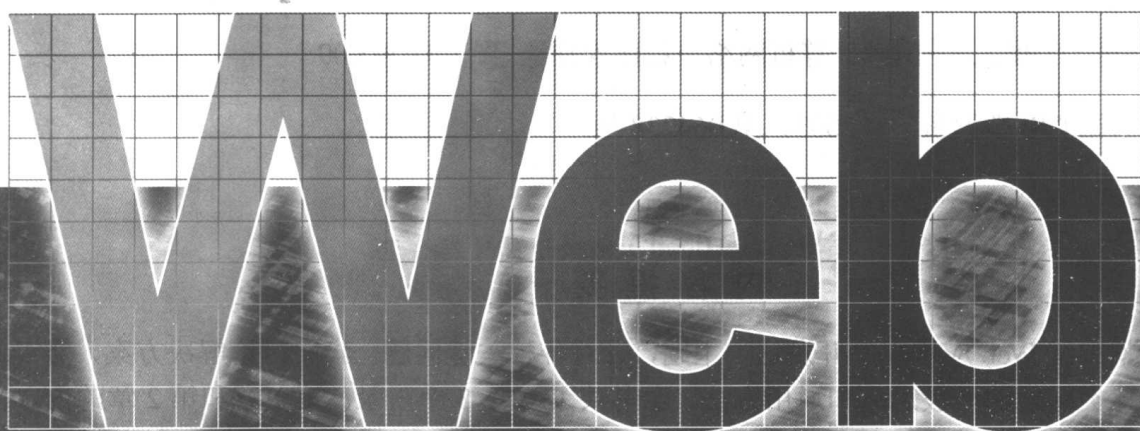




应用程序开发 ——算法分析与应用

■ 侯志荣 编著

The word "Web" is rendered in a large, bold, sans-serif font. The letters are filled with a dark, textured pattern that resembles a close-up of a circuit board or a digital grid. The letters are set against a lighter, grid-like background that also has a textured appearance.

应用程序开发 ——算法分析与应用

■ 侯志荣 编著

人民邮电出版社

图书在版编目 (CIP) 数据

Web 应用程序开发: 算法分析与应用 / 侯志荣 编著. —北京: 人民邮电出版社, 2003.9
ISBN 7-115-11499-4

I. W... II. ①侯... III. 计算机网络—程序设计 IV. TP393.09

中国版本图书馆 CIP 数据核字 (2003) 第 064238 号

内 容 提 要

本书紧紧围绕 Web 应用程序开发这一主题, 针对各种关键问题, 集中讨论解决方案, 努力探求解决该类问题的一般思路和通用方法。全书共 15 章、3 个附录。第 1 章和第 2 章的内容是 Web 开发的基础, 介绍 Web 应用程序的体系结构、开发环境配置以及 HTTP 等基础理论知识; 第 3 章至第 12 章的内容是进行 Web 开发各阶段的算法分析, 以专题形式讨论了 Web 应用程序开发中的各种具体问题及其解决方案; 第 13 章到第 15 章介绍几个具体 Web 开发应用实例。

本书适合 Web 应用程序开发人员和 Web 系统管理员阅读。

Web 应用程序开发——算法分析与应用

- ◆ 编 著 侯志荣
责任编辑 刘 浩
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132692
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 25.5
字数: 618 千字 2003 年 9 月第 1 版
印数: 1-5 000 册 2003 年 9 月北京第 1 次印刷

ISBN7-115-11499-4/TP · 3545

定价: 42.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

互联网发展到今天,已经成为人类社会生活中不可缺少的一部分,其中 WWW 起到了巨大的推动作用。如今,Web 已成为社会信息交流的一个平台,大量网站在互联网中兴起,各种基于 Web 的商业应用正方兴未艾。借微软公司的话说:“The future is the web”。

考察当前的应用程序开发,可以发现一个绝对的倾向:人们开始偏向基于 Web 浏览器的瘦客户应用程序,即基于“B/S”模式的软件,或者说 Web 应用程序。与传统的桌面程序相比,Web 应用程序的优点在于无需开发客户端程序,从而可以大大降低成本,同时也降低了程序的使用难度。

Web 技术日新月异,从 HTML 到 XML,从 Perl 到 JSP,Web 应用程序开发者不断面临新的技术挑战,在频繁的变化面前不免产生疑问:究竟该学什么?PHP 好还是 JSP 好?作为一个真正的程序员,应该能够摆脱“语言”的束缚,而掌握“算法”这个精髓。所谓算法,就是解决问题的基本思路和方法。环顾今日之计算机图书市场,各种 Web 编程语言的书籍随处可见,但系统讨论 Web 应用程序开发中的关键问题及其解决方案的技术书籍并不多见。

