

21世纪

高等院校计算机系列教材

ASP 实用技术

——网络数据库应用系统设计

李禹生 刘 兵 等编著



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校计算机系列教材

ASP 实用技术——网络 数据库应用系统设计

李禹生 刘 兵 等编著

中国水利水电出版社

内 容 提 要

本书全面介绍网络数据库应用系统设计中使用的 ASP 实用技术,主要包括:网络数据库基础知识与 HTML 基础、客户端脚本语言 VBScript、ASP 技术基础、ASP 对象、SQL 与 ADO 组件模型、ASP 中的 ActiveX 组件以及网络数据库应用系统集成技术。

本书采用任务驱动、实例引导的方式,讲解应用 ASP 技术进行网络数据库应用系统设计的基本技能。作为贯穿全书的网络数据库应用系统实例——网上商店,具有良好的通用性。读者可以在这个实例的引导下,逐步掌握网络数据库应用系统设计方法与基本技术。

本书注重理论与实际相结合,强调实用性和实践性,内容丰富、结构完整,讲述过程循序渐进,深入浅出。每一章均配有相关习题,读者通过习题可以归纳所学内容,巩固所学知识。

本书适合作为高等院校计算机、网络工程及相关专业的教材,也可作为企事业单位相关技术人员和计算机应用人员的培训教材或参考书。

本书所配电子教案、程序源代码及相关素材均可从中国水利水电出版社网站下载,网址为:

www.waterpub.com.cn

图书在版编目(CIP)数据

ASP 实用技术:网络数据库应用系统设计/李禹生等编著. —北京:中国水利水电出版社,2004

(21 世纪高等院校计算机系列教材)

ISBN 7-5084-2298-8

I. A... II. 李... III. 主页制作-程序设计-高等学校-教材
IV. TP393.092

中国版本图书馆 CIP 数据核字(2004)第 076502 号

书 名	ASP 实用技术——网络数据库应用系统设计
作 者	李禹生 刘 兵 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	北京市天竺颖华印刷厂
规 格	787mm × 1092mm 16 开本 18.75 印张 417 千字
版 次	2004 年 8 月第 1 版 2004 年 8 月第 1 次印刷
印 数	0001—5000 册
定 价	26.00 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

前 言

随着计算机技术的发展,近十年来,Internet 技术被应用于各个社会领域,成为当今应用范围最广、影响最深刻的一项科学技术。其中的 Web 技术采用浏览器和超文本链接,得到了众多 Internet 使用者的青睐。同时,Web 技术也得到了众多计算机应用系统开发者的认真研究,形成了一系列基于 Web 技术的计算机应用系统。在数据库应用领域,Web 技术为网络数据库应用系统提供了一种全新的应用模式。

将 Web 技术与数据库技术相结合,使得 Web 技术与数据库技术都发生了质的变化。由于数据库技术的支撑,Web 页面由静态网页发展成为动态网页;由于 Web 技术的支撑,数据库应用系统实现了数据环境 and 应用环境的分离,使得客户端可以用相对统一的浏览器实现跨平台的零客户端应用。比较几种网络数据库应用技术,Microsoft 公司推出的动态服务器网页(ASP, Active Server Pages)应该是一种非常优秀的技术。

本书全面介绍网络数据库应用系统设计中使用的 ASP 实用技术。其主要特点为:采用任务驱动、实例引导的方式进行讲解,使读者在学习过程中通过模仿例题、完成作业的方式,最终实现一个功能简化了的网上商店系统设计,从而掌握应用 ASP 技术进行网络数据库应用系统设计的基本技能。

全书共分 8 章,开篇介绍网络数据库基础知识与 HTML 基础,同时分析网上商店功能需求、给出数据库设计要素。然后,分 5 章分别详细介绍客户端脚本语言 VBScript、ASP 技术基础、ASP 对象、SQL 与 ADO 组件模型以及 ASP 中的 ActiveX 组件。最后,介绍网络数据库应用系统集成。全书内容丰富、结构完整,讲述过程循序渐进,深入浅出,每一章均配有相关习题,读者通过习题可以归纳所学内容,巩固所学知识。本书所用实例具有较好的通用性,读者可在学习过程中逐步完成,从而获得网络数据库应用系统开发全过程的实践。

本书由李禹生编写第 1、2、8 章,刘兵编写第 3 至 7 章,段凯负责网上商店实例设计。全书由李禹生统稿。谭立烽教授与谢兆洪教授审阅了书稿,廖明潮、欧阳峥峥参与了编写大纲的讨论,在此一并表示感谢。

