China-pub.com
下载

第32章 HTTP: World Wide Web

作者：Neal S. Jamison

本章内容包括：

- 万维网(World Wide Web)
- 统一资源定位器(URL)
- Web服务器与浏览器
- 理解HTTP
- 高级主题
- Web语言
- Web的未来

万维网被称为1990年至今最引人注目的应用。没有什么技术或工具像它一样被广泛应用。Web的增长现象是互联网技术重要性和潜力的体现。

32.1 万维网(WWW)

万维网、信息高速公路、网络，无论怎样称呼， Web无疑是个人计算机产生以来最轰动的发明。从90年代初的小实验网络发展到今天拥有 2000万用户， Web在短暂的时间迈出了巨大的一步。在1995年，我们在几千个Web站点间漫无目的地遨游时，还可以知道自己在某个地方。而今天却必须依靠智能搜索引擎的帮助，导航寻找需要信息，甚至通过它在线订购和付款。

本节讨论万维网(WWW)及它短暂但辉煌的历史。

32.1.1 Web简史

World Wide Web最初在CERN，量子物理欧洲实验室开发。为了提高物理学家之间文件的共享及通信。 1993年，国家超级计算机中心 (NCSA)开发出了第一个图形 Web浏览器 —— Mosaic。Web客户方的发展加速了World Wide Web的发展。

Web的创建者和维护者

Tim Berbers-Lee为创建Web的CERN雇员。他编写了第一个Web服务器，定义了Web语言及协议。同时也编写了第一个基本浏览器。

第一个流行的Web服务器(NCSA HTTPd)由Bob McCool在美国国家超级计算机应用中心创建。该服务器是当前最流行的 Web服务器Apache Web服务器的前身。第一个图形界面的浏览器也在NCSA创建，它的开发者是Marc Andreessen，他后来创立了网景通信公司，开发出了众所周知的Netscape Navigator。

Tim Berners-Lee现在是万维网论坛(W3C)的经理， W3C是负责Web及其协议和标准持续发展的主要组织。关于W3C的详细信息及它们的主要工作参见站点： http://www.w3c.org。

另一个重要组织是互联网工程任务组 (IETF)。在组织的指导原则中写道“ 互联网工程任

China-pub.com 下载

务组是一个松散的自由组织，它由致力于互联网和互联网技术发展的人组成”（见http://www.ietf.org/)。因此，它在Web的发展过程中尤其在HTTP的发展中扮演重要的角色。关于IETF在互联网中的作用，请访问站点：http://www.ics.uci.edu/pub/ietf/http/。

还存在许多其他组织致力于互联网和 Web的发展和标准化。它们包括互联网体系结构论坛(IAB)、互联网社团(ISOC)及互联网研究任务组(IRTF)。

32.1.2 Web的发展

在1994年中期， Web由大约3000个Web站点组成，约有 300万人使用 Web。今天，据Nielsen/NetRatings的估计(http://www.nielsen-netratings.com)互联网用户已超过1亿人， Web服务器已超过500万。

注意 在上面提到的统计数字中，“人”指主机。很难统计使用Web的实际人口数；统计主机数相对较为简单，而且任何人都会同意每台机器至少有一个使用者。

Web最初由一群科学家和工程师开发出的不成熟但非常有希望的媒介发展成为可靠、安全的环境，可实现电子商务及其他重要的功能。随着计算机和网络技术的发展，我们可以预见使用Web的人口将不断增长，其功能也将不断增强。

32.2 统一资源定位器

在Web上寻找信息的关键在于了解 Web服务器和客户端如何定位服务器和文件的位置。Web使用统一资源定位器策略(URL)标识Web页和其他资源。

下面是一个URL示例：

http://www.w3c.org/Protocols/index.html

这个URL可以将用户带到万维网论坛的Web站点。它可以分为以下几部分：

Protocol://(协议)

Servername.domain(服务器名.域)

directory/(目录)

file(文件)

