名词解释[编辑](javascript:;)

Web Server中文名称叫网页服务器或[web服务器](http://baike.baidu.com/view/460250.htm" \t "_blank)。[WEB服务器](http://baike.baidu.com/view/460250.htm)也称为WWW(WORLD WIDE WEB)服务器，主要功能是提供网上信息浏览服务[1]  。

解析

Web服务器可以解析(handles)[HTTP协议](http://baike.baidu.com/view/70545.htm)。当Web服务器接收到一个[HTTP请求](http://baike.baidu.com/view/641736.htm" \t "_blank)(request)，会返回一个HTTP响应(response)，例如送回一个HTML页面。为了处理一个请求(request)，Web服务器可以响应(response)一个[静态页面](http://baike.baidu.com/view/3293095.htm" \t "_blank)或图片，进行页面跳转(redirect)，或者把动态响应(dynamic response)的产生委托(delegate)给一些其它的程序例如CGI脚本，JSP(JavaServer Pages)脚本，servlets，ASP(Active Server Pages)脚本，服务器端(server-side)JavaScript，或者一些其它的服务器端(server-side)技术。无论它们(译者注：脚本)的目的如何，这些服务器端(server-side)的程序通常产生一个HTML的响应(response)来让浏览器可以浏览。

web服务

通俗的讲，Web服务器传送(serves)页面使浏览器可以浏览，然而应用程序服务器提供的是客户端应用程序可以调用(call)的方法(methods)。确切一点，你可以说：Web服务器专门处理HTTP请求(request)，但是应用程序服务器是通过很多协议来为应用程序提供(serves)[商业逻辑](http://baike.baidu.com/view/1569065.htm)(business logic)。

主要web server产品[编辑](javascript:;)

kangle

[kangle](http://baike.baidu.com/view/5939987.htm)[web服务器](http://baike.baidu.com/view/460250.htm)（简称：kangle）是一款跨平台、功能强大、安全稳定、易操作的高性能web服务器和[反向代理](http://baike.baidu.com/view/1165595.htm" \t "_blank)[服务器软件](http://baike.baidu.com/view/4438667.htm)。除此：kangle也是一款专为做[虚拟主机](http://baike.baidu.com/view/7383.htm" \t "_blank)研发的web服务器。实现虚拟主机独立进程、独立身份运行。用户之间安全隔离，一个用户出问题不影响其他用户。安全支持php、asp、net、java、ruby等多种动态开发语言。[1]

nginx

[**Nginx**](http://baike.baidu.com/view/926025.htm)（发音同 engine x）是一款轻量级的Web服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。由俄罗斯的程序设计师Igor Sysoev所开发，供俄国大型的入口网站及搜索引擎Rambler（俄文：Рамблер）使用。其特点是占有内存少，[并发](http://baike.baidu.com/view/684757.htm" \t "_blank)能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：新浪、[网易](http://baike.baidu.com/view/6043.htm" \t "_blank)、腾讯等。

apache

[Apache](http://baike.baidu.com/subview/28283/5418752.htm)是世界使用排名第一的Web[服务器](http://baike.baidu.com/view/899.htm)软件。它可以运行在几乎所有广泛使用的[计算机平台](http://baike.baidu.com/view/2269685.htm)上，由于其[跨平台](http://baike.baidu.com/view/469855.htm)和安全性被广泛使用，是最流行的Web服务器端软件之一。同时Apache音译为[阿帕奇](http://baike.baidu.com/view/19261.htm" \t "_blank)，是北美印第安人的一个部落，叫阿帕奇族，在美国的西南部。也是一个基金会的名称、一种[武装直升机](http://baike.baidu.com/view/115674.htm)等等。

WebServer[编辑](javascript:;)

Web服务器可以解析(handles)HTTP协议。当Web服务器接收到一个HTTP请求(request)，会返回一个HTTP响应(response)，例如送回一个HTML页面。为了处理一个请求(request)，Web服务器可以响应(response)一个静态页面或图片，进行页面跳转(redirect)，或者把动态响应(dynamic response)的产生委托(delegate)给一些其它的程序例如CGI脚本，JSP(JavaServer Pages)脚本，servlets，ASP(Active Server Pages)脚本，服务器端(server-side)JavaScript，或者一些其它的服务器端(server-side)技术。无论它们(译者注：脚本)的目的如何，这些服务器端(server-side)的程序通常产生一个HTML的响应(response)来让浏览器可以浏览。

要知道，Web服务器的代理模型(delegation model)非常简单。当一个请求(request)被送到Web服务器里来时，它只单纯的把请求(request)传递给可以更好的处理请求(request)的程序(译者注：服务器端脚本)。Web服务器除了仅仅提供一个可以执行服务器端(server-side)程序和返回(程序所产生的)响应(response)的环境，而不会提供任何功能。服务器端(server-side)程序通常具有事务处理(transaction processing)，数据库连接(database connectivity)和消息(messaging)等功能。

虽然Web服务器不支持事务处理或[数据库连接池](http://baike.baidu.com/view/84055.htm" \t "_blank)，但它可以配置(employ)各种策略(strategies)来实现[容错性](http://baike.baidu.com/view/2700299.htm" \t "_blank)(fault tolerance)和可扩展性(scalability)，例如[负载平衡](http://baike.baidu.com/view/162341.htm" \t "_blank)(load balancing)，缓冲(caching)。集群特征(clustering—features)经常被误认为仅仅是应用程序服务器专有的特征。

应用程序服务器(the Application Server)

