

高等学校计算机应用规划教材

网站设计与 Web 应用开发技术 (第三版)

吴伟敏 编著

清华大学出版社

北 京

内 容 简 介

本书从 Web 基本概念和网站的规划设计及建设方法入手,着重介绍 HTML、CSS、JavaScript 和服务器端开发技术的基本原理和开发方法,并对将来网站开发领域的方向进行了描述。全书内容在编排上由浅入深,并辅以大量的实例进行说明。全书共分为 7 章,内容包括 WWW 简介、网站设计与网站运行环境配置、HTTP 协议与 HTML 语言、层叠样式表(CSS)、JavaScript 语言与客户端开发、服务器端开发——动态网页技术基础和 Web 的未来。

本书内容丰富,结构清晰,具有很强的实用性,既可作为高等院校学习网站设计及 Web 开发技术的教材,也可作为 Web 开发人员及自学者的参考用书。

本书配套的电子课件、习题答案和实例源文件可以通过 <http://www.tupwk.com.cn/downpage> 网站下载,也可以扫描前言中的二维码进行下载。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

网站设计与 Web 应用开发技术 / 吴伟敏 编著. —3 版. —北京:清华大学出版社, 2020.2
高等学校计算机应用规划教材
ISBN 978-7-302-54924-6

I. ①网… II. ①吴… III. ①网站—设计—高等学校—教材②网页制作工具—程序设计—高等学校—教材 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2020)第 025221 号

责任编辑:胡辰浩

装帧设计:孔祥峰

责任校对:成凤进

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:25 字 数:640 千字

版 次:2009 年 1 月第 1 版 2020 年 4 月第 3 版 印 次:2020 年 4 月第 1 次印刷

印 数:1~3000

定 价:76.00 元

产品编号:079726-01

前 言

没有哪一项技术能和今天的互联网技术一样发展迅速,它对人们工作和生活的影响面之广、影响程度之深,使得人们不能不重视它。在长期关于网站开发的教学生涯中,笔者注意到虽然很多人希望通过学习掌握开发技术,但由于没有建立正确的见解和思考的方法,部分人出现了事倍功半的学习结果,乃至无法胜任或完成开发任务而最终不得不放弃。笔者通过观察和分析,得出以下几个观点,希望读者能够了解和思考。

1. 对于计算机及其相关技术发展的思考

由于技术的发展基于功能越来越完善的平台,因此其发展水平体现了提升速度呈指数级增加的特征。在这个新思想、新技术以小时为单位而迅速更新的年代,对希望学习信息技术,特别是网站开发技术的开发者提出了极高的要求。因为学习者所面临的是今天所学的技术,可能今后不再使用,而真正需要学习的技术今天还没有出现的现状,对此问题的深入思考一定有助于读者更好地理解该学什么和该怎么学。如果能透过纷乱的现象看清开发工作中所存在的问题,从更深的层次把握开发技术的本质,就一定能更好地掌握技术的实质,能更好地适应将来的变化并能满足不断提升的要求。

2. 对于学习方法的思考

网上有大量关于 Web 应用开发的文档,如 HTML、CSS、JavaScript、服务器端开发语言等,这些知识非常容易获取和查询,但是否获得了这些文档就能成为优秀的网站开发者呢?答案是:不一定。虽然在有关文档中所列出的用法是固定的,但据此而进行的拓展往往是无穷的,有经验的开发者可以灵活实现,充分发挥其功能。所谓的“经验”是从哪里获取的呢?其实有经验的人也经历过没有经验的阶段,因而如何快速跨越获取和累积“经验”的鸿沟,是一个非常值得思考的问题。

基于上面的思考,在本书中将介绍 Web 的发展历史、工作原理、开发框架、网站策划设计、网站运行环境构建、HTTP 协议、HTML 语言、层叠样式表(CSS)、CSS 滤镜应用、CSS3 开发、JavaScript 开发、服务器端开发技术基础、XML 技术、WebAssembly、移动开发和混合开发模式等内容。希望这样的内容安排能为大多数希望学习和掌握 Web 技术的读者有所帮助,使他们能够更好地了解网站及其相关技术的走向和本质。对于一个初学者,本书能引导其快速入门并迅速成为合格的开发者;对于初级开发人员,本书可以答疑解惑,提供开发的总体框架和思路,拓展问题的实现手段和方法。

由于本书旨在为读者今后学习和开发高级网站打下良好的基础,因此为了更好地掌握本书所介绍的知识,读者最好已熟练掌握了至少一门编程语言。

完整地学习 Web 技术需要具备 3 个层面的知识。本书据此设计了 3 个层次：Web 基本概念及网站基础、Web 开发基础和 Web 高级应用。本书的知识体系结构如图 1 所示，将按照循序渐进的原则，逐步引领读者从基础到各个知识点进行学习，为今后的深入学习奠定基础。

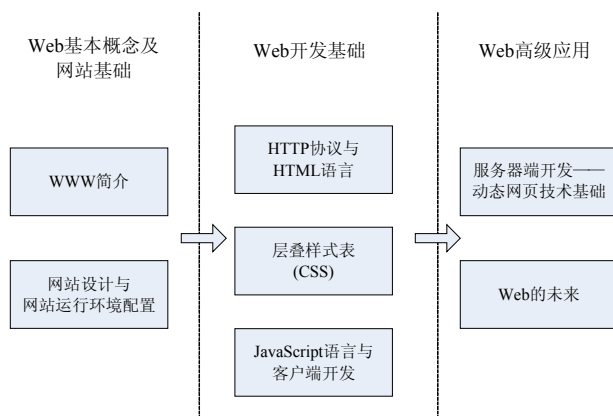


图 1 本书的知识体系结构

概括起来，本书具有以下主要特点。

- 结构清晰、内容详实。在每一章的开头都概要说明了本章所介绍的内容，使读者能快速了解本章的要点；介绍每一个知识点时，会辅以实例，并说明此实例的功能、运行的方式，然后给出执行的结果；在各章的最后都有对应的小结，总结本章介绍的内容，前后呼应，系统性较强。
- 强调实用性，突出基本原理和方法。为了让读者打下坚实的基础，学会掌握不断涌现的新技术，本书采用了将网站设计思想与网页制作技术相结合的理念，让读者学会从全局的角度出发来考虑和解决当前问题，并通过所掌握的学习方法能解决未来实际工作中遇到的问题。全书按照 Web 开发的方法与顺序，从基本概念和策划设计入手，循序渐进地介绍了进行 Web 开发的步骤、技巧，并在各章配有精心选择的应用实例。这些实例既有较强的代表性和实用性，又能够综合应用所介绍的知识，使读者能够全面、准确地掌握 Web 开发的全过程，并启发读者达到举一反三的目的。
- 每一章最后都附有思考和练习。这些习题紧扣该章介绍的内容。通过思考和练习能使读者更好地掌握本章所讲解的基本概念，提高读者的学习效果和开发技能。

本书共分为 7 章，内容包括 WWW 简介、网站设计与网站运行环境配置、HTTP 协议与 HTML 语言、层叠样式表(CSS)、JavaScript 语言与客户端开发、服务器端开发——动态网页技术基础和 Web 的未来。

第 1 章“WWW 简介”，介绍 Internet 与 WWW 的发展历程、网站相关的基本概念及其开发技术以及 Web 的基本框架。第 2 章“网站设计与网站运行环境配置”，说明在网站建立之前做好策划工作的必要性，并给出了一些基本原则；为了让网站正常运行，需要在正式开发前做好详细的设计工作；本章还介绍建立网站开发和运行基本环境的方法。第 3 章“HTTP 协议与 HTML 语言”，介绍 HTTP 的基本概念及运行原理、HTML 文档的构成和常用元素的基本用法，网站交互的设计和实现思路，HTML 高级特性和使用方法。第 4 章“层叠样式表(CSS)”，介绍 CSS 的基本用法、滤镜的使用以及 CSS3 的基本用法。第 5 章“JavaScript 语言与客户端开发”，

介绍 JavaScript 脚本语言的基本概念、基本语法、常用对象和网页特效的制作方法。第 6 章“服务器端开发——动态网页技术基础”，介绍服务器端开发的几种典型方法、动态网页的基本原理以及不同实现技术的特点分析。第 7 章“Web 的未来”，简单介绍 XML、WebAssembly、移动开发和混合开发的基本特征。

有一定网络和网站基础知识的读者可跳过第 1 章的学习，具备网站设计、架设和管理经验的读者可跳过第 2 章的学习。

本书内容由浅入深，并注重读者学习和开发能力的培养，通过辅以大量的实例分析和说明，深入、详细地讲解网站设计与 Web 应用开发技术，因此本书既可作为各类高等院校学习网站设计及 Web 技术的教材，也可作为 Web 开发人员及自学者的参考用书。

本书除封面署名的作者外，南京邮电大学的潘慧、查飞琴和薛涛等参与了本书第 7 章的编写，在此深表感谢。此外，还要感谢负责全书校稿及编辑工作的江苏产业技术研究院的徐欣。

感谢笔者的好友夏兰、徐汝鉴，他们为本书的编写提出了许多指导性的意见；借此还要感谢吴革新、刘迪庐，他们也为本书的出版提供了很多宝贵的建议；另外，为本书编写提供帮助的还有吴殊同、吴晓谦等。正是因为这么多人的大力支持和倾情奉献，本书才得以顺利出版。

由于本书涉及的内容非常广泛，在深度和广度上很难做到完美，加之笔者水平有限，书中肯定存在错误和不足，敬请读者批评指正，我们的信箱是 huchenhao@263.net，电话是 010-62796045。

本书配套的电子课件、实例源文件和习题答案可以通过 <http://www.tupwk.com.cn/downpage> 网站下载，也可以扫描下面的二维码下载。



作 者
2019 年 10 月

目 录

第 1 章 WWW 简介	1	2.4 网站的建立——IIS的安装与配置	58
1.1 Internet与WWW	1	2.4.1 IIS的安装	59
1.1.1 Internet的发展	1	2.4.2 使用IIS建立站点	60
1.1.2 Internet技术基础	4	2.4.3 IIS的配置	62
1.1.3 Internet提供的服务	7	2.4.4 其他Web服务器	64
1.2 WWW概述	9	2.5 网站运行的基础——安全	65
1.2.1 WWW的起源	9	2.5.1 网站安全威胁	66
1.2.2 Web的实质	11	2.5.2 防范策略	67
1.2.3 Web的技术基础	12	2.6 网站开发过程	69
1.2.4 Web的高级技术	15	2.6.1 瀑布模型	69
1.2.5 WWW的发展	20	2.6.2 敏捷开发模型	70
1.3 Web应用开发的架构	23	2.7 网站评估	72
1.3.1 Web应用的需求	23	2.7.1 准备工作	73
1.3.2 应用发展的需求	25	2.7.2 数据分析	73
1.4 本章小结	30	2.7.3 小结	74
1.5 思考和练习	30	2.8 本章小结	74
第 2 章 网站设计与网站运行		2.9 思考和练习	74
环境配置	31	第 3 章 HTTP 协议与 HTML 语言	75
2.1 网站设计的总体流程	31	3.1 HTTP协议	75
2.2 网站建立的前期工作——		3.1.1 HTTP概述	75
网站策划	32	3.1.2 HTTP的宏观工作原理	79
2.3 网站的设计	35	3.1.3 HTTP协议基础	81
2.3.1 网站的CI形象设计	36	3.1.4 HTTP应用开发方法	86
2.3.2 网站的总体结构设计	38	3.1.5 HTTP应用的开发	89
2.3.3 网站的版面设计	40	3.1.6 安全超文本传输协议、安全套接层	
2.3.4 网页的色彩设计	45	及传输层协议	91
2.3.5 网站的导航设计	48	3.2 HTML基础	93
2.3.6 网站信息的可用性设计	50	3.2.1 HTML简介	93
2.3.7 网站的交互设计	51	3.2.2 HTML标记语法及文档结构	99

3.3 HTML的基本语法	110	4.6 CSS典型用法实例	204
3.3.1 标题和段落	110	4.6.1 边框的用法	204
3.3.2 列表	116	4.6.2 动画	205
3.3.3 超链接	120	4.6.3 语音应用	211
3.3.4 表格	126	4.6.4 制作可交互的360度全景展示	211
3.3.5 图像、音频、视频及嵌入元素	132	4.6.5 自动适应移动设备横竖屏显示方式的实现方案	214
3.3.6 iframe框架应用	139	4.7 本章小结	215
3.3.7 表单	141	4.8 思考和练习	215
3.3.8 canvas应用	148	第5章 JavaScript语言与客户端开发	216
3.4 本章小结	152	5.1 JavaScript简介	216
3.5 思考和练习	152	5.1.1 什么是JavaScript	217
第4章 层叠样式表(CSS)	153	5.1.2 JavaScript的作用	218
4.1 CSS概述	153	5.1.3 JavaScript语言的组成	219
4.2 将CSS引入网站	156	5.1.4 将JavaScript引入HTML文档的方式	220
4.2.1 CSS的定义	156	5.1.5 一个简单的实例	221
4.2.2 CSS的浏览器兼容性	158	5.1.6 JavaScript的版本与兼容性	222
4.2.3 不同层次的CSS定义	159	5.2 JavaScript基本语法	223
4.2.4 书写规范	161	5.2.1 JavaScript的语句	223
4.3 CSS选择器	162	5.2.2 数据类型	224
4.3.1 标签选择器	162	5.2.3 变量	225
4.3.2 类别选择器	163	5.2.4 运算符与表达式	226
4.3.3 ID选择器	165	5.2.5 功能语句	228
4.3.4 通用选择器	167	5.2.6 函数	232
4.3.5 后代选择器	168	5.3 对象化编程	235
4.3.6 交集选择器	168	5.3.1 对象的基本知识	235
4.3.7 并集选择器	169	5.3.2 事件处理	237
4.3.8 伪类选择器	170	5.3.3 JavaScript的内部对象	240
4.3.9 伪元素选择器	174	5.3.4 JavaScript的自定义类及对象	248
4.3.10 样式表的继承性与层叠性	176	5.4 浏览器对象模型与文档对象模型	253
4.3.11 对div+CSS方案的思考	177	5.4.1 navigator对象	255
4.4 CSS的布局及盒子模型	178	5.4.2 window对象	259
4.4.1 CSS的布局基础	179	5.4.3 screen对象	265
4.4.2 CSS的盒子模型	182	5.4.4 event对象	267
4.4.3 CSS布局	186	5.4.5 history对象	269
4.4.4 CSS布局技巧	193	5.4.6 location对象	270
4.5 CSS滤镜	198	5.4.7 document对象	271
4.5.1 CSS3滤镜的种类及定义方式	199	5.4.8 link对象	273
4.5.2 滤镜实例	201		

