WIKIPEDIA

Web server

Web server refers to server software, or hardware dedicated to running said software, that can serve contents to the World Wide Web. A web server processes incoming network requests over HTTP and several other related protocols.[1]

The inside and front of a Dell PowerEdge web server, a computer designed for rack mounting

Contents

Overview

The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to the text content.

A user agent, commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. The resource is typically a real file on the server's secondary storage, but this is not necessarily the case and depends on how the web server is implemented.

While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting web forms, including uploading of files.

Many generic web servers also support server-side scripting using Active Server Pages (ASP), PHP, or other scripting languages. This means that the behaviour of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this function is used to generate HTML documents dynamically ("on-the-fly") as opposed to returning static documents. The former is primarily used for retrieving or modifying information from databases. The latter is typically much faster and more easily cached but cannot deliver dynamic content.

Multiple web servers may be used for a high traffic website; here, Dell servers are installed together being used for the Wikimedia Foundation.

Web servers are not only used for serving the World Wide Web. They can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring or administering the device in question. This usually means that no additional software has to be installed on the client computer since only a web browser is required (which now is included with most operating systems).

History

In 1989 Sir Tim Berners-Lee proposed a new project to his employer CERN, with the goal of easing the exchange of information between scientists by using a hypertext system. The project resulted in Berners-Lee writing two programs in 1990:

- A browser called WorldWideWeb
- The world's first web server, later known as CERN httpd, which ran on 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 scientific organizations and universities, and subsequently to the industry.

In 1994 Berners-Lee decided to constitute the World Wide Web Consortium (W3C) to regulate the further development of the many technologies involved (HTTP, HTML, etc.) through a standardization process.

The world's first web server, a NeXT Computer workstation with Ethernet, 1990. The case label reads: "This machine is a server. DO NOT POWER IT DOWN!!"

Path translation

Web servers are able to map the path component of a 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.

Sun's Cobalt Qube 3 - a computer server appliance (2002, discontinued)

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

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

The client's user agent 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 an Apache server, this is commonly `/home/www` (on Unix machines, usually `/var/www`). The result is the local file system resource:

```
/home/www/path/file.html
```

The web server then reads the file, if it exists, and sends a response to the client's web browser. The response will describe the content of the file and contain the file itself or an error message will return saying that the file does not exist or is unavailable.

Kernel-mode and user-mode web servers

A web server can be either incorporated into the OS kernel, or in user space (like other regular applications).

Web servers that run in user-mode have to ask the system for 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. Executing in user mode can also mean useless buffer copies which are another handicap for user-mode web servers.

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 (and TCP port) and it can serve only a certain maximum number of *requests per second* (RPS, also known as queries per second or QPS) depending on:

- its own settings,
- the HTTP request type,
- whether the content is static or dynamic,
- whether the content is cached, and
- the hardware and software limitations of the OS of the computer on which the web server runs.

When a web server is near to or over its limit, it becomes unresponsive.

Causes of overload

At any time web servers can be overloaded due to:

- Excess legitimate web traffic. Thousands or even millions of clients connecting to the web site in a short interval, e.g., Slashdot effect;
- Distributed Denial of Service attacks. A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer or network resource unavailable to its intended users;
- Computer worms that sometimes cause abnormal traffic because of millions of infected computers (not coordinated among them)
- XSS viruses can cause high traffic because of millions of infected browsers or web servers;
- Internet bots Traffic not filtered/limited on large web sites with very few resources (bandwidth, etc.);
- 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) partial unavailability. This can happen because of required or urgent maintenance or upgrade, hardware or software failures, back-end (e.g., database) failures, etc.; in these cases the remaining web servers get too much traffic and become overloaded.

Symptoms of overload

The symptoms of an overloaded web server are:

- Requests are served with (possibly long) delays (from 1 second to a few hundred seconds).
- The web server returns an HTTP error code, such as 500, 502, 503, 504, 408, or even 404, which is inappropriate for an overload condition.
- The web server refuses or resets (interrupts) TCP connections before it returns any content.
- In very rare cases, the web server returns only a part of the requested content. This behavior can be considered a bug, even if it usually arises as a symptom of overload.

Anti-overload techniques

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

- Managing network traffic, by using:
  - Firