

浏览器/服务器应用开发

浏览器 / 服务器 应用开发

林海
编著

科



社

科学出版社

1173172
L58

461561

浏览器/服务器应用开发

林 海 编著

科 学 出 版 社

2000

JS198/09
内 容 简 介

本书介绍了从客户机/服务器(Client/Server)模式派生出的浏览器/服务器(Browser/Server)应用模式及工作原理、结构和主要的应用要素,着重介绍了基于ASP(Active Server Pages)技术的应用开发解决方案。为了使读者有一个完整的概念,书中对HTML语言、HTTP协议、脚本语言、控件等做了必要的阐述。全书共分5章。包括浏览器/服务器应用框架、应用开发环境、浏览器端和服务器主要应用要素,以及基于ASP技术的B/S应用开发实例。

本书的特点是内容全面,力求实用,从技术概念到具体解决方案逐渐展开,适合不同层次的读者自学。

本书可供高等学校计算机、电子工程、信息工程等专业师生参考,对从事计算机网络及相关系统工作的广大科技工作者和开发应用人员具有重要参考价值。

图书在版编目(CIP)数据

浏览器/服务器应用开发/林海编著,-北京:科学出版社,2000

ISBN 7-03-008131-5

I. 浏… II. 林… III. ①浏览器-软件开发②服务器-软件开发
IV. IP393

中国版本图书馆CIP数据核字(1999)第71311号

科学出版社 出版

北京东黄城根北街16号
邮政编码:100717

新蕾印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

2000年3月第一版 开本:787×1092 1/16

2000年3月第一次印刷 印张:17 1/2

印数:1—5 000 字数:400 000

定价:27.00元

(如有印装质量问题,我社负责调换(环伟))

前 言

传统的数据库应用系统采用客户机/服务器(Client/Server)模式,既有高度的交互性、高效的用户界面等优点,但也存在数据交互和交互规则移至客户端、代价高、维护成本高、缺少中央控制、多媒体要素不易扩展等缺点。

基于瘦客户机的浏览器/服务器(Browser/Server)模式克服了以上缺点,由于 Web 服务器将浏览器端的界面布局,人机交互形式以及界面间逻辑关系仅在服务器端定义,从而保证了界面的完整清晰,易维护,并使前台客户机的负载减到最小。HTTP1.1 较 HTTP1.0 的 Web 传输协议,其查询请求与查询结果已简化,减少了不必要的网络连接和数据传输(HTTP1.0 每一请求需要一次 TCP 连接,而 HTTP1.1 允许服务器和浏览器一次连接传送所有数据)。采用超文本标记语言 HTML4.0 标准,支持更多的多媒体元素,具有很强的多媒体扩展功能,前台界面更具吸引力。ActiveX 充分利用了操作系统所提供的资源,是模块重用、高效率运行的较好的解决方案。

实现 Web 数据库的应用一般有两种方法:一种是 Web 服务器端提供中间件连接 Web 服务器与数据库服务器;另一种是把应用程序下载到客户端直接访问数据库。最常用的中间件技术有通用网关接口(CGI)和应用程序编程接口(API)两种。CGI 程序有效率低、速度慢等缺点。API 很大程度上克服了 CGI 的缺点,但其兼容性差和开发难度大也许多开发人员望而却步。客户端访问数据库主要包括 Java Script 和数据库连接器(IDC)等。用 Java Script 和 IDC 开发 Web 数据库虽简单但同时也存在功能有限、不能完全控制其过程等缺点,所以它们也无法胜任较复杂的 Web 数据库的开发。微软推出的 ASP(Active Server Pages)技术就是一种开发相对容易、功能强大且高效的 Web 数据库开发方法。ASP 实际上是一种在服务器端开发脚本语言的环境。利用它可以开发出动态、交互、高性能 Web 服务器端的应用,且能较容易实现复杂的 Web 应用。利用 ASP 开发的脚本程序全部运行在服务器端,极大地减少了对客户端的要求,而且又有效地保护了开发人员的知识产权不受别人侵犯。

本书较全面地讲述了与 ASP 有关的开发工具、ActiveX、ADO 等技术背景,在理论上对浏览器/服务器模式的优缺点和技术发展趋势有一定的介绍。全书共分五章,内容如下:

第一章全面概述了客户机/服务器的发展历程及其衍生出的浏览器/服务器应用模式,阐述了浏览器/服务器的应用的基础,即 HTML 语言和构筑于 TCP/IP 协议之上的 HTTP 协议,简要概述了浏览器和服务器的基本工作原理,对扩展浏览器和服务器功能的脚本语言、ActiveX 控件和组件也进行了简要概述。

第二章简要说明了基于 ASP 技术的浏览器/服务器应用开发环境,介绍了几种浏览器和服务器软件和 Web 数据库应用环境,对浏览器端的商品化控件和服务器的扩展应用软件也有所介绍。

第三章全面阐述了利用浏览器端计算资源的两大应用要素:脚本语言(VBScript 和 JavaScript)和控件。在对象层次上将浏览器端的资源描述为“浏览器对象模型”,使开发者

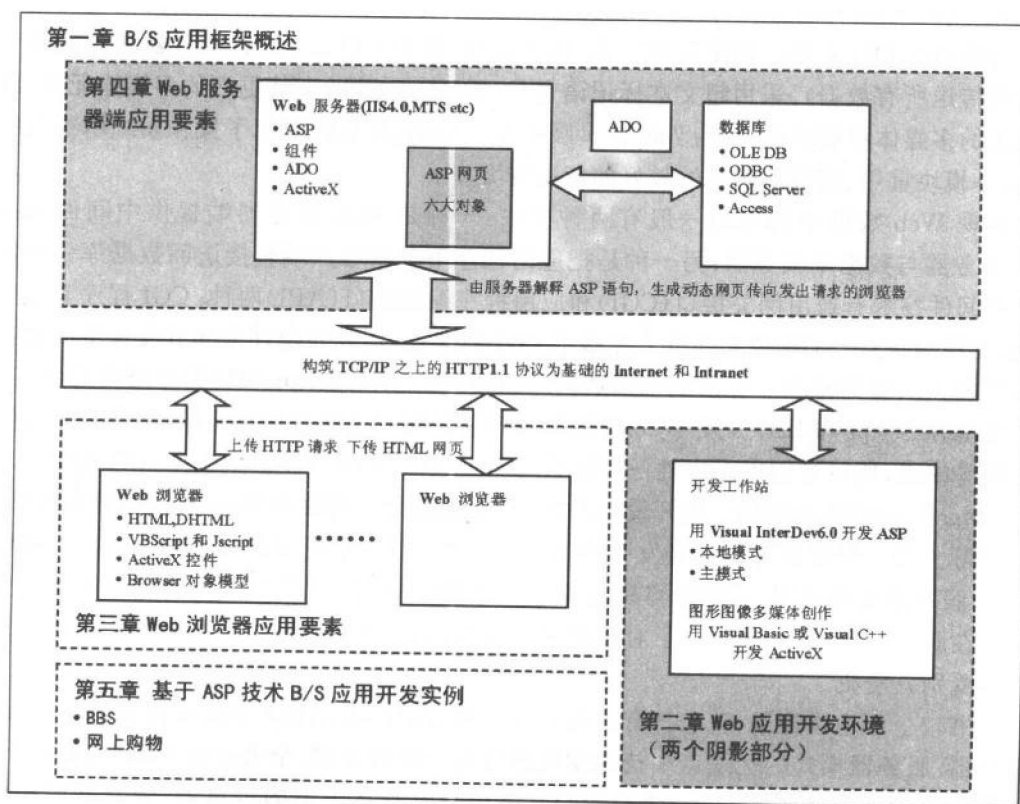
非常方便地利用浏览器和操作系统所提供的资源;ActiveX 控件使在 Windows 环境下代码和模块重用成为现实,本章讲述了 ActiveX 控件的使用和开发。

第四章详细阐述了 ASP 的基本概念和 ASP 内置的六大对象、服务器端的十种常用组件,全面介绍了与数据库相连的 ADO 组件。本章对组件的开发和使用,组件和事件服务器的相互关系都有所阐述。

第五章的两个应用实例讲述了基于 ASP 技术的 Web 应用开发中所遇到的典型问题,也是对读者掌握 ASP 技术的一个检阅。

附录部分对 HTML,SQL,VBScript 和 JavaScript 进行了系统的归纳和总结,使读者在具体开发当中可以方便快捷地进行查阅。

本书所讲述的 B/S 应用框架及对应的本书的章节内容可参阅下图:



参加本书编写工作的还有元燕辉、濮澍、白云、金锋,金叶英参与了本书的文稿打印工作。

限于作者水平,本书中出现的错误和疏漏,恳请读者批评指正。

作者

1999 年 10 月

目 录

第一章 浏览器/服务器应用框架	1
1.1 Web 应用框架——浏览器/服务器	1
1.1.1 终端/主机(Terminal/Mainframe)结构	1
1.1.2 客户机/服务器(Client/Server)结构	2
1.1.3 浏览器/服务器(Browser/Server)结构	4
1.2 Web 服务器	5
1.2.1 Web 服务体系结构	5
1.2.2 几种 Web 服务器软件系统	6
1.3 Web 浏览器	8
1.3.1 Web 浏览器技术	8
1.3.2 其它常见的浏览器	11
1.4 Web 基础协议	12
1.4.1 TCP/IP 协议	12
1.4.2 HTTP 协议	15
1.5 Web 动态网页	19
1.5.1 超文本标记语言 HTML	20
1.5.2 动态 HTML 原理	28
1.5.3 脚本语言	30
1.5.4 ActiveX 概述	32
1.5.5 Java 和 JavaApplet	34
1.5.6 从 CGI 到 ASP	35
第二章 浏览器/服务器应用开发环境	38
2.1 选择解决方案	38
2.2 Web 浏览器	38
2.3 Web 服务器	38
2.3.1 Internet 信息服务器	38
2.3.2 个人 Web 服务器	42
2.4 后台数据库	43
2.4.1 Microsoft SQL Server	43
2.4.2 Microsoft Access	43
2.4.3 使用 ODBC 连接数据库	44
2.5 安装和使用开发工具 Visual InterDev 6.0	46
2.5.1 Visual InterDev6.0 简介	46
2.5.2 Visual InterDev6.0 的主要特征	48
2.6 其它软件介绍	53
2.6.1 多媒体播放控件 MS Media Player6.0	53