5.4.9 form对象	275	6.4.2 Django的特点	344
5.4.10 cookie对象	284	6.4.3 Django实例	345
5.5 JS开发框架技术	286	6.5 更多的服务器开发技术及其比较	348
5.5.1 框架技术简介	286	6.5.1 CGI	348
5.5.2 jQuery框架	290	6.5.2 ISAPI/NSAPI	350
5.5.3 Flex	294	6.5.3 PHP	350
5.5.4 框架开发实例	294	6.5.4 不同开发技术之间的比较	352
5.6 JavaScript实例	298	6.6 本章小结	354
5.6.1 document.write()的副作用	298	6.7 思考和练习	354
5.6.2 带动画效果的进度条	299	第7章 Web的未来	355
5.6.3 旋转变幻文字效果	300	7.1 Web的发展路径	355
5.6.4 指针式时钟的实现	302	7.2 XML技术	357
5.6.5 一个益智小游戏的实现	305	7.2.1 XML介绍	357
5.7 Ajax技术	310	7.2.2 XML的文档格式	359
5.7.1 Ajax介绍	310	7.2.3 XML相关技术介绍	362
5.7.2 Ajax应用与传统的Web应用的比较	311	7.2.4 XML的开发工具	366
5.8 本章小结	314	7.2.5 XML的使用前景	367
5.9 思考和练习	314	7.2.6 JSON	368
第6章 服务器端开发——动态网页		7.3 WebAssembly技术	368
技术基础	315	7.3.1 WebAssembly概述	369
6.1 动态网页基本原理	315	7.3.2 WebAssembly的历史	369
6.2 ASP及.NET技术	317	7.3.3 WebAssembly的运行原理	370
6.2.1 ASP	317	7.3.4 WebAssembly的应用	371
6.2.2 ASP.NET简介	320	7.3.5 WebAssembly的现状和发展趋势	372
6.2.3 .NET战略	320	7.4 移动开发与混合开发模式	372
6.2.4 ASP.NET应用的开发实例	322	7.4.1 移动开发简介	372
6.3 Java技术	330	7.4.2 移动应用开发的三种模式	375
6.3.1 Java技术概述	330	7.4.3 混合应用开发框架介绍	376
6.3.2 Applet与Application	332	7.5 本章小结	379
6.3.3 Servlet	334	7.6 思考和练习	379
6.3.4 JSP	336	参考文献	380
6.3.5 J2EE	338	附录 HTML5 代码规范	382
6.4 Python网站开发技术	340		
6.4.1 Python Web应用开发框架	340		

WWW简介

互联网在世界范围内的迅速崛起使得它已经成为一种应用最为广泛的大众媒体，其应用范围和参与人群都在急剧增长。日益增加的网上购物、各种网络系统和形形色色的网站已经改变了人们的日常工作、生活、娱乐等行为方式，这一切改变中最为重要的支撑技术就是 Web 技术。

本章旨在引导读者了解 Internet 与 WWW 的发展历程，熟悉 Web 的基本概念及其相关技术，了解开发、运行、调试本书示例程序的软硬件环境。本章还将简要介绍各种不同的 Web 开发方法。

本章要点：

- 理解 Internet 与网站技术的发展历程
- Web 的基本概念
- Web 技术基础及高级技术介绍
- Web 应用开发基础

1.1 Internet 与 WWW

1.1.1 Internet 的发展

诞生于 1946 年的世界上第一台计算机“埃尼阿克”(ENIAC)是一场计算技术的革命，数字信息时代也由此拉开了序幕。在之后的若干年中，计算机的处理能力基本按照每 18 个月就翻一番的规律发展，由于这个定律首先是由美国英特尔公司的戈登·摩尔提出并应用的，因此这个定律被称为“摩尔定律”。

早期的计算机是独立的，之后为了能在计算机之间方便地进行通信和共享资源，诞生了网络，由此宣告了网络时代的到来。Internet 最早来源于美国国防部高级研究计划署 DARPA(Defense Advanced Research Projects Agency)的前身 ARPA 建立的 ARPAnet，而 ARPAnet 则源于当时美国国防部为了保证美国国防力量在受到第一次核打击后仍能具有生存和反击能力而设计的分散指挥系统。该网于 1969 年投入使用，最初由加州大学、犹他大学和斯坦福研究院的 4 台计算机以分组交换的原理构成。从 20 世纪 60 年代开始，ARPA 就开始向美国国内大学的计算机系和一些私人有限公司提供经费，以促进基于分组交换技术的计算机网络的研究。1968 年，ARPA 为 ARPAnet 网络项目立项，这个项目基于这样一种主导思想：网络必须能够经受住故障的考验并维持正常工作，一旦发生战争，当网络的某一部分因遭受攻击而失去工作能力时，

网络的其他部分应当能够维持正常通信。最初, ARPAnet 主要用于军事研究目的, 它具有以下五大特点:

- 支持资源共享;
- 采用分布式控制技术;
- 采用分组交换技术;
- 使用通信控制处理机;
- 采用分层的网络通信协议。

1972年, ARPAnet 在首届计算机后台通信国际会议上首次与公众见面, 并验证了分组交换技术的可行性, 由此, ARPAnet 成为现代计算机网络诞生的标志。

ARPAnet 在技术上的另一个重大贡献是 TCP/IP 协议簇的开发和使用。1980年, ARPA 投资把 TCP/IP 加进 UNIX(BSD 4.1 版本)的内核中, 在 BSD 4.2 版本以后, TCP/IP 协议即成为 UNIX 操作系统的标准通信模块。1982年, Internet 由 ARPAnet、MILNET 等几个计算机网络合并而成。作为 Internet 的早期骨干网, ARPAnet 奠定了 Internet 存在和发展的基础, 较好地解决了异构环境下网络互联的一系列理论和技术问题。

1983年, ARPAnet 分裂为两部分: ARPAnet 和纯军事用的 MILNET。同年1月, ARPA 把 TCP/IP 协议作为 ARPAnet 的标准协议, 其后, 人们称呼这个以 ARPAnet 为主干网的网际互联网为 Internet。TCP/IP 协议簇在 Internet 中不断被研究、试验, 并改进成为使用方便、效率极好的协议簇。

与此同时, 局域网和其他广域网的产生和蓬勃发展对 Internet 的进一步发展起到了重要的作用。其中, 最引人注目的就是美国国家科学基金会(National Science Foundation, NSF)建立的美国国家科学基金网 NSFnet。1986年, NSF 建立了六大超级计算机中心, 为了使全国的科学家、工程师能够共享这些超级计算机设施, NSF 建立了自己的基于 TCP/IP 协议簇的计算机网络 NSFnet。NSF 在全国建立了按地区划分的计算机广域网, 并将这些地区网络和超级计算中心相连, 最后将各超级计算中心互联起来。地区网的构成一般是由一批在地理上局限于某一地域, 在管理上隶属某一机构或在经济上有共同利益的用户的计算机互联而成, 连接各地区网上主通信节点计算机的高速数据专线构成了 NSFnet 的主干网。这样, 当一个用户的计算机与某一地区相连以后, 它除了可以使用任一超级计算中心的设施, 可以同网上任一用户通信, 还可以获得网络提供的大量信息和数据。这一成功使得 NSFnet 于 1990年6月彻底取代了 ARPAnet 而成为 Internet 的主干网。

NSFnet 对 Internet 的最大贡献是使 Internet 向全社会开放, 而不再像以前那样仅仅为计算机研究人员、政府职员和政府承包商所使用。然而, 随着网上通信量的迅猛增长, NSF 不得不采用更新的网络技术来适应发展的需要。1990年9月, 由 Merit、IBM 和 MCI 公司联合建立了一个非营利性的组织——ANS。ANS 的目的是建立一个全美范围的 T3 级主干网, 它能以 45Mb/s 的速率传送数据, 相当于每秒传送 1400 页的文本信息。到 1991 年底, NSFnet 的全部主干网都已同 ANS 提供的 T3 级主干网相通。

1969年12月, 当 ARPAnet 最初建成时只有 4 个节点, 到 1972年3月也仅增加到 23 个节点, 直到 1977年3月总共也只有 111 个节点。但是近几十年来, 随着社会科技、文化和经济的发展, 特别是计算机网络技术和通信技术的大发展, 以及人类社会从工业社会向信息社会过渡的趋势越来越明显, 人们对信息的认识, 对开发和使用信息资源的重视越来越强烈, 这些都强烈刺

激了 ARPAnet 和后来的 NSFnet 的发展,使连入这两个网络的主机和用户数目急剧增加。1988 年,由 NSFnet 连接的计算机数就猛增到 56 000 台,此后每年以 2~3 倍的惊人速度向前发展;1994 年,Internet 上的主机数目达到了 320 万台,连接了世界上的 35 000 个计算机网络;2000 年,全球已有超过一亿名用户,而这个数字此后以每年 15%~20% 的速度递增。中国互联网络信息中心的数据显示,截至 2014 年 6 月,中国的互联网用户数已达 6.86 亿,中国是全球最大的互联网市场,而且未来这个数量还将以更快的速度增加。Internet 发展过程中的重要阶段如表 1-1 所示。

表 1-1 Internet 发展过程中的重要阶段

	1969 年	1982 年	1986 年	20 世纪 80 年代后期
网络名称	ARPAnet(美国国防部高级研究计划署网)	ARPAnet 与 MILNET 合并形成 Internet 雏形	NSFnet(国家科学基金网)取代 ARPAnet 成为 Internet 基础	Internet 形成并迅速发展

在 Internet 蓬勃发展的同时,其本身随着用户需求的转移也在不断发生着产品结构上的变化,现已成为全球重要的信息传播工具。我国于 1994 年 5 月正式接入 Internet,发展至今已 25 年多的时间。据 2019 年中国互联网络信息中心(CNNIC)在北京发布的《第 43 次中国互联网络发展状况统计调查》显示,截至 2018 年 12 月底,我国网民规模达 8.29 亿,全年共计新增网民 5653 万人,网络普及率达到 59.6%,增长率为 3.8%,如图 1-1 所示。



图 1-1 中国网民规模和互联网普及率

另外,值得注意的现象是:移动互联网接入流量自 2015 年以来连续三年实现翻番增长,2018 年移动互联网的接入流量消费累计达到 711.1 亿 GB,比上年同期累计增长 189.1%。我国手机网民规模达 8.17 亿,网民中使用手机上网人群的占比由 2016 年的 97.5% 提升至 98.6%;与此同时,使用电视上网的网民比例也提高了 2.9 个百分点,达 31.1%;台式电脑和笔记本电脑的使用率均出现下降,如图 1-2 所示。

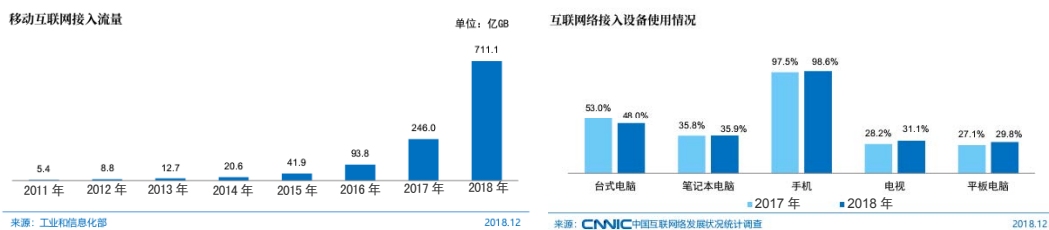


图 1-2 移动上网流量及上网所使用的设备统计数据

截至 2018 年 12 月,我国手机网络支付用户规模达 5.83 亿,年增长率为 10.7%,手机网民使用率达 71.4%。线下网络支付使用习惯持续巩固,网民在线下消费时使用手机网络支付的比例由 2017 年底的 65.5%提升至 67.2%。在跨境支付方面,支付宝和微信支付已分别在 40 个以上国家和地区合规接入。我国在基础资源、5G、量子信息、人工智能、云计算、大数据、区块链、虚拟现实、物联网标识、超级计算等领域发展势头向好。在 5G 领域,核心技术研发取得了突破性进展,政企合力推动产业稳步发展;在人工智能领域,科技创新能力得到加强,各地规划及政策相继颁布,有效地推动了人工智能与经济社会发展的深度融合;在云计算领域,我国政府高度重视以其为代表的新一代信息产业的发展,企业积极推动战略布局,云计算服务已逐渐被国内市场认可和接受。

在 Internet 上,按从事的业务分类包括了广告公司、航空公司、农业生产公司、艺术、导航设备、书店、化工、通信、计算机、咨询、娱乐、财贸、各类商店、旅馆等 100 多类,覆盖了社会生活的方方面面,构成了一个信息社会的缩影。由于越来越多计算机的加入,Internet 上的资源变得越来越丰富。从 2018 年的统计数据可以看出,Internet 已超出一般计算机网络的概念,它不仅是传输信息的媒体,而且已成为一个全球规模的信息服务系统。它是人类有史以来第一个真正的世界性的“信息仓库”,是一个全球范围的交流场所。人们的生活越来越离不开网络,其中起到核心作用的正是 Web 技术。

1.1.2 Internet 技术基础

1. TCP/IP

1972 年出现了网际互联的核心技术 TCP/IP 协议,该协议包括近 100 个协议,而其中最主要的是 TCP 协议和 IP 协议,其中 TCP(Transmission Control Protocol)是传输控制协议,它的作用是保证信息在网络间可靠地传送,保证接收到的信息在传输途中不被损坏;而 IP(Internet Protocol)是网际协议,保证信息从一个地方传送到另一个地方,不管中间要经过多少节点和不同的网络。TCP/IP 模型的网络协议如图 1-3 所示。