根据我们的定义，作为应用程序服务器，它通过各种协议，可以包括HTTP，把[商业逻辑](http://baike.baidu.com/view/1569065.htm" \t "_blank)暴露给(expose)客户端应用程序。Web服务器主要是处理向浏览器发送HTML以供浏览，而应用程序服务器提供访问商业逻辑的途径以供客户端应用程序使用。应用程序使用此商业逻辑就象你调用对象的一个方法(或过程语言中的一个函数)一样。

应用程序服务器的客户端(包含有[图形用户界面](http://baike.baidu.com/view/185360.htm" \t "_blank)(GUI)的)可能会运行在一台PC、一个Web服务器或者甚至是其它的应用程序服务器上。在应用程序服务器与其客户端之间来回穿梭(traveling)的信息不仅仅局限于简单的显示标记。相反，这种信息就是程序逻辑(program logic)。 正是由于这种逻辑取得了(takes)数据和方法调用(calls)的形式而不是静态HTML，所以客户端才可以随心所欲的使用这种被暴露的[商业逻辑](http://baike.baidu.com/view/1569065.htm" \t "_blank)。

在大多数情形下，应用程序服务器是通过组件(component)的[应用程序接口](http://baike.baidu.com/view/592964.htm" \t "_blank)(API)把商业逻辑暴露(expose)(给客户端应用程序)的，例如基于J2EE(Java 2 Platform, Enterprise Edition)应用程序服务器的EJB(Enterprise JavaBean)[组件模型](http://baike.baidu.com/view/3852154.htm)。此外，应用程序服务器可以管理自己的资源，例如看大门的工作(gate-keeping duties)包括安全(security)，事务处理(transaction processing)，资源池(resource pooling)， 和消息(messaging)。就象Web服务器一样，应用程序[服务器配置](http://baike.baidu.com/view/928821.htm)了多种可扩展(scalability)和容错(fault tolerance)技术。

一个例子

例如，设想一个在线商店(网站)提供实时定价(real-time pricing)和有效性(availability)信息。这个站点(site)很可能会提供一个表单(form)让你来选择产品。当你提交查询(query)后，网站会进行查找(lookup)并把结果内嵌在HTML页面中返回。网站可以有很多种方式来实现这种功能。我要介绍一个不使用应用程序服务器的情景和一个使用应用程序服务器的情景。观察一下这两中情景的不同会有助于你了解应用程序服务器的功能。

不带应用程序服务器的web服务器

在此种情景下，一个Web服务器独立提供在线商店的功能。Web服务器获得你的请求(request)，然后发送给服务器端(server-side)可以处理请求(request)的程序。此程序从数据库或文本文件(flat file，译者注：flat file是指没有特殊格式的非二进制的文件，如properties和XML文件等)中查找定价信息。一旦找到，服务器端(server-side)程序把结果信息表示成(formulate)HTML形式，最后Web服务器把会它发送到你的[Web浏览器](http://baike.baidu.com/view/206703.htm)。

简而言之，Web服务器只是简单的通过响应(response)HTML页面来处理HTTP请求(request)。

Web

情景2和情景1相同的是Web服务器还是把响应(response)的产生委托(delegates)给脚本(译者注：服务器端(server-side)程序)。然而，你可以把查找定价的[商业逻辑](http://baike.baidu.com/view/1569065.htm)(business logic)放到应用程序服务器上。由于这种变化，此脚本只是简单的调用应用程序服务器的查找服务(lookup service)，而不是已经知道如何查找数据然后表示为(formulate)一个响应(response)。 这时当该[脚本程序](http://baike.baidu.com/view/69991.htm)产生HTML响应(response)时就可以使用该服务的返回结果了。

在此情景中，应用程序服务器提供(serves)了用于查询产品的定价信息的[商业逻辑](http://baike.baidu.com/view/1569065.htm)。(服务器的)这种功能(functionality)没有指出有关显示和客户端如何使用此信息的细节，相反客户端和应用程序服务器只是来回传送数据。当有客户端调用应用程序服务器的查找服务(lookup service)时，此服务只是简单的查找并返回结果给客户端。

通过从响应产生(response-generating)HTML的代码中分离出来，在应用程序之中该定价(查找)逻辑的可重用性更强了。其他的客户端，例如收款机，也可以调用同样的服务(service)来作为一个店员给客户结帐。相反，在情景1中的定价查找服务是不可重用的因为信息内嵌在HTML页中了。

总而言之，在情景2的模型中，在Web服务器通过回应HTML页面来处理HTTP请求(request)，而应用程序服务器则是通过处理定价和有效性(availability)请求(request)来提供应用程序逻辑的。

警告

现在，XML Web Services已经使应用程序服务器和Web服务器的界线混淆了。通过传送一个XML有效载荷(payload)给服务器，Web服务器现在可以处理数据和响应(response)的能力与以前的应用程序服务器同样多了。

另外，现在大多数应用程序服务器也包含了Web服务器，这就意味着可以把Web服务器当作是应用程序服务器的一个子集(subset)。虽然应用程序服务器包含了Web服务器的功能，但是开发者很少把应用程序服务器部署(deploy)成这种功能(capacity)(译者注：这种功能是指既有应用程序服务器的功能又有Web服务器的功能)。相反，如果需要，他们通常会把Web服务器独立配置，和应用程序服务器一前一后。这种功能的分离有助于提高性能(简单的Web请求(request)就不会影响应用程序服务器了)，分开配置(专门的Web服务器，集群(clustering)等等)，而且给最佳产品的选取留有余地。