2.6.2	VRML 浏览控件 MS VRML2.0 Browser Control	53
2.6.3	Microsoft Index Server 2.0	54
2.6.4	事务服务器 MTS	54
2.6.5	Microsoft Site Server	54
第三章 Web 浏览器端应用要素		56
3.1	脚本语言	56
3.1.1	脚本语言和编程语言的比较	56
3.1.2	浏览器和服务器脚本之间的比较	57
3.1.3	VBScript	58
3.1.4	JavaScript	62
3.2	浏览器对象模型	65
3.2.1	对象模型	65
3.2.2	Document 对象的方法、属性和集合	71
3.2.3	Document 对象的事件	73
3.2.4	Window 对象事件	75
3.2.5	对事件进行脚本编辑	76
3.3	ActiveX 控件的使用和开发	77
3.3.1	用 ActiveX 控件扩展浏览器的功能	78
3.3.2	用 VBScript 控制 ActiveX 控件	84
3.3.3	使用 Visual C++ 开发控件	87
第四章 Web 服务器端应用要素		97
4.1	动态服务器网页 ASP	97
4.1.1	ASP 的基本概念	97
4.1.2	如何编写 ASP	102
4.2	ASP 内置对象	116
4.2.1	Application 对象	118
4.2.2	Session 对象	120
4.2.3	Request 对象	125
4.2.4	Response 对象	128
4.2.5	Server 对象	130
4.2.6	ObjectContext 对象	131
4.3	服务器端组件	131
4.3.1	广告轮显 AdRotator	132
4.3.2	数据库访问 ADO	133
4.3.3	浏览器兼容类型 BrowserType	134
4.3.4	内容循环 ContentRotator	134
4.3.5	计数器 Counter	135
4.3.6	文件系统对象 FileSystemObject	136
4.3.7	个人信息 MyInfo	136
4.3.8	网页超链 NextLink	137
4.3.9	网页计数器 PageCounter	138
4.3.10	口令认证 PermissionChecker	138

4.4 通过 ADO 与数据库相连	139
4.4.1 ADO 应用模型	139
4.4.2 ADO 功能	141
4.4.3 ADO 与数据库的连接	141
4.4.4 ADO 中的七个对象	144
4.4.5 ADO 的性能考虑	157
4.5 开发用户定制的 ActiveX 组件	159
4.5.1 进程内和进程外 ActiveX 组件	159
4.5.2 ActiveX 组件的创建、注册	160
4.5.3 在 ASP 页面调用自制 ActiveX 组件	163
4.5.4 在服务器端使用组件	164
4.6 组件与事务服务器	165
4.6.1 为什么需要 MTS	165
4.6.2 使用 MTS 管理组件和组件关联	167
第五章 浏览器/服务器应用开发实例	169
5.1 创建 BBS 系统	169
5.1.1 建立数据库与 ODBC 连接	170
5.1.2 用户登录的处理	171
5.1.3 在 ASP 中灵活的运用 VBScript 脚本	176
5.1.4 如何在 BBS 系统中实现翻页功能	179
5.1.5 用 JavaScript 帮助实现按主题搜索	181
5.1.6 在文章中嵌入多媒体信息	182
5.1.7 把一般的文本格式转化为 HTML 语言	185
5.2 创建一个电子商务系统	186
5.2.1 电子商务系统流程	186
5.2.2 系统初始化	187
5.2.3 用 ASP 实现商品查询与浏览	188
5.2.4 购物车的实现	191
5.2.5 网上付款	195
5.2.6 用安全套接字来(SSL)保护您的信息	195
5.2.7 使用 AdRotator 实现随机的商品广告	196
附录 A HTML 速查手册	198
A.1 HTML 语法	198
A.2 HTML 标记和属性	199
附录 B 常用 SQL 命令	219
B.1 SQL 命令的种类	219
B.1.1 定义数据库结构(DDL)	219
B.1.2 操纵数据(DML)	219
B.1.3 选择数据(DQL)	219
B.1.4 数据控制语言(DCL)	220
B.1.5 数据管理命令	220
B.1.6 事务性控制命令	220