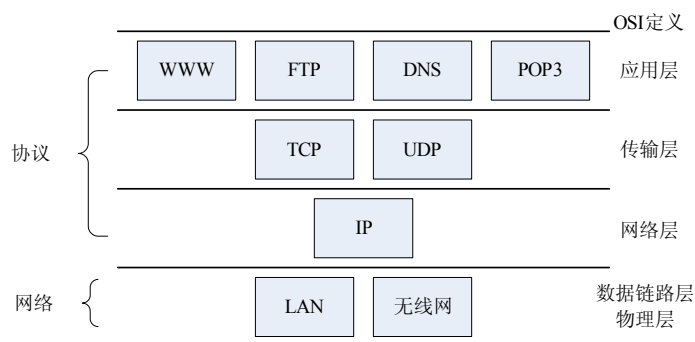


图 1-3 TCP/IP 模型的网络协议

IP 是 TCP/IP 体系结构中非常重要的协议,它是为计算机网络相互连接进行通信而设计的协议,该协议基于分组交换技术,包含如下规则:

- 目前 Internet 上采用 IPv4 协议的计算机都配置了一个由 4 个数字组成的 Internet 地址,每个数字不超过 256,如 202.96.101.201。

- 一个信息被划分成若干个分组。
- 每个分组被填入一个 IP 信封。
- IP 信封外包含一个发送地址和一个收信地址，再加一个顺序号。

在 Internet 上每台主机都有专门的地址，称为 IP 地址，只有有了地址，信息才可以正确送达。因此为了访问互联网中的计算机，必须有一种寻址方法来定位，IP 地址就成为互联网上的主机和路由器的标识方法，正如日常生活中发送纸质邮件需要地址一样。IP 地址是从左到右表示的，最左边部分识别网络中的最大部分，IP 是由管理 IP 地址的专门机构分配的，它包括网络号和主机号。这一编码组合是唯一的，没有两台有同一 IP 地址的计算机。

在互联网中，IP 协议是能使连接到网上的所有计算机网络实现相互通信的一套规则，它规定了计算机在互联网上进行通信时应当遵守的规则。任何厂家生产的计算机系统，只要遵守 IP 协议就可以与互联网互联互通。

2. IPv6

IPv6(Internet Protocol Version 6)是 IETF(Internet Engineering Task Force, 互联网工程任务组)设计的用于替代现行版本 IP 协议(IPv4)的下一代 IP 协议。

当前所使用的第二代互联网 IPv4 技术，其核心技术属于美国。它的最大问题是网络地址资源有限。从理论上讲，IPv4 可以实现为 1600 万个网络和共计 40 亿台主机编址。但采用 A、B、C 三类编址方式后，可用的网络地址和主机地址的数目大打折扣。实际上，IPv4 的地址已于 2011 年 2 月 3 日分配完毕，其中北美占有 3/4，约 30 亿个，而人口最多的亚洲只有不到 4 亿个。

一方面是地址资源数量的限制，另一方面是随着电子技术及网络技术的发展，“万物互联”时代的到来将可能使人们身边的每一样东西都连入互联网。在这种需求的推动下，IPv6 应运而生。单从数量级上来说，IPv6 所拥有的地址容量约是 IPv4 的 8×10^{28} 倍，达到 2^{128} (包括地址为零的和全 1 的)个。这不但解决了网络地址资源数量的问题，同时也为物联网的推进在 IP 地址不足的问题上扫清了障碍。

由于 Internet 的规模以及网络中数量庞大的 IPv4 用户和设备，IPv4 到 IPv6 的过渡不可能一次性实现。而且，许多企业和用户的日常工作越来越依赖于 Internet，它们无法容忍在协议过渡过程中出现的问题。所以 IPv4 到 IPv6 的过渡必须是一个循序渐进的过程，在体验 IPv6 带来的好处的同时仍能兼容网络中原先使用 IPv4 的设备。实际上，IPv6 在设计的过程中就已经考虑到了 IPv4 到 IPv6 的过渡问题。中国 IPv6 地址数量如图 1-4 所示，在 2018 年呈现了加速发展的趋势。

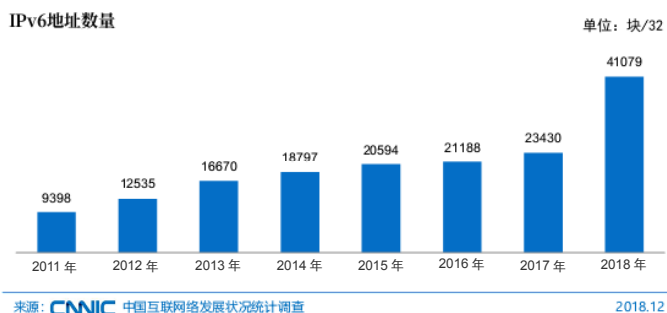


图 1-4 中国 IPv6 地址数量(截至 2018 年底)

3. 域名系统

如果上网就必须使用 IP 地址,这将是非常痛苦的。值得庆幸的是,作为一个 World Wide Web 用户,实际上并不需要对 IP 地址有很深的了解,也不需要记住很多枯燥的 IP 地址,这归因于一种 Internet 上的计算机的命名方案,我们称之为域名系统(Domain Name System, DNS)。它可以将形如 www.njupt.edu.cn 的域名与其所对应的 IP 地址进行对应和转换。因此,用户就可以使用域名来取代 IP 地址了。在语法上,每台计算机的域名由一系列字母和数字构成的段组成。例如,某个服务器的域名为 www.njupt.edu.cn,其中, cn 代表中国, edu 代表教育部门, njupt 代表南京邮电大学, www 代表 WWW 服务。

DNS 是一个分布式的数据库,利用它能进行域名的解析,一般存放于 DNS 服务器上,为了定义 Internet 上的主机而提供的一个层次性的命名系统,如图 1-5 所示。

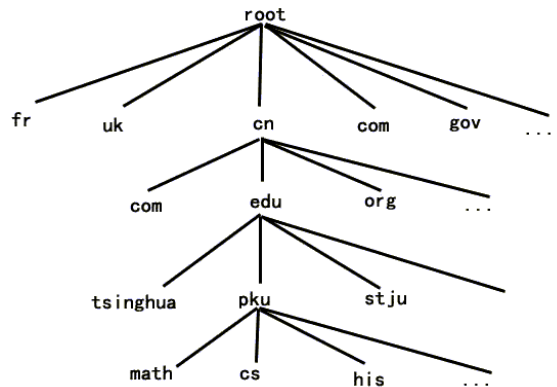


图 1-5 Internet 域名空间

域名的解析过程如下。

- DNS 客户向本地的 DNS 服务器发出查询请求。
- 如果该 DNS 本身具有客户想要查询的数据,则直接返回给客户;如果没有,则该服务器和其他命名服务器联系,从其他服务器上获取信息,然后返回给用户。

各种域名扩展名是有意义的, DNS 常见的扩展名及其含义如表 1-2 所示。

表 1-2 常见域名系统的含义

域名扩展名	含 义
edu	教育及学术单位
com	公司或商业组织
gov	政府单位
mil	军事单位
org	基金会等非官方单位
net	网络管理服务机构
int	国际性组织
apra	APRAnet Internet 的起源
国别名(国家/及地区代码)	依据 ISO 标准定义,例如, cn 代表中国

1.1.3 Internet 提供的服务

Internet 的飞速发展和广泛应用得益于它所提供的多种服务, 这些服务为人们的信息交流带来了极大的便利, 下面介绍 Internet 所提供的几种主要服务。

1. WWW 服务

WWW 是环球信息网的缩写(亦作“Web”“WWW”或“W3”, 英文全称为“World Wide Web”), 中文名为“万维网”“环球网”等, 常被简称为 Web。它是一种基于超文本的信息查询方式, 由欧洲粒子物理研究中心(CERN)研制。可将 Internet 上不同来源的信息有机地组织在一起, 使用这项服务时可利用已开发的具有友好用户界面的浏览器, 方便信息的浏览; 利用 WWW 服务还可以提供更多的功能, 如 Telnet、FTP、Gopher、News、E-mail 等; WWW 还可以实现诸如搜索引擎、网络新闻、博客、网络视频、网络游戏、微博、社交网站、网络购物、网上银行、论坛、Web 邮件、网上支付、网上炒股等多项延伸服务。

2. 文件传输服务(FTP)

FTP 服务解决了远程传输文件的问题, 无论两台计算机相距多远, 只要它们都连入 Internet 并且都支持 FTP 协议, 则这两台计算机之间就可以进行文件的传送。FTP 实质上是一种实时的联机服务, 用户首先要登录到目标服务器上, 之后可以在服务器目录下寻找所需的文件, FTP 几乎可以传送任何类型的文件, 如文本文件、二进制文件、图像文件、声音文件等。一般的 FTP 服务器都支持匿名登录, 用户在登录到这些服务器时无须事先注册用户名和口令, 只要以 anonymous 为用户名和合法的 E-mail 地址作为口令就可以访问 FTP 服务。

3. 电子邮件服务(E-mail)

E-mail 是 Internet 上使用最广泛和最受欢迎的服务, 它是网络用户之间进行快速、简便、可靠且低成本联络的现代通信手段。电子邮件使网络用户能够发送和接收文字、图像和语音等多种形式的信息。使用电子邮件的前提是拥有自己的电子信箱, 即 E-mail 地址, 实际上是在邮件服务器上建立一个用于存储邮件的磁盘空间。电子邮件地址的典型格式为 username@mailserver.com, 其中 mailserver.com 部分代表邮件服务器的域名, username 代表用户名, 符号@读作“at”, 意为“在”。例如某 E-mail 地址为 master@njupt.edu.cn, 其含义表示为在计算机 njupt.edu.cn 上用户名为 master 的电子邮件地址。利用电子邮件可以获得其他各种服务(如 FTP、Gopher、Archie、WAIS 等)。当用户希望从这些信息中心查询资料时, 只需要向其指定的电子信箱发一封含有一系列信息查询命令的电子邮件, 该邮件服务器程序将自动读取、分析该邮件中的命令, 若无错误则将检索结果通过邮件方式发给用户。

4. 视音频业务

基于 Internet 的语音传输是利用基于 IP 数据网进行的语音传输。语音(模拟信号)首先由数字信号处理器(DSP)将其转换为数字信号, 然后, 数字信号被压缩成更便于网络传输的数据包, 之后, 通过 Internet 将数据包传送到目的地, 在目的地以相反的过程解压缩、解包、数/模转换, 送达对方话筒。由于 Internet 中采用“存储转发”的方式传递数据包, 并不独占电路, 并且对

语音信号进行了大比例的压缩处理,因此,IP电话占用带宽仅为8~10kb/s,还不到模拟电话所需带宽的1/8,再加上Internet上数据传输的计费方式与距离的远近无关,这大大降低了语音通信的费用。

基于数字视频通信的会议电视已经发展了多年,在视频点播、远程教育、视频监控、视频会议、视频直播方面有了广泛的应用。由于Internet的无连接数据包转发机制主要为突发性的数据传输而设计,不适用于对连续媒体流的传输,因此为了在Internet上有效、高质量地传输视频流,需要多种技术的支持,主要包括视频的压缩、编码技术,应用层质量控制技术,连续媒体分布服务技术,媒体同步技术和数字版权管理技术、组播等。

近年流行的视频直播业务,一般采用组播的网络方式来实现。所谓组播就是利用一种协议将IP数据包从一个信息源传送到多个目的地,将信息的拷贝发送到一组地址,送达所有想要接收它的接收者。IP组播是将IP数据包“尽最大努力”传输到一个构成组播群组的主机集合,群组的各个成员可以分布于各个独立的物理网络上。IP组播群组中成员的关系是动态的,主机可以随时加入和退出群组,群组的成员关系决定了主机是否接收送给该群组的组播数据包,不是某群组的成员主机也能向该群组发送组播数据包。同单播或广播相比,组播效率非常高,因为任何给定的链路至多用一次,可以节省网络带宽和资源。其技术实现过程为:首先用户发出直播请求,服务器根据直播信息,将该直播频道的播放地址(一般是一个组播URL,而非组播文件)传递给用户,然后用户根据该地址加入对应的组播组,即可接收视频直播内容。

5. 电子商务

电子商务是指利用计算机网络进行的商务活动,它将顾客、销售商、供货商和雇员联系在一起,实现商务活动的电子化、网络化、自动化。在互联网开放的网络环境下,买卖双方在任何可连接网络的地点间进行各种商务活动,实现两个或多个交易者间的生产资料交换及所衍生出来的交易过程、金融活动和相关的综合服务活动的一种商业运营模式。

在Internet开放的网络环境下,基于浏览器/服务器应用方式,买卖双方可以不谋面地进行各种商贸活动,实现消费者的网上购物、商户之间的网上交易、在线电子支付以及各种商务活动、交易活动、金融活动和相关的综合服务活动。各国政府、学者、企业界人士根据自己所处的地位和对电子商务参与的角度和程度的不同,给出了许多不同的定义。电子商务分为ABC、B2B、B2C、C2C、B2M、M2C、B2A(即B2G)、C2A(即C2G)、O2O等。

6. 对等网服务(P2P)

P2P是英文Peer-to-Peer(对等)的简称,有时也被称为“点对点”。“对等”技术是一种网络新技术,它依赖于网络服务使用者的计算能力和带宽,而不是依赖于有限的几台服务器。目前该方法在加强网络交流、文件交换、分布式计算等方面大有前途。

简单而言,P2P在网络客户之间直接建立联系,使得网络上的沟通变得容易,共享和交互变得更直接,真正地消除或减少了中间商。其另一个重要特点是改变互联网现在以服务器为中心的状态,实现“非中心化”,并把控制权交还给用户。P2P看起来似乎很新,但是正如B2C、B2B是将现实世界中很平常的东西移植到互联网上一样,P2P并不是什么新东西。在现实生活中存在大量P2P模式的面对面或者通过其他方式的交流和沟通。

即使从网络的角度来看,P2P也不是新概念,P2P是互联网整体架构的基础。互联网最基

本的协议 TCP/IP 中并没有客户机和服务器的概念，所有的设备在通信中都是平等的。在十年之前，互联网上的所有系统都同时具有服务器和客户机的功能。当然，后来发展的那些架构在 TCP/IP 之上的服务的确采用了客户机/服务器的结构，如浏览器和 Web 服务器、邮件客户端和邮件服务器等。但对于服务器来说，它们之间仍然是对等联网的。以 E-mail 为例，互联网上并没有一个巨大的、唯一的邮件服务器来处理所有的 E-mail，而是对等联网的邮件服务器相互协作把 E-mail 传送到相应的服务器上去。