本书适合作为高等院校计算机、网络工程及相关专业的教材,也可作为企事业单位相关技术人员和计算机应用人员的培训教材或参考书。

由于作者水平有限,书中难免存在疏漏与不妥之处,敬请读者与同行批评指正。

编 者

2004 年 6 月

目 录

前言

第1章 网络数据库应用系统概述	(1)
本章学习目标	(1)
1.1 网络数据库应用系统模式	(1)
1.1.1 文件/服务器模式	(1)
1.1.2 客户机/服务器模式	(2)
1.1.3 浏览器/服务器模式	(4)
1.2 Web 数据库技术	(4)
1.2.1 三层结构	(4)
1.2.2 CGI 技术	(5)
1.2.3 API 技术	(6)
1.2.4 PHP 技术	(6)
1.2.5 ASP 技术	(8)
1.3 网络数据库访问接口技术	(9)
1.3.1 JDBC 技术简介	(9)
1.3.2 ODBC 技术基础	(10)
1.3.3 ODBC 的配置方法	(12)
1.4 数据库管理系统	(16)
1.4.1 关系数据库管理系统概述	(16)
1.4.2 Access 数据库管理系统	(18)
1.4.3 MS SQL Server 数据库管理系统	(20)
1.5 网络数据库应用系统实例——网上商店	(23)
1.5.1 网上商店功能分析	(23)
1.5.2 网上商店数据库设计	(26)
1.5.3 网上商店功能模块设计	(28)
本章小结	(29)
习题	(29)
第2章 超文本标记语言	(31)
本章学习目标	(31)
2.1 HTML 概述	(31)
2.2 HTM 文档结构	(32)
2.2.1 <HTML> 标记段	(33)
2.2.2 <HEAD> 标记段	(33)

2.2.3	<BODY> 标记段	(34)
2.3	HTML 的格式标记	(35)
2.3.1	文本格式标记符	(35)
2.3.2	图像格式标记符	(39)
2.3.3	滚动对象标记符	(40)
2.4	HTML 的超文本链接标记符	(41)
2.4.1	统一资源定位器 URL	(41)
2.4.2	在文本上设定超文本链接指针	(42)
2.4.3	在图像上设定超文本链接指针	(43)
2.5	HTML 的表格标记符	(44)
2.5.1	<TABLE> 标记段	(44)
2.5.2	表格元素标记符	(45)
2.5.3	表格标题标记 <caption> ... </caption>	(47)
2.5.4	表格对象应用举例	(48)
2.6	HTML 的框架标记符	(50)
2.6.1	<FRAMESET> 标记段	(51)
2.6.2	<frame> 标记和 <noframes> ... </noframes> 标记段	(53)
2.6.3	页面框架结构应用举例	(54)
2.7	HTML 的表单标记段	(55)
2.7.1	表单标记段 <FORM> ... </FORM>	(56)
2.7.2	表单标记段中使用的标记	(57)
2.7.3	HTM 文档中表单对象应用实例	(60)
2.8	网上商店用户登录页面设计	(63)
2.8.1	登录页面的结构与功能设计	(65)
2.8.2	登录页面的基本标记应用说明	(66)
	本章小结	(67)
	习题	(67)
第3章	客户端脚本语言	(69)
	本章学习目标	(69)
3.1	脚本语言概述	(69)
3.2	脚本语言使用规则	(70)
3.2.1	脚本语言程序的嵌入规则	(70)
3.2.2	脚本语言程序的执行规则	(72)
3.3	VBScript 语法	(73)
3.3.1	注释与分行	(73)
3.3.2	常量与变量	(74)
3.3.3	运算符与表达式	(77)
3.3.4	数组	(78)
3.4	VBScript 程序设计	(79)