在上述示例中：

- 协议是HTTP。
- 全称域命名为www.w3c.org。
- 目录名为Protocols。
- 文件为index.html。

注意 大多数Web服务器都配置为可自动提供缺省主页。在大多数情况下，缺省主页为index.html，其他可能的缺省主页为：home.html、default.html,、home.htm及index.htm。使用这一属性，URL: http://www.w3c.org/Protocols/将返回Protocols目录下的index.html文件。

其他常见的URL为：

ftp://*服务器域名/目录/文件*。

ftp://*用户名@服务器域名/目录/文件*。

China-pub.com
下载

telnet://*服务器域名*。

news://*新闻服务器域名/新闻组*。

以上URL分别表示通过匿名FTP请求文档，使用用户名访问FTP请求文档，使用telnet访问服务器，请求访问usenet新闻组等。

用户也可以使用URL向服务器传递数据。典型应用为向服务器方函数传递参数。例如：

http://*服务器域名/目录/文件*/file.html? *用户名＝Jamison & uid=300*

此URL向file.html主页传递一对参数：用户名Jamison和UID 300。

有时，需要在URL中包含特殊字符如空隔或斜杠 (/)。此时，这些特殊字符必须重新编码以避免服务器出现问题。编码过程 (有时指16进制编码)包括将特殊字符用其16进制的数取代。例如：假设用户需要在URL中列出用户全名：

http://*服务器域名/目录*/file.html? *用户名*＝Neal%20 Jamison

在示例中， Neal和Jaminson间的空隔由与空隔等价的16进制的数取代。在URL中传递的信息通常使用通用网关接口(CGI)程序处理。关于CGI的详细信息参见本章32.6节。

32.3 Web服务器与浏览器

Web服务器是Web的内容提供者。它响应客户端请求，并向客户端提供某种形式的数据。通常，这些数据采用超文本标记语言 (HTML)。服务器也可提供其他形式的数据如：图像、声音、应用程序，甚至是视频。 Web浏览器是Web的客户端。浏览器包括与Web服务器建立通信所需的软件及转换，并显示从服务器方返回数据的软件。表 32-1列出了目前Web中主要的服务器软件。表32-2列出了主流浏览器。

表32-1 主流Web服务器

服 务 器	对应URL
Apache	http://www.apache.org
Microsoft Internet Information Server (IIS)	http://www.microsoft.com/ntserver/web/exec/feature/Datasheet.asp?RLD=71
Netscape Enterprise Server	http://home.netscape.com/enterprise/

表32-2 主流Web浏览器

浏 览 器	对应URL
Netscape Navigator	http://home.netscape.com/browsers/
Microsoft Internet Explorer	http://www.microsoft.com/windows/ie/
Opera	http://opera.nta.no/

Apache与互联网哲学

互联网的主流Web服务器“免费”是不值得惊讶的。互联网就是由黑客与科学家自由的思想和免费的软件构成。据 1999年7月统计，超过56%的Web服务器使用Apache。其次是使用微软IIS中的Web服务器，占22%。

Apache及其他免费软件产品如Perl编程语言和Linux操作系统的成功，再一次掀起了自由软件或源码公开的热潮。目前， Netscape Navigator(http://www.mozilla.org)和AOL Web服务器(http://www.aolserver.com)也加入到这一浪潮中。其他已加入的公

China-pub.com 下载

司包括IBM和Sun公司等。

关于自由软件哲学的详细信息，请参见站点： http://www.gnu.org/philosophy/。

Apache的详细信息可在 http://www.apache.org/上可找到。 Linux信息可在http://www.linux.com上找到。

Web服务器与浏览器的通信协议是超文本传输协议。

32.4 理解HTTP

http协议使Web服务器和浏览器可以通过Web交换数据。它是一种请求/响应协议，即服务器等待并响应客户方请求。 HTTP不维护与客户方的连接，它使用可靠的 TCP连接，通常采用TCP 80端口。客户/服务器传输过程可分为四个基一步骤： 1) 浏览器与服务器建立连接； 2) 浏览器向服务器请求文档； 3) 服务器响应浏览器请求； 4) 断开连接。HTTP是一种无状态协议，它不维护连接的状态信息。