B.2	SQL 语句	220
B.3	SQL 子句	222
附录 C	VBScript 参考	224
C.1	VBScript 数据类型	224
C.2	VBScript 变量	225
C.3	VBScript 流程控制语句	226
C.4	VBScript 过程	232
附录 D	JavaScript 参考	234
D.1	JavaScript 语句	234
D.2	JavaScript 操作符	236
D.3	JavaScript 保留字	237
D.4	JavaScript 对象	238
D.5	JavaScript 方法	249
D.6	JavaScript 事件处理器	260
D.7	JavaScript 属性	263

第一章 浏览器/服务器应用框架

1.1 Web 应用框架——浏览器/服务器

当计算机的应用开始从数值计算向数据库方面发展的时候,许多公司和机构都致力于建立一种完美的数据库运行的体系结构。最早数据库应用是在一台主机上进行的。它的基本思想是在计算机上建立一个或多个应用程序,然后对本机上的文件系统进行操作。这种思想随着网络的出现,就显得力不从心了。网络的出现给计算机带来了一次重大的改革,为了适应网络环境下的数据库应用,人们依据计算机的应用特点和应用功能的分划方式,提出了一系列应用框架。本书主要讨论的是,在 Web(World Wide Web)方式下,浏览器/服务器的应用框架及相关的技术解决方案。

1.1.1 终端/主机(Terminal/Mainframe)结构

这种体系结构使得用户能够尽可能地使用共享资源,这里不仅包括了数据的共享,也包括了应用的共享。终端所要做的事只是输入命令(通常是以命令行的形式),显示结果,而有关的一切数据处理都交给了主机去完成。在这样的结构中,由于终端不需要做什么处理工作,所以它可以很简单,也不必随着信息业的发展频繁更新,因此也称“哑”终端;但是主机就不一样了,它几乎包揽了有关数据的一切处理工作,这就要求它具有十分高的性能,并且能够随着信息量(包括数据库的加大和终端的增加)的增加不断更新(见图 1-1)。

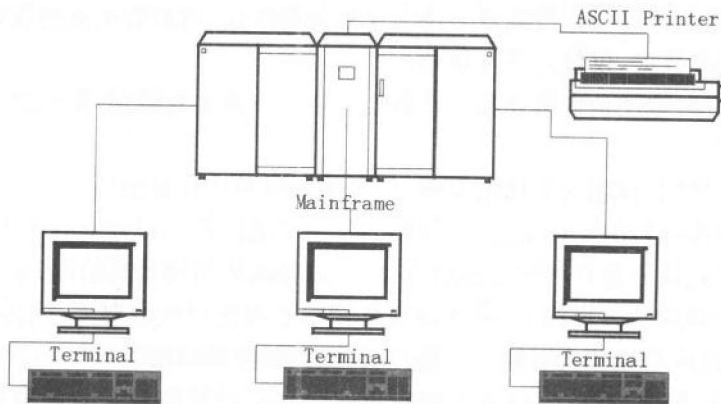


图 1-1 终端/主机模式:服务器存储所有数据,进行所有处理运算

【提示】

优点:资源得到充分的共享

缺点:用户界面不友好

主机负载过大

1.1.2 客户机/服务器(Client/Server)结构

传统的客户机/服务器结构划分为四层。从客户机到服务器分别为:数据捕获(Data Capture)、业务规则(Business Rules)、应用接口(Application Interface)、应用服务器(Application Server),如图 1-2 所示。

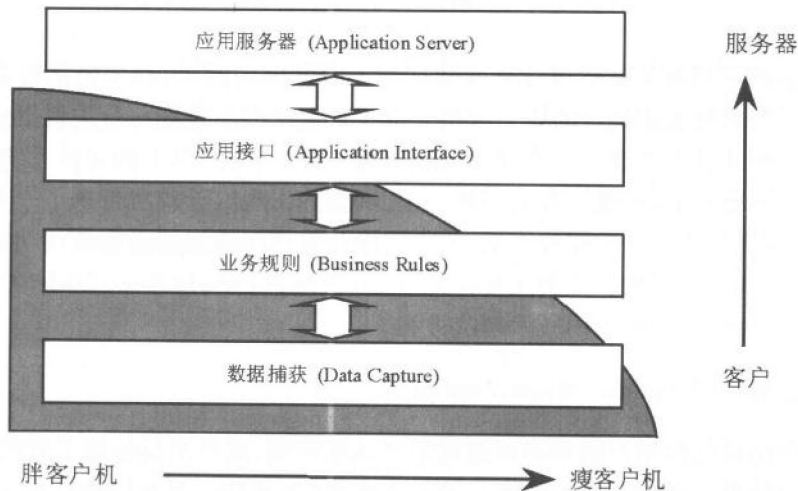


图 1-2 传统的客户/服务器四层结构

1)数据捕获层:人机交互的捕获和转换,包括键盘、鼠标、虚拟现实接口设备和未来的语音、手势和神经传导,以及输出设备如显示器、打印机、磁带机等。在该层的编程操作则是将数据以预定的格式展现出来。

2)业务规则层:该层负责将数据转换为业务关联数据,增加有关业务规则的信息。业务规则层并不处理数据,仅仅是涉及规则。

3)应用接口层:该层负责将业务关联数据转换为技术关联数据,如 SQL 语句,以供应用服务器层使用。

4)应用服务器层:该层负责处理数据,如存储数据和计算出结果。

这种体系结构最初的出现是为了缓解终端/主机结构中主机繁重的工作负担,通过在客户和服务器之间划分各自所包含的层次,则可以提高应用的计算效率。于是将主机要做的一部分工作转移到了客户机上,显然这也就增加了对客户机的要求,但是随着 PC 业的发展,这一点已经构不成什么问题了。客户机/服务器体系结构给整个的企业计算带来了一个全新的标准,然而,与此同时它的出现也使得企业计算的开发者面临许多新的挑战(见图 1-3)。

为了应用客户机/服务器这种体系结构,原来完全放在主机上的应用程序被分成了两部分:客户机端的应用程序和服务器端的应用程序,即所谓的两层结构。

运行在客户机端的应用程序包括了发给用户的信息的表达方式和接受用户输入的机制。另外它也包括了一些在客户机上的数据交互,例如随着用户选择的不同而不同的图形界面。在现在的应用中,通常要对送往服务器端的数据进行有效性检查,也就是说在客户

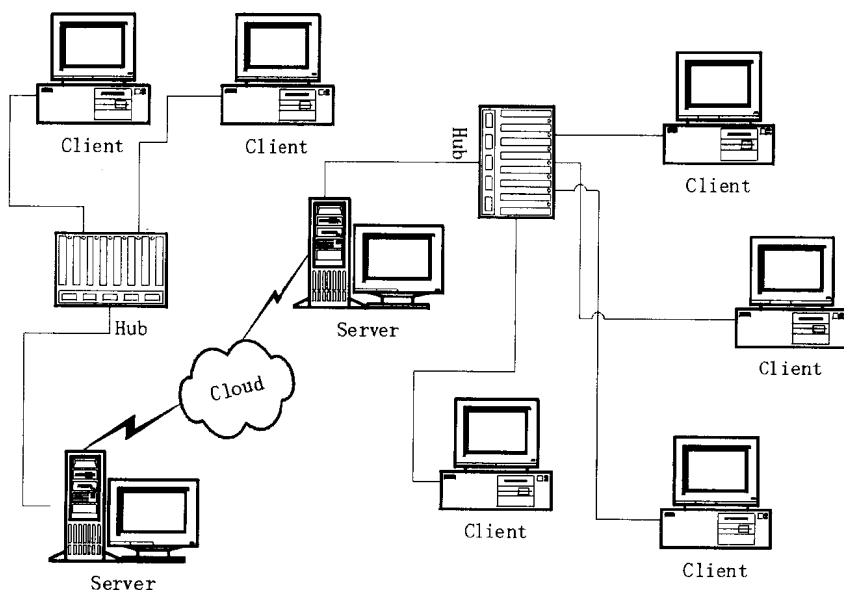


图 1-3 客户机/服务器模式：在客户和服务端划分应用

机端要具备有效性检验的手段,这常被称为业务规则。运行在服务器端的应用程序一般是在所有其它用户都能存取的中心位置存放数据。过去那种功能极强的主机就退化成数据库服务器了,而被退化掉的应用部分被转移到了客户机上。

这种转移带来了许多问题。客户机变得臃肿了(极端情况就是包含数据捕获、业务规则、应用接口的所谓“胖客户机”)。随着应用程序复杂性的增加,需要提供给用户的机器的性能也需要改善:更快的处理器、更大的硬盘、更多的内存。尽管PC机制造商能够为相同数量的钱提供越来越好的计算机,但是经常性的升级对任何企事业单位来说都是一笔昂贵的开销。就目前来说,用于对成千上万的PC机进行升级所耗费的人力和物力,多于仅用来对主机进行升级所耗费的人力和物力。除了对客户机进行装备所耗费的代价外,分发到成千上万的机器上的应用程序(包括业务规则)的增值对于企事业单位来说也是个很大的花费。以前IT工作人员能够在一个中心区域维护所有的公共软件,现在他们必须要在公司的各个角落维护公共软件。很多的估计表明装备一个客户机/服务器体系所需的费用是仅仅购买机器费用的八倍或更多。有些公司提供一些仅用于维护分散在企业各个地方的公共软件的系统,它们往往又大又昂贵。其中的问题在于要维护的公共软件不是像文字处理那样的个人软件,而是像定单账目和清算账目那样的公共软件。试想象一下,如果要有计划地对公共数据库做更新,同时客户机上的上千个拷贝也要做出相应的更新,该有多困难!部门级局域网的使用可以大大降低要更新的拷贝。然而局域网就受到了限制,因为超大的客户机应用软件需要相应的局域网的支持,否则用户会将软件拷贝到自己的机子上,这样需要维护的软件拷贝又回到了原来的数目。如何使得哪怕只有几百个客户机的软件拷贝做出相应的改变仍是个十分令人头疼的问题。

