**WEB SERVER**

A **web server** is a [computer program](http://en.wikipedia.org/wiki/Computer_program) that delivers ([serves](http://en.wikipedia.org/wiki/Server_(computing))) content, such as this [web page](http://en.wikipedia.org/wiki/Web_page), using the [Hypertext Transfer Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol). The term web server can also refer to the [computer](http://en.wikipedia.org/wiki/Computer) or [virtual machine](http://en.wikipedia.org/wiki/Platform_virtualization) running the program.

Overview

The primary function of a web server is to deliver web pages ([HTML](http://en.wikipedia.org/wiki/HTML) documents) and associated content (e.g. [images](http://en.wikipedia.org/wiki/Image), [style sheets](http://en.wikipedia.org/wiki/Style_sheet_(web_development)), [JavaScripts](http://en.wikipedia.org/wiki/JavaScript" \o "JavaScript)) to [clients](http://en.wikipedia.org/wiki/Client_(computing)). A client, commonly a [web browser](http://en.wikipedia.org/wiki/Web_browser) or [web crawler](http://en.wikipedia.org/wiki/Web_crawler), makes a request for a specific resource using [HTTP](http://en.wikipedia.org/wiki/HTTP) and, if all goes well, the server responds with the content of that resource. The resource is typically a real file on the server's [secondary memory](http://en.wikipedia.org/wiki/Secondary_memory), but this is not necessarily the case and depends on how the web server is [implemented](http://en.wikipedia.org/wiki/Implementation).

While the primary function is to serve content, a full implementation of HTTP also includes a way of receiving content from clients. This feature is used for submitting [web forms](http://en.wikipedia.org/wiki/Web_form), including[uploading](http://en.wikipedia.org/wiki/Upload) of files.

Many generic web servers also support [server-side scripting](http://en.wikipedia.org/wiki/Server-side_scripting) (e.g. [Apache HTTP Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server) and [PHP](http://en.wikipedia.org/wiki/PHP)). This means that the behaviour of the web server can be [scripted](http://en.wikipedia.org/wiki/Scripting_language) in separate files, while the actual server software remains unchanged. Usually, this functionality is used to create HTML documents on-the-fly as opposed to return fixed documents. This is referred to as [dynamic](http://en.wikipedia.org/wiki/Dynamic_web_page" \o "Dynamic web page)and [static](http://en.wikipedia.org/wiki/Static_web_page) content respectively. The former is primarily used for retrieving and/or inserting information to [databases](http://en.wikipedia.org/wiki/Database). The latter is, however, typically much faster.

Highly niched web servers can be found in devices such as [printers](http://en.wikipedia.org/wiki/Printer_(computing)) and [routers](http://en.wikipedia.org/wiki/Router) in order to ease administration using a familiar [user interface](http://en.wikipedia.org/wiki/User_interface) in the form of a web page.

[[edit](http://en.wikipedia.org/w/index.php?title=Web_server&action=edit&section=2)]History of web servers

[](http://en.wikipedia.org/wiki/File:First_Web_Server.jpg)

[http://en.wikipedia.org/skins-1.5/common/images/magnify-clip.png](http://en.wikipedia.org/wiki/File:First_Web_Server.jpg)

The world's first web server.

In 1989 [Tim Berners-Lee](http://en.wikipedia.org/wiki/Tim_Berners-Lee) proposed to his employer [CERN](http://en.wikipedia.org/wiki/CERN) (European Organization for Nuclear Research) a new project, which had the goal of easing the exchange of information between scientists by using a hypertext system. As a result of the implementation of this project, in 1990 Berners-Lee wrote two programs:

* a [browser](http://en.wikipedia.org/wiki/Web_browser) called [WorldWideWeb](http://en.wikipedia.org/wiki/WorldWideWeb" \o "WorldWideWeb);
* the world's first web server, later known as [CERN httpd](http://en.wikipedia.org/wiki/CERN_httpd), which ran on [NeXTSTEP](http://en.wikipedia.org/wiki/NeXTSTEP" \o "NeXTSTEP).

Between 1991 and 1994 the simplicity and effectiveness of early technologies used to surf and exchange data through the World Wide Web helped to port them to many different operating systems and spread their use among lots of different social groups of people, first in scientific organizations, then in universities and finally in industry.

In 1994 Tim Berners-Lee decided to constitute the [World Wide Web Consortium](http://en.wikipedia.org/wiki/World_Wide_Web_Consortium) to regulate the further development of the many technologies involved (HTTP, HTML, etc.) through a standardization process.

