器或存储过程形式实现。由于商业或企业规则不能封装，不能进行集中配置和管理，有时会有一些限制或麻烦。

为了克服以上问题，引入了一种经过改进的客户/服务器配置结构，即三层客户/服务器结构，如图 1－6 所示。在这个结构中有三个逻辑层，客户层是面向用户服务的，服务器层是面向数据服务的，而中间层是面向商业或企业规则的。通过这种划分可以将程序代码划分成不同的逻辑组件，每个组件都可以分配给能发挥最佳性能的一台机器。需要说明的是，中间件不一定要对应一个物理层，即中间件不一定需要一台独立的服务器，中间件是一个方案，需要相应的软件支持，它可以安排在后台服务器上。

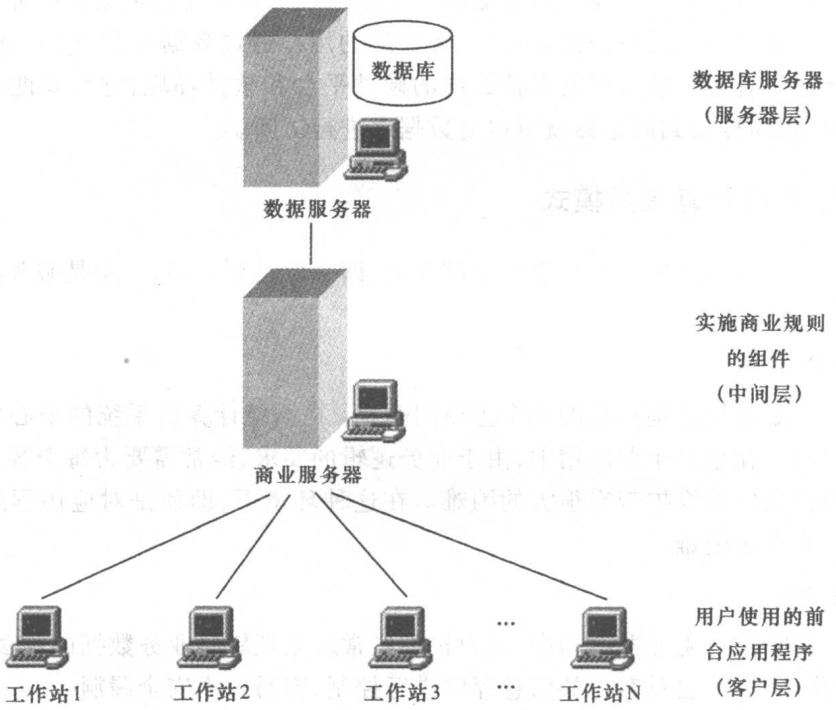

图 1－6　三层客户/服务器结构

这种多层客户/服务器结构也称做分布式客户/服务器结构，它强调的是组件开发，将原来很多客户端的处理逻辑剥离出来，形成相对独立的组件模块，这些模块安排在服务器上，供所有的客户端应用程序访问。

从三层结构的思想，可以概括出以下四个方面的优点。

1. 可重复使用

进行组件的设计和实施,可以在以后不同的应用程序中共享它们。

2. 性能改善

由于可在客户工作站以外的其他计算机上配置组件,所以能将计算负担从性能不高的客户机转移到功能强大的服务器。利用这种配置和设计的灵活性,可以充分利用计算机资源,使应用程序的执行达到最佳状态,并由此获得更好的性能。

3. 易于管理

将应用程序的服务封装到各种组件之后,可将大型、复杂的应用程序划分为更易管理的模块。

4. 易于维护

由于将组件集中起来,以便重复使用,所以一旦需要对某种组件进行修改,便可更容易地重新进行设计和配置,这样更能随时适应商业或企业规则的变化。

1.1.6 Web 网络应用模式

因特网(Internet)是一个全球性的计算机网络系统,它可将分布在世界各地的各种计算机系统及各种网络用户连接在一起,使他们通过采用共同的网络通信协议在不同的网络和操作系统间交换数据。