【提示】

优点:高效的交互性

友好的用户界面

缺点:数据交互和业务规则全移到了客户机上

客户机越来越庞大

缺乏集中控制

花费大

需要 LANS 辅助

在维护多个企业 LANS 仍旧很困难

1.1.3 浏览器/服务器(Browser/Server)结构

上述的客户/服务器框架,还存在着客户机和服务器间的通信问题,有一种被称为“中间件”(Middleware)的软件技术专门解决这一问题。但是,不同的开发商所开发的中间件并不是完全兼容,这就导致许多网络通信上的问题,这些问题的解决将会使客户/服务器开发变的更加容易。

随着 Web 技术的发展,由于 HTTP(HyperText Transfer Protocol:超文本传输协议)协议为所有的应用提供了统一的基础,解决了通信层的问题,接下来的问题就是在 Web 环境下如何划分我们的应用。

在基于 Web 的客户/服务器框架下,第一层为人机界面,一般为基于 HTML(HyperText Markup Language:超文本标记语言)的浏览器,有一些客户端控件或脚本技术用来丰富用户界面。

第二层,业务规则层,一般由内嵌 HTML 网页的脚本和控件来处理。当然,浏览器并没有固有的规则,它们仅存在于网页之中。由于这些规则要通过网络传输,会带来安全性的问题,采用安全套接层 SSL(Secure Sockets Layers)可以减少安全风险。在某些情况下,为增加安全性或当浏览器不具有处理这些规则的能力,规则层则可以位于服务器端。

第三层由服务器来完成将业务数据向技术数据转换。这可能是由 HTTP 请求触发一段程序来产生结构化数据,或由 ASP(Active Server Pages)来完成这一任务。

第四层为应用服务器,同样位于服务器端,该层处理和产生最后的结果。这部分可以为数据库或其它业务对象,结果是,挖掘出的信息发往浏览器或存储由浏览器发来的信息。

因此,在 Web 模式下,第一层和第二层位于浏览器端,而服务器端则包含第三和第四层。

基于 Web 的浏览器/服务器(B/S)结构是由客户/服务器(C/S)结构转化而来的,由上也可看出,B/S 模式可以看作是传统 C/S 模式的一个子集。

以下对 Browser/Server 模式与传统 Client/Server 模式进行了比较。

(1)Browser/Server 模式的优点

1)由于 Web 支持底层的 TCP/IP 协议,使 Web 网与目前使用的几乎所有局域网都可以做到无缝连接,从而彻底解决了异构系统间的连接问题;

2)由于 Web 采用了“瘦客户端”,使系统具开放性得到很大的改善,系统对将要访问系统的用户数的限制有所放松;

- 3)系统的相对集中性使得系统的维护和扩展变得更加容易。比如,数据库存储空间不够,可再加一个数据库服务器;系统要增加功能,可以新增一个应用服务器来运行新功能;
- 4)界面统一(全部为浏览器方式),操作相对简单;
- 5)业务规则和数据捕获的程序容易分发。

(2)Client/Server 模式的优点

- 1)Client/Server 方式有很强的实时处理能力,与 Web 方式相比,Client/Server 结构更适合于对数据库的实时处理和大批量的数据更新;
- 2)Client/Server 方式的面向对象技术十分完善,并且有众多与之配套的开发工具,这些开发工具已经做得非常优秀,这一点对目前 Web 技术来说绝对是个挑战。
- 3)由于 Client/Server 方式必须安装客户端软件,系统相对封闭,这反而使它的保密性能优于 Web 方式。

【提示】

表 1-1 给出了终端/主机、客户机/服务器、浏览器/服务器三种模式比较的一览表,从表中可以看出,浏览器/服务器模式具有灵活方便而不丢失强大的交互功能等优点。

表 1-1 三种模式总结

	技 术		
	终端/主机	客户机/服务器	浏览器/服务器
应用程序位于	主 机	用户的 PC 机	主机或服务器
客户端需要的软件	终 端	定制的客户软件	浏览器
用 户 界 面	字符集(Character)	图形用户界面(GUI)	图形用户界面(GUI)
每次传输对网络的要求	几 次	一次或几次	一 次

1.2 Web 服务器

这一节和下一节,将简要地介绍一下 Browser/Server 体系结构中的两大要素:Web 服务器和 Web 浏览器。读者如果熟悉 Client/Server 体系结构及其应用程序的开发,可以将它的两个要素客户机和服务器与 Web 浏览器和 Web 服务器进行对照,看看有什么区别,这样容易加深理解。如果从未开发过 Client/Server 应用程序,建议将本章第一节中有关 Browser/Server 的介绍仔细看一下,特别要注意理解四层结构的涵义。