[[edit](http://en.wikipedia.org/w/index.php?title=Web_server&action=edit&section=3)]Common features

1. [**Virtual hosting**](http://en.wikipedia.org/wiki/Virtual_hosting) to serve many web sites using one [IP address](http://en.wikipedia.org/wiki/IP_address).
2. [**Large file support**](http://en.wikipedia.org/wiki/Large_file_support) to be able to serve files whose size is greater than 2 GB on 32 bit [OS](http://en.wikipedia.org/wiki/Operating_system).
3. [**Bandwidth throttling**](http://en.wikipedia.org/wiki/Bandwidth_throttling) to limit the speed of responses in order to not saturate the network and to be able to serve more clients.
4. [**Server-side scripting**](http://en.wikipedia.org/wiki/Server-side_scripting) to generate [dynamic web pages](http://en.wikipedia.org/wiki/Dynamic_web_page), but still keeping web server and web site implementations separate from each other.

[[edit](http://en.wikipedia.org/w/index.php?title=Web_server&action=edit&section=4)]Path translation

Web servers are able to map the path component of a [Uniform Resource Locator](http://en.wikipedia.org/wiki/Uniform_Resource_Locator) (**URL**) into:

* a local file system resource (for static requests);
* an internal or external program name (for dynamic requests).

For a *static request* the URL path specified by the client is relative to the Web server's root directory.

Consider the following URL as it would be requested by a client:

http://www.example.com/path/file.html

The client's web browser will translate it into a connection to www.example.com with the following HTTP 1.1 request:

GET /path/file.html HTTP/1.1

Host: www.example.com

The web server on www.example.com will append the given path to the path of its root directory. On [Unix](http://en.wikipedia.org/wiki/Unix) machines, this is commonly /var/www. The result is the local file system resource:

/var/www**/path/file.html**

The web server will then read the file, if it exists, and send a response to the client's web browser. The response will describe the content of the file and contain the file itself.

[[edit](http://en.wikipedia.org/w/index.php?title=Web_server&action=edit&section=5)]Load limits

A web server (program) has defined load limits, because it can handle only a limited number of concurrent client connections (usually between 2 and 80,000, by default between 500 and 1,000) per [IP address](http://en.wikipedia.org/wiki/IP_address) (and TCP port) and it can serve only a certain maximum number of requests per second depending on:

* its own settings;
* the HTTP request type;
* content origin (static or dynamic);
* the fact that the served content is or is not [cached](http://en.wikipedia.org/wiki/Cache);
* the [hardware](http://en.wikipedia.org/wiki/Personal_computer_hardware) and [software](http://en.wikipedia.org/wiki/Computer_software) limits of the OS where it is working.

When a web server is near to or over its limits, it becomes overloaded and thus unresponsive.

[[edit](http://en.wikipedia.org/w/index.php?title=Web_server&action=edit&section=6)]Kernel-mode and user-mode web servers

A web server can be either implemented into the [OS](http://en.wikipedia.org/wiki/Operating_system) [kernel](http://en.wikipedia.org/wiki/Kernel_(computing)), or in [user space](http://en.wikipedia.org/wiki/User_space) (like other regular applications).

An [in-kernel web server](http://en.wikipedia.org/wiki/In-kernel_web_server) (like [TUX](http://en.wikipedia.org/wiki/TUX_web_server) on Linux or Microsoft [IIS](http://en.wikipedia.org/wiki/Internet_Information_Services) on Windows) will usually work faster because, as part of the system, it can directly use all the hardware resources it needs, such as:

* non-paged memory;
* CPU time-slices;
* network adapters buffers.

Web servers that run in user-mode have to ask the system the permission to use more memory or more CPU resources. Not only do these requests to the kernel take time, but they are not always satisfied because the system reserves resources for its own usage and has the responsibility to share hardware resources with all the other running applications.

Also applications cannot access the system internal buffers, which is causing useless buffer copies that create another handicap for user-mode web servers. As a consequence, the only way for a user-mode web server to match kernel-mode performances is to raise the quality of its code to much higher standards than the code used into another web server that runs in the kernel.