3.4.1	顺序结构程序设计	(79)
3.4.2	分支结构程序设计	(80)
3.4.3	循环结构程序设计	(83)
3.5	VBScript 函数与过程	(88)
3.5.1	VBScript 内部函数	(88)
3.5.2	VBScript 自定义函数及其应用	(95)
3.5.3	VBScript 过程及其应用	(96)
3.6	JavaScript 简介	(98)
3.6.1	概述	(98)
3.6.2	JavaScript 的语法知识	(100)
本章小结		(107)
习题		(108)
第4章	ASP 技术基础	(109)
本章学习目标		(109)
4.1	ASP 技术概述	(109)
4.1.1	ASP 的基本概念	(109)
4.1.2	ASP 脚本语言	(112)
4.1.3	ASP 的对象模型	(112)
4.2	VBScript Window 对象	(115)
4.2.1	VBScript Window 对象的属性	(115)
4.2.2	VBScript Window 对象的事件	(119)
4.2.3	VBScript Window 对象的方法	(120)
4.3	VBScript Document 对象	(124)
4.3.1	VBScript Document 对象的属性	(124)
4.3.2	VBScript Document 对象的事件	(128)
4.3.3	VBScript Document 对象的方法	(131)
4.4	VBScript 的 Location、History 和 Navigator 对象	(132)
4.4.1	Location 对象的属性	(132)
4.4.2	Location 对象的方法	(133)
4.4.3	History 对象的属性	(134)
4.4.4	History 对象的方法	(134)
4.4.5	Navigator 对象的属性	(135)
4.4.6	Navigator 对象的方法	(136)
4.5	cookie 属性	(136)
4.5.1	什么是 cookie	(136)
4.5.2	建立 cookie 属性	(137)
4.5.3	cookie 的限制	(139)
4.5.4	网上商店中 cookie 属性的运用	(139)
本章小结		(142)

习题	(142)
第5章 ASP 对象	(143)
本章学习目标	(143)
5.1 ASP 对象概述	(143)
5.2 Application 对象	(144)
5.2.1 Application 对象变量的定义与使用	(144)
5.2.2 Application 对象的方法	(146)
5.2.3 Application 对象的事件	(147)
5.2.4 Application 对象的集合	(148)
5.3 Session 对象	(149)
5.3.1 Session 变量的使用	(150)
5.3.2 Session 对象属性	(152)
5.3.3 Session 对象的方法	(154)
5.3.4 Session 对象的事件	(155)
5.3.5 Session 对象的集合	(157)
5.4 Request 对象	(158)
5.4.1 Request 对象的集合	(158)
5.4.2 Request 对象的应用	(168)
5.5 Response 对象	(172)
5.5.1 Response 对象属性	(173)
5.5.2 Response 对象的方法	(175)
5.5.3 Response 对象的应用	(177)
5.6 Server 对象	(179)
5.6.1 Server 对象的属性	(179)
5.6.2 Server 对象的方法	(181)
5.6.3 Server 对象的应用	(183)
本章小结	(184)
习题	(184)
第6章 SQL 与 ADO 组件模型	(185)
本章学习目标	(185)
6.1 结构化查询语言(SQL)	(185)
6.1.1 SQL 常用语句	(185)
6.1.2 SQL 基本操作符	(193)
6.1.3 SQL 常用聚合函数	(197)
6.2 ADO 组件模型	(199)
6.3 Connection 对象	(202)
6.3.1 Connection 对象属性	(202)
6.3.2 Connection 对象的方法	(204)
6.3.3 数据库连接方式	(207)

6.3.4	Connection 对象的应用	(209)
6.4	Command 对象	(210)
6.4.1	Command 对象属性	(210)
6.4.2	Command 对象的方法	(212)
6.4.3	Command 对象的应用	(213)
6.5	Recordset 对象	(215)
6.5.1	Recordset 对象属性	(215)
6.5.2	Recordset 对象的方法	(220)
6.5.3	Recordset 对象的应用	(229)
6.6	网上商店应用实例	(233)
	本章小结	(236)
	习题	(237)
第7章	ASP 中的 ActiveX 组件	(238)
	本章学习目标	(238)
7.1	什么是 ActiveX 组件	(238)
7.2	广告轮显组件	(240)
7.2.1	创建广告轮显对象实例	(240)
7.2.2	创建轮显列表文件	(240)
7.2.3	使用重定向文件	(242)
7.2.4	使用广告轮显组件的属性和方法	(243)
7.3	使用页面计数器组件	(245)
7.3.1	创建页面计数器组件对象实例	(245)
7.3.2	使用页面计数器组件的方法	(245)
7.4	使用计数器组件	(247)
7.4.1	创建计数器组件对象实例	(247)
7.4.2	使用计数器对象的方法	(247)
7.5	使用文件存取组件	(250)
7.5.1	文件存取组件概述	(250)
7.5.2	创建文本文件	(251)
7.5.3	读取文本文件	(252)
7.5.4	文件操作	(253)
7.5.5	文件夹操作	(255)
7.5.6	驱动器操作	(256)
7.6	上传组件	(257)
7.6.1	上传组件的安装	(257)
7.6.2	上传组件的使用方法	(258)
	本章小结	(264)
	习题	(264)