1.2.1 Web 服务体系结构

后台服务器是任何好 Web 站点的心脏和灵魂。Web 服务器不仅可以按 HTML 文档形式提供静态 Web 页,而且也能够执行一些可极大地增加站点内容的应用程序。执行 Web 服务器上内容的构想并不是一个新思路。以前,Web 服务器使用通用网关接口(CGI: Common Gateway Interface)来获取站点内容。CGI 是一种允许 Web 服务器开始一个可执

行体,并使用进程来完成任务(如发送 E-mail)的技术。CGI 应用程序为 Web 站点增加了很多功能,但是由于 CGI 应用程序以独立进程的方式运行,所以速度很慢。而且,Web 服务器和 CGI 应用程序之间必须通过进程交换数据,因此它们之间的通信速度下降得很快(见图 1-4)。

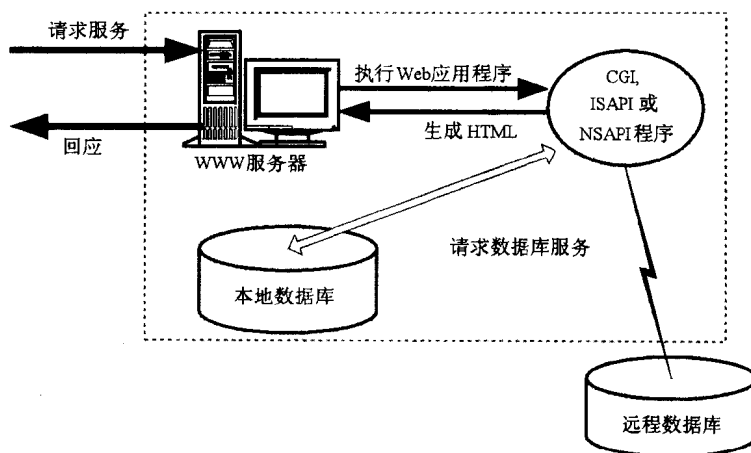


图 1-4 Web 服务的框架结构:使用 CGI 等技术

在 Browser/Server 体系结构中,Web 服务器的工作方式是:

- 1)接受请求;
- 2)执行服务器端的应用程序,如果必要,请求数据库服务;
- 3)产生 HTML 文档;
- 4)回应请求。

图 1-4 的虚线框里面的部分代表整个 Web 服务的框架结构。由于采用了 CGI 编程,则给编程人员带来很多的麻烦,而且因为 CGI 程序是个独立的程序,所以在和服务器的数据交换还会出现许多问题,像上面提到的速度问题只是其中的一个方面。在 HTML 语言得到了广泛应用和脚本语言在 Web 中的应用取得了很大成功后,人们开始寻求一种快速简洁的 Web 服务方案。

已有一种解决方案,即动态服务器网页技术(ASP: Active Server Pages),它考虑利用一种新的模型来替代上述传统的 Web 服务体系结构(见图 1-5)。原来的 CGI 等程序部分被 ASP 构建的模板所代替了,ASP 允许将静态的模板数据和服务器端可执行的脚本绑在单独的一个文件中,所以整个的 Web 服务过程变成了对 ASP 中的模板数据的填充替换过程。在本书的第四章对 ASP 技术有进一步的阐述。

1.2.2 几种 Web 服务器软件系统

1. Internet 信息服务器

Microsoft 信息服务器(Internet Information Server 或 IIS)对那些使用 Windows NT 技术构建的站点来说是最主要的 Web 服务器,它体现了在简单使用 CGI 创建内容的服务

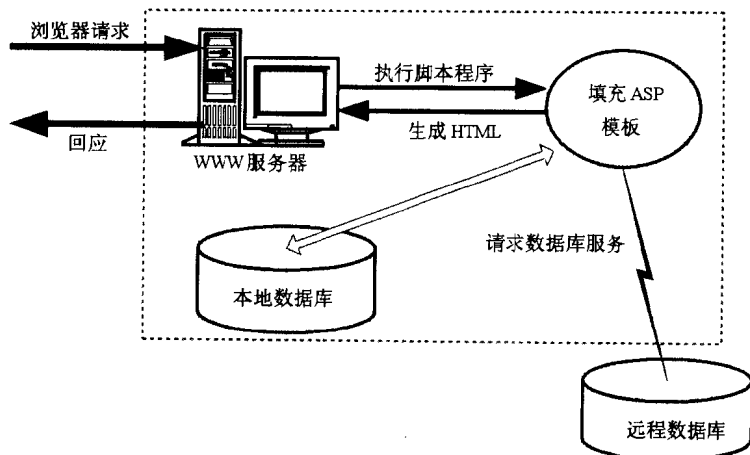


图 1-5 Web 服务的框架结构:使用 ASP 技术

器上的重大改进。IIS 支持一组使用 Internet 服务器应用程序编程接口 ISAPI(Internet Server Application Programming Interface)的新应用。IIS 能够以流行的模拟 CGI 方式执行 ISAPI 应用程序,但是它们与 Internet 服务器在相同的内存空间中运行。事实上,ISAPI 是作为动态链接库构造的,它们比等价的 CGI 应用程序要快 20 倍。Microsoft Active Server Pages(ASP)就是一个 ISAPI 应用程序的例子。