WWW(World Wide Web),万维网是因特网上最为流行的信息服务。随着 WWW 的迅速扩展,WWW 上可用数据源的数量也在迅速增长。因此,人们正试图把 WWW 上的数据源集成为一个完整的 Web 数据库,从而使这些数据资源得到充分利用。

有时也将 WWW 称做 Web,用户通过自己计算机上的 Web 浏览器软件来浏览 Web 服务器上的信息。我们所看到的信息页面称做主页(Home Page),主页是用一种称做超文本(hypertext)的文本形式组织的,有一种专门用于组织超文本的 HTML 语言,即在文本文件中加入了一些特殊的 HTML 标记,它可以用来控制主页的格式等。

以前的 Web 主页只能提供类似这样的静态信息,用户用浏览器所看到的所有信息,都是技术人员事先编辑好并放在 Web 服务器上供用户阅读的。

在传统的 Web 服务器中,文本和其他多媒体信息都是以文件的形式来进行存储和管理的,随着信息量的不断增加,系统的速度等性能受到越来越大的影响。另一方面,WWW 的应用领域在不断拓展,静态的 Web 页面也越来越不能满足对信息服务的动态性、实时性和交互性的要求。

而数据库技术经过几十年的发展,其功能越来越强大。各种数据库系统,例如,Oracle、DB2、Sybase、SQL Server 等,都具有对大批量数据进行有效的组织管理和快速的查询检索功能。

因此,将 Web 技术与数据库技术相结合,开发动态的 Web 数据库应用势在必行。

实现 Web 数据库的最常用方法与第 1.1.5 小节"分布式计算模式"中介绍的组件方法非常类似,它是在 Web 服务器端提供中间件来连接 Web 服务器和数据库服务器,如图 1-7 所示。

中间件负责管理 Web 服务器和数据库服务器之间的通信并提供应用程序服务,它能够直接访问数据库或调用外部程序或利用程序代码来访问数据库,因此它可以提供与数据库相关的

动态 HTML 页面，或执行用户查询，并将查询结果格式化成 HTML 页面，然后通过 Web 服务器返回给用户浏览器。

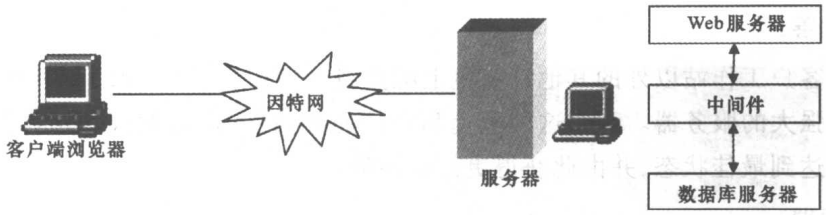

图 1-7　使用中间件的 Web 数据库应用模式

除了在 Web 服务器端采用中间件以外，还可以通过 Web 浏览器把应用下载到客户端运行，在客户端直接访问数据库，如图 1-8 所示。客户端应用包括：Java Applet、ActiveX、Plug-in 等，其中最典型的就是 Java Applet。

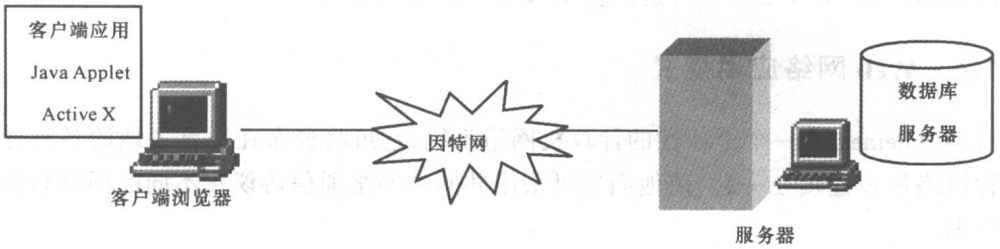

图 1-8　直接访问 Web 数据库的模式

利用 Java Applet 可以方便地实现与用户的交互，还能提供丰富的图形功能和声音、视频等多媒体功能。特别是由于 Java 是一种与平台无关的编程语言，因而具有极强的可移植性。在 Java Applet 中访问数据库，可以使用 JDBC(Java DataBase Connectivity)技术，通过 JDBC 提供的 API 来实现对分布在网上的不同数据库的各种操作。另外还可以把对数据库的访问转交给专用服务器来完成，而 Java Applet 通过与专用服务器的 Socket 通信来传递数据库操作的请求和结果。

1.2　数据库应用系统开发方法概述

掌握一个开发工具容易，而运用开发工具开发出一个好的系统却不那么容易。“软件危机”这个词出现已经有二十多年了，它说明软件难于开发、难于维护，特别是像信息系统这样的软件成功率就更低。信息系统开发方法目前主要可以分为结构化生命周期法、快速原型法和面向对象方法等。

在本节将对结构化生命周期法、快速原型法做一些概要的介绍，然后在第 1.3 节中介绍面向对象方法。任何一种方法都不是万能的，要领会其精髓，只有在实践中去体会、去摸索适合各种信息系统的项目开发方法。

1.2.1 结构化生命周期法

结构化生命周期法是基于软件工程思想的一种信息系统设计方法，它的基本思想是把系统开发看作是工程项目(如建筑工程)，需要经过用户需求、可行性分析、立项批准、设计、施工(编程实施)、验收和最后交付使用等过程。信息系统的生命周期可以分为如下5个阶段：

(1) 确定系统需求；
(2) 系统开发；
(3) 系统安装配置；
(4) 系统运行；
(5) 系统切换。

其中的系统开发又可以分为如下七个步骤：

(1) 系统调查及可行性分析；
(2) 系统分析(需求分析)；
(3) 概要设计(总体设计)；
(4) 详细设计(模块设计)；
(5) 系统实现(编程)；
(6) 系统调试与试运行(测试)；
(7) 系统运行、评价与维护(运行)。

以上7个步骤可以分为3个时期，即系统定义时期、系统开发时期和系统运行维护时期，结构化生命周期方法的开发过程如图1-9所示。

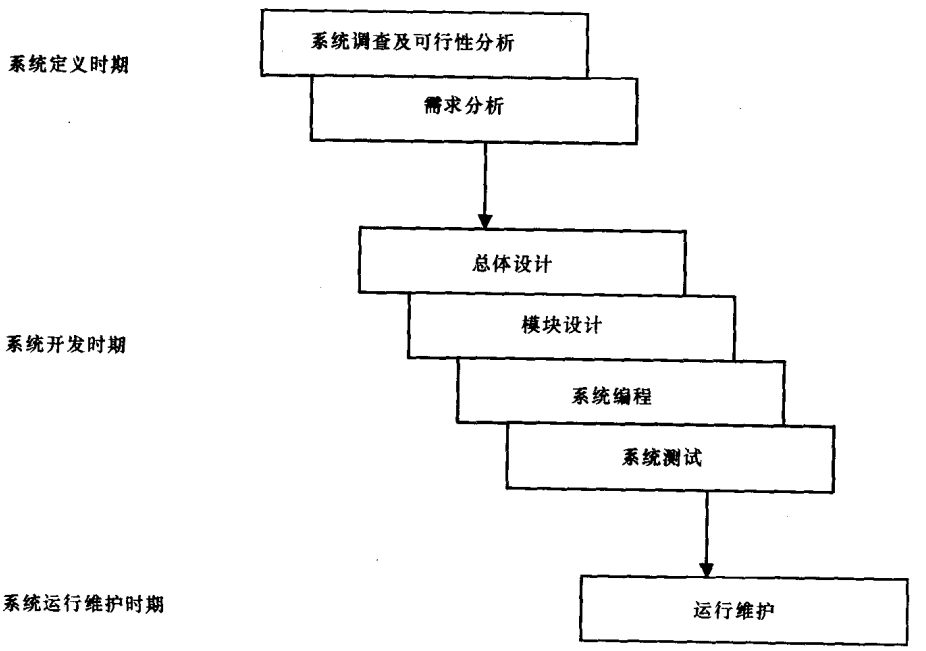

图1-9 结构化生命周期法的开发过程