第8章 网络数据库应用系统集成	(265)
本章学习目标	(265)
8.1 网络系统架构	(265)
8.1.1 基于 Intranet 的网络架构	(265)
8.1.2 基于 Internet 的网络架构	(266)
8.2 服务器功能配置	(267)
8.2.1 Web 服务器功能配置	(267)
8.2.2 数据库服务器功能配置	(272)
8.3 网络数据库系统安全	(275)
8.3.1 访问控制	(276)
8.3.2 数据库安全控制	(277)
8.3.3 数据库的安全策略	(279)
8.4 网上商店网站测试与维护	(280)
8.4.1 网站界面测试	(280)
8.4.2 网站性能测试	(281)
8.4.3 网站的维护	(282)
本章小结	(285)
习题	(285)
参考文献	(287)

第1章 网络数据库应用系统概述

本章学习目标

本章介绍网络数据库应用系统的相关基础知识,包括网络数据库应用系统模式,Web数据库技术,数据库访问接口和数据库管理系统概念;介绍 Access DBMS 和 MS SQL Server DBMS 基础知识及其设计数据库与数据表的方法;介绍本书实例“网上商店”的功能与数据库设计要点,为后续章节的学习奠定基础。通过本章的学习,读者应掌握以下主要内容:

- 理解 F/S、C/S、B/S 三种网络数据库应用系统模式的应用特征。
- 理解 CGI、API 与 PHP 的技术要点,掌握 ASP 技术特征与工作原理。
- 掌握 ODBC 技术以及配置方法。
- 理解 DBMS 概念,掌握 Access 和 MS SQL Server 数据库设计方法。
- 理解网上商店实例功能与数据库设计参数。

1.1 网络数据库应用系统模式

自 20 世纪 90 年代以来,Internet 被应用于各个社会领域,成为当今应用范围最广、影响最深刻的一项科学技术。其中的 Web 技术采用浏览器和超文本链接,得到了众多 Internet 使用者的青睐。同时,Web 技术也得到了众多计算机应用系统开发者的认真研究,形成了一系列基于 Web 技术的计算机应用系统。在数据库应用领域,Web 技术为网络数据库应用系统提供了一种全新的应用模式。

Web 不是传统意义上的物理网络,而是在超文本基础上形成的信息网,是 Internet 的重要组成部分。将 Web 技术与数据库技术相结合,使得 Web 技术与数据库技术都发生了质的变化。由于数据库技术的支撑,Web 页面由静态网页发展成为动态网页;由于 Web 技术的支撑,数据库应用系统实现了数据环境和应用环境的分离,使得客户端可以用相对统一的浏览器实现跨平台的零客户端应用。

纵观网络数据库应用系统模式的发展过程,可以看到,基于 Web 技术的网络数据库应用系统具有良好的技术优势,因而必将形成强大的生命力。

1.1.1 文件/服务器模式

建立计算机网络的主要目的是实现资源共享和计算机之间的通信。资源共享包括硬件资源共享、软件资源共享、数据资源共享和通信信道资源共享。其中,如何有效地实现数据

资源共享是数据库应用系统的重要功能之一。

数据资源共享的方式随着网络结构的不同而不同。文件/服务器模式(F/S 模式, File Server Model)是一种基于局域网络结构的网络数据库应用系统模式,其基本构架为:系统中所有的数据资源和操作资源均集中于服务器端,工作站上的所有应用均通过向服务器提出申请、获得相应资源而得到满足。

根据不同的网络结构,工作站与服务器的分配、数据资源与操作资源的分配与管理可以采用不同的模型。

1. 文件服务器模型

以 Novell Netware 局域网操作系统为代表的是文件服务器模型的典型。它以一台或几台服务器作为共享资源的主体,在文件服务器中提供高速存取的大容量磁盘,用于存放网络中各个客户端共享的文件和目录,包括各种应用程序和数据库。工作站根据所拥有的权限使用程序、访问数据库。作为客户端的工作站上不安装应用程序,属于一种零客户端方案。

文件服务器模型充分实现了网络资源共享、数据资源统一管理与调配,是一种局域网络环境下的优秀网络数据库应用系统模式。

但是,采用这种方式构成的数据库应用系统,网络通信量特别大,而且由于局域网多数采取多路复用、载波侦听的通信方式,所以效率比较低。

2. 工作组模型

以 Microsoft 公司的 Windows for Workgroup、Windows 95/98、Windows NT 和 IBM 公司的 OS/2 LAN Server 为代表,构成了工作组模型的典型例子。

在工作组模型中,某个拥有特权的系统管理员创建一个用户工作组,并赋予组中的成员以对等(Peer to Peer)方式工作。工作组中的每一台计算机既可作为服务器,又可作为工作站。每台计算机具有自己的账户,并管理着自己所属的共享资源。