事实上，网络上现有的许多服务都可以归入 P2P 的行列。即时讯息系统，如 ICQ、AOL Instant Messenger、Yahoo Pager、微软的 MSN Messenger 以及国内的 QQ 和微信等；下载工具，如 BitTorrent、BitSpirit、eMule(电驴)、PP 点点通、卡盟、迅雷等；大量的视频传输工具等都是流行的 P2P 应用。它们允许用户互相沟通和交换信息、交换文件，甚至于实现远程协助等复杂应用。

P2P 网络的一个重要的目标就是让所有的客户机都能提供资源，包括带宽、存储空间和计算能力。因此，当有大量节点接入时，很容易超出系统的设计容量，这是服务容量固定的客户机/服务器结构所不能承受的，此时客户机数量的增加就意味着服务质量的下降。而 P2P 网络的分布特性通过在多节点上复制数据，也增加了服务的健壮性。而且在纯 P2P 网络中，节点不需要依靠一个中心服务器来提供服务，此时系统也不容易出现单点崩溃。

在具有上述优点的同时，P2P 技术也有流量大、占用大量网络带宽的缺点，但以下技术可以使这个问题在一定程度上得到缓解。

- P4P(Proactive network Provider Participation for P2P)技术，这是 P2P 技术的升级版，目的是为了加强 ISP 与客户端程序的通信，降低骨干网的数据传输压力，并提高文件传输的性能。P4P 与 P2P 最大的不同在于它可以有针对性地选择传输节点，而不是像 P2P 那样，随机选择。这样就可以把 P2P 节点的传输区域控制在某个范围，可以最大限度地解决大型节点和网络出口负载，从而缓解骨干网的拥堵。
- PCDN 技术。这项技术在 CDN 节点的边缘构建了基于用户的 P2P 自治域，通过集中的分布式架构将 P2P 的流量严格限制在同一边缘节点的区域内。这项技术的原理与 P4P 技术非常相似，即通过控制 P2P 流量传输的范围来降低其对骨干网的压力。
- P2P 服务器模式。即把服务器而不是客户机当成 CDN 网络的节点，达到 CDN 网络优化和加速的目的。服务器之间实现 P2P 连接后，就不用再到中心节点的存储上寻找内容，从而提高了网络传输的效率。

1.2 WWW 概述

1.2.1 WWW 的起源

Web 源于欧洲粒子物理研究所(CERN)的 Tim Berners-Lee 于 1989 年提出的链接文档构想，由日内瓦粒子物理实验室研发。后来它在 TCP/IP、MIME、Hypertext 等技术之上进一步发展，并形成了 HTTP(HyperText Transfer Protocol)、HTML(HyperText Markup Language)、URL(Uniform Resource Location)等多项新技术。

什么是 Web? 它是 World Wide Web 的简称, Web 的本意是蜘蛛网, 有时被称为网页, 中文译为“万维网”, 现广泛译为“网络”和“互联网”等。实际上, Web 是运行在 Internet 上的所有 HTTP 服务器软件和它们所管理的对象的集合, 包括 Web 页面/Web 文档和程序。由于 Web 技术涉及的面很广, 因此为了能有一个比较清楚的认识, 在此首先对 Web 的历史进行简单介绍。

Web 现在变得越来越复杂, 但刚开始时一切却非常简单。最初为了连接几个顶尖研究机构, 美国设计了最早的“Internet”, 以便共同开展科学研究。不论是图书馆管理员、原子能物理学家, 还是计算机科学家, 都必须学习相当复杂的系统。1962 年, 麻省理工学院(MIT)的 J. C. R. Licklider 首先提出了他的“Galactic Network”(超大网络)思想——设想了全球计算机互联的一系列概念, 其中的资源和信息能够在任何站点上被处理。这个简单的设想经过多年的发展和努力, 最终形成了现在的 Web。

最初, 研究人员认为传输控制协议(Transmission Control Protocol, TCP)只适用于大型系统, 因为 TCP 就是为大型系统设计的。不过, 麻省理工学院 David Clark 的研究小组却发现, 这个协议也可以在工作站之间实现大面积的互联。Clark 的这项研究为 Web 的发展解决了底层网络通信的问题, 为 Web 的流行奠定了基础。

如前所述, 随着主机数量的快速增加, 去记忆数量众多且毫无意义的数字地址编号就非常困难了, 人们开始设想为主机指定有意义的名字来改善上述问题, 这就是域名系统(Domain Name System, DNS)。另外, ARPAnet 决定从使用网络控制协议(Network Control Protocol, NCP)变为使用 TCP/IP(Transmission Control Protocol/Internet Protocol, 传输控制协议/Internet 协议), 而 TCP/IP 是军方使用的标准协议。

到了 20 世纪 80 年代中期, Internet 已经实际成为一个连接不同研究人员的平台, 并且其他网络也开始出现: 如美国国家航空航天局(National Aeronautics and Space Administration)创建了 SPAN、美国能源部(U.S. Department of Energy)建立了 MFENet 等。1980 年欧洲粒子物理研究所(European Organization for Nuclear Research, CERN)的 Tim Berners-Lee 负责了 Enquire(Enquire Within Upon Everything)项目。1989 年, Tim Berners-Lee 提出了一个很有意思的概念: 他认为, 与其简单地引用其他人的工作, 为什么不干脆直接链接过去呢?例如在读一篇文章时, 读者可以直接单击打开所引用的文章。

超文本当时相当流行, 它利用了之前在文档和文本处理方面的研究成果。Berners-Lee 发明了标准通用标记语言(Standard Generalized Markup Language, SGML)的一个子集, 它被称为超文本标记语言(HyperText Markup Language, HTML)。HTML 的妙处在于, 它能把应该如何展现文本与具体实现显示的方法相分离。Berners-Lee 不仅创建一个称为超文本传输协议(HyperText Transfer Protocol, HTTP)的简单协议, 还同时开发了第一个 Web 浏览器, 该浏览器名为 World Wide Web。1990 年 11 月, 第一台 Web 服务器 nxoc01.cern.ch 开始运行, Tim Berners-Lee 在自己编写的图形化 Web 浏览器“World Wide Web”上看到了最早的 Web 页面。1991 年, CERN 正式发布了 Web 技术标准。目前, 与 Web 相关的各种技术标准都是由著名的 W3C 组织(World Wide Web Consortium)管理和维护的。

注意:

W3C 是英文 World Wide Web Consortium 的缩写, 中文意思是 W3C 理事会或万维网联盟。

W3C 于 1994 年 10 月在麻省理工学院计算机科学实验室成立。创建者是万维网的发明者 Tim Berners-Lee。该组织是对网络标准进行制定的一个非营利性组织，像本书后面章节中将要介绍的 HTML、XHTML、CSS、XML 等的标准都是由 W3C 制定的。W3C 会员(大约 500 名会员)包括生产技术产品及服务的厂商、内容供应商、团体用户、研究实验室、标准制定机构和政府部门，他们共同协作，致力于在万维网发展方向上达成共识。

1.2.2 Web 的实质

自 Web 诞生之日起，人们就没有给它下过一个精确的定义，但是我们可以通过以下方式来理解它。首先，Internet 是一个网络的网络，也可以说是一个全球范围的网中网。它由成千上万的计算机共同组成，它们各自扮演不同的角色，但总的来看可以分为客户机和服务器。客户机就是我们通常所使用的计算机；而服务器是一种高性能计算机，作为网络的节点，用于存储、处理网络上大量的数据和信息，因此也被称为网络的灵魂。此外，现在流行的所谓云，实际上可以认为是服务器的集合，其所提供的服务，则包括邮件服务、文件服务、DNS 服务、Web 服务和计算资源服务等。

Web 应用是 Internet 所提供的众多应用中的一种，其作用是将本地的信息以超文本的方式组织起来，方便用户在 Internet 上搜索和浏览，并能提供一定的交互。因此 Web 或者是 WWW 服务，实际上是由 Internet 中被称为 Web 服务器的计算机所提供的，从这个意义上来看，可以将 Web 应用看成是 Internet 应用的一个子集，如图 1-6 所示。

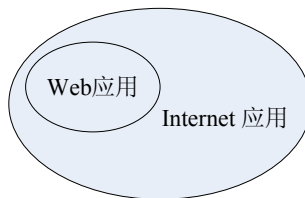


图 1-6 Internet 和 Web 的包含关系

注意：

Internet 是 Web 的基础平台，Web 是 Internet 平台上的一种应用或服务，它使人们能方便、快捷地发布和获取信息。至于这些信息是如何在 Internet 的网络层上进行传输的，对于一般的 Web 用户而言是透明的。

在 Web 出现初期，人们各自建立网页、互相建立链接，用户是沿着链接浏览的，这是真正的“网”。但是当 Yahoo 和 Lycos 等网站建立了搜索引擎和门户站点后，用户上网的方式就被改变了，在一个节点上可以获取几乎所有的信息。由此出现了所谓的“目标站点”模式，当人们逐条阅读内容时，还存在一个“网”的概念吗？而这些站点在起到积极作用的同时，也控制了信息的流动并包含了过时的信息，有时还包含一些广告。

而基于 P2P 应用的出现，则把控制权重新交还给用户。人们共享硬盘上的文件、目录甚至整个硬盘。所有人都共享了他们认为最有价值的东西，这将使互联网上信息的价值得到极大的提升。

而博客乃至微博以及社交媒体等的流行并与移动终端相结合的现实，则最大限度地将网络的应用大幅度延伸到人们日常生活的每个角落，并通过为所有用户提供这种控制权，使得内容发布的方式得到了极大的改变。

1.2.3 Web 的技术基础

从技术层面上来看, Web 架构的精华主要有 3 点: 用统一资源定位技术(URL)实现全球资源的精确定位; 用应用层协议(HTTP)实现分布式的信息传送; 以超文本技术(HTML)实现信息的表示。这 3 个特点无一不与信息的分发、获取和利用有关。其实, Tim Berners-Lee 早就明确无误地告诉我们: “Web 是一个抽象的(假想的)信息空间。”也就是说, 作为 Internet 上的一种应用架构, Web 的首要任务就是向人们提供信息和信息服务。

很可惜, 在 Web 应用日新月异的今天, 许多技术开发人员似乎已经忘记了 Web 架构的设计初衷。他们在自己开发的网站或 Web 应用中大肆堆砌各种所谓的“先进”技术, 但最终用户能够在这些网站或应用中获得的有价值的信息却寥寥无几。这个问题绝不像评论者常说的“有路无车”或“信息匮乏”那么简单。一个 Web 开发人员倘若忘记了 Web 技术的最终目标是提供信息和信息服务, 他的愚蠢程度就丝毫不亚于一个在足球场上只知道卖弄技巧, 却忘记了射门得分的大牌球星。从这个角度来说, 评价一种 Web 开发技术优劣的标准只有一个, 那就是看这种技术能否在最恰当的时间和最恰当的地点, 以最恰当的方式, 为最需要信息的人提供最恰当的信息服务。

Web 技术利用了一种称为超文本的技术, 即它使用了文件中突出显示的词句或图形生成链接来指向其他文件、图形、声音等资源。它可以从一个文件中的任何一点指向另一个信息资源, 从而可以实现快速的信息浏览。同时超文本技术具有良好的图形用户界面, 使得用户能很容易地浏览互联网中的信息。

注意:

Web 正是通过各种技术来实现其功能的, 这些技术无论是现在已有的还是将来即将出现的, 它们都共同构成了 Web 的技术基础。无论这些技术多么复杂、功能多么强大, 都可以将之囊括到资源的定位、传输和表示方面。

Web 技术中其实还包括其他更多的技术, 这里介绍其中最主要的 3 个。

1. 统一资源定位技术

统一资源定位符(Uniform Resource Locator, URL)通过定义资源位置的抽象标识来定位网络资源。资源被定位后, 便可对其进行各种操作, 例如, 访问、更新、替换、查找属性等。

总体来说, URL 可按下列格式进行书写:

```
<scheme>:<scheme-specific-part>
```

其中, <scheme>指所用的 URL 方案名。<scheme-specific-part>意义的解释与所用方案有关。方案名由字符组成, 可包括字母(a~z)、数字(0~9)、加号(+)、句点(.)和连词符(-), 字母大小写是不分的。

对于 Internet, <scheme>指的是 Internet 协议名, 可包括 http、ftp、gopher、mailto、new、nntp、telnet、wais、file 等, 这个列表以后还会不断扩充。

HTTP URL 方案用于表示可通过 HTTP 协议进行访问的 Internet 资源。HTTP URL 的格式如下:

```
http://<host>[:<port>]/<path>?<searchpart>
```

其中, <host>和<port>为标准格式, :<port>如果省略, 则默认端口值为 80。<path>为 HTTP

选择器, 而<searchpart>为查询字符串, 它们都是可选的, 如果这两项不存在, 则主机或端口后的斜杠也应该省略。例如: `http://www.edu.cn:80/index.aspx`, `http` 是协议, `www.edu.cn` 是主机名, `80` 是端口号, `index.aspx` 是要访问的资源名(此处是一个文件的形式)。

2. 超文本标记语言

超文本标记语言(HyperText Markup Language, HTML)是一种用来制作超文本文档的简单标记语言。HTML 在诞生之初, 其目的非常简单。当时 Tim Berners-Lee 将他设计的初级浏览器和编辑系统在网上合二为一, 创建了一种快速小型超文本语言来为他的这个想法服务。他也设计了数十种乃至数百种未来使用的超文本格式, 并想象智能客户代理通过服务器在网上进行轻松谈判并翻译文件。这同 Macintosh 的 Claris XTND 系统极为相似, 不同的是它可以在任何平台和浏览器上运行。

Berners-Lee 当时所设计的语言极其简易, 它以纯文本为基础, 因此任何编辑器和文字处理器都可以编辑, 并且它仅有不多的标签(Tag)组成, 任何人都可以轻松掌握。网络从此迅猛发展, 开启了大众在网上浏览和发布信息时代。

超文本传输协议规定了浏览器在运行 HTML 文档时所遵循的规则和进行的操作。HTTP 协议的制定使浏览器在运行超文本时有了统一的规则和标准。用 HTML 编写的超文本文档称为 HTML 文档, 它能独立于各种操作系统平台, 自 1990 年以来 HTML 就一直被用作 WWW/Web/万维网(World Wide Web)的信息表示语言, 是全球广域网上描述网页内容和外观的标准。使用 HTML 语言描述的文件, 需要通过 Web 浏览器显示出效果。HTML 包含了一对打开和关闭的标记, 在其中包含有属性和值。标记描述了每个在网页上的组件, 例如文本段落、文字、图形、动画、声音、表格、链接等对象。HTML 必须使用特定的程序, 即 Web 浏览器来完成翻译和执行的功能, 通常编写者可以使用任何编辑器对 HTML 文件进行编辑, 一些浏览器(如 Chrome、Firefox 和 Internet Explorer 等)则提供了交互式的 HTML 编辑器。

HTML 是一种用于创建文档的标记语言, 通过在文档中包含相关信息的链接来实现通过单击这个链接来访问其他文档、图像或多媒体对象, 并获得关于链接项的附加信息。有关 HTML 语言更详细的介绍, 将放在后面专门的章节中进行。

3. 超文本传输协议

超文本传输协议(HyperText Transfer Protocol, HTTP)是一种通信协议, 它允许将超文本标记语言(HTML)文档从 Web 服务器传送到 Web 浏览器。其中设计了一套相当简单的规则, 用来支持超媒体系统在网络上的分布, 它的出现使 Web 成为可能。如果希望真正理解 Web, 那么理解 HTTP 是基础。

HTTP 采用的是客户机/服务器(C/S)结构, 定义了客户机/服务器之间进行“对话”的简单请求-应答规则, 客户端的请求程序与运行在服务器端的接收程序建立连接, 如图 1-7 所示。客户端发送请求给服务器, HTTP 规则定义了如何正确解析请求信息, 服务器用应答信息回复该请求, 应答信息中包含了客户端所希望得到的信息, HTTP 规则当然也定义了如何正确解

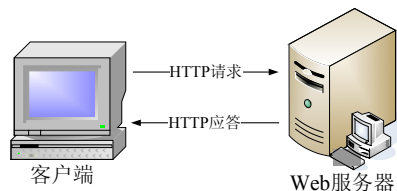


图 1-7 HTTP 的基本原理

析应答信息,但并没有定义网络如何建立连接、管理及信息如何在网络上发送,这些事情交给底层协议 TCP/IP 来完成。这也就是我们经常说“Web 是站在巨人的肩膀上”的原因,它的真实含义是“HTTP 是建立在 TCP/IP 之上的”,HTTP 属于应用层的协议,是 TCP/IP 的一个应用,从 TCP/IP 来看,Web(HTTP)和 Telnet、FTP、Gopher、WAIS 等没有什么区别。

注意:

HTTP 规则实际上定义了客户机和服务器之间请求与应答的格式,使用这种规范,传输过程能够得以顺利完成。

4. 浏览器

除了上面提到的三大技术外,浏览器在 Web 领域也起到了重要的作用。提到 Web 浏览器,大多数人会想到无处不在的 Microsoft Internet Explorer,直到像 Firefox、Safari 和 Opera 之类的浏览器日益兴起,这种情况才稍有变化。尽管许多新手可能认为 Internet Explorer 是市面上的第一个浏览器,但事实并非如此。实际上,第一个 Web 浏览器出自 Berners-Lee 之手,这是他为 NeXT 计算机创建的(这个 Web 浏览器原来名为 WorldWideWeb,后来改名为 Nexus),并在 1990 年发布给 CERN 的人员。Berners-Lee 和 Jean-Francois Groff 将 WorldWideWeb 移植到 C,并把这个浏览器改名为 libwww。20 世纪 90 年代初出现了许多浏览器,包括 Nicola Pellow 编写的一个行模式浏览器(这个浏览器允许任何系统的用户都能访问 Internet,从 UNIX 到 Microsoft DOS 都涵盖在内),还有 Samba,这是第一个面向 Macintosh 的浏览器。

1993 年 2 月, Illinois-Urbana-Champaign 大学超计算应用国家中心的 Marc Andreessen 和 Eric Bina 为 UNIX 发布了 Mosaic。几个月之后, Aleks Totic 为 Macintosh 发布了 Mosaic 的一个版本,这使得 Mosaic 成为第一个跨平台浏览器,它很快得到普及,并成为最流行的 Web 浏览器。后来这个技术卖给了 Spyglass,之后又归入 Microsoft 门下,最后成为现在的 Internet Explorer。

1993 年,堪萨斯大学的开发人员编写了一个基于文本的浏览器,叫作 Lynx,它成为字符终端的标准。1994 年,挪威奥斯陆的一个小组开发了 Opera。1996 年,这个浏览器得到了广泛使用。1994 年 12 月, Netscape 发布了 Mozilla 的 1.0 版,这标志着第一个营利性质的浏览器诞生。2002 年又发布了一个开源的版本,其发展为现在流行的 Firefox 浏览器,于 2004 年 11 月发布。

Microsoft 发布 Windows 95 时,把 Internet Explorer 1.0 作为 Microsoft Plus!包的一部分同时发布。尽管这个浏览器与操作系统集成在一起,但大多数人还是坚持使用 Netscape、Lynx 或 Opera。之后的 IE 2.0 有了很大起色,增加了对 cookie、安全套接字层(Secure Socket Layer, SSL)和其他新兴标准的支持。该版本还可以用于 Macintosh,使之成为 Microsoft 的第一个跨平台浏览器。不过,大多数用户还是很执着,仍然使用他们用惯了的浏览器。

不过到了 1996 年夏天, Microsoft 发布了 3.0 版本。几乎一夜之间,人们纷纷拥向 Internet Explorer。当时 Netscape 的浏览器还是要收费的,而 Microsoft 却免费提供了 Internet Explorer。关于浏览器领域谁主沉浮的问题, Internet 群体发生了两极分化,很多人担心 Microsoft 会像在桌面领域一样,在 Web 领域也一统天下。有些人则考虑到安全问题,而且不出所料, IE 3.0 版发布 9 天之后,就报告了第一个安全问题。但是到 1999 年发布 Internet Explorer 5 时,它已经逐步成为使用最广的浏览器。但根据 market share 的统计, 2018—2019 年全世界网民所使用

的六大浏览器和占比以及从逐月各大浏览器占比的变化情况如图 1-8 所示，其中目前使用数量最多的是 Google 的 Chrome，占比达 66.46%，而 IE 占比为 7.35%。

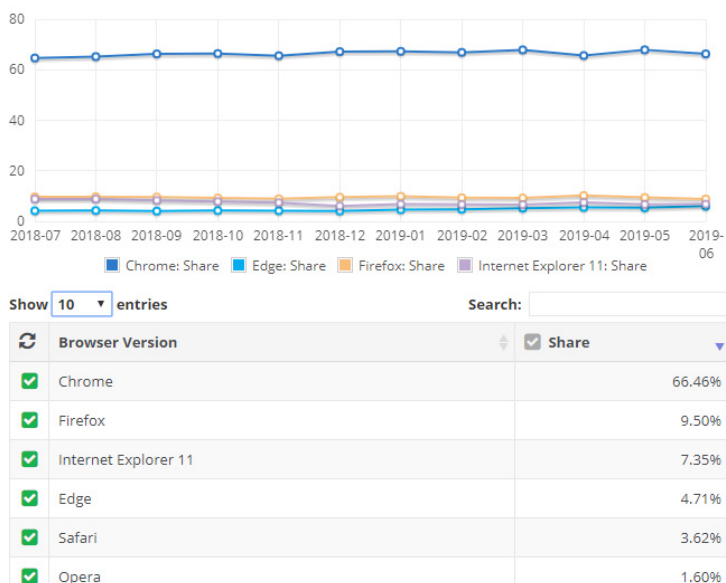


图 1-8 2018—2019 年全球浏览器占比及变化情况

1.2.4 Web 的高级技术

最初，所有 Web 页面都是静态的。用户请求一个资源，服务再返回这个资源。什么都不动，也不会出现屏幕的闪动。坦率地讲，对于部分 Web 网站来说就是这样实现的，这些网站的 Web 页面只是数字化的文本，一旦生成，就内容固定，再发布到各处。在浏览器发展的最初阶段，Web 页面的这种静态特性是可以满足需要的。当时网站的作用主要是交换研究论文和学术资料，教学机构也只是通过 Internet 在线发布课程信息。最初企业界还没有发现这个新“渠道”能提供巨大的商机。

从开发技术的角度来看，刚开始时，大量网站主页显示的信息较少，无非是一些联系信息或者只是一些文档等。不过不久之后，Web 用户就开始产生了新的需求，同时个人计算机的普及也推动了 Web 的快速发展，用户越来越希望得到更方便、具有动态性和交互性的网站体验。

1. CGI

要让 Web 更为动态，早期所采用的方法是通用网关接口(Common Gateway Interface, CGI)。与静态 Web 内容获取的方式不同，使用 CGI 可以创建一个具有响应用户请求的程序。假设需要在网站中显示所销售的商品，此时 CGI 脚本的功能主要是访问商品数据库，并以网页的方式将结果呈现给用户。通过使用简单的 HTML 表单和 CGI 脚本，就可以创建动态的应用，客户可以通过浏览器来浏览或购买商品。编写 CGI 脚本可以使用多种语言，从 Perl 到 Visual Basic 等都可以，这使得掌握不同语言的人都能直接编写 CGI 程序。

不过，要创建动态的 Web 页面，CGI 并不是最安全的方法。CGI 使得在系统中可以执行具有较高运行权限的应用程序。倘若某个用户有恶意企图，就可以利用这一点让系统运行恶意程

序, 从而带来安全问题。尽管存在这个缺陷, 但如今 CGI 仍在一定场合下被使用。

2. Applet

很显然, CGI 有待改进。1995 年 5 月, Sun 公司(后被合并到 Oracle 公司)的 John Gage 和 Andreessen 发布了一种新的编程语言——Java。Netscape Navigator 为这种新语言提供了支持, 最初是为了支持机顶盒(读者可能认为, 为了抢占在居室电子化方面的发展先机, 最早涉足的公司是 Microsoft 和 Sony)。就像所有革命一样, Java 和 Internet 的出现恰到好处, 在适当的时间、适当的地点推出, Java 在 Web 上发布仅几个月, 就已经有数以千计的人下载 Java。由于 Netscape 的 Navigator 支持 Java, 因此动态 Web 页面掀开了新的一页——Applet 时代到来。

Applet 允许开发人员编写小应用, 这些小应用可以嵌入在 Web 页面上。只要用户使用支持 Java 的浏览器, 就可以在浏览器的 Java 虚拟机(Java Virtual Machine, JVM)中运行 Applet。尽管 Applet 可以做很多事情, 但它们也存在一些限制, 即通常不允许读/写文件系统, 不能加载本地库, 而且可能无法启动客户端上的程序。不过, Applet 可以在一个沙箱安全模型中运行, 这有助于防止用户运行恶意代码。

对许多人来说, 最初接触 Java 编程语言就是从 Applet 开始的, 当时这是创建动态 Web 应用的一种绝好的办法。Applet 允许在浏览器中创建一个“胖”客户端应用, 不过必须在平台的安全限制范围内。当时, 在很多领域都广泛使用了 Applet; 但是, Web 群体并没有完全被 Applet “征服”。“胖”客户端的开发人员都很熟悉一个问题: 必须在客户端上部署适当的 Java 版本。因为 Applet 在浏览器的虚拟机中运行, 所以开发人员必须确保客户端安装了适当版本的 Java。尽管这个问题并非无法解决, 但确实妨碍了 Applet 技术的进一步推广。而且如果 Applet 写得较糟糕, 很可能对客户主机造成影响, 这使许多客户对于是否采用基于 Applet 的解决方案犹豫不决。

3. JavaScript

Netscape 创建了一种脚本语言, 并最终称之为 JavaScript(建立原型时本来叫作 Mocha, 正式发布之前曾经改名为 LiveWire 和 LiveScript, 不过最后终于确定为 JavaScript)。设计 JavaScript 旨在让不太熟悉 Java 的网页设计人员和程序员能够更轻松地开发 Applet(当然, Microsoft 也推出了与 JavaScript 相对应的脚本语言, 称为 VBScript)。Netscape 邀请 Brendan Eich 来设计和实现这种新语言, Brendan Eich 认为在这种情况下需要的是一种动态类型的脚本语言。由于缺乏开发工具, 缺少有用的错误消息和调试工具, JavaScript 备受非议。尽管如此, JavaScript 也仍然不失为一种创建动态 Web 应用的强大方法。

JavaScript 是一种基于对象和事件驱动并具有安全性能的脚本语言, 有了 JavaScript, 可使网页变得生动。使用它的目的是与 HTML、Java 脚本语言一起实现在一个网页中链接多个对象, 与网络客户交互作用, 从而可以开发客户端的应用。它是通过嵌入方式在标准的 HTML 语言中实现的。

最初, 创建 JavaScript 是为了帮助开发人员动态地修改页面上的标记, 以便为客户提供更丰富的体验。人们越来越认识到, 页面也可以当作对象, 因此文档对象模型(Document Object Model, DOM)应运而生。刚开始, JavaScript 和 DOM 紧密地交织在一起, 但最后它们还是“分道扬镳”, 并各自发展。DOM 是页面的一个面向对象模型, 可以用某种脚本语言(如 JavaScript

或 VBScript)进行修改。关于这部分内容的详细介绍读者可以查阅后面章节的内容。

最后,万维网协会(World Wide Web Consortium, W3C)介入,完成了 DOM 的标准化,而欧洲计算机制造商协会(European Computer Manufacturers Association, ECMA)则批准了将 JavaScript 作为 ECMAScript 规范。根据这些标准编写的页面和脚本在遵循相应原则的任何浏览器上都应该有相同的外观和表现。

在最初的几年中,JavaScript 的发展比较坎坷,这是许多因素造成的。首先,浏览器支持很不一致(即使是今天,同样的脚本在不同浏览器上也可能有不同的表现),而且客户可以自由地把 JavaScript 关闭(由于存在一些已知的安全漏洞,因此往往鼓励用户把 JavaScript 关掉)。由于开发 JavaScript 有一定难度,且使用 JavaScript 完成的代码是对用户公开的,这使得许多开发人员避之如不及,很少使用这种语言,有些开发人员干脆不考虑 JavaScript,认为这是图形设计人员使用的一种“玩具”语言。许多人曾试图使用、测试和调试复杂的 JavaScript,并为此身心俱疲,所以大多数人在经历了这种痛苦之后,最终还是满足于创建简单的基于表单的应用。