This is more difficult under Windows than under Linux where the user-mode overhead is 6 times smaller than under Windows.[[1]](http://en.wikipedia.org/wiki/Webserver#cite_note-0)

[[edit](http://en.wikipedia.org/w/index.php?title=Web_server&action=edit&section=7)]**Overload causes**

At any time web servers can be overloaded because of:

* **Too much legitimate web traffic.** Thousands or even millions of clients hitting the web site in a short interval of time. (e.g. [Slashdot effect](http://en.wikipedia.org/wiki/Slashdot_effect));
* [**DDoS**](http://en.wikipedia.org/wiki/DDoS)**.** Distributed Denial of Service attacks;
* [**Computer worms**](http://en.wikipedia.org/wiki/Computer_worm) that sometimes cause abnormal traffic because of millions of infected computers (not coordinated among them);
* [**XSS viruses**](http://en.wikipedia.org/wiki/Computer_virus#Cross-site_scripting_virus) can cause high traffic because of millions of infected browsers and/or [web servers](http://en.wikipedia.org/wiki/Web_servers);
* [**Internet web robots**](http://en.wikipedia.org/wiki/Internet_bot)**.** Traffic not filtered/limited on large web sites with very few resources (bandwidth, etc.);
* [**Internet**](http://en.wikipedia.org/wiki/Internet)**(network) slowdowns**, so that client requests are served more slowly and the number of connections increases so much that server limits are reached;
* **Web servers (**[**computers**](http://en.wikipedia.org/wiki/Computer)**) partial unavailability.** This can happen because of required or urgent maintenance or upgrade, hardware or software failures, [back-end](http://en.wikipedia.org/wiki/Back-end) (e.g. [DB](http://en.wikipedia.org/wiki/Database)) failures, etc.; in these cases the remaining web servers get too much traffic and become overloaded.

[[edit](http://en.wikipedia.org/w/index.php?title=Web_server&action=edit&section=8)]**Overload symptoms**

The symptoms of an overloaded web server are:

* requests are served with (possibly long) delays (from 1 second to a few hundred seconds);
* [500, 502, 503, 504 HTTP errors](http://en.wikipedia.org/wiki/List_of_HTTP_status_codes) are returned to clients (sometimes also unrelated [404 error](http://en.wikipedia.org/wiki/HTTP_404) or even [408 error](http://en.wikipedia.org/wiki/List_of_HTTP_status_codes) may be returned);
* [TCP](http://en.wikipedia.org/wiki/Transmission_control_protocol) connections are refused or reset (interrupted) before any content is sent to clients;
* in very rare cases, only partial contents are sent (but this behavior may well be considered a [bug](http://en.wikipedia.org/wiki/Software_bug), even if it usually depends on unavailable system resources).

[[edit](http://en.wikipedia.org/w/index.php?title=Web_server&action=edit&section=9)]**Anti-overload techniques**

To partially overcome above load limits and to prevent overload, most popular web sites use common techniques like:

* **managing network traffic**, by using:
  + [**Firewalls**](http://en.wikipedia.org/wiki/Firewall) to block unwanted traffic coming from bad IP sources or having bad patterns;
  + **HTTP traffic managers** to drop, redirect or rewrite requests having bad [HTTP](http://en.wikipedia.org/wiki/HTTP) patterns;
  + [Bandwidth management](http://en.wikipedia.org/wiki/Bandwidth_management) **and** [traffic shaping](http://en.wikipedia.org/wiki/Traffic_shaping)**, in order to smooth down peaks in network usage;**
* deploying [**web cache**](http://en.wikipedia.org/wiki/Web_cache) techniques;
* using different [domain names](http://en.wikipedia.org/wiki/Domain_name) to serve different (static and dynamic) content by separate Web servers, i.e.:
  + http://images.example.com
  + http://www.example.com
* using different domain names and/or computers to separate big files from small and medium sized files; the idea is to be able to fully [cache](http://en.wikipedia.org/wiki/Cache) small and medium sized files and to efficiently serve big or huge (over 10 - 1000 MB) files by using different settings;
* using many Web servers (programs) per computer, each one bound to its own [network card](http://en.wikipedia.org/wiki/Network_card) and [IP address](http://en.wikipedia.org/wiki/IP_address);
* using many Web servers (computers) that are grouped together so that they act or are seen as one big Web server, see also: [**Load balancer**](http://en.wikipedia.org/wiki/Load_balancer);
* adding more hardware resources (i.e. [**RAM**](http://en.wikipedia.org/wiki/RAM), [disks](http://en.wikipedia.org/wiki/Disk_storage)) to each computer;
* tuning OS parameters for hardware capabilities and usage;
* using more efficient [computer programs](http://en.wikipedia.org/wiki/Computer_program) for web servers, etc.;
* using other [workarounds](http://en.wikipedia.org/wiki/Workaround), especially if dynamic content is involved.