这种网络的管理比较松散,安全性比较差,数据资源的冗余控制也比较难以实现。

3. 域模型

Microsoft 公司的 Windows NT Server 和 IBM 公司的 OS/2 LAN Server 除了提供工作组模型外,还提供了域模型的网络应用方式。

在这种应用方式中,将一个网络中的某些计算机连接成为一个域,每个域中的工作站再分成为若干个工作组。一个域中可以有若干个服务器和工作站,其中至少有一台服务器为域服务器(域控制器)负责域的集中管理。系统管理员创建并管理用户账户及数据库。

利用域模型可以实现对网络的集中管理。域模型是一种安全、高效的网络使用模型,能够构成性能良好的、基于局域网络的数据库应用系统。

1.1.2 客户机/服务器模式

文件/服务器模式的基本思想是将所有资源集中于服务器端,包括数据资源与操作资源,工作站不拥有任何资源。工作站运行所需要的程序、数据等,均由服务器提供。因此,网络通信过于拥挤、服务器负担过重,成为文件/服务器模式的缺憾。

使用网络技术,采用客户机/服务器模式(C/S 模式, Client/Server Model)的数据库应用系统,不仅可以实现对数据库资源的共享,而且可以提高数据库的安全性。

传统客户机/服务器模式的数据库应用系统是两层的,其基本思想是:服务器提供数据的存储和管理等功能;客户端运行相应的应用程序。通过网络获得服务器的服务,使用服务器上的数据库资源。客户端和服务器通过网络连接成一个互相协调的系统。

客户机/服务器模式提出了一种新的资源共享方式。即,将不同的应用程序安装在不同的工作站上,形成客户端。客户端运行本地程序,访问存储在服务器端的数据库,获取数据资源,完成相关处理后回写至数据库中。这就形成了客户机/服务器模式的基本运行机理。

根据客户机/服务器构架方式,可以将其分为二层 C/S 模式和三层 C/S 模式。

1. 二层 C/S 模式

二层 C/S 模式把数据处理任务分配给客户机(Client)和数据库服务器(Data Base Server)共同承担。数据库服务器安装数据库管理系统(DBMS, Data Base Management System),承担数据库数据管理、响应客户机请求并根据请求完成数据操作;客户机安装应用程序,承担客户程序运行、数据的处理和输入/输出操作。

二层 C/S 模式结构如图 1-1 所示。

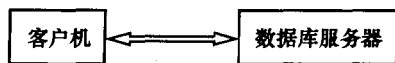


图 1-1 二层 C/S 模式结构

C/S 模式有如下优点:

- 充分发挥了客户机和服务器两方面的处理能力。
- 减少网络信息流量。
- 服务器可高效、安全地处理数据库,客户机可处理 GUI 界面及本地 I/O。
- C/S 提供了开放式分布计算环境。

2. 三层 C/S 模式

在客户端数量过大、请求过于频繁时,数据库服务器负担非常沉重。因此提出在客户机与数据库服务器之间增加功能服务器的方案。即,将客户端请求的接受、响应与分析赋予功能服务器完成,由功能服务器将客户端请求转换为 SQL 请求交数据库服务器处理,再由功能服务器将数据处理结果发还客户端。

如此,即构成三层 C/S 模式,如图 1-2 所示。在三层 C/S 模式结构中,客户机驻留用户界面层(也称为表示层)软件负责用户与应用层之间的对话任务。功能服务器存放业务逻辑层(也称为功能层)软件,响应客户机请求,完成业务处理或复杂计算。在出现数据库访问任务时,根据客户机的要求向数据库服务器发出 SQL 指令。数据库服务器存放数据库服务层(也称为数据层)软件,用来执行功能层送来的 SQL 指令,实现对数据库的读、写、删、改及查询等操作,操作完成后通过功能服务器向客户机返回操作结果。

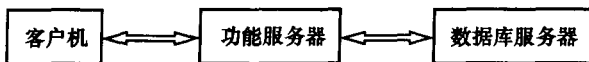


图 1-2 三层 C/S 模式结构

由于客户机上必须驻留表示层软件,不再是零客户端方案。当功能服务器仅配置很弱的功能软件时,三层 C/S 模式结构就会还原为二层 C/S 模式结构。此时,由于客户机上驻留的表示层软件任务繁重会形成一个庞大的应用软件,被称为胖客户结构。