本节讨论HTTP协议的标准版本。

32.4.1 HTTP/1.1

注意 在编写本书时，HTTP/1.1是当前的标准，因此，本书讨论 1.1版本。它在RFC 2616中描述，RFC文档可从http://www.w3c.org/Protocols/中找到。

为了使服务器与客户端通信成为可能， HTTP协议建立了一种由请求和响应消息组成的Web语言。

1. 客户请求

客户请求包含以下信息：

- 请求方法
- 请求头
- 请求数据

请求方法是用于特定URL或Web页面的程序。表32-3列出了可用的请求方法。

表32-3 HTTP请求方法

方 法	描 述
GET	请求指定的文档
HEAD	仅请求文档头
POST	请求服务器接收指定文档作为可执行的信息
PUT	用从客户端传送的数据取代指定文档中的内容
DELETE	请求服务器删除指定页面
OPTIONS	允许客户端查看服务器的性能
TRACE	用于测试—允许客户端查看消息回收过程

头信息是可选项，它用于向服务器提供客户端的其他信息。请求头在表 32-4中显示。

表32-4 HTTP请求头

头	描 述	头	描 述
Accept	客户端接收的数据类型	User-Agent	客户方软件类型
Authorization	认证消息，包括用户名和口令	Referer	用户获取的 Web页面

China-pub.com
下载

如果客户采用某种方法获取数据(如POST)，数据就放在头(header)之后；否则客户机等待从服务器传来的响应。

2. 服务器响应

服务器响应包括以下关键部分：

- 状态码
- 响应头
- 响应数据

HTTP定义了多组返回给浏览器的状态码。表32-5中详细列出了这些状态码信息。

表32-5 HTTP状态码

客户方错误(1xx)		402	需要付费
100	继续	403	禁止
101	交换协议	404	未找到
成功(2xx)		405	方法不允许
200	OK	406	不接受
201	已创建	407	需要代理认证
202	接收	408	请求超时
203	非认证信息	409	冲突
204	无内容	410	失败
205	重置内容	411	需要长度
206	部分内容	412	条件失败
重定向(3xx)		413	请求实体太大
300	多路选择	414	请求URI太长
301	永久转移	415	不支持媒体类型
302	暂时转移	服务器错误	
303	参见其他	500	服务器内部错误
304	未修改	501	未实现
305	使用代理	502	网关失败
客户方错误(4xx)		504	网关超时
400	错误请求	505	HTTP版本不支持
401	未认证		

响应头向客户方提供服务器和/或请求文档的信息。表32-6列出了所有头信息。所有的头均以空行结束。

表32-6 HTTP响应头

方 法	描 述
Server	Web服务器信息
Date	当前日期/时间
Last Modified	请求文档最近修改时间
Expires	请求文档过期时间
Content-length	数据长度(字节)
Content-type	数据MIME类型
WWW-authenticate	用于通知客户方需要的认证信息(如用户名、口令等)

如果有客户方请求的数据，数据放在响应头之后，否则服务器断开连接。

China-pub.com 下载

32.4.2 MIME与Web

多用途互联网邮件扩充 (MIME)在Web中用于指定大量数据的类型 (如文件或Web页面)。MIME使用户可发送多种格式的数据，而不单单是文本数据。由于使用了 MIME，用户可以发送和接收包含非ASCII码数据如声音、视频、图像应用等的页面。

当Web浏览器与服务器建立连接时，它们协商 MIME类型。浏览器向服务器发送它所能接收的MIME类型，这部分信息位于请求头标中。服务器通知客户方它发送的数据包含的 MIME类型。

表32-7列出了在Web中常见的MIME类型。

表32-7 Web中常见的MIME类型

MIME类型	描 述
text/plain	纯ASCII码文本
text/html	HTML文本
image/gif	GIF图像
image/jpeg	JPEG图像
application/msword	Microsoft Word
video/mpeg	MPEG视频
audio/wave	Wave音频
application/x-tar	Tar压缩数据

32.4.3 HTTP通信示例

我们已经讲述了服务器和浏览器通信的机制及可共享的数据类型，下面举例说明协议的工作原理。

1. 请求

在本例中，浏览器请求文档的 URL为http://www.hostname.com/index.html。所有的请求均以空行结束。

```
GET /index.html HTTP/1.1
Accept: text/plain
Accept: text/html
User-Agent: Mozilla/4.5 (WinNT)
              (blank line)
```

浏览器使用Get方法请求文档/index.html。浏览器声明它只能接收纯文本和 html数据，它使用Mozilla/4.5(Netscape)引擎。

2. 响应

服务响应包括状态码、一些头信息 (以空行结束)及请求数据，假设数据存在，则响应信息如下：

```
HTTP/1.1 200 OK
Date Sunday, 15-Jul-99 12:18:03 GMT
Server: Apache/1.3.6
MIME-version: 1.0
Content-type: text/html
```

China-pub.com
下载


```
Last-modified: Thursday, 02-Jun-99 20:43:56 GMT
Content-length: 1423
               (blank line)
<HTML>
<HEAD>
<title>Example Server-Browser Communication</title>
</HEAD>
<BODY>
...
```

假设文档未找到，响应信息如下：

```
HTTP/1.1 404 NOT FOUND
Date Sunday, 15-Jul-99 12:18:03 GMT
Server: Apache/1.3.6
```

32.5 高级主题

本节简单讨论一些与Web相关的高级主题，这些高级主题包括：服务器方功能和安全信息机制。

32.5.1 服务器方功能

Web服务器可以向浏览器提供范围较广的多种类型的数据，包括HTML主页、视频、声音和图像等。这些数据可来自静态主页和文件，也可以根据请求动态产生。动态内容可根据对主页的实际请求动态生成。例如，对电话号码的表的查询结果就需根据用户的查询条件动态产生。用于产生动态数据的技术包括：

- 通用网关接口
- 应用程序接口
- Java Servlets
- 服务器方JavaScript
- 服务器方Include

关于这些技术的详细信息，参见32.6节。

32.5.2 SSL和S-HTTP

安全套接字层(Secure Socket Layer，SSL)和安全HTTP(S-HTTP)是Web上传输敏感信息的协议。

SSL由网景通信公司开发，采用私钥加密算法传送敏感数据。它主要用于加强连接的安全。基于SSL的服务器用https标识，以取代URL中的http。

关于SSL的详细信息，参见http://home.netscape.com/security/techbriefs/ssl.html。

S-HTTP是HTTP的增强版。它主要用于加强发送信息的安全。并非所有的浏览器和服务器都支持S-HTTP。

32.6 Web语言

HTTP提供使Web服务器与浏览器通信的一组规则。但是，使我们对通信感兴趣的是 Web

China-pub.com 下载

编程语言，它向Web的使用者提供他们需要的信息。最通用的 Web编程语言是HTML。但是，还存在其他一些Web编程语言，它们或者与HTML结合使用，或单独使用。本节将讨论常见的Web编程语言。

32.6.1 HTML

超文本标记语言是所有浏览器都可以理解的标准语言。它是一组标明 Web页面内容的标记组成。HTML与平台无关，因此，可以高效地从一个计算机环境传输到另一个计算机环境。这些特性使HTML成为Web中最通用的语言。

注意 HTML是标准通用标记语言(Standard Generalized Markup Language，SGML)的一个应用。SGML是电子文档标记的国际标准。使用SGML，用户可以创建类似于HTML的文档类型定义(Document Type Definitions，DTD)。