本书紧紧围绕 Web 应用程序开发这一主题,针对各种关键问题,集中讨论解决方案,努力探求解决该类问题的一般思路和通用方法。全书共 15 章,3 个附录,分为 3 部分:第 1 章和第 2 章为基础内容,介绍 Web 应用程序的体系结构、开发环境配置以及 HTTP 等基础理论知识;第 3 章至第 12 章进行 Web 开发各阶段的算法分析,以专题形式讨论了 Web 应用程序开发中的各种具体问题及其解决方案;第 13 章到第 15 章为具体应用实例,首先介绍 Web 应用程序的建模方法,然后给出 Web 应用程序开发的具体实例,最后讨论了 Web 应用程序的扩展。

本书给出了大量程序实例,但并非程序代码的汇集,而是希望通过简单的例子来验证算法的可行性。本书不可避免地使用了具体的编程语言,主要目的在于演示如何使用具体的编程语言来实现算法程序,探讨 Web 应用程序开发的关键问题及其一般解决方案才是本书的宗旨。笔者相信,只要把握了问题的基本特点及其一般处理思路,Web 应用程序开发者就可以迅速运用具体的编程语言实现程序代码,解决实际问题。

本书适合所有 Web 应用程序开发人员和 Web 系统管理员阅读。无论采用何种编程语言,使用何种操作系统,本书都能够让读者从中受益。如果是 Web 应用程序开发的初学者,将本书与一般技术书籍相结合,必将有助于提高学习速度,加深对 Web 技术的理解;如果读者已经熟悉某种 Web 编程语言,本书的内容将有助于从全局高度深入地把握 Web 应用程序开发。

本书由侯志荣、张辞编著,张辞完成了第 2 章和全部附录的编写工作,并绘制了本书全部图表,侯志荣撰写了本书其余章节的内容,并对全书统一定稿。由于笔者水平有限,加之时间紧促,错误疏漏之处在所难免,望读者批评指正(可发信至 ridgel@sina.com)。

本书代码可在 <http://www.ucbook.com> 处下载。

编者

2003 年 6 月

目 录

第 1 章 Web 应用程序开发概述	1
1.1 Web 发展历史	1
1.2 Web 应用程序的体系结构	2
1.2.1 2 层体系结构	2
1.2.2 3 层体系结构	3
1.2.3 4 层体系结构	3
1.3 Web 开发技术与开发工具	4
1.3.1 Web 开发技术	4
1.3.2 Web 开发工具	7
1.4 Web 开发环境配置	8
1.4.1 ASP 开发环境的配置	8
1.4.2 Perl 开发环境配置	10
1.4.3 PHP 开发环境配置	19
1.4.4 JSP 开发环境配置	27
1.4.5 常见 Web 开发技术的性能比较	28
1.5 Web 应用程序的特点	30
1.6 本章小结	31
第 2 章 HTTP	32
2.1 HTTP 简介	32
2.2 HTTP 的通信机制	32
2.3 HTTP 请求	34
2.3.1 HTTP 请求格式	34
2.3.2 HTTP 请求方法	35
2.4 HTTP 应答	36
2.4.1 HTTP 应答格式	36
2.4.2 HTTP 应答码	37
2.5 内容类型	37
2.5.1 服务器使用内容类型的方法	38
2.5.2 内容类型的格式	38
2.5.3 非标准类型	39
2.6 安全连接	39
2.7 本章小结	40

第 3 章 用户接口设计	41
3.1 单向用户接口	41
3.1.1 辅助窗口设计	41
3.1.2 层设计	43
3.1.3 提示窗口设计	45
3.2 双向用户接口	46
3.2.1 对话框设计	46
3.2.2 表单设计	47
3.3 用户接口设计原则	65
3.4 本章小结	65
第 4 章 客户端信息获取	66
4.1 HTTP 头信息	66
4.1.1 HTTP 头信息的数据格式	66
4.1.2 获取 HTTP 头信息的方法	66
4.2 Cookie	68
4.2.1 Cookie 的工作原理	69
4.2.2 Cookie 的基本参数	70
4.2.3 Cookie 的数据格式	70
4.2.4 Cookie 的存取方法	71
4.2.5 Cookie 应用示例	74
4.3 查询字符串	76
4.3.1 查询字符串的数据格式	77
4.3.2 获取查询字符串的方法	77
4.4 表单内容	78
4.4.1 表单内容的数据格式	78
4.4.2 获取表单内容的方法	79
4.5 客户计算机信息	81
4.6 本章小结	83
第 5 章 会话管理	84
5.1 HTTP 的缺陷	84
5.2 会话机制剖析	85
5.3 会话管理解决方案	85
5.3.1 Perl 的会话管理解决方案	85
5.3.2 PHP 的会话管理解决方案	89
5.3.3 ASP 的会话管理解决方案	96
5.3.4 JSP 的会话管理解决方案	99
5.4 关于会话的几个试验	101