如果为功能服务器配置非常强大的功能,则客户机上驻留的表示层软件任务轻松,会成为一份小巧的应用软件,被称为瘦客户结构。瘦客户结构与胖客户结构孰优孰劣,不可一概

而论,应该根据技术的发展、应用系统的具体环境进行具体分析。

1.1.3 浏览器/服务器模式

浏览器/服务器模式(B/S 模式, Browser/Server Model)是 Web 技术和数据库技术相结合形成的一种技术,采用这种技术,可以实现数据库应用系统开发环境和应用环境的分离。

基于 B/S 模式的数据库应用系统通常采用三层结构:“浏览器—Web 服务器—数据库服务器”,如图 1-3 所示。

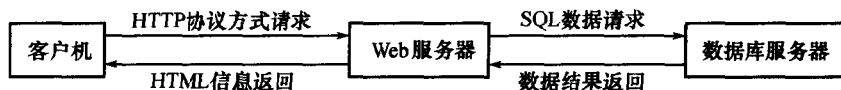


图 1-3 B/S 模式结构

在 B/S 模式结构中,客户机驻留的用户界面层软件为浏览器,用户通过浏览器的表单(Form)等形式向 Web 服务器发送 HTTP 协议方式的请求。

功能服务器配置成为 Web 服务器,响应客户机发出的 HTTP 协议方式请求。当客户机提出访问数据库的请求时,Web 服务器根据实际要求向数据库服务器发出 SQL 数据请求。

数据库服务器依然存放数据库服务层软件,执行 Web 服务器送来的 SQL 数据请求,进行对数据库的读、写、删、改及查询等操作,操作完成后,将数据结果返回给 Web 服务器。

Web 服务器在收到数据结果返回后,将其转换为 HTML 或各类 Script 格式,形成 HTML 信息返回给浏览器,并结束一次操作事务。

可以看到,客户机上只需安装浏览器,不需安装数据库应用系统程序,因此 B/S 模式结构称为零客户端结构。

另外,由于 B/S 模式结构中的客户机与 Web 服务器之间采用 HTTP 协议进行信息传送,因此适用于广域网结构的 Internet 或 Intranet,具有良好的发展前景。

1.2 Web 数据库技术

随着 Internet/Intranet 技术的普及和发展,客户机/服务器结构的数据库应用系统正由二层向三层或多层发展。如果客户端仅需浏览器软件,中间层提供 Web 服务,后台提供数据库服务,即形成了 Web 数据库应用系统结构。

1.2.1 三层结构

实质上,Web 数据库应用系统采用三层的客户机/服务器结构:第一层为客户层,通常安装浏览器软件;第二层为中间层,需要配置相应的 Web 服务器以及相关技术支撑软件;第三层为数据库层,必须配备相应的数据库服务器和数据库管理系统。

浏览器作为用户输入查询条件和显示查询结果的交互界面,用户可以通过填写表单或输入关键字的方式来与 Web 服务器交互。当用户单击表单上的“提交”按钮时,表单中的数据便被发送到 Web 服务器。

Web 服务器负责客户端发送信息的接收,它将数据传送至要被处理的脚本或应用程序,并在数据库中查询数据或将数据传送到数据库中。最后,Web 服务器将数据库服务器的返回结果插入到 HTML 页面,传送至客户端以响应用户。

从技术发展角度看,最经典的数据库访问技术是 CGI(Common Gateway Interface),随后出现了服务器 API(Application Programming Interface)技术,近来流行的是 ASP(Active Server Pages)技术和 PHP(PHP:Hypertext Preprocessor)技术。

1.2.2 CGI 技术

CGI 是 Web 服务器与外部扩展程序交互的一个标准接口。按 CGI 标准编写的外部扩展程序可以处理客户端(一般是 Web 浏览器)输入的协同工作数据,完成客户端与服务器的交互操作。服务器并不关心外部扩展程序用什么语言(如 C、C++、Perl、Delphi 或 Visual Basic 等)编写,它只负责接收用户的输入信息,并把 CGI 程序所产生的结果(HTML 文档或其他符合 HTTP 规定的文件)回传给用户。

CGI 在实际应用中相当有用,可以通过编写 CGI 程序来访问数据库,客户端用户可通过它和 Web 服务器进行数据查询。应用 CGI 技术访问数据库的步骤大致如下:

- 1) 客户端通过浏览器向 Web 服务器发出 HTTP 请求。
- 2) Web 服务器接收客户对 CGI 的请求,设置环境变量或命令行参数,然后创建一个子进程来启动 CGI 程序,把客户的请求传给 CGI 程序。
- 3) CGI 程序向数据库服务器发出请求,数据库服务器执行相应的查询操作。
- 4) 数据库服务器将查询结果返回给 CGI 程序。
- 5) CGI 程序将查询结果转换成 HTML 格式并返回给 Web 服务器。
- 6) Web 服务器将格式化的结果送客户端浏览器显示。