HTML使用标记指明信息的表现形式，标记的语法格式如下：

<tag>*信息*</tag>

类型开始于<tag>结束于</tag>，并且标记可以嵌套。

HTML Web页面示例如下：

```
<HTML>
<head>
<title>Sample Page</title>
</head>
<body>
<h1>Hello World</h1>
<b>Bold text</b><br>
<i>Italics</i><br>
<u>Underlined text</u><br>
<li>List Item 1
<li>List Item 2
</body>
</HTML>
```

上例说明了HTML的简单性。在信息两边的标记指定信息的风格、字体类型、颜色等。关于HTML的详细信息，用户可访问站点http://www.builder.com。

32.6.2 XML

扩展标记语言 (XML)是SGML的子集，它使通用的 SGML直接用于Web。XML可看作SGML除去复杂且很少使用的特性后剩下的部分。 XML文档示例如下：

```
<?xml version="1.0'' standalone="yes"?>
<conversation>
<greeting>Hello, world!</greeting>
<response>Hello to you too!</response>
</conversation>
```

虽然XML相对较新，但它已经得到支持。微软的IE 5支持XML，Mozilla(网景公司源码公开的浏览器)也有支持XML的版本，而且正在开发基于 Mozilla的浏览器DocZilla，它可读取

China-pub.com
下载

HTML、XML和SGML(参见http://www.doczilla.com)。

关于XML的详细信息，参见 W3C站点http://www.w3c.org/xml或访问站点http:// www.ucc.ie/xml/。

32.6.3 CGI

通用网关接口(CGI)是Web服务器向基于CGI的程序传递数据的标准。CGI程序可以使用任何语言编写，但它通常使用 C或Perl语言编写。CGI脚本的典型用法是从Web页面上获取信息，处理这些信息然后向用户提供他所需的信息。

关于CGI的详细信息，参见http://www.w3.org/CGI。

Perl

Perl是源码公开的编程语言，通常用于 Web。因为其文本处理能力出众，所以最早用于Unix系统，系统管理员使用它来简化日常工作。在 Web应用中，Perl语言的文本处理能力再一次发挥作用。

下面的示例使用Perl CGI脚本向用户输出信息：

```
#!/usr/local/bin/perl
#
#   helloworld.pl: CGI output sample program.
#

# Print the CGI response header, required for all HTML output
# Follow with an extra \n, to send a blank line
print ''Content-type: text/html\n\n'' ;

# Print simple HTML to STDOUT
print <<EOF ;
<html>
<head><title>CGI Results</title></head>
<body>
<h1>Hello, world.</h1>
</body>
</html>
EOF

exit ;
```

关于Perl的详细信息，请访问站点http://www.perl.org/perl.html。

32.6.4 Java

Java语言由Sun公司开发，与C++类似，也是一种面向对象的程序设计语言。 Java比C++更容易使用而且更加健壮。因此，它是 Web开放的最佳选择。关于 Java的详细信息，参见http://java.sun.com。

Applet与Servlet

应用到Web上的Java主要有两种方式：在客户端或在服务器端。客户方 Java程序(称为

China-pub.com 下载

Applet)从服务器方下载，则客户方执行，其执行区域称为 Sandbox。因为Applet通常较大，下载到客户方往往需要较长时间。出于安全等方面的考虑，在客户方执行时， Applet的功能也受到限制。例如，Applet不可以写访问客户计算机，也不能运行本地程序。

另一种运行Java的方法是在服务器方执行，称为 Servlet。虽然， Servlet可能需要更多的Web服务器处理时间，但它大大削减了将数据传送到客户方的时间，并取消了对于 Applet的安全限制。Servlet的性能比CGI程序好。在执行完成后， Servlet仍保留在内存区中，使下次执行更快。CGI程序不驻留内存。