5.4.1	试验一 PHP 会话的基本工作方式	101
5.4.2	试验二 PHP 会话中 Cookie 的作用	103
5.4.3	试验三 PHP 的 session_destroy()函数与 session_unset()函数	103
5.4.4	实验四 ASP 会话的工作原理	105
5.4.5	实验五 JSP 会话测试	107
5.5	本章小结	109
第 6 章	数据访问	110
6.1	数据库访问	110
6.1.1	数据库应用程序接口	111
6.1.2	SQL 语言	115
6.1.3	数据库访问程序的基本结构	116
6.1.4	连接数据库	117
6.1.5	查询数据记录	124
6.1.6	修改数据记录	127
6.1.7	显示数据记录	128
6.2	文件操作	142
6.2.1	文件处理概述	142
6.2.2	文件基本操作	143
6.2.3	文件远程处理	147
6.3	XML 处理	156
6.3.1	XML 简介	156
6.3.2	创建 XML 文件	159
6.3.3	显示 XML 文件	165
6.3.4	XML 文件内容操作	173
6.4	电子邮件	178
6.4.1	发送纯文本邮件	179
6.4.2	发送带附件的邮件	183
6.4.3	接收电子邮件	184
6.5	Office 文档	186
6.5.1	Excel 文档	186
6.5.2	Word 文档	191
6.6	本章小结	193
第 7 章	参数传递	194
7.1	基于表单隐藏域的参数传递方法	194
7.1.1	基本原理	194
7.1.2	应用实例	195
7.2	基于 Cookie 的参数传递方法	197
7.2.1	基本原理	198

7.2.2 应用实例	198
7.3 基于查询字符串的参数传递方法	201
7.3.1 基本原理	201
7.3.2 应用实例	201
7.4 基于服务器内存变量的参数传递方法	203
7.4.1 基本原理	203
7.4.2 应用实例	204
7.5 基于服务器文件的参数传递方法	207
7.5.1 基本原理	207
7.5.2 应用实例	207
7.6 参数传递方法比较	210
7.7 本章小结	210
第 8 章 用户跟踪	211
8.1 用户身份确认方法	211
8.2 用户在线状态跟踪	212
8.2.1 基本思路	212
8.2.2 算法设计	213
8.2.3 程序实现	214
8.3 用户当前位置跟踪	217
8.3.1 基本思路	218
8.3.2 算法设计	218
8.3.3 程序实现	218
8.4 用户访问路线跟踪	221
8.4.1 基本思路	221
8.4.2 算法设计	221
8.4.3 程序实现	222
8.5 本章小结	224
第 9 章 流量统计	225
9.1 页面流量统计	225
9.1.1 系统首页的流量统计	225
9.1.2 系统每一页的流量统计	227
9.1.3 用户访问页面的次数统计	228
9.2 IP 流量统计	229
9.3 Web 服务器日志分析	230
9.3.1 Web 服务器的日志格式	230
9.3.2 基于 Web 服务器日志的访问流量统计	231
9.3.3 基于 Web 服务器日志的 IP 流量统计	234
9.4 流量数据的图形化表示方法	235

9.4.1 图形数字表示方法	235
9.4.2 条形图表示方法	237
9.4.3 饼图表示方法	241
9.5 本章小结	245
第 10 章 时间处理	246
10.1 时间获取	246
10.1.1 客户端时间获取	246
10.1.2 服务器时间获取	248
10.2 时钟问题	248
10.2.1 基本思路	248
10.2.2 数字时钟	249
10.2.3 图形时钟	252
10.3 计时方法	257
10.3.1 停留时间	257
10.3.2 倒计时	258
10.3.3 定时	262
10.4 日历设计	266
10.4.1 功能简介	266
10.4.2 关键问题	266
10.4.3 设计实例	267
10.5 本章小结	269
第 11 章 安全控制	270
11.1 Web 应用程序安全概述	270
11.1.1 Web 应用程序面临的安全威胁	270
11.1.2 Web 应用程序安全控制的基本框架	271
11.1.3 防火墙在 Web 应用程序安全中的应用	272
11.1.4 安全原则在 Web 应用程序安全中的应用	274
11.2 客户端安全控制	274
11.2.1 浏览器插件安全控制	275
11.2.2 Cookie 安全控制	276
11.2.3 脚本程序安全控制	278
11.3 服务器安全控制	281
11.3.1 用户身份验证	282
11.3.2 服务器资源访问控制	290
11.3.3 CGI/API 程序安全控制	297
11.4 网络传输安全控制	301
11.4.1 SSL 的工作原理	301
11.4.2 为 Web 应用程序配置 SSL	302