按应用环境的不同,CGI 可分为标准 CGI 和 WinCGI 两种。标准 CGI 通过环境变量或命令行参数来传递 Web 服务器获得的用户请求信息,Web 服务器与浏览器间的通信采用标准输入/输出方式。当 Web 服务器接收到浏览器发来的 HTTP 请求时,首先对该请求进行分析,并设置所有环境变量或命令行参数,然后创建一个子进程启动 CGI 程序。CGI 程序执行完后,利用标准输出将执行结果返回 Web 服务器。CGI 程序的输出类型可以是 HTML 文档、图形、图像、纯文本或声音等。

标准 CGI 之所以采用标准输入/输出方式进行数据通信,是由其最初的开发环境 UNIX 操作系统所决定的。但许多 Windows 环境下的编程工具(如 Delphi 和 Visual Basic 等)并不支持标准输入/输出方式,因此无法用这些工具来开发基于标准 CGI 的应用程序,于是有些 Web 服务器引入了 WinCGI。WinCGI 也称间接 CGI 或缓冲 CGI,它最主要的特点是:Web 服务器与 CGI 程序间的数据交换通过缓冲区进行,而不是通过标准输入/输出进行。当 Web 服务器接收到浏览器的 HTTP 请求时,先创建一个子进程启动缓冲程序,该缓冲子进程与 Web 服务器通信,它通过标准输入/输出、环境变量和命令行参数来获得有关的数据,并将这些数据保存在一个输入缓冲区中;缓冲子进程再创建一个子进程启动 CGI 程序,CGI 程序读取输入缓冲区中的内容,处理浏览器的请求,并将要输出的内容存入输出缓冲区;缓冲程序通过环境变量或命令行参数等方式传递输入缓冲区和输出缓冲区的地址(或临时文件名)到 CGI 子进程。

在整个处理过程中,缓冲子进程与 CGI 子进程之间应保持同步,以监测 CGI 程序执行的状态。当缓冲子进程得到 CGI 子进程的输出时,设置有关环境变量并终止该 CGI 子进程,然后采用标准输出与 Web 服务器通信,并通过 Web 服务器将 CGI 程序的输出结果返回给浏览器。Web 服务器进程与缓冲进程必须保持同步,以监测缓冲子进程执行的状态。

CGI 的优点是跨平台性能极佳,几乎可以在任何操作系统(如 DOS、Windows、UNIX 和 OS/2 等)上实现。

CGI 的缺点也显而易见,CGI 程序一般都是一个独立的可执行程序,与 Web 服务器各自占据着不同的进程;且一个 CGI 程序只能处理一个请求,对每个请求,CGI 都会产生一个新的进程,同一时刻发出的请求越多,服务器产生的进程就越多,耗费掉的系统资源也越多。这样,在用户访问的高峰期,网站就会表现出响应时间延长、处理缓慢等情况,严重的甚至会导致整个网站崩溃。另外,CGI 的功能有限,开发较为复杂,且不具备事务处理功能,这在一定程度上限制了 CGI 的应用。

1.2.3 API 技术

API 以动态链接库(DLL)的形式提供,是驻留在 Web 服务器上的本机代码,作用类似于 CGI,可起到扩展 Web 服务器功能的作用。目前流行的服务器 API 有 Microsoft 的 ISAPI(Internet Server API)、Netscape 的 NSAPI(Netscape Server API)和 O'Reilly 的 WSAPI(WebSite API)。各种服务器 API 均与相应的 Web 服务器紧密联系在一起,程序员可利用服务器 API 来开发 Web 服务器与数据库服务器的接口程序。

服务器 API 可实现 CGI 程序所能提供的全部功能,其工作原理和 CGI 大体相同,都是通过交互式页面取得用户的输入信息,然后交服务器后台处理,但各自在实现机制上却大相径庭。

服务器 API 与 CGI 最大的区别在于:组成服务器 API 的程序均以动态链接库的形式存在,而 CGI 程序一般都是可执行程序。在服务器 API 调用方式中,被用户请求激活的 DLL 和 Web 服务处于同一进程中,在处理完某个用户请求后并不会马上消失,而是和 Web 服务器一起继续驻留在内存中,等待处理其他用户的 HTTP 请求,直到超过设定时间后一直没有用户请求为止。