4. Servlet、JSP、ASP 和 PHP 等

尽管 Applet 是基于 Web 的,但“胖”客户端应用存在的许多问题在 Applet 身上也有所体现。在当时的网速条件下,要下载一个复杂 Applet 的完整代码,可能要花较长的时间,这往往是用户所不能忍受的。开发人员还要考虑客户端上的 Java 版本,有些虚拟机还有更多的要求。理想情况下只需提供静态的 Web 页面,毕竟这正是 Internet 的本来目标。当然,尽管静态页面是静态的,但是如果能在服务器上动态地生成内容,再将所生成的内容返回,这就更好了。

在 Java 问世一年左右,Sun 引入了 Servlet。Java 代码不用像 Applet 那样在客户端浏览器中运行;它在一个应用服务器上运行。这样,开发人员就能充分利用现有的业务应用,而且,如果需要升级为最新的 Java 版本,只需要考虑服务器端的升级就行了。正如 Java 所推崇的“一次编写,到处运行”,这一点使得开发人员可以选择最先进的应用服务器和服务器环境,这也是这种新技术的另一个优点。如此,Servlet 就可以取代 CGI 脚本了。

Servlet 向前迈出了很大一步,它提供了对整个 Java 应用编程接口(API)的完全访问,而且提供了一个完备的库可以处理 HTTP。不过,Servlet 并不是十全十美的,使用 Servlet 来设计界面可能很困难。在一个典型的 Servlet 交互中,先要从用户得到一些信息,完成某种业务逻辑,然后使用一些“打印行”创建 HTML,为用户显示结果。以下是一个简单的 Servlet 代码片段。

```
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<!DOCTYPE html>");
out.println("<head>");
out.println("<title>Servlet SimpleServlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Hello World</h1>");
out.println("<p>Imagine if this were more complex.</p>");
out.println("</body>");
out.println("</html>");
out.close();
```

Servlet 不仅容易出错, 很难生成可视化显示, 而且还无法做到人尽其才。一般来说, 服务器端代码的编写者往往是软件开发人员, 由于只是对算法和编译器很精通, 他们并不能设计出精美网站的图形和页面布局。使用这种模式进行开发, 业务开发人员不仅要编写业务逻辑, 还必须考虑怎样创建一致的设计。因此, 很有必要将表示与业务逻辑分离, 其实这里需要的就是 Java Server Pages(JSP)。

在某种程度上, JSP 是对 Microsoft 的 Active Server Pages(ASP)做出的一个回应。Microsoft 从 Sun 在 Servlet 规范上所犯的错误中吸取了教训, 并创建了 ASP 来简化动态页面的开发。Microsoft 增加了一些支持工具, 并与其 Web 服务器紧密集成。JSP 和 ASP 都具有将业务处理与表示布局相分离的特征, 从这个意义上讲, 二者是相似的。虽然存在一些技术上的差别(Sun 也从 Microsoft 那里吸取了教训), 但它们有一个最大的共同点, 即都允许 Web 设计人员能够把重点放在布局上, 而软件开发人员可以集中开发业务逻辑。以下代码展示了一段简单的 JSP 源码。