11.5 本章小结	307
第 12 章 性能优化	308
12.1 Web 应用程序性能测试	308
12.1.1 Web 应用程序的性能测试概述	308
12.1.2 基于 ApacheBench 的性能测试	309
12.1.3 基于 WAST 的性能测试	312
12.2 环境优化	314
12.2.1 Web 服务器	315
12.2.2 程序运行模式	316
12.3 程序优化	317
12.3.1 输出方式	317
12.3.2 注释语句	320
12.3.3 脚本使用方式	321
12.3.4 包含文件	323
12.3.5 分支结构	325
12.3.6 循环结构	327
12.4 数据库优化	330
12.4.1 连接方式	330
12.4.2 SQL 语句	334
12.4.3 字段输出	337
12.4.4 分页显示	340
12.5 本章小结	341
第 13 章 Web 应用程序建模	342
13.1 统一建模语言——UML	342
13.1.1 UML 概述	342
13.1.2 UML 中的事物	342
13.1.3 UML 中的关系	344
13.1.4 UML 中的图	345
13.1.5 UML 建模工具	345
13.2 基于 UML 的 Web 应用程序建模	346
13.2.1 体系结构分析	346
13.2.2 用户角色分析	347
13.2.3 系统需求分析	348
13.2.4 事务流程分析	348
13.2.5 Web 页面建模	349
13.3 Web 建模语言——WebML	352
13.3.1 WebML 概述	352
13.3.2 结构模型	353

13.3.3 超文本模型	355
13.3.4 站点视图	361
13.3.5 WebML 建模流程	362
13.4 本章小结	363
第 14 章 案例研究——Web 办公系统	364
14.1 系统分析	364
14.1.1 功能描述	364
14.1.2 体系结构	364
14.1.3 用户分析	365
14.1.4 系统用例	365
14.2 模块设计	366
14.2.1 模块功能与组织方式	366
14.2.2 文档交流模块建模	367
14.3 关键问题与解决方案	369
14.3.1 用户管理方案	369
14.3.2 用户身份自动识别问题	370
14.3.3 收文人指定方法	370
14.4 本章小结	370
第 15 章 Web 应用程序的扩展	372
15.1 WAP 应用程序开发	372
15.1.1 体系结构	372
15.1.2 环境配置	373
15.1.3 开发步骤	375
15.1.4 程序示例	376
15.2 基于 Web 的 MATLAB 应用程序开发	377
15.2.1 体系结构	378
15.2.2 环境配置	378
15.2.3 开发步骤	380
15.2.4 程序示例	382
15.3 本章小结	385
附录 A HTTP 应答码	386
附录 B 环境变量	389
附录 C SSL 3.0	390

第 1 章 Web 应用程序开发概述

本章概述了 Web 应用程序开发的基础知识, 主要包括 Web 发展历史、Web 应用程序的体系结构、Web 开发技术和开发工具以及 Web 开发环境配置等内容。本章旨在让读者对 Web 应用程序开发有一个总体认识和宏观把握, 尤其是掌握 Web 应用程序的体系结构与开发环境配置方法。

1.1 Web 发展历史

根据 W3C (World Wide Web Consortium) 协会的资料, Web 的历史可以追溯到 1945 年。1945 年, Vannevar Bush 发表了一篇关于一种电子图片设备的论文, 指出该设备可以在微缩胶片的文档间建立和使用链接 (Links)。这就是超级链接的雏形。

20 世纪 60 年代, Doug Engelbart 提出了 “ONLine System” (NLS) 原型, 它使用超文本 (Hypertext) 进行浏览、编辑并用电子邮件的方式发送。1965 年, 在纽约召开的第 20 届国家会议 (National Conference) 上, Ted Nelson 正式提出 “Hypertext” 一词。1967 年, Andy van Dam 等人开发了一套正式的超文本编辑系统。

1980 年, Tim Berners-Lee 编写了一个文本程序——“Enquire-Within-Upon-Everything”, 该程序允许在任何两个节点之间建立链接, 每个节点具有一个标识、类型, 并且拥有一个双向链接列表。实际上, 这就具备了我们今天的 HTML 系统的基本特点。