基于服务器 API 的所有进程均可获得服务器上的任何资源,而且当它调用外部 CGI 程序时,需要的开销也较单纯的 CGI 少,因此服务器 API 的运行效率明显高于 CGI。

服务器 API 的出现解决了 CGI 的低效问题,但用 API 编程比开发 CGI 程序更加困难。API 开发需要多线程、进程同步、直接协议编程及错误处理之类的专门技术。为了解决复杂与高效之间的矛盾,Netscape 与 Microsoft 均为各自的 Web 服务器提供了基于 API 的高级编程接口。Netscape 提供的是 LiveWire,而 Microsoft 提供的是 IDC(Internet Database Connector)。

1.2.4 PHP 技术

PHP 是 Hypertext Preprocessor(超文本预处理器)的英文缩写。PHP 是一种脚本编程语言,通常嵌入或结合 HTML 使用。对于开发三层结构的 Web 数据库应用系统,PHP 是开发中间层中的应用逻辑的理想工具。

PHP 是一种 HTML 内嵌式的语言 (类似 IIS 上的 ASP)。而 PHP 独特的语法混合了 C、Java、Perl 以及 PHP 式的新语法。它可以比 CGI 或者 API 更快速地执行动态网页。典型的 PHP 嵌入式工作机理如图 1-4 所示。

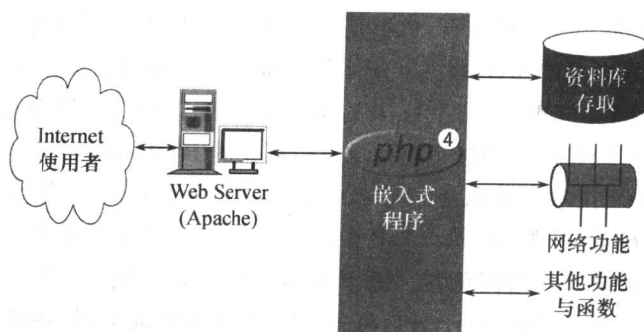


图 1-4 典型的 PHP 嵌入式工作机理

PHP 已经作为许多大中型规模的 Web 数据库应用程序的组件出现。而且,由于它所具有的出色性能,PHP 正在越来越多地受到众多 Web 数据库应用系统开发者的青睐。

PHP 的当前主流版本为 PHP4,下面以 Zend Technoligise 公司发布的 PHP4 脚本编程环境体系结构(如图 1-5 所示,图像来自 Zend Technoligise 公司)为例,说明 PHP 工作机理。

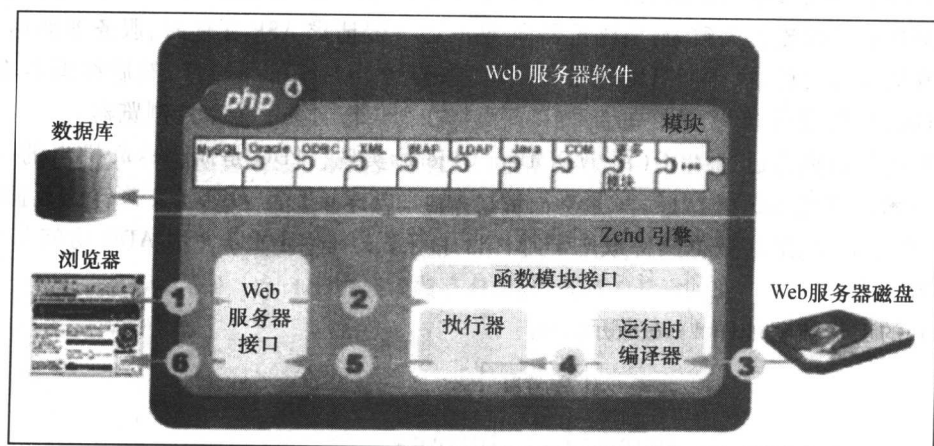


图 1-5 PHP4 脚本编程环境体系结构

- (1) Internet 使用者在客户端采用 HTTP 通过浏览器向 Web 服务器发出 PHP 脚本请求。
- (2) Web 服务器将请求传递到 Zend 引擎的 Web 服务器接口。
- (3) Web 服务器接口调用 Zend 引擎并传递参数给引擎。
- (4) PHP 脚本由引擎从磁盘中获取。
- (5) 脚本由运行时编译器编译。
- (6) 已编译的代码由引擎的执行器运行,实现对后台数据库的访问,执行器的输出返回到 Web 服务器接口。
- (7) Web 服务器接口返回输出给 Web 服务器(Web 服务器又将其输出作为一个 HTTP 响应返回给位于客户端的 Internet 使用者)。