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Hello World</title>
</head>
<body>
<h1>Hello World</h1>
<p>This code is more familiar for Web developers.</p>
</body>
</html>
```

当然, Microsoft 和 Sun 并没有垄断服务器端解决方案。除了 JSP 和 ASP 以外还有许多其他的方案, 如 PHP、ColdFusion 等。有些开发人员喜欢独特的工具, 而有一些开发人员则倾向于更简单的语言。从目前来看, 所有这些解决方案完成的目标都是一样的, 它们都是要动态生成 HTML。

5. Flash

并不是只有 Microsoft 和 Sun 在努力寻找办法来解决动态 Web 页面问题。1996 年夏天, FutureWave 发布了一个名叫 FutureSplash Animator 的产品。这个产品起源于一个基于 Java 的动画播放器, FutureWave 很快被 Macromedia 兼并, Macromedia 则将这个产品改名为 Flash。

Flash 是交互式矢量图和 Web 动画的标准。网页设计者使用 Flash 可以创作出既漂亮又可改变尺寸的导航界面以及其他奇特的效果。Flash 通常也指 Macromedia Flash Player(现称为 Adobe Flash Player)。2012 年 8 月 15 日, Flash 退出 Android 平台, 正式告别移动端。Adobe Flash 最新版本也宣布支持 3D, 这将会是 Flash 未来发展的趋势, 也会是网页游戏的主流技术。2015 年 12 月 1 日, Adobe 将动画制作软件 Flash Professional CC 2015 升级并更名为 Animate CC 2015.5, 从此与 Flash 技术划清了界限。

利用 Flash, 设计人员可以创建令人惊叹的动态应用, 可以在 Web 上发布高度交互性的应

用,几乎与“胖”客户端应用相差无几。但是不同于 Applet、Servlet 和 CGI 脚本,Flash 可以不需要编程技巧,很容易上手。不过,这种易用性也是有代价的(有关胖客户端应用在本章的后面将有较为详细的介绍)。

像许多解决方案一样,Flash 需要客户端软件支持。尽管许多流行的操作系统和浏览器上都内置有其所需的播放器插件,但并非所有的浏览器都有。虽然能免费下载,但由于担心同时携带病毒,许多用户会拒绝安装这个软件,这一点限制了此解决方案的通用性。在某些情况下,Flash 应用可能还需要较大的网络带宽才能很好地工作,这也限制了 Flash 的推广(因此产生了某些网页上出现的所谓“跳过动画”的链接)。尽管有些网站选择建立多个版本的 Web 应用,分别适应于不同的连接速度,但是许多公司都无法承受支持两个或更多网站所增加的开发开销。

注意:

创建 Flash 应用需要专用的软件和浏览器插件。而 Applet 可以用文本编辑器编写,而且有一个免费的 Java 开发包(Java Development Kit, JDK); Flash 则不同,使用完整的 Flash 工具包需要支付较高的费用。尽管这些因素不是难以逾越的障碍,但它们确实减慢了 Flash 在动态 Web 应用道路上的前进脚步。

此外,由于 Flash 具有:存在一定的安全漏洞;其中所包含的文字无法被搜索引擎识别;需要专业人员制作、修改;占用 CPU 资源比较大,会影响页面响应速度;更新时需要手动安装等问题,因此出现了 HTML5 逐步取代 Flash 的趋势。使用 HTML5 的优势在于:符合 W3C 的标准;跨平台、多设备支持;方便搜索引擎抓取和搜索;便于游戏开发;具有更好的互动性;能直接支持音视频;标签代码更加简洁、清晰;更新及时、方便等。

6. Silverlight

Microsoft Silverlight 中文名称为“微软银光”,是微软所发展的 Web 前端应用开发解决方案,亦是微软丰富型互联网应用(Rich Internet Application)策略的主要应用开发平台之一。其以浏览器的外挂组件方式,提供 Web 应用中多媒体(含影音流与音效流)与高度交互性前端应用的解决方案,同时它也是微软 UX(用户体验)策略中的一环,更是微软试图将美术设计和程序开发人员的工作明确区分与协同合作发展应用的尝试之一。

对于互联网用户来说,Silverlight 是一个安装简单的插件程序。用户只要安装了这个插件程序,就可以在 Windows 和 Macintosh 等操作系统的多种浏览器中运行相应版本的 Silverlight 应用,享受视频分享、在线游戏、广告动画、交互丰富的网络服务等。

对于开发设计人员而言,Silverlight 是一种融合了微软的多种技术的 Web 呈现技术。它提供了一套开发框架,并通过使用基于向量的图像、图层技术,支持任何尺寸图像的无缝整合,对基于 ASP.NET、Ajax 在内的 Web 开发环境实现了无缝连接。Silverlight 使开发设计人员能够更好地协作,有效地创建能在 Windows 和 Macintosh 上的多种浏览器中运行的内容丰富、界面绚丽的 Web 应用——Silverlight 应用。

简而言之,Silverlight 是一个跨浏览器、跨平台的插件,为网络带来下一代基于 .NET 媒体体验的交互式应用。对运行在 Macintosh 和 Windows 上的主流浏览器,Silverlight 提供了统一而丰富的用户体验。通过 Silverlight 这个小小的浏览器插件,视频、交互性内容,以及其他应用能很好地融合在一起。

1.2.5 WWW 的发展

WWW 发展迅猛, 将来许多新的技术会带来革命性的进步, 以下是一些变化的方向。

1. DHTML 革命

Microsoft 和 Netscape 发布其各自浏览器的第 4 版时, Web 开发人员有了一个新的选择, 开发了动态 HTML(Dynamic HTML, DHTML)技术。有些人可能认为 DHTML 不是一个 W3C 标准, 它更像是一种销售手段。实际上, DHTML 结合了 HTML、层叠样式表(Cascading Style Sheets, CSS)、JavaScript 和 DOM。这些技术的结合使得开发人员可以动态地修改 Web 页面的内容和结构。

最初开发人员对 DHTML 的反响很好。不过, 它需要的浏览器版本还没有得到广泛应用。尽管主流浏览器都支持 DHTML, 但是它们的实现却存在差异, 因此开发人员必须知道用户使用的是什么浏览器; 否则就意味着, 需要大量代码来检查浏览器的类型和版本, 这进一步增加了开发的开销。有些人对于尝试这种方法很是迟疑, 因为 DHTML 还没有一个官方的标准。不过, 应该相信将来一定会更好。

2. XML 技术

20 世纪 90 年代中期, 基于 SGML, 衍生出了 W3C 的可扩展标记语言(eXtensible Markup Language, XML), 自此, XML 变得极为流行。许多人把 XML 视为解决所有计算机开发问题的灵丹妙药, 以至于 XML 几乎无处不在。实际上, Microsoft 早已经宣布, 将来的 Office 将支持 XML 文件格式。

如今, 我们至少有 4 种 XML 衍生语言可以创建 Web 应用(W3C 的 XHTML 不包括在内), 分别是: Mozilla 的 XUL; XAMJ, 这是结合 Java 的一种开源语言; Macromedia 的 MXML; 以及 Microsoft 的 XAML。下面分别对这 4 种语言进行详细介绍。

- XUL: XUL(拼作“zool”)代表 XML 用户接口语言(XML User Interface Language), 由 Mozilla Foundation 推出。流行的 Firefox 浏览器和 Thunderbird 邮件客户都是用 XUL 编写的。利用 XUL, 开发人员能构建功能很丰富的应用, 可以与 Internet 连接, 也可以不连接。为了让熟悉 DHTML 的开发人员尽快地学会, XUL 设计为可以为诸如窗口和按钮等标准界面部件提供跨平台支持。虽然它本身不是一个标准, 但它基于标准, 如 HTML 4.0、CSS、DOM、XML 和 ECMAScript 等。XUL 应用可以在浏览器上运行, 也可以安装在一个客户主机上。当然, XUL 也不是没有缺点。XUL 需要 Gecko 引擎, 而且目前 Internet Explorer 还没有相应的插件。尽管 Firefox 在浏览器市场已经有了一定的份额, 但少了 Internet Explorer 的支持还是会受到很大影响, 这使得大多数应用都无法使用 XUL。目前开展的很多项目都是力图在多个平台上使用 XUL, 包括 Eclipse。
- XAML: XAML(拼作“zammel”)是 Microsoft 推出的 Vista 操作系统的一个组件。XAML 是可扩展应用标记语言(eXtensible Application Markup Language)的缩写, 它为使用 Vista 创建用户界面定义了一个标准。与 HTML 类似, XAML 使用标签来创建标准元素, 如按钮和文本框等。XAML 建立在 Microsoft 的 .NET 平台之上, 而且可以编译为 .NET 类。开发人员应当很清楚 XAML 的局限所在, 其作为一个 Microsoft 产品, 要求必须使

用 Microsoft 的操作系统。在许多情况下,这可能不成问题,但是有些公司使用的不是 Microsoft 的操作系统,总不能削足适履吧。在 Vista 交付的日期不断推迟的过程中, XAML 也有了很大变化,它不再只是一个播放器。据说,在未来,我们可能会看到一个全新的 XAML。

- **MXML:** Macromedia 创建了 MXML,作为与其 Flex 技术一同使用的一种标记语言, MXML 与 HTML 很相似,可以以一种声明的方式来设计界面。与 XUL 和 XAML 类似, MXML 提供了更丰富的界面组件,如 DataGrid 和 TabNavigator,利用这些组件可以创建功能丰富的 Internet 应用。不过, MXML 不能独立使用,它依赖于 Flex 和 ActionScript 编程语言来编写业务逻辑。MXML 与 Flash 有同样的一些限制,表现为,它是专用的,而且依赖于价格昂贵的开发和部署环境。尽管将来 .NET 可能会对 MXML 提供支持,但现在 Flex 只能在 Java 2 企业版(Java 2 Enterprise Edition, J2EE)应用服务器上运行,如 Tomcat 和 IBM 的 WebSphere,这就进一步限制了 MXML 的广泛采用。
- **XAMJ:** 让人欣喜的是,开源群体又向有关界面设计的 XML 衍生语言世界增加了新的成员。XAMJ 作为另一种跨平台的语言,为 Web 应用开发人员又提供了一个工具。这种衍生语言基于 Java,由于 Java 是当前最流行的面向对象语言之一, XAMJ 也因此获得了面向对象语言的强大功能。XAMJ 实际上想要替代基于 XAML 或 HTML 的应用,力图寻找一种更为安全的方法,既不依赖于某种特定的框架,也不需要高速的 Internet 连接。XAMJ 是一种编译型语言,建立在“Clientlet”体系结构之上,尽管基于 XAMJ 的程序也可以是独立的应用,但一般来讲都是基于 Web 的应用。

谈到“以 X 开头的东西”时,是一定要涉及 W3C XForms 规范的。XForms 支持一种更丰富的用户界面,而且能够将数据与表示解耦合。毋庸置疑, XForms 数据是 XML,这样就能使用现有的 XML 技术,如 XPath 和 XML Schema。标准 HTML 能实现的功能, XForms 都能实现,而且 XForms 还有更多功能,包括动态检查阈值、与 Web 服务集成等。不同于其他的许多 W3C 规范, XForms 不需要新的浏览器,可以使用已有的许多浏览器实现。与大多数 XML 衍生语言一样, XForms 是一种全新的方法,所以对于这种方法何时得以采纳,目前还不能确定。

注意:

XML 技术正在快速进步中,目前,很多应用只是将 XML 作为一种数据交换或数据存储的手段,其实 XML 的功能远不止这些。

3. XHTML 技术

2000 年底, W3C 公布了 XHTML(Extensible HyperText Markup Language, 可扩展超文本标记语言) 1.0 版本。XHTML 1.0 是一种在 HTML 4.0 基础上优化和改进的新语言,目的是基于 XML 应用。这是一种增强型的 HTML,是更严谨、更纯净的 HTML 版本。其所具有的可扩展性和灵活性可以适应未来网络应用的更多需求。XML 虽然数据转换能力强大,甚至完全可以替代 HTML,但面对成千上万已有的基于 HTML 语言设计的网站,直接采用 XML 还为时过早。因此,在 HTML 4.0 的基础上,用 XML 的规则对其进行扩展,就得到了 XHTML,它的表现方式和 HTML 类似,但在语法上更加严格。所以,建立 XHTML 的目的在某种程度上是实现 HTML 向 XML 的过渡,它结合了部分 XML 的强大功能及大多数 HTML 的简单特性。在网站设计中

推崇的 Web 标准就是基于 XHTML 的应用(即通常所说的 CSS+DIV)。在与 CSS(层叠样式表)结合后, XHTML 能发挥真正的威力。在实现样式与内容分离的同时,又能有机地组合网页代码,还可以混合各种 XML 应用,比如 MathML、SVG 等。

XHTML 比 HTML 的语法更加严格,体现在:

- 所有的标签必须要闭合,也就是说开始标签要有相应的结束标签;
- 所有标签必须小写;
- 所有的参数值,包括数字,都必须放在双引号中;
- 图片必须使用 ALT 属性来提供说明文字。

4. HTML5

HTML 标准自 1999 年 12 月发布了 HTML 4.01 后,后继的 HTML5 和其他标准就被束之高阁。为了推动 Web 标准化运动的发展,一些公司联合起来,成立了一个叫作 Web Hypertext Application Technology Working Group(Web 超文本应用技术工作组,WHATWG)的组织。WHATWG 致力于 Web 表单和应用,而 W3C 专注于 XHTML 2.0。在 2006 年,双方决定进行合作,创建 HTML 的下一个版本。2014 年 10 月 29 日,该标准规范最终完成。HTML5 会逐步取代 HTML 4.01 和 XHTML 1.0 标准,以期能在互联网应用迅速发展的时候,使网络标准达到符合当代的网络需求,为桌面和移动平台带来无缝衔接的丰富内容。HTML5 还有望成为梦想中的“开放 Web 平台”的基石,进一步推动更深入的跨平台 Web 应用。当前 W3C 正致力于开发用于实时通信、电子支付、应用开发等方面的标准规范,还会创建一系列的隐私、安全防护措施。

相比之前的标准,HTML5 的变化主要在于:

- 取消了一些过时的 HTML 4.0 标签:其中包括纯粹显示效果的标签,如和<center>,它们已经被 CSS 取代;HTML5 吸取了 XHTML 2.0 的一些用法,包括一些用来改善文档结构的功能,如新的 HTML 标签 header、footer、dialog、aside、figure 等的使用,将使内容创作者更加语义化地创建文档,之前的开发人员在实现这些功能时一般都是使用 div。
- 将内容和展示相分离:b 和 i 标签依然保留,但它们的意义已经和之前有所不同,这些标签的意义只是为了将一段文字标识出来,而不是为了设置粗体或斜体样式;u、font、center、strike 这些标签则被完全去掉了。
- 一些全新的表单输入对象:包括日期、URL、E-mail 地址,其他对象则增加了对非拉丁字符的支持。HTML5 还引入了微数据,这一使用机器可以识别的标签标注内容的方法,使语义 Web 的处理更为简单。总的来说,这些与结构有关的改进使内容创建者可以创建更干净、更容易管理的网页,这样的网页对搜索引擎,对读屏软件等更为友好。
- 全新的、更合理的 Tag:多媒体对象将不再全部绑定在 object 或 embed Tag 中,而是视频有视频的 Tag,音频有音频的 Tag。
- 本地数据库:这个功能将内嵌一个本地的 SQL 数据库,以加速交互式搜索、缓存以及索引功能。同时,那些离线 Web 程序也将因此获益匪浅。
- Canvas 对象:将给浏览器带来直接在上面绘制矢量图的能力,这意味着用户可以脱离 Flash 和 Silverlight,直接在浏览器中显示图形或动画。
- 浏览器中的真正程序:将提供 API 实现浏览器内的编辑、拖放,以及各种图形用户界面的能力。内容修饰 Tag 将被剔除,而使用 CSS。

- 移动端开发方面：在移动应用中取代 Flash，开发过程友好、跨平台、可适配多种终端。
- 其突出的特点就是强化了 Web 页面的表现性，追加了本地数据库。
- 搜索引擎方面：新增了页面语义化元素，让搜索引擎更容易抓取网页信息。
- 多媒体应用方面：新增了专门的多媒体元素，可以很方便地在网上插入音频、视频等元素，网页加载时也不容易产生较大的延迟。

HTML5 是一项非常有前途的技术，本书将在第 3 章和第 4 章介绍这项技术。

5. Ajax 技术

Ajax(异步 JavaScript 和 XML)是个新产生的术语，它从两方面提供强大的性能。这两个特性在多年来一直被网络开发人员所忽略，直到 Gmail、Google suggest 和 Google Maps 的横空出世才使人们开始意识到其重要性。这两个特性是：

- 无须重新加载整个页面便能向服务器发送请求；
- 对 XML 文档的解析和处理。

Ajax 描述了一组技术，它使浏览器可以为用户提供更为自然的浏览体验。在 Ajax 之前，Web 站点强制用户采用“提交/等待/重新显示”的操作流程，用户的动作总是与服务器的“思考时间”同步。Ajax 提供与服务器异步通信的能力，从而使用户从请求/响应的循环中解脱出来。借助于 Ajax，可以在用户单击按钮时，使用 JavaScript 和 DHTML 立即更新 UI，并向服务器发出异步请求，以更新或查询数据库。当请求返回时，就可以使用 JavaScript 和 CSS 来相应地更新 UI，而不是刷新整个页面。最重要的是，用户甚至不知道浏览器正在与服务器通信，Web 站点看起来是即时响应的。