1990 年是 Web 发展史上值得纪念的一年。9 月, Tim 的老板 Mike Sendall 同意购买一台 NeXT (可能是早期的一种计算机系统) 并支持 Tim 编写一套全球范围的超文本系统。10 月, Tim 开始了在 NeXT 系统下的基于图形用户界面的 Web 浏览器和编辑器的开发工作, 并将这个项目命名为 “World Wide Web (WWW)”。随着 World Wide Web 项目的推进, 这一年的 11 月, 世界上第一台 Web 服务器 `nxoc01.cern.ch` (后来称作 `info.cern.ch`) 与第一份网页 `http://nxoc01.cern.ch/hypertext/WWW/TheProject.html` 诞生了。

1994 年 5 月, 在瑞士的日内瓦, 第一届国际 WWW 会议召开。同年 10 月 1 日, 著名的 W3C 协会成立。从此, Web 就进入了一个有组织的发展阶段。今天, 如果要了解 Web, W3C 协会的网站 `http://www.w3.org` 仍然是最好的信息来源。

1995 年以后, Web 的发展更加迅速, 并且开始从学术殿堂进入寻常百姓的生活。杨致远及其同伴创建的 Yahoo! 网站一夜成名, 创造了 Web 发展史上的一个奇迹。直到今天, Yahoo! 仍然是因特网上最著名的网站之一。随后的亚马逊网上书店 (`http://www.amazon.com`) 通过 Web 开展图书营销活动, 也取得了显著成就, 成为电子商务网站的先驱。亚马逊的成功, 对 Web 的发展产生了深远的影响。同时, 其经营模式与经营理念也成为了市场营销领域的一个重要研究课题。

1995 年 Web 进入我国, 随后在中国掀起了一股网络发展的高潮。特别是 1998 年以来,

新浪、搜狐、网易三大门户网站相继诞生。随后的几年里，为了推动中国信息化的进程，有关单位先后在全国范围内开展了“政府上网工程”、“企业上网工程”和“家庭上网工程”三大系列活动。如今，随着电子商务、远程教育以及电子政务等概念的提出，中国的网络发展出现了前所未有的兴盛局面。

Web 应用程序（Web Applications）一词什么时候最早出现已不得而知。笔者在微软公司发布的 Web 开发工具 IntelDev 的帮助文档中第一次发现这个术语。Web 应用程序最显著的特点就是在原来的 HTML 文本系统中加入了对数据库或文件访问的支持，Web 服务器不仅能响应 Web 浏览器的文档请求，而且能够执行特定的应用程序，并将程序结果以 HTML 的形式返回给 Web 浏览器。通常，我们把基于 Web 的软件系统称作 Web 应用程序。

1.2 Web 应用程序的体系结构

Web 应用程序的体系结构描述了构成 Web 应用程序的逻辑功能模块，它直接影响到 Web 应用程序的系统设计、技术方案以及编程方法。因此，Web 应用程序开发人员必须熟悉 Web 应用程序的各种体系结构，才能正确实现所需求的系统功能。通常，Web 应用程序可以采用 4 种基本的体系结构：2 层结构、3 层结构以及两类 4 层结构。

1.2.1 2 层体系结构

采用 2 层体系结构的 Web 应用程序由 Web 浏览器与 Web 服务器两部分组成，如图 1-1 所示。Web 浏览器与 Web 服务器之间的通信遵循 HTTP。Web 浏览器向 Web 服务器请求文档，Web 服务器则根据该请求返回相应的文档。如果该文档不存在，Web 服务器将返回错误提示信息。最早的 Web 网站就采用了这种简单的 2 层结构。采用这种结构的典型例子就是国外一些针对某些专题的 FAQs（Frequently Ask Questions）以及一些软件（比如 Apache Web Server）的 HTML 帮助系统。

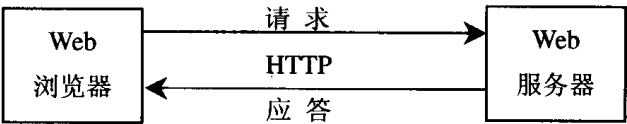


图 1-1 Web 应用程序的 2 层体系结构

在 Web 发展早期，由于 Web 服务器中的文档是事先编写好的，任何用户请求文档时返回的结果都一样，所以这种网站被称作“静态网站”，这里的“静态”是指文档内容不变化。随着 Web 技术的发展，人们开始提出一些增强 Web 浏览器功能的新技术，比如让浏览器执行一些简单的脚本程序。这样一来，Web 浏览器从 Web 服务器接收到文档后，就可能根据用户的操作呈现不同的结果。网上一些流行的心理测试就是通过浏览器执行嵌入在 HTML 文档中的 Javascript 程序来实现的。无论是简单的 HTML 文档，还是嵌入脚本程序的 HTML 文档，两层体系结构的 Web 应用程序的基本逻辑结构都一致。Web 服务器的任务就是响应 Web 浏览器的文档请求。

1.2.2 3 层体系结构

采用 3 层体系结构的 Web 应用程序由 Web 浏览器、Web 服务器和数据信息 3 部分组成, 如图 1-2 所示。其中, 最典型的数据信息就是数据库。Web 浏览器与 Web 服务器之间的通信仍然遵循 HTTP; Web 服务器与数据信息(比如数据库)之间的通信遵循 CGI(Common Gateway Interface) 或者 Server API (Application Interface) 规范。在采用 3 层体系结构的 Web 应用程序中, Web 浏览器向 Web 服务器请求某个 HTML 文档或者请求执行某个 CGI 脚本程序。Web 服务器根据 Web 浏览器请求文档的类型执行相应的操作: 如果 Web 浏览器请求的是某个 HTML 文档, Web 服务器就返回该文档; 如果 Web 浏览器请求的是某个 CGI 程序或者 API 程序, Web 服务器则执行(或调用外部程序执行)该程序, 然后将程序执行结果返回给 Web 浏览器。

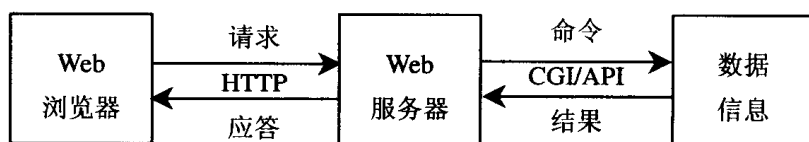


图 1-2 Web 应用程序的 3 层体系结构

通过 CGI 脚本或者 Server API 程序来扩展 Web 服务器的功能可以说是 Web 技术发展史上的一个里程碑。因为, CGI 脚本或者 Server API 程序可以根据用户的不同请求, 返回不同的结果。典型的例子就是在线查询, 用户输入不同的查询关键字, Web 服务器就返回相应的查询结果。习惯上, 我们把采用这种体系结构的 Web 网站称作“动态网站”。目前, 互联网中绝大部分网站和大多数 Web 应用系统都采用了这种体系结构。由于采用 3 层结构的 Web 应用程序基本上可以实现传统桌面软件的绝大部分功能, 所以 Web 就不仅是发布信息的平台, 而且也是软件开发的平台。

1.2.3 4 层体系结构

4 层体系结构是对 3 层体系结构的扩展。根据不同的应用环境, 采用 4 层体系结构的 Web 应用程序又可分为两种不同的类型: 类型 I 和类型 II。类型 I 的基本结构如图 1-3 所示, 它由 Web 浏览、Web 服务器、应用服务器以及数据信息 4 部分组成。在采用这种结构的 Web 应用程序中, Web 浏览器向 Web 服务器提出请求, Web 服务器分析请求: 如果 Web 浏览器请求的是简单的 HTML 文档, Web 服务器就返回相应的文档; 如果 Web 浏览器请求的是特殊文档, Web 服务器就将该请求交给应用服务器执行, 应用服务器根据请求, 访问相应的数据信息, 然后把执行结果返回给 Web 服务器, 并通过 Web 服务器将结果以 HTML 的形式返回给 Web 浏览器。

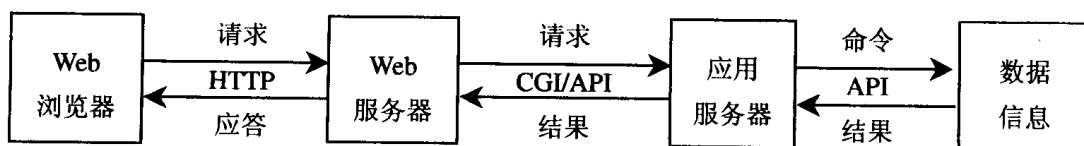


图 1-3 Web 应用程序的 4 层体系结构 (类型 I)

基于 Web 的 Matlab 应用程序就是采用类型 I 这种 4 层体系结构的 Web 应用程序。其中，应用服务器是 Matlab Web Server。本书第 15 章将详细介绍这一应用形式。

图 1-4 所示描述的是另一种类型的 4 层体系结构，它由客户设备、应用服务器、Web 服务器以及数据信息 4 部分组成。这种体系结构与前面 3 种结构最大的不同在于：它取消了 Web 浏览器，而代之以客户设备。手机和掌上电脑都是生活中常见的客户设备。客户设备与应用服务器之间的通信遵循专门的通信协议，而应用服务器与 Web 服务器之间的通信仍然遵循 HTTP，Web 服务器与数据信息之间的通信仍然遵循 CGI 或 Server API 规范。

