

高等院校计算机应用系列教材

网站设计与Web应用 开发技术(第四版)(微课版)

吴伟敏 编著

清华大学出版社
北京



内 容 简 介

本书从移动互联网基本概念和项目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方法,并对将来移动互联网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,内容包括移动互联网简介、移动互联网项目策划设计与运行环境配置、HTTP协议及其开发与前端开发基础、层叠样式表(CSS)、JavaScript语言与客户端开发、服务器端开发——动态页面技术基础和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生学习移动互联网设计及相关开发技术的教材,也可作为移动互联网开发人员及自学者的参考用书。

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,也可以扫描前言中的二维码进行下载。扫描前言中的视频二维码可以直接观看教学视频。

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。举报:010-62782989, beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

网站设计与Web应用开发技术:微课版/吴伟敏编
著.--4版.--北京:清华大学出版社,2024.9.
(高等院校计算机应用系列教材).--ISBN 978-7-302-
-66984-5

I . TP393.092

中国国家版本馆 CIP 数据核字第 2024TN5357 号

责任编辑:胡辰浩

封面设计:高娟妮

版式设计:茆博文化

责任校对:孔祥亮

责任印制:刘 菲

出版发行:清华大学出版社

网 址: <https://www.tup.com.cn>, <https://www.wqxuetang.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-83470000 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:三河市铭诚印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24 字 数:615千字

版 次:2009年1月第1版 2024年10月第4版 印 次:2024年10月第1次印刷

定 价:79.00元

产品编号:107186-01



前

言

移动互联网技术发展迅速，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。在长期关于项目开发的教学生涯中，笔者注意到虽然很多人希望通过学习掌握开发技术，但由于缺乏正确的见解和思考的方法，有的导致学习事倍功半，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

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，特别是移动互联网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，此外还需要考虑日益强大的AI技术对开发者的冲击和挑战，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，在本书中将介绍移动互联网的发展历史、工作原理、开发框架、项目策划设计、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、AI技术等内容。希望这样的内容安排能为大多数希望学习和掌握移动互联网开发技术的读者有所帮助，使他们能够更好地了解移动互联网及其相关技术的走向和本质。对于初学者，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，拓展问题的解决手段和方法。

由于本书旨在为读者今后学习和开发高级项目打下良好的基础，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。

完整地学习移动互联网开发技术需要具备3个层面的知识。本书据此设计了3个层次：移动互联网基本概念、开发基础和高级应用开发。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。

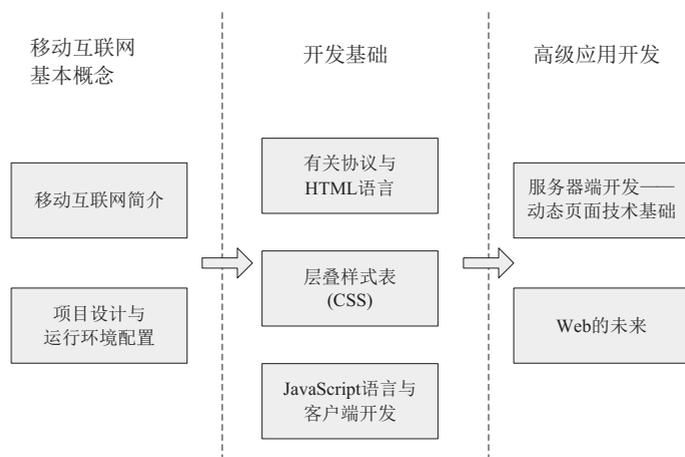


图1 本书的知识体系结构

概括起来，本书具有以下主要特点。

- 结构清晰、内容翔实。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，并能通过所掌握的学习方法来解决未来实际工作中遇到的问题。全书按照项目开发的方法与顺序，从基本概念和策划设计入手，循序渐进地介绍了进行项目开发的步骤、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、准确地掌握项目开发的全过程，并使读者能够举一反三。
- 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：移动互联网简介、移动互联网项目策划设计与运行环境配置、HTTP协议及其开发与前端开发基础、层叠样式表(CSS)、JavaScript语言与客户端开发、服务器端开发——动态页面技术基础和Web的未来。

第1章“移动互联网简介”，介绍移动互联网与WWW的发展历程、移动互联网相关的基本概念及其开发技术以及基本框架。第2章“移动互联网项目策划设计与运行环境配置”，说明在项目建立之前做好策划工作的必要性，并给出了一些基本原则；为了让系统正常运行，需要在正式开发前做好详细的设计工作；本章还介绍创建服务器和配置基本开发环境的方法。第3章“HTTP协议及其开发与前端开发基础”，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，交互元素的设计和实现思路，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、常用对象和页面特效的制作方法。第6章“服务器端开发——动态页面技术基础”，介绍服务器端开发的几种典型方法、动态页面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、人工智能时代的项目开发等。

有一定网络和网站基础知识的读者可跳过第1章的学习，具备项目设计、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、详细地讲解项目设计与移动互联网应用开发技术，因此本书既可作为各类高等院校学生学习网站设计及移动互联网技术的教材，也可作为移动互联网开发人员及自学者的参考用书。

本书除封面署名的作者外，南京邮电大学的毕梦娜、倪超和朱逸雯等参与了本书第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北京元年科技股份有限公司吴洲华。

感谢笔者的好友柳文娟，她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，他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。

在编写本书的过程中参考了相关文献，在此向这些文献的作者深表感谢。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，书中难免存在不足，敬请读者批评指正。我们的电话是010-62796045，信箱是992116@qq.com。

本书配套的电子课件、实例源文件和习题答案可以到<http://www.tupwk.com.cn/downpage>网站下载，也可以扫描下方二维码获取。扫描下方二维码可以直接观看教学视频。

扫描下载



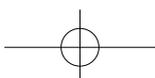
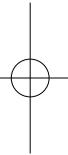
配套资源

扫一扫



看视频

作者
2024年5月



目 录

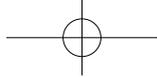
第1章 移动互联网简介	1	2.4 建立项目的准备——环境安装与配置	59
1.1 移动互联网与WWW	1	2.4.1 IIS的安装	59
1.1.1 移动互联网的发展	1	2.4.2 使用IIS建立站点	61
1.1.2 Internet技术基础	4	2.4.3 IIS的配置	62
1.1.3 Internet提供的服务	7	2.4.4 其他Web服务器	63
1.2 移动互联网概述	9	2.5 项目运行的基础——安全	64
1.2.1 移动互联网的发展	9	2.5.1 安全威胁	64
1.2.2 移动互联网是什么?	11	2.5.2 防范策略	66
1.2.3 移动互联网的技术基础	12	2.6 项目开发过程	68
1.2.4 移动互联网的关键技术	15	2.6.1 瀑布模型	68
1.2.5 移动互联网的未来	22	2.6.2 敏捷开发模型	69
1.3 移动互联网应用开发的需求与实现架构	24	2.7 项目评估	72
1.3.1 移动互联网应用的需求	24	2.7.1 准备工作	72
1.3.2 应用发展的需求	25	2.7.2 数据分析	72
1.4 本章小结	30	2.7.3 小结	73
1.5 思考和练习	31	2.8 本章小结	73
第2章 移动互联网项目策划设计与运行环境配置	32	2.9 思考和练习	74
2.1 项目建设的总体流程	32	第3章 HTTP协议及其开发与前端开发基础	75
2.2 项目建立的前期工作——策划	33	3.1 HTTP协议	75
2.3 项目设计	36	3.1.1 HTTP概述	75
2.3.1 项目的CI形象设计	37	3.1.2 HTTP的宏观工作原理	80
2.3.2 项目的总体结构设计	39	3.1.3 HTTP协议基础	82
2.3.3 项目的版面设计	40	3.1.4 HTTP应用开发方法	88
2.3.4 项目的色彩设计	46	3.1.5 HTTP应用的开发	90
2.3.5 项目的导航设计	49	3.1.6 安全超文本传输协议、安全套接层及传输层协议	92
2.3.6 项目信息的可用性设计	51	3.2 前端开发基础	94
2.3.7 项目的交互设计	52	3.2.1 前端开发简介	94

3.2.2	HTML标记语法及文档结构	100
3.3	HTML的基本应用	111
3.3.1	标题和段落	111
3.3.2	列表	115
3.3.3	超链接	119
3.3.4	表格	125
3.3.5	图像、音频、视频及嵌入元素	131
3.3.6	iframe框架应用	137
3.3.7	表单	138
3.3.8	canvas应用	145
3.4	本章小结	148
3.5	思考和练习	149
第4章	层叠样式表(CSS)	150
4.1	CSS概述	150
4.2	将CSS引入网站	153
4.2.1	CSS的定义	154
4.2.2	CSS的浏览器兼容性	155
4.2.3	不同层次的CSS定义	156
4.2.4	书写规范	158
4.3	CSS选择器	159
4.3.1	标签选择器	159
4.3.2	类别选择器	160
4.3.3	ID选择器	162
4.3.4	通用选择器	164
4.3.5	后代选择器	164
4.3.6	交集选择器	165
4.3.7	并集选择器	166
4.3.8	伪类选择器	167
4.3.9	伪元素选择器	170
4.3.10	样式表的继承性与层叠性	172
4.3.11	对div+CSS方案的思考	173
4.4	CSS的布局及盒子模型	174
4.4.1	CSS的布局基础	174
4.4.2	CSS的盒子模型	177
4.4.3	CSS布局	181
4.4.4	CSS布局技巧	188
4.5	CSS滤镜	193
4.5.1	CSS3滤镜的种类及定义方式	193
4.5.2	滤镜实例	195

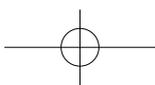
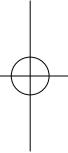
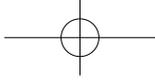
4.6	CSS典型用法实例	198
4.6.1	边框的用法	198
4.6.2	动画	199
4.6.3	语音应用	205
4.6.4	制作可交互的360度全景展示	206
4.6.5	自动适应移动设备横竖屏显示方式的实现方案	209
4.7	本章小结	210
4.8	思考和练习	210

第5章 JavaScript语言与客户端开发 211

5.1	JavaScript简介	211
5.1.1	什么是JavaScript	212
5.1.2	JavaScript的作用	213
5.1.3	JavaScript语言的组成	214
5.1.4	将JavaScript引入HTML文档的方式	215
5.1.5	一个简单的实例	216
5.1.6	JavaScript的版本与兼容性	217
5.2	JavaScript基本语法	219
5.2.1	JavaScript的语句	219
5.2.2	数据类型	219
5.2.3	变量	220
5.2.4	运算符与表达式	221
5.2.5	功能语句	223
5.2.6	函数	227
5.3	对象化编程	229
5.3.1	对象的基本知识	230
5.3.2	事件处理	232
5.3.3	JavaScript的内部对象	235
5.3.4	JavaScript的自定义类及对象	242
5.4	浏览器对象模型与文档对象模型	247
5.4.1	navigator对象	249
5.4.2	window对象	252
5.4.3	screen对象	258
5.4.4	event对象	259
5.4.5	history对象	261
5.4.6	location对象	262
5.4.7	document对象	264
5.4.8	link对象	266



5.4.9	form对象	267	6.5.2	Node.JS开发实例	329
5.5	JS开发框架技术	275	6.6	更多服务器端开发技术及比较	331
5.5.1	框架技术简介	275	6.6.1	CGI	331
5.5.2	jQuery框架	280	6.6.2	ISAPI/NSAPI	333
5.5.3	Flex	283	6.6.3	PHP	333
5.5.4	框架开发实例	284	6.6.4	不同开发技术之间的比较	335
5.6	JavaScript实例	288	6.7	本章小结	337
5.6.1	document.write()的副作用	288	6.8	思考和练习	337
5.6.2	带动画效果的进度条	288			
5.6.3	旋转变幻文字效果	289	第7章	Web的未来	338
5.6.4	指针式时钟的实现	291	7.1	Web的发展路径	338
5.6.5	一个益智小游戏的实现	294	7.2	XML技术	340
5.7	Ajax技术	298	7.2.1	XML介绍	340
5.7.1	Ajax介绍	298	7.2.2	XML的文档格式	342
5.7.2	Ajax应用与传统的Web应用的比较	300	7.2.3	XML相关技术介绍	345
5.8	本章小结	302	7.2.4	XML的开发工具	349
5.9	思考和练习	302	7.2.5	XML的使用前景	350
			7.2.6	JSON	350
			7.3	WebAssembly技术	351
第6章	服务器端开发		7.3.1	WebAssembly概述	351
	——动态页面技术基础	303	7.3.2	WebAssembly的历史	352
6.1	动态页面基本原理	303	7.3.3	WebAssembly的运行原理	352
6.2	.NET技术	305	7.3.4	WebAssembly的应用	353
6.2.1	ASP.NET简介	305	7.3.5	WebAssembly的现状和发展趋势	354
6.2.2	.NET战略	305	7.4	移动互联网开发与混合开发模式	355
6.2.3	ASP.NET应用的开发实例	307	7.4.1	移动互联网开发简介	355
6.3	Java技术	313	7.4.2	移动互联网应用开发的三种模式	357
6.3.1	Java技术概述	314	7.4.3	混合应用开发框架介绍	358
6.3.2	Servlet	315	7.5	人工智能时代的项目开发	361
6.3.4	JSP	317	7.6	本章小结	364
6.3.5	J2EE	319	7.7	思考和练习	364
6.4	Python开发技术	320			
6.4.1	Python Web应用开发框架	320	参考文献		365
6.4.2	Django的特点	324			
6.4.3	Django实例	325	附录	HTML5代码规范	367
6.5	Node.JS开发技术	328			
6.5.1	Node.JS概述	328			





第 1 章

移动互联网简介

移动互联网将移动通信终端与互联网结合为一体，使用户能够使用手机、PAD或其他无线终端设备，通过速率较高的移动网络，在移动状态下随时、随地(如在地铁、公交车等)访问Internet以获取信息，使用商务、娱乐等各种网络服务。移动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络应用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移动互联网与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应用开发方法。

本章要点：

- 移动互联网与网站技术的发展历程
- 移动互联网的基本概念
- 移动互联网技术基础及高级技术介绍
- 移动互联网应用开发基础

1.1 移动互联网与WWW

1.1.1 移动互联网的发展

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(Defense Advanced Research Projects Agency, DARPA)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。

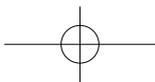
1972年，ARPAnet在首届计算机后台通信国际会议上首次与公众见面，并验证了分组交换技术的可行性，由此，ARPAnet成为现代计算机网络诞生的标志。

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的，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。

当ARPAnet最初于1969年12月建成时只有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快速发展,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,以及对开发和利用信息资源的重视越来越强烈,这些都刺激了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2023年12月,中国网民规模达10.92亿,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

年份	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,发展至今已有30年的时间。中国互联网络信息中心(CNNIC)发布的第53次《中国互联网络发展状况统计报告》显示,截至2023年12月,我国网民规模达10.92亿人,较2022年12月增长2480万人,互联网普及率达77.5%,如图1-1所示。

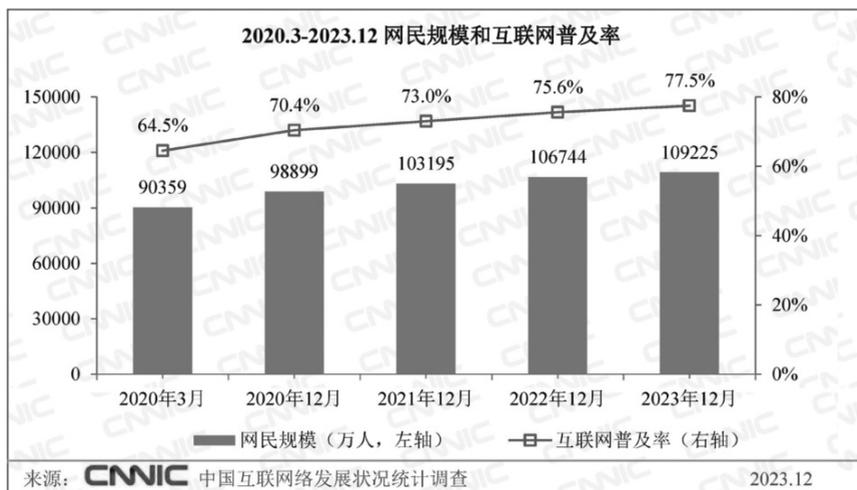


图1-1 2020.3—2023.12网民规模和互联网普及率

另外,值得注意的现象是:2023年,我国移动互联网接入流量达3015亿GB,同比增长15.2%;截至2023年我国网民使用手机上网的比例达99.9%;使用台式电脑、笔记本电脑、电视

和平板电脑上网的比例分别为33.9%、30.3%、22.5%和26.6%。移动上网流量及上网所使用的设备统计数据如图1-2所示。

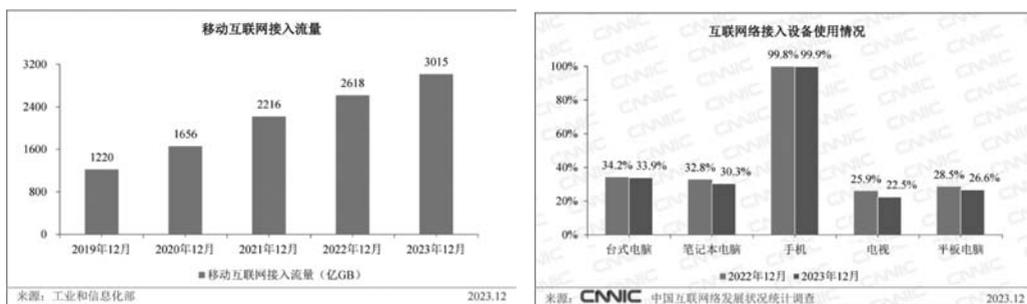


图1-2 移动上网流量及上网所使用的设备统计数据

截至2023年12月,我国网络支付用户规模达9.54亿人,较2022年12月增长4243万人,占网民整体的87.3%。在移动网络发展方面,我国移动电话基站总数达1162万个,其中5G基站总数达334.7万个,占移动基站总数的29.1%;移动互联网应用蓬勃发展,国内市场上监测到的活跃App数量达261万款,进一步覆盖网民日常学习、工作、生活。我国各类互联网应用持续发展,多类应用用户规模获得一定程度的增长。一是网络视频、即时通信、短视频的用户规模仍稳居前三。截至12月,网络视频、即时通信、短视频用户规模分别达10.68亿人、10.60亿人和10.53亿人,用户使用率分别为97.7%、97.0%和96.4%。二是网约车、在线旅行预订、互联网医疗等用户规模实现较快增长。截至2023年12月,网约车、在线旅行预订、互联网医疗的用户规模较2022年12月分别增长9057万人、8629万人、5139万人,增长率分别为20.7%、20.4%和14.2%,成为用户规模增长最快的三类应用。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。

按从事的相关业务分类,网络涉及的业务包括广告公司、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,网上的资源变得越来越丰富。从2023年的统计数据可以看出,移动互联网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移动互联网,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。

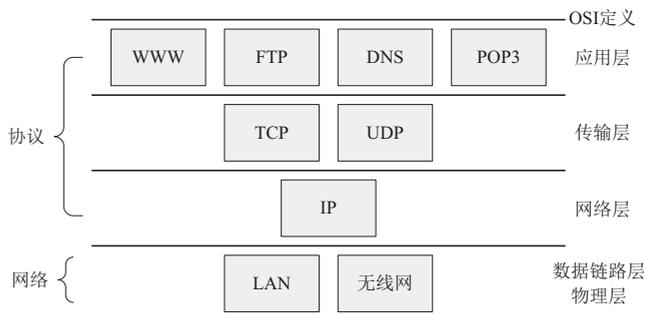


图1-3 TCP/IP模型的网络协议

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，其中最左边的部分定义了最大的网络范围，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} 个(包括地址为零的和全1的)。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。据中国互联网络信息中心截至2023年12月的统计数据，我国IPv6地址数量为68042块/32，IPv6活跃用户数达7.62亿，呈现出稳步增长的态势，我国IPv6地址数量如图1-4所示。

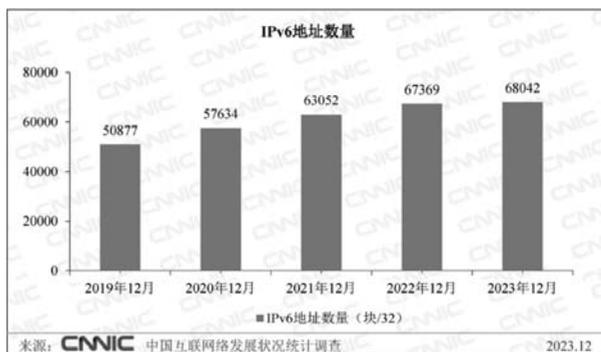


图1-4 我国IPv6地址数量(截至2023年12月)

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。

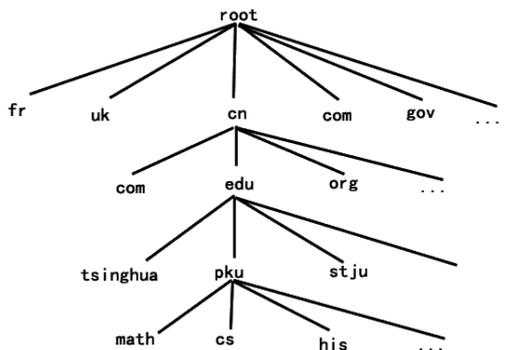


图1-5 Internet域名空间

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

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 常见的域名扩展名及其含义

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
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(European Organization for Nuclear Research, CERN)研制。WWW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, 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(Digital Signal Processor, 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系统已经发展了多年,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、在线会议等业务,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,将信息的副本发送到一组地址,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传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,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)传送给用户,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。

在移动互联网时代的网络环境下,基于B/S等应用模式,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,对电子商务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。在十几年之前,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QQ和微信等;下载工具,如BitTorrent、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过重的问题,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。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 移动互联网概述

1.2.1 移动互联网的发展

首先,早期的Web源于欧洲粒子物理研究所(CERN)的蒂姆·伯纳斯·李(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)的约瑟夫·利克莱德(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，麻省理工学院大卫·克拉克(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。大卫·克拉克的这项研究为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，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称来改善上述问题，这就是域名系统(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(CERN)的蒂姆·伯纳斯·李负责了Enquire(Enquire Within Upon Everything)项目。1989年，蒂姆·伯纳斯·李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。蒂姆·伯纳斯·李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。蒂姆·伯纳斯·李不仅创建了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，蒂姆·伯纳斯·李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蒂姆·伯纳斯·李。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。

然后,随着移动通信网络的全面覆盖,移动互联网伴随着移动网络通信基础设施的升级换代快速发展,尤其是在2009年国家开始大规模部署3G移动通信网络,2014年又开始大规模部署4G移动通信网络。两次移动通信基础设施的升级换代,有力地促进了中国移动互联网快速发展,服务模式和商业模式也随之大规模创新与发展。4G移动电话用户扩张带来用户结构不断优化,支付、视频广播等各种移动互联网应用普及,带动数据流量呈爆炸式增长。

整个移动互联网发展历史可以归纳为四个阶段:萌芽阶段、培育成长阶段、高速发展阶段和全面发展阶段。

(1) 萌芽阶段(2000—2007年)。萌芽阶段的移动应用终端主要是基于WAP(无线应用协议)的应用模式。受限于移动2G网速和手机智能化程度,中国移动互联网发展处在一个简单WAP应用期。

(2) 培育成长阶段(2008—2011年)。3G移动网络建设掀起了中国移动互联网发展新篇章,移动网速的大幅提升初步解决了手机上网带宽瓶颈,移动智能终端丰富的应用软件让移动上网的娱乐性得到大幅提升。

(3) 高速发展阶段(2012—2013年)。随着智能手机的崛起,具有触摸屏功能的智能手机解决了传统键盘机交互上的不便,移动互联网应用呈现了爆发式增长。

(4) 全面发展阶段(2014年至今)。4G网络的建设将移动互联网发展推上了快车道。由于上网网速瓶颈限制得到基本解除,故移动应用场景得到极大丰富。移动互联网应用开始全面发展。桌面互联网时代,门户网站是企业开展业务的标配;移动互联网时代,手机App应用是企业开展业务的标配,4G网络催生了许多公司利用移动互联网开展业务。特别是由于4G网速大大提高,促进了实时性要求较高、流量较大、需求较大类型的移动应用快速发展,许多手机应用开始大力推广移动视频应用。

截至2023年12月,中国累计建成5G基站337.7万个,占移动基站总数的29.1%;中国5G移动电话用户达8.05亿户,占移动电话用户总数的46.6%;千兆宽带接入用户达1.63亿户,占用户总数的25.7%。

1.2.2 移动互联网是什么?

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功能。因此Web或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。

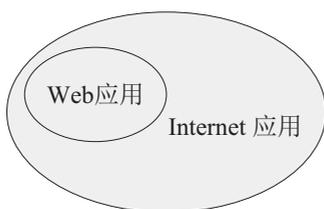


图1-6 Internet和Web的包含关系

❖ 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。

随着移动互联网的普及，网络的应用已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。大量新奇的应用，逐渐渗透到人们生活、工作的各个领域，进一步推动着移动互联网的蓬勃发展。电子阅读、手机游戏、移动视听、移动搜索、移动社区、移动商务、移动支付、位置服务等丰富多彩的移动互联网应用发展迅猛，正在深刻改变信息时代的社会生活，移动互联网正在迎来新的发展浪潮，带来新的应用生态。

1.2.3 移动互联网的技术基础

从技术层面上来看，早期互联网架构的精华主要有3点：用统一资源定位技术(URL)实现全球资源的精确定位；用应用层协议实现分布式的信息传送；以超文本标记语言(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，蒂姆·伯纳斯·李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，作为移动互联网上的一种应用架构，其首要任务就是向人们提供信息和信息服务。

很可惜，在移动互联网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。一个开发人员倘若忘记了其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、视频等资源。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交互。同时设计精良的图形用户界面,使得用户能很容易地查看信息,这一技术架构一直延伸到今天的移动互联网中。

❖ 注意:

移动互联网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,它们都共同构成了移动互联网的技术基础。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。

移动互联网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。

对于移动互联网,<scheme>指的是协议名,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协议进行访问的网络资源。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.njupt.edu.cn:80/index.aspx,http是协议,www.njupt.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。当时蒂姆·伯纳斯·李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。

蒂姆·伯纳斯·李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,并且它仅由不多的标签(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的时代。

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 and 标准。用HTML编写的超本文档称为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Edge等)则提供了交互式的HTML调试器。

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服务器传送到显示终端。

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。

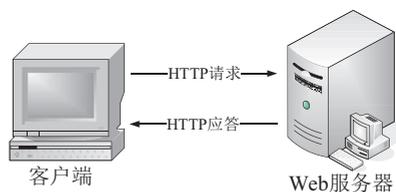


图1-7 HTTP的基本原理

❖ 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浏览器出自蒂姆·伯纳斯·李之手,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。蒂姆·伯纳斯·李和让-弗朗索瓦·格罗夫(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, 伊利诺伊大学厄巴纳-香槟分校(University of Illinois-Urbana-Champaign)超计算应用国家中心的马克·安德森(Marc Andreessen)和埃里克·比纳(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, 它成为字符终端浏览器的标准。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, 这标识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。

根据statcounter的统计数据, 2023年全世界网民所使用的六大浏览器和占比以及各大浏览器占比的逐月变化情况如图1-8所示, 其中目前使用数量最多的是Google的Chrome, 占比达65.31%, Safari为18.31%, 而Edge占比为5.07%。

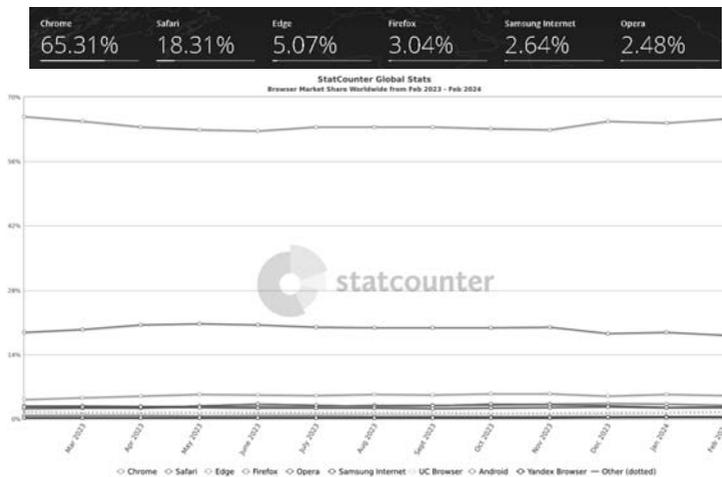


图1-8 2023—2024年全球浏览器占比及变化情况

1.2.4 移动互联网的关键技术

最初的页面都是静态的。用户请求一个资源, 服务再返回这个资源。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, 再发布到多处。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,刚开始时所显示的信息较少,无非是一些联系信息或者只是一些文档等。不过不久之后,用户就开始产生了新的需求,同时移动设备的普及也推动了应用的快速发展,用户越来越希望得到更方便、具有动态性和交互性的交互体验。

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可以创建能够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. 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邀请布兰登·艾奇(Brendan Eich)来设计和实现这种新语言,布兰登·艾奇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页面中链接多个对象,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)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程序有一定难度，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退避三舍，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。

随后诞生了Node.js技术，Node.js发布于2009年5月，由赖安·达尔(Ryan Dahl)开发，是一个基于Chrome V8引擎的JavaScript运行环境，使用了一个事件驱动、非阻塞式I/O模型，让JavaScript运行在服务器端的开发平台，它让JavaScript成为与PHP、Python、Perl、Ruby等服务器端语言平起平坐的脚本语言。这改变了传统的JavaScript仅能应用于客户端的现状，也带来了全新的全栈开发(前后端统一)的理念。

使用Node.js的技术方案，具有如下的特点。

(1) 事件驱动。Node.js的设计思想中以事件驱动为核心，它提供的绝大多数API都是基于事件的、异步的风格。事件驱动的优势在于充分利用了系统资源，执行代码无须阻塞等待某种操作完成，有限的资源可以用于其他任务。在服务器开发中，并发的请求处理是一个大问题，阻塞式的函数会导致资源浪费和事件延迟。通过事件注册、异步函数，开发人员可以提高资源的利用率，性能也会改善。

(2) 异步、非阻塞I/O。在Node.js提供的支持模块中，包括文件操作在内的许多函数都是异步执行的。同时为了方便服务器开发，Node.js的网络模块特别多，包括HTTP、DNS、NET、UDP、HTTPS、TLS等。开发人员可以在此基础上快速构建Web服务器。异步I/O的大致流程如下。

- 发起I/O调用。用户通过JavaScript代码调用Node核心模块，将参数和回调函数传入核心模块。Node核心模块会将传入的参数和回调函数封装成一个请求对象。然后将这个请求对象推入I/O线程池等待执行。JavaScript发起的异步调用结束，JavaScript线程继续执行后续操作。
- 执行回调。I/O操作完成后会将结果存储到请求对象的result属性上，并发出操作完成的通知。每次事件循环时会检查是否有完成的I/O操作，如果有就将请求对象加入I/O观察者队列中，之后当作事件处理。处理I/O观察者事件时会取出之前封装在请求对象中的回调函数，执行这个回调函数，并将result当作参数，以完成JavaScript回调的目的。

(3) 性能出众。Node.js在设计上以单线程模式运行。事件驱动机制是Node.js通过内部单线程高效地维护事件循环队列来完成的，没有多线程的资源占用和上下文切换。这意味着面对大规模的HTTP请求，Node.js是凭借事件驱动来完成的。

(4) 单线程。Node.js是以单线程为基础的，这正是Node.js保持轻量级和高性能的关键。这里的单线程是指主线程为“单线程”，所有阻塞的部分交给一个线程池处理，然后这个主线程通过一个队列跟线程池协作。所写的js代码部分不用再关心线程问题，代码也主要由一堆回调函数来构成，然后主线程在循环过程中适时调用这些代码。

单线程除了保证Node.js高性能，还保证了绝对的线程安全，使开发者不用担心同一变量同时被多个线程读写而造成的程序崩溃。

3. Servlet、JSP、ASP.NET、PHP和Python等

在Java问世一年左右，当时的Sun公司引入了Servlet。Java代码在应用服务器上运行。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.NET还有许多其他的方案，如PHP、Python及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页面内容。

4. 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，即动态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。

5. 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，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。

- **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。

6. 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。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，体现在以下几点。

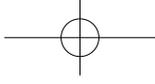
- 所有的标签必须要闭合，也就是说开始标签要有相应的结束标签。
- 所有标签必须小写。
- 所有的参数值，包括数字，都必须放在双引号中。
- 图片必须使用ALT属性来提供说明文字。

7. 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，使网络标准满足当代的网络需求，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的变化主要体现在以下几点。

- 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，将有助于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及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章介绍这项技术。

8. Ajax技术

Ajax(异步JavaScript和XML)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。

1.2.5 移动互联网的未来

移动互联网发展迅猛，将来许多新的技术会带来革命性的进步，以下是一些可能带来变化的重要因素。

1. 6G技术

未来互联网将实现更高速度、更低延迟、更广覆盖的通信网络，5G技术已经逐步商用，并在工业、医疗、农业等领域实现了广泛应用。6G的数据传输速率可能达到5G的50~100倍，时延缩短到5G的十分之一，在峰值速率、时延、流量密度、连接数密度、移动性、频谱效率、定位能力等方面远优于5G。6G网络将是一个地面无线与卫星通信集成的全连接世界。通过将卫星通信整合到6G移动通信，实现全球无缝覆盖，网络信号能够抵达任何一个偏远的乡村，让深处山区的病人能接受远程医疗，让孩子们能接受远程教育。未来，6G技术将成为互联网的重要技术支撑，其可用于更广泛的应用场景，包括虚拟现实、增强现实、自动驾驶、工业自动化等，将大大提升互联网的智能化和物联网化程度。

2. 人工智能技术

人工智能(Artificial Intelligence, AI)是当前互联网发展的重要趋势，其可以实现更高效、更智能、更人性化的服务，推动智慧城市、智慧医疗、智慧交通等领域的发展。未来，人工智能技术将逐步实现感知、认知、决策、执行等多个环节的自动化和智能化，成为未来互联网的重

要支撑。

2023年,以AIGC(ChatGPT)为首的AI下游应用迅速铺开,标志着“场景革命4.0”的到来,推动完成从“万物互联”到“万物赋能”的跨越;随后2024年诞生了可以根据用户的文本提示创建逼真视频的Sora,标志了OpenAI的开发重点正逐步过渡到图像的生成。AI在移动互联网中的应用包括图像识别、语音识别、自然语言理解、推荐系统、对话系统等多个方面,正不断地渗透到人类社会的各个领域。

3. 边缘计算技术

边缘计算是指在靠近物或数据源头的一侧,采用网络、计算、存储、应用核心能力为一体的开放平台,就近提供最近端服务。其应用程序在边缘侧发起,产生更快的网络服务响应,满足行业在实时业务、应用智能、安全与隐私保护等方面的基本需求。边缘计算处于物理实体和工业连接之间,或处于物理实体的顶端。而云端计算,仍然可以访问边缘计算的历史数据。

随着物联网的发展,越来越多的设备连接到互联网上,传统的云计算架构已经无法满足其要求。边缘计算技术将成为未来互联网的重要支撑,其可以将计算和存储资源放置在离设备更近的地方,以提高响应速度和数据处理能力。未来,边缘计算技术将和5G、6G等技术相结合,形成更加智能化和高效的通信与计算环境。

4. 区块链技术

区块链是一种链式存储、不可篡改、安全可信的去中心化分布式账本,它结合了分布式存储、点对点传输、共识机制、密码学等技术,通过不断增长的数据块链记录交易和信息,确保数据的安全和透明性。它起源于比特币,最初由中本聪(Satoshi Nakamoto)在2008年提出,作为比特币的底层技术。

区块链技术具有去中心化、不可篡改、匿名性等特点,已经在金融、供应链、物联网等领域得到广泛应用。从诞生初期的比特币网络开始,区块链逐渐演化为一项全球性技术,吸引了全球的关注和投资。随后,以太坊等新一代区块链平台的出现进一步扩展了应用领域。未来,区块链技术将成为互联网的重要支撑,其可以实现数字资产的交易、数字身份的认证、数据的安全共享等功能,推动数字经济的发展。

5. 元宇宙技术

元宇宙(Metaverse)是指人类运用数字技术构建的,由现实世界映射或超越现实世界,可与现实世界交互的虚拟世界,具备新型社会体系的数字生活空间。“元宇宙”本身并不是新技术,而是集成了一大批现有技术,包括5G、云计算、人工智能、虚拟现实、区块链、数字货币、物联网、人机交互等。

元宇宙的特点包括:社会与空间属性、科技赋能的超越延伸、人-机与人工智能共创、真实感与现实映射性、交易与流通。元宇宙本质上是对现实世界的虚拟化、数字化过程,需要对内容生产、经济系统、用户体验以及实体世界内容等进行大量改造。但元宇宙的发展是循序渐进的,是在共享的基础设施、标准及协议的支撑下,由众多工具、平台不断融合、进化而最终成形的。元宇宙基于扩展现实技术提供沉浸式体验,基于数字孪生技术生成现实世界的镜像,基于区块链技术搭建经济体系,将虚拟世界与现实世界在经济系统、社交系统、身份系统上密切融合,并且允许每个用户进行内容生产和世界编辑。

元宇宙将给人们的生活和社会经济发展带来多方面的变化：从技术创新和协作方式上，进一步提高社会生产效率；催生出一系列新技术新业态新模式，促进传统产业变革；推动文创产业跨界衍生，极大刺激信息消费；重构工作生活方式，大量工作和生活将在虚拟世界发生；推动智慧城市建设，创新社会治理模式。

1.3 移动互联网应用开发的需求与实现架构

网络应用与其开发技术是相互依存而发展的。在移动互联网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。其中，基于Web的应用已经成为一种主流的解决方案。下面就从移动互联网应用开发需求和实现架构的角度，来分析移动互联网应用的开发。

1.3.1 移动互联网应用的需求

自从网络诞生以来，经过几十年的发展，其应用的架构经历了从静态、活动到动态页面的发展历程。

1. 静态页面

静态页面是存储于服务器的页面文件，其内容在开发该页面时就已定义好，并且始终不变。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htm或者html，如图1-9所示。

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，静态页面的开发难度不大。

2. 活动页面

提出活动页面是为了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JavaScript脚本、ActiveX控件、其他插件、H5交互元素等来构成页面的活动属性。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实现一定的交互功能，只要用户程序在运行，该页面就可不断变化保持最新，如图1-10所示的全景数字博物馆交互界面。



图1-9 静态页面

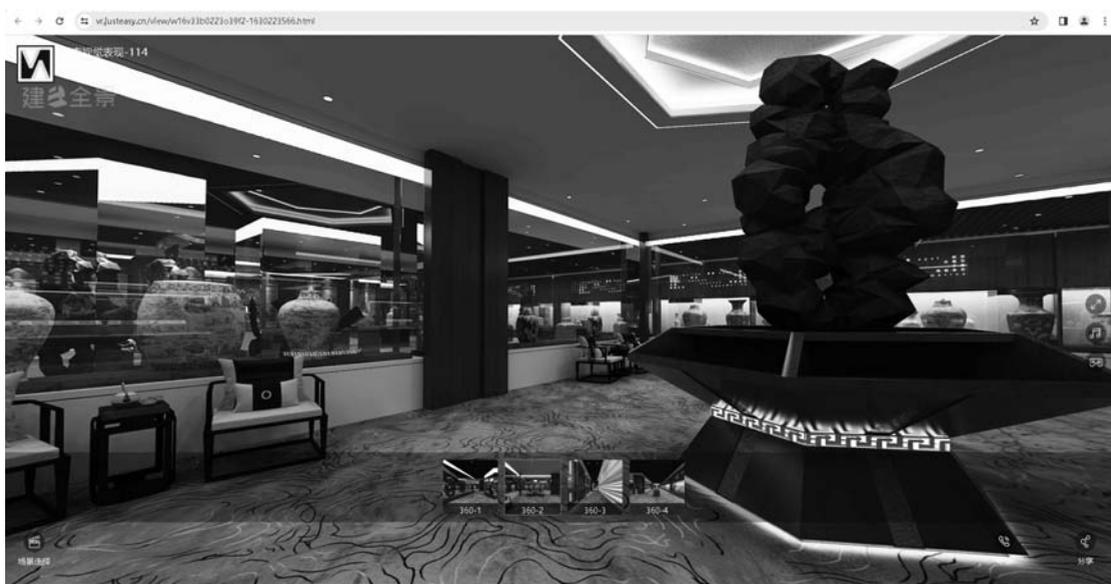


图1-10 活动页面

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JavaScript、ActiveX、H5和插件等。

3. 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、内容等方面的个性化定制。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服务器时，由服务器创建的。当浏览器向服务器发出请求时，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1-11所示的某电商首页。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1-11 动态页面

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(Client/Server，客户机/服务器模式)，再到当前流行的B/S模式(Browser/Server，浏览器/服务器模式)、SOA(面向服务的架构)、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(富互联网应用)、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(业务逻辑)。在这种模式下,服务器仅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。

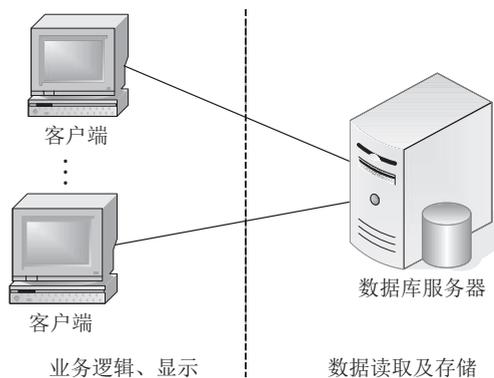


图1-12 两层结构

❖ 注意:

由于客户端的计算任务相对较多,因此通常将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)。

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。但在移动互联网时代,某些移动App是适合采用这种C/S模式的,比较典型的如大型游戏等移动应用。

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。

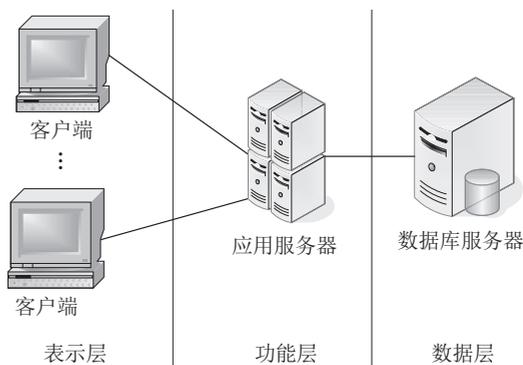


图1-13 三层结构

一般情况下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)。

3. B/S模式

随着Web的兴起，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。但由于此模式下Web服务器的负担最大，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、Node.js或Python等脚本语言、普通的CGI程序、ISAPI或NSAPI等来实现。例如，利用Linux+PHP+MySQL+Apache来构成整个服务体系。

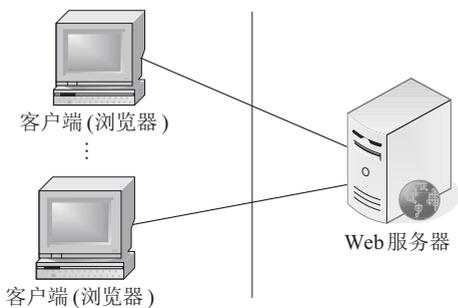


图1-14 静态模式

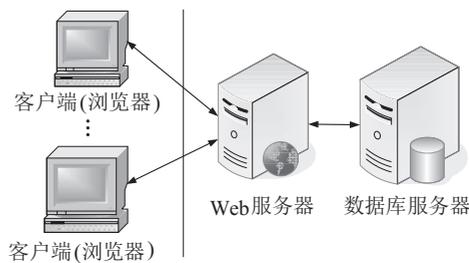


图1-15 一般动态模式

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服务器的负担。

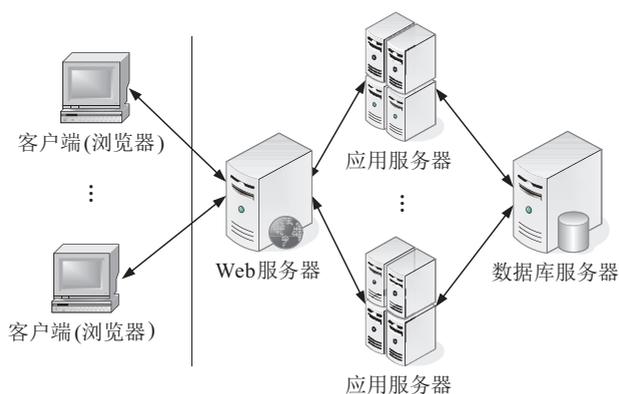


图1-16 多层动态模式

多层动态模式具有负载均衡与容错的功能，这可以通过各种技术来实现，比如，首先部署提供反向代理和负载均衡功能的Nginx，再将ASP.Net脚本结合COM/COM+、将CGI或ISAPI结合COM/COM+、将PHP脚本结合CORBA构件技术。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模式的巨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下一代网络应用。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。富互联网应用结合了声音、视频和实时对话的综合通信技术，使其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：立即部署、跨平台、采用逐步下载

来检索内容和数据；可以充分利用被广泛采纳的互联网标准等。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. 应用框架技术

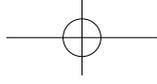
Web应用框架(Web Application Framework)是一种开发框架技术，用来支持动态网站、网络应用及网络服务的开发。框架技术有助于减轻网页开发时共通性活动的工作负载，例如许多框架提供数据库访问接口、标准样板以及会话管理等，可提升代码的可再用性。框架技术可分为基于请求的(request-based)和基于组件的(component-based)两大阵营。前者的代表有Struts和Spring MVC等，后者的代表则有JSF、Tapestry等。

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，这个框架只需要做一点修改，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底层机制交流的框架中，服务器端程序要收集客户端的数据，即Get或Post方式提交的数据，并对其进行转换、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。现代开发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) 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下运行的程序或手机App，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。由于应用需求的日益复杂，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或一些访问量不大的中小型规模的应用。

(4) 对于一些用户规模较大的应用场合，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，比如在电子商务网站中，对数据的严谨性要求也非常严格，因此，在这些情况下，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

移动互联网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移动互联网和应用开发的一些基础知识。为了让读者对Web技术有一个较为全面的认识，本章从移动互联网需求的发展、移动互联网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)，如<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. 本章提到的移动互联网开发技术中哪项技术最为强大？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. 简述使用URL访问网站时的网络传输全过程。
7. 对于某一特定的移动互联网应用建设需求(比如面对大量用户的电商应用)，应该如何选择合适的架构方式？