32.6.5 JavaScript

JavaScript由网景公司开发，使Web开发人员可在他们的主页中添加交互功能。常用的客户方变量有点击数、按钮及其他交互性内容。 JavaScript还可以实现CGI程序的功能。例如：Web格式的错误检查：与以往将数据发送回服务器来确认完整性及正确性不同，通过JavaScript，可以在客户方实现上述操作，这样可获得更高的性能。示例如下：

```
<html>
<head>
<script language="JavaScript">
<!-- Hide

function testsomething(form) {
  if (form.something.value == '''')
    alert(''I said enter something!'')
  else {
   alert(''Thank you!'');
  }
}

// -->
</script>
</head>

<body>
<form name="example">
Enter something:<br>
<input type="text'' name="something"><br>
<input type="button'' name="button'' value="Test Input''
   onClick="testsomething(this.form)">
<P>
</body>
</html>
```

服务器方Java Script是Java Script的一个版本。它运行于服务器方，实现与 CGI类似的功能。Netscape Enterprise Server拥有较强的绑定服务器方 JavaScript能力。关于JavaScript的详细信息，参见http://developer.netscape.com/tech/。

32.6.6 动态服务器页面

微软的动态服务器页面在 Web页面上完成与CGI类似的功能。与CGI不同， ASP仅需要简

China-pub.com
下载

单的脚本语言如JavaScript、VBScript或Visual Basic就可实现。

关于ASP及其支持语言的详细信息，参见 http://msdn.microsoft.com/ workshop /server/asp/ASPover.asp。

32.7 Web的未来

本节讨论Web的发展趋势，虽然Web仅出现几年，但它呈现出强劲的发展势头。在其增长的背后是强大的发展潜力。随着 Web使用人数的不断增长，其功能也不断增加。 HTTP-ng、IPv6、IIOP等技术给Web带来无限生机。

32.7.1 HTTP-ng

作为下一代HTTP协议， HTTP-ng的安全性更好，速度更高。它将更好地支持商业应用。其改进包括：

- 模块化强
- 网络效率更高
- 安全性更好
- 结构简单

关于HTTP-ng的详细信息，参见http://www.w3.org/Protocols/HTTP-NG/。

32.7.2 IIOP

IIOP(互联网对象请求处理协议)为用户提供更有效的 Web访问方式。与 HTTP不同，它只能在Web上传递文本， IIOP可以传递更复杂的数据，如数组和其他对象。 IIOP通过Web实现CORBA(通用对象请求体系结构)。CORBA使程序员开发与平台无关的应用。

关于IIOP及CORBA的详细信息，请访问对象管理组织 (OMG)的主页http://www.omg.org。

32.7.3 IPv6

下一代互联网正在孕育之中。网际协议版本 6(IPv6)也称IPng，在许多方面对正在使用的网际协议IPv4进行了改进：

- 增加了地址空间
- 增加了安全性/私用性
- 提高了网络服务质量

关于IPv6的详细信息，参见http://www.ipv6.org。

32.7.4 IPP

互联网打印协议(IPP)由Novell和Xerox提出并由IETF开发。IPP的基础是HTTP/1.1，它为用户提供多种打印功能：

- 确定可用的打印机
- 提交和取消打印作业。
- 查询打印作业的状态。

关于IPP的详细信息请查阅LETF的主页http://www. ietf. org/html. charters/ipp-charter.html.。

China-pub.com 下载

32.8 小结

本章介绍了万维网及其实现技术。首先我们讲述了统一资源定位器 (URL)及如何使用URL从Web服务器上获取信息，然后讨论了 Web服务器、浏览器及使两者传输信息的协议。其中，我们详细讲述了 HTTP/1.1。随后，我们浏览了多种 Web语言：HTML、Perl和Java等。最后，我们展望Web的未来，介绍了可能改变 Web的革命性技术： HTTP-ng、IIOP、IPv6等。Web URL贯穿本章。读者可以通过这些URL访问相关站点，获取需要的信息。