大家比较熟悉的 WAP 应用就是采用这种结构模型的 Web 应用程序。在 WAP 应用中，客户设备通常是手机，应用服务器一般是 WAP 网关。客户设备与 WAP 网关的通信采用 WAP 协议，手机屏幕显示的文档则由 WML 实现。

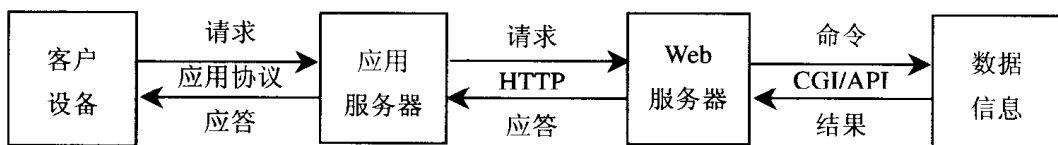


图 1-4 Web 应用程序的 4 层体系结构（类型 II）

1.3 Web 开发技术与开发工具

“工欲善其事，必先利其器”。Web 应用程序开发人员必须掌握一定的 Web 开发技术和相应的开发工具才能“构造”出真正的 Web 应用程序。同样，熟悉这些技术和工具也是阅读本书的基本条件。本节将介绍一些常用的 Web 开发技术和开发工具。

1.3.1 Web 开发技术

1. HTML (HyperText Markup Language)

HTML 是 Web 文档的描述语言，它是整个 Web 的基础。严格地讲，HTML 不能算作一门编程语言，因为它没有自己的数据类型、也没有分支、循环等控制结构。HTML 由一系列标记构成，所以 HTML 语言也被称作标记语言。HTML 标记放在<>之内，每个标记可以具有一个或几个控制属性。由 HTML 编写的文档被称作 HTML 文档，即通常所说的网页。Web 浏览器在显示 HTML 文档时，将滤除 HTML 标记，只显示除标记以外的其他内容。历史上，HTML 有多个版本，目前的版本为 HTML 4.0。

2. CSS(Cascading Style Sheet)

一般把 CSS 翻译为层叠样式表，简称样式表。CSS 的最新版本为 CSS 2.0。CSS 的出现，为实现 HTML 文档内容和样式（即内容显示格式）的分离提供了有力的技术手段。利用 CSS，Web 开发人员可以在 HTML 文档中安排内容，而在另一份样式文档中设计这些内容的显示方式，比如字体颜色、字体大小、行距等。另外，CSS 的出现还有利于 Web 应用程序开发中的

分工合作，因为有了 CSS，就可以让美工编写 CSS 文档，而 HTML 程序员只编写文档内容，他们分工合作就可以有效地完成 Web 文档的开发。

3. 客户端脚本程序

最初的 Web 浏览器只能显示文本、图片等静态内容。后来，为了扩展网页的功能，人们在网页中引入了嵌入脚本程序。相应地，Web 浏览器也加入了执行这些脚本程序的解释器。常用的脚本语言有 Javascript、VBScript、JScript 以及 Perlscript，其中 Javascript 最为流行。通过在网页中嵌入脚本程序，就可以实现许多动态特性，有些网站甚至利用脚本程序来开发一些简单的网络游戏。必须注意，由于网页中嵌入的脚本程序是由请求该网页的 Web 浏览器解释执行的，所以 Web 浏览器的差异可能导致脚本程序不能正常运行。这就是采用客户端脚本程序所带来的风险。

4. DOM(Document Objects Model)

DOM 即文档对象模型。文档对象模型的基本思想是将 Web 浏览器窗口以及其中的内容都看作对象。根据面向对象的思想，每一对象都具有特定的属性和方法。通过调用对象的方法，Web 开发人员就可以执行相应的文档操作。另外，借用面向对象编程中的“事件驱动”概念，DOM 为网页定义了一些事件，主要包括鼠标事件、键盘事件和系统事件 3 类。这样一来，Web 程序员就可以在程序中利用这些事件来响应用户的操作。不过，不同的浏览器（比如 Internet Explorer 和 Netscape Navigator）定义的 DOM 有所区别。

5. DHTML(Dynamic Hypertext Markup Language)

DHTML 实际上不是一种具体的技术，也不是一种编程语言，它是一个标准，一个 Web 客户端程序开发的标准。DHTML 与 HTML 的区别就在于 Dynamic 一词。DHTML 的主要思想就是增加 HTML 文档的吸引力和与用户的“交互性”。用脚本程序来操纵 HTML 文档的样式与内容（充分利用 CSS 和 DOM）是 DHTML 最主要的实现方式。DHTML 主要包括两个方面：变化方式与变化内容。变化方式决定网页特效如何变化，变化内容指定变化对象。笔者将它们称作 DHTML 的“经”和“纬”，它们的组合就构成了 DHTML 的基本动态效果，如表 1-1 所示。例如，用户用鼠标单击网页中的某张图片，该图片就由红色变为蓝色。这个动态效果就属于事件驱动的样式变化，事件就是鼠标单击，样式就是图片颜色。将表 1-1 中的这些基本动态效果结合起来还可以构成更加复杂的动态效果。

表 1-1

DHTML 的基本动态效果

	样 式	内 容
时间驱动	时间驱动的样式变化	时间驱动的内容变化
事件驱动	事件驱动的样式变化	事件驱动的内容变化

6. ActiveX

ActiveX 是微软公司提出的一项增加 Web 浏览器功能的新技术。其基本原理是利用一些专门的 Web 浏览器插件来实现网页中的某些特殊效果，这些插件就被称作 ActiveX 控件。显

然，它要求用户的计算机中必须首先安装这些插件。如果没有安装这些插件，系统将提示用户从互联网上下载并安装。这就明显存在一个问题：如果这些插件中包含了恶意的攻击代码，那么用户的计算机就可能遭受损害。尽管许多从网络下载的插件都经过了安全检查，但是仍然存在安全风险，ActiveX 技术因此受到人们的批评，而逐渐被抛弃。

7. Java Applet

Java 语言的横空出世给网络开发带来了巨大影响，其独特魅力就在于跨平台特性。Java 语言的跨平台原理其实并不复杂，与传统的高级语言不同，Java 程序编译后，并不形成可执行的二进制代码，而是形成一种中间代码，有文献将这种代码称作“字节码”（Byte Code）。中间代码最终交给（JVM Java 虚拟机）去执行。因此，无论何种平台，只要安装了 JVM，就可以执行编译后的 Java 程序。

Java Applet 也被称作 Java 小程序，它是 Java 程序的一类。Java Applet 与 ActiveX 有些类似，主要用来增强 Web 浏览器的功能。Java Applet 的工作原理是：Web 浏览器下载 HTML 文档的同时，将 Java Applet 程序（严格地说，是编译后的字节码）下载到用户的计算机中，然后调用用户计算机的 JVM 执行该程序，并最终在用户的 Web 浏览器窗口中显示执行结果。Java Applet 的安全性比 ActiveX 容易控制，但是同样存在缺点，它的执行速度通常比较慢。Java Applet 启动前网页中会出现一块灰色的区域，总是让人感觉不怎么愉快。

8. CGI(Common Gateway Interface)

前面曾经提到 CGI，一般将其翻译成公共网关接口。CGI 实际上是一种标准，它提供了一种使 Web 服务器可编程扩展的方法。这种扩展程序通常被称作 CGI 程序。CGI 代码可以给 Web 服务器增加了许多新功能，从而使之具有几乎无限的灵活性。从理论上讲，CGI 程序可以用任何一种语言编写，比如 VB、C、Pascal、Dephi 等。但实际应用中，由于 Perl 语言在字符串处理方面具有独特优势，因此成为 CGI 编程中最流行的语言。因此，大多数时候人们一说到 CGI 程序都是指用 Perl 编写的网关程序。

CGI 程序的运行原理是：Web 浏览器请求 Web 服务器执行某一 CGI 程序，Web 服务器启动一个进程，调用 CGI 解释程序（如 Perl 解释器）执行该程序，然后把执行结果返回给 Web 浏览器。CGI 程序的运行原理也就决定了 CGI 程序的缺陷：Web 服务器将为每个 CGI 程序请求启动一个进程，当存在大量用户请求时，Web 服务器性能就会显著下降。

9. ISAPI (Internet Server Application Programming Interface)

ISAPI 即 Internet 服务器程序接口，虽然微软公司的 Internet Information Server(IIS)完全支持这一标准，但是 ISAPI 并不是特别为微软公司开发的。它是一个通过使用 DLL 允许你扩展服务器功能的通用标准。ISAPI 的提出主要是为了克服 CGI 的不足。ISAPI 程序与 CGI 程序的主要区别在于：ISAPI 程序一旦被载入，就会一直运行在 Web 服务器的内存空间中，直到 Web 服务器退出服务。这就避免了使用 CGI 时不断调用 CGI 程序而带来的负载问题。ISAPI 的优越性在服务器繁忙时显得特别突出，有资料表明，在一个繁忙的服务器上，ISAPI 程序在性能上比与其功能类似的 CGI 代码高出 4~5 倍。当然，ISAPI 也有自己的局限性。由于 ISAPI 程序运行在服务器内存空间中，所以 ISAPI 程序的运行错误可能导致 Web 服务器的崩