2. 个人 Web 服务器

Personal Web Server(PWS)是专为使用 Microsoft Windows 95(Windows 98)和 NT 工作站设计的服务器。它对于脱机开发站点是理想的服务器。事实上,个人 Web 服务器支持 IIS 完全版本中的许多特性,包括虚拟目录和 Active Server Pages。PWS 与 Visual InterDev 一起发送,并且与 Visual InterDev 一起用于创建站点。

3. 其它 Web 服务器软件

(1) Netscape Enterprise Server 3.6

Enterprise Server 3.6 具有一些高端特性,如基于 LDAP(一种目录服务标准,属 X.500 的一个子集)的目录服务器、Web 集群技术、对所有主要大型数据库的支持。Enterprise Server 的浏览器管理方式使你可以从任何地方管理你的 Web 服务器。除运行于 NT 平台上,它还支持多数 Unix 商业版本。

Enterprise Server 提供了一个强大的开放式开发平台,它支持从 CGI 到 JavaScript 和 NSAPI 的各种开发接口,在网点管理器(Site Manager)和应用程序管理器(Application Manager)中提供了功能丰富的 Live Wire 模块。在战略上,Netscape 提出了强大的应用框架:开放网络环境(ONE)和 Internet 基础类库(IFC)。

(2) Apache 1.3.4

Apache 是全球用得最广泛的 Web 服务器。它是源代码公开的免费软件,该软件的最新版本可以从 Apache 的 Web 站点免费获得,其可自解压的安装文件还不到 3MB。

Apache 有 Unix, Linux, NT 多种版本, 在 NT 上安装 Apache 意味着要在 DOS 界面下处理基于文本方式的命令文件。Apache 的许多优势包括简单的虚拟主机设置、广泛的平台支持、灵活性、注重标准以及公开的源代码。

【提示】

ASP 是服务器端的脚本编写环境, 可用它来创建动态 Web 页或生成功能强大的 Web 应用程序。ASP 页是包括 HTML 标记、文本和脚本命令的文件。并且, ASP 页可以调用 ActiveX 组件来执行任务。

1.3 Web 浏览器

目前市场上的 Web 浏览器主要有 Microsoft 公司的 Internet Explorer (IE) 和 Netscape 公司的 Netscape Navigator。这两种浏览器的工作原理比较相象, 但是它们内部的构建过程不大一样, 特别是两者对 HTML 语言(稍后将会提到)的支持也有区别。作为使用者可能并不关心浏览器的内部技术, 只关心它提供了什么技术。在这一方面, 两种浏览器的开发公司都以能够支持更多的技术为目的, 在努力改善着自己的产品。所以仅考虑浏览器所能支持的技术, 两者区别不是很大, 只是在 IE 被捆绑到 Windows 操作系统中后, 具有了一些新的特征。下面我们就以 IE 4.0 为例阐述有关的 Web 浏览器技术。

1.3.1 Web 浏览器技术

1. IE 4.0 的体系结构

IE 4.0 采用 Active 控件(Controls)和 Active 文档接口将组件(Components)连接起来。图 1-6 给出了 IE 高层框架结构。

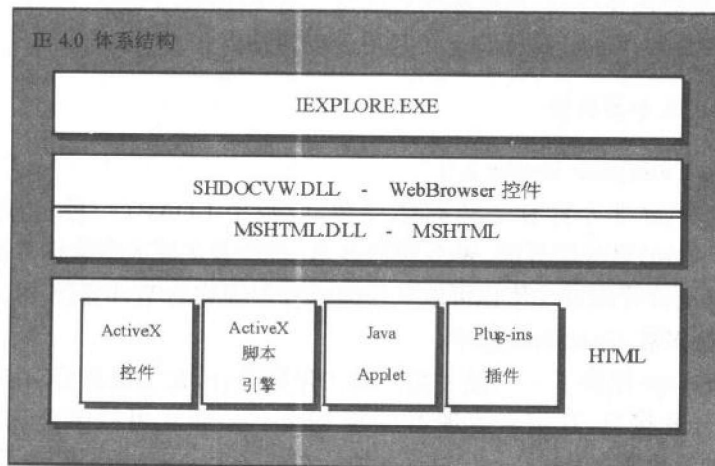


图 1-6 IE4.0 的体系结构

在最高层为 IExplore. exe, 当 IE 4.0 浏览器启动后, 一个特别小的应用程序就被实例化了。该执行程序使用 IE 4.0 组件来完成导航、历史记录、收藏、HTML 文档解释和显示