1.3 Web 应用开发的架构

最初，Web 应用技术与应用的开发技术是独立发展的。在各种日益复杂的应用需求的推动下，两者日益融合，但又各具特色。目前基于 Web 的应用已经成为一种主流的解决方案。下面就从 Web 应用架构的角度，来分析 Web 应用开发的架构方法。

1.3.1 Web 应用的需求

自从 Web 诞生以来，经过十几年的发展，Web 应用的架构经历了静态页面、活动页面以及动态页面的转变等。

1. 静态页面

静态页面是存储于服务器的文件，其内容在生成该文档时就已定义好，并且始终不变。Web 上最早的内容就是这种形式的，这些页面中可以包含多种媒体元素，信息资源的表现形式也是多样化的，而且页面间可以通过超链接进行关联，便于用户浏览和检索。典型的静态页面在访问时可以看到其 URL 中的资源的扩展名通常为 html，如图 1-9 所示。

静态页面的内容是编著者创作页面时确定好的，一旦将所有内容保存在服务器中并发布，任何人访问时都可以看到相同的内容。虽然静态页面已经可以为用户提供通过远程来访问信息

资源的良好途径，但这种架构方式实际上存在很大的局限性。对于信息资源的用户而言，是不能和页面进行交互的，页面的内容也不会因为用户所做的操作而发生变化。对于信息资源的提供者而言，静态页面必须手工制作，手工更新和发布，其开发难度不大。

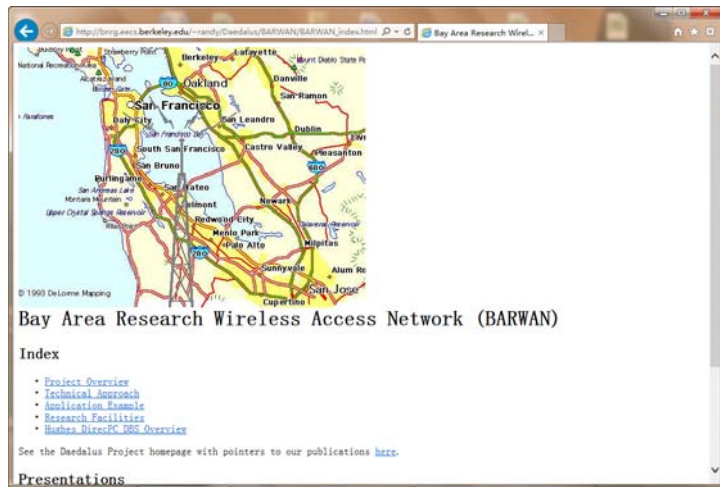


图 1-9 静态页面

2. 动态页面

对于类似网上购物、股市行情等需要实时更新的信息而言，静态页面是难以胜任的。同时静态页面也无法实现显示形式、内容等的个性化定制。动态页面就是针对这些问题应运而生的。

动态页面是在浏览器访问 Web 服务器时，由 Web 服务器创建的。当浏览器向服务器发出请求时，Web 服务器运行一个应用，创建动态文档，并返回给浏览器作为应答。因此，不同用户在不同时刻访问同一个动态页面时，可能会得到不同的结果，从这个角度来看，动态页面的内容是变化的，如图 1-10 所示。Web 服务器还可以将数据库中的最新数据返回给用户。但是动态页面需要由服务器实时生成，服务器的负荷较前面提到的静态方式要大；同时，其开发难度也较大，这对开发人员提出了更高的要求。



图 1-10 动态页面

3. 活动页面

提出活动页面是为了在保持动态页面优点的基础上,又能避免服务器负担过重的问题。即在传统 HTML 文档的基础上,加入诸如 Java Applet、VBScript 脚本、ActiveX 控件、Flash 插件等活动元素。首先,由服务器提供 HTML 文档和相关的活动元素,它们经客户端下载后在客户端运行,浏览器执行这些活动程序后再获得所需的信息,因此所显示的内容并不完全由服务器产生。用户通过这些元素可以和 Web 服务器进行交互,只要用户程序在运行,该页面就可不断变化保持最新,如图 1-11 所示。



图 1-11 活动页面

由于这些元素在客户端运行,因此可以实现快速的响应和显示,但它们对客户端计算机的硬件配置和浏览器软件提出了一定的要求。此外,实现活动页面也需要一系列新技术的支持,可包括 ActiveX、Java Applet 和 Flash 插件等。

从目前的情况来看,Internet 市场仍具有巨大的发展潜力,未来其应用将涵盖从办公室共享信息到市场营销、服务等广泛领域。另外,Internet 带来的电子贸易正改变着现今商业活动的传统模式。

1.3.2 应用发展的需求

从应用开发模式发展的角度来看,从最早的单机应用,到后来的 C/S 模式(客户机/服务器模式),再到当前流行的 B/S 模式(Browser/Server)、SOA、云计算等,是由简单的两层结构逐步演变为三层甚至是多层的。此外,RIA、分布式应用、设计模式和各种高级的架构模式等也在需求的推动下得到日益广泛的应用。

1. 两层结构

所谓的两层结构(Two-Tier)指的是客户机、服务器,即 C/S 结构,其结构如图 1-12 所示。通常来说,数据库位于服务器端,而客户端应用提供了与用户接口的界面,同时还包含了对服

务器上的数据进行操作的一系列规则(商业逻辑)。在这种模式下,服务器仅需要承担数据访问的任务,而客户端程序不仅需要完成业务逻辑,即数据处理的任务,还需要负责数据的显示形式,即展示问题。

注意:

通常将 C/S 这种模式的部署方式形象地称为“胖客户端/瘦服务器”(Fat Client/Thin Server)。

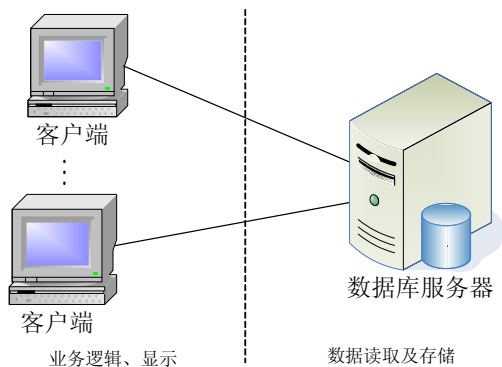


图 1-12 两层结构

C/S 模式使得多个客户可以同时访问服务器上的数据库。但是,两层结构也有不足之处。在这种结构中,所有的数据处理规则都与某个应用相关联。一旦业务逻辑发生变化,必须重新修改和发布客户端的应用。如果客户机的数量巨大,这个工作将变得十分繁重和费时。因此,两层模式难以适应大规模分布式的应用需求。

2. 三层结构

三层结构(Three-Tier)旨在解决两层结构所存在的问题。从功能的角度将整个应用的功能分成表示层、功能层和数据层三部分,其结构如图 1-13 所示。其解决方案是,对这三层进行明确分割,并在逻辑上使其独立。原来的数据层作为 DBMS 已经独立出来,所以关键是要将表示层和功能层分离成各自独立的程序,并且还要使这两层间的接口简洁明了。

一般情况是只将表示层配置在客户机中,与两层 C/S 结构相比,其程序的可维护性要好得多,但是其他问题并未得到解决。客户机的负荷较重,其业务处理所需的数据要从服务器传给客户机,所以系统的性能容易变差。如果将功能层和数据层分别放在不同的服务器中,则服务器和服务器之间也要进行数据传送。但是,由于在这种形态中三层是分别放在各自不同的硬件系统上的,因此灵活性很高,能够适应客户机数量的增加和负荷的变动。例如,在追加新业务处理时,可以相应增加功能层服务器的数量。因此,系统规模越大,这种形态的优点就越显著。

注意:

由于服务器承担了大部分的处理工作,因此常常将这种模式称为“瘦客户端/胖服务器”(Thin Client/ Fat Server)。

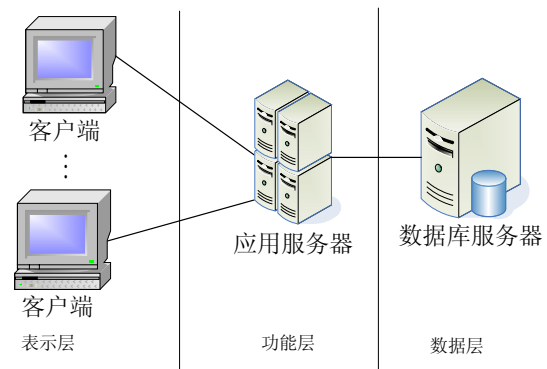


图 1-13 三层结构

3. 基于 Web 的 B/S 模式

随着 Web 的广泛运用，人们发现在某些情况下可以使用 Web 来取代以往的应用。此时，将 Web 浏览器作为表示层；Web 服务器上的各种服务器端应用充当功能层；而数据层使用数据库服务器。为了与传统的三层结构相区别，将它称为 B/S 模式。以下对该模式进行分析。

(1) 静态模式

它的服务器端基本上只由 Web 服务器构成，它要发布的内容以文件的形式保存在 Web 服务器上，只能通过 HTML 文件提供静态的 Web 内容，所有的服务内容必须预先定义并编辑好，其结构如图 1-14 所示。用户可以通过 URL 直接定位到这些定制好的 HTML 文件进行存取，这一模式比较简单，并且可靠性比较高，实现起来也比较容易，但是提供的内容比较单调，且时效性及可维护性均较差，现在较大型的网站已很少采用。

(2) 一般动态模式

一般动态模式是当前使用得比较多的一种结构模式。这种模式在服务器端增加了一台数据库服务器，其结构如图 1-15 所示。它可以为用户提供动态的信息服务，通过定制页面模板，添加到后台数据库的信息可即时发布给发起请求的客户机，保证了信息的时效性。但由于它增加了 Web 服务器的负担，因此降低了 Web 服务器的稳定性。具体的实现方式大致上可通过 ASP.NET、JSP、PHP 等脚本语言、普通的 CGI 程序或 ISAPI 或 NSAPI 等来实现。例如，利用 Linux+PHP+MySQL+Apache 来构成整个服务体系。

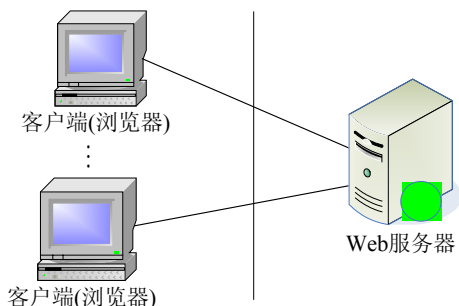


图 1-14 静态模式

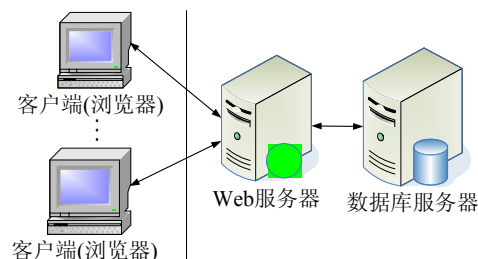


图 1-15 一般动态模式

(3) 多层动态模式

多层动态模式是在 Web 服务器和后台数据库服务器之间增加了一层应用服务器，其结构如图 1-16 所示。这是一种先进的结构模式，在国外的一些大型知名网站上有所应用，像 Microsoft 的站点以及国外的一些大型电子商务站点均采用了这种结构模式。由于将一些复杂的企业逻辑及数据库的连接服务等封装到中间层中，因此减轻了 Web 服务器

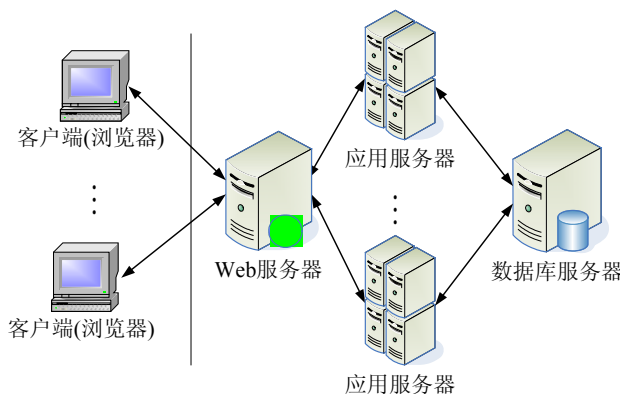


图 1-16 多层动态模式

的负担。

多层动态模式具有负载平衡与容错的功能，这可以通过各种技术来实现，比如，通过 ASP 脚本结合 COM/COM+、CGI 或 ISAPI 结合 COM / COM+、PHP 脚本结合 CORBA 构件技术来实现。大多数构件均是已编译的可执行代码，在执行速度上要比单纯的脚本语言快得多。这种结构属于典型的分布式 Web 应用系统。

B/S 模式的最大优势在于将应用部署到 Web 上，能创建跨平台的应用，避免多次创建和分发同一个软件的多个版本。服务器端的应用使用 Web 服务器上生成的 HTML 文档，这样几乎可以被所有平台上的用户浏览。

对于不同角色的服务器，由于作用不同，其要求也不尽相同：

- 对于应用服务器而言，由于需要处理大量的业务逻辑，因此需要更好、更快、更强大的 CPU 来支持；
- 数据库服务器由于需要快速地进行磁盘检索和数据缓存，因此需要更快的硬盘和更大的内存；
- 文件服务器由于需要存储用户上传的文件资源，因此需要更大的硬盘存储空间；
- 分布式缓存服务器则可以使用集群的方式，部署大内存的服务器作为专门的缓存服务器，可以在理论上实现不受内存容量限制的缓存服务。

此外，还可以进一步采用数据库读写分离、反向代理和 CDN、分布式文件系统和分布式数据库系统、NoSQL 和非数据库查询技术、业务拆分等方式来进行优化。

(4) RIA

富互联网应用(RIA)是下一代的将桌面应用的交互式用户体验与传统的 Web 应用的部署灵活性和成本分析结合起来的网络应用。富互联网应用中的富客户端技术通过提供可承载已编译客户端应用(以文件形式，用 HTTP 传递)的运行环境，客户端应用使用异步客户机/服务器架构连接现有的后端应用服务器，这是一种安全、可升级、具有良好适应性的新的面向服务的模型，这种模型由采用的 Web 服务所驱动。结合了声音、视频和实时对话的综合通信技术，使富互联网应用具有前所未有的网上用户体验。

RIA 具有的桌面应用的特点包括：在消息确认和格式编排方面提供互动用户界面；在无刷新页面的情况下提供快捷的界面响应时间；提供通用的用户界面特性如拖放式(drag and drop)以及在线和离线操作能力。RIA 具有的 Web 应用的特点包括如立即部署、跨平台、采用逐步下载来检索内容和数据以及可以充分利用被广泛采纳的互联网标准等。RIA 具有通信的特点则包括实时互动的声音和图像。客户机在 RIA 中的作用不仅是展示页面，还可以在幕后与用户请求异步地进行计算、传送和检索数据、显示集成的用户界面和综合使用声音及图像，这一切都可以在不依赖于客户机连接的服务器或后端的情况下进行。

可实现这类应用的技术包括 Adobe 的 Flex、微软的 Silverlight、Oracle 的 JavaFX 和 Java SWT、XUL、Bindows、Curl、Laszlo 以及 MUILIB 等。

4. Web 应用框架

Web 应用框架(Web Application Framework)是一种开发框架技术，用来支持动态网站、网络应用及网络服务的开发。框架技术有助于减轻网页开发时共通性活动的工作负载，例如许多框架提供数据库访问接口、标准样板以及会话管理等，可提升代码的可复用性。框架技术可分为

基于请求的(request-based)和基于组件的(component-based)两大阵营。前者的代表有 Struts 和 Spring MVC 等, 后者的代表则有 JSF、Tapestry 等。

基于请求的框架较早出现, 它用于描述一个 Web 应用结构的概念, 其和传统的静态 Internet 站点一样, 是将其机制扩展到动态内容的延伸。对一个提供 HTML 和图片等静态内容的网站, 网络另一端的浏览器发出以 URL 形式指定的资源的请求, Web 服务器解读请求, 检查该资源是否存在于本地, 如果是则返回该静态内容, 否则通知浏览器没有找到。Web 应用升级到动态内容领域后, 这个模型只需要做一点修改, 即 Web 服务器收到一个 URL 请求(相较于静态情况下的资源, 动态情况下更接近于对一种服务的请求和调用)后, 判断该请求的类型, 如果是静态资源, 则按上面所述进行处理; 如果是动态内容, 则通过某种机制(CGI、调用常驻内存的模块、递送给另一个进程如 Java 容器等)运行该动态内容对应的程序, 最后由程序给出响应, 返回浏览器。在这样一个直接与 Web 底层机制交流的模型中, 服务器端程序要收集客户端即 Get 或 Post 方式提交的数据、转换、校验, 然后以这些数据作为输入运行业务逻辑后生成动态的内容(包括 HTML、JavaScript、CSS、图片等)。

基于组件的框架则采取了另一种思路, 它把长久以来软件开发应用的组件思想引入到 Web 开发中。服务器返回的原本文档形式的网页被视为由一个个可独立工作、重复使用的组件构成。每个组件都能接受用户的输入, 负责自己的显示。上面提到的服务器端程序所做的数据收集、转换、校验的工作都被下放给各个组件。现代 Web 框架基本上都采用了模型、视图、控制器相分离的 MVC 架构, 基于请求和基于组件的两种类型的框架大都会有一个控制器将用户的请求分派给负责业务逻辑的模型, 运算的结果再以某个视图表现出来。所以两大分类框架的区别主要在于视图部分, 基于请求的框架仍然把视图也即网页看作一个文档整体, 程序员要用 HTML、JavaScript 和 CSS 这些底层的代码来写“文档”, 而基于组件的框架则把视图看作由积木一样的构件拼成, 积木的显示不用程序员操心(当然它也是由另一些程序员开发出来的), 只要设置好它绑定的数据和调整好它的属性, 就可以把它从编写 HTML、JavaScript 和 CSS 界面的工作中解放出来。

5. 实际应用中的选取原则

(1) 首先, 确定是应该使用 C/S 模式还是 B/S 模式。对于某些应用场合, C/S 模式还是存在优势的。比如开发一个在 Windows 下运行的程序, 或开发一个在局域网内并且只针对少量用户的程序, 或者一个管理程序、后台运行程序, 未必一定强求使用多层模式, 因为在这种情况下 B/S 并不能带来什么突出的好处, 反而会增加工作量与维护量。

(2) 选择了 B/S 模式进行 Web 应用开发时, 要根据 Web 网站的规模、用户访问量以及要求的响应时间等几个指标来规划网站的结构模式。由于 Web 技术的发展, 前面所说的静态模式现在已很少采用, 它已不能满足当前的用户基本需求了。

(3) 对于一般动态模式, 要分情况对待。对于访问量很低, 信息量不大, 或对系统稳定性要求不是很高的情况, 可以采用这种模式。因为这种模式对编程人员的素质要求不是很高, 并且开发周期快, 比较适用于企业内部的 Intranet 或一些访问量不大的中小型网站。

(4) 对于一些大型的门户网站或大型的电子商务网站, 由于用户访问量非常大, 并且对系统的安全性以及稳定性要求都十分严格, 在电子商务网站中, 对数据的严谨性要求也非常严格, 因此, 在这几种情况下, 多层动态模式就更加适合。

(5) RIA 能实现比 HTML 更加健壮、反应更加灵敏和更具有令人感兴趣的可视化。RIA 允许使用一种像 Web 一样简单的方式来部署富客户端程序。对于那些采用 C/S 架构的胖客户端技术运行复杂应用系统的机构和采用基于 B/S 架构的瘦客户端技术部署 Web 应用系统的机构来说, RIA 确实提供了一种廉价的选择。在应用时, 需要结合客户端资源和客户端的交互需求进行设计, 由于可以与前面几种模式结合应用, 因而可以产生多种运用模式, 但通常这种应用对客户端的运算能力有一定的要求。

(6) 针对较为复杂的应用, 可以考虑在开发过程中运用框架, 而基于请求的和基于组件的方法则各有优劣。不过后者看上去有很大的吸引力, 普通的 Web 开发人员只要使用专门的公司或开源组织提供的组件就可以轻松开发出好用且漂亮的界面。要编写一个没有潜在问题的、跨浏览器的、显示美观、有足够灵活性并且可以调整的服务器端组件则需要高水平的技能、丰富的经验和较多的时间, 即使付出这些成本, 也不能完全避免使用者失望的情况。综合来看, 基于请求的框架要求程序员自己动手的地方比较多, 但也因此可以更精细地控制 HTML、CSS 和 JavaScript 这些最终决定应用界面的代码, 特别是如果要在界面上有创新, 尝试新的视觉效果和用户操作, 必然应选择基于请求的框架。基于组件的框架可以提高开发界面的效率, 前提是选用的组件质量要优秀。

总而言之, 技术只有与实际应用的需求紧密结合才能具有持续不断发展的生命力。针对特定应用而言, 任何超前或落后的技术都将产生负面效应乃至失败。

1.4 本章小结

Web 应用开发是目前计算机应用的热点之一。本章首先讲解了有关 Internet 和 Web 的一些基础知识。为了让读者对 Web 技术有一个较为全面的认识, 分别从 Web 需求的发展、Web 应用发展的层面讨论了 Web 技术的本质。本章的内容为读者深入掌握 Web 技术奠定了基础。

1.5 思考和练习

1. Internet 与 WWW 有什么关系?
2. 统一资源定位符(URL)——<https://www.alipay.com/aip/index.html> 中, 既包含了 HTTP, 又包含了 WWW, 它们之间是什么关系?
3. 本章提到的开发技术中哪项技术更好? 我们应该学习哪项技术呢?
4. 简述 DNS 应用系统的主要作用、系统组成以及其基本工作原理。
5. HTML、DHTML、XHTML、XML、HTML5 之间存在什么异同?
6. 简述 URL 访问网站时的网络传输全过程。
7. 对于某一特定的网站建设需求, 应该如何选择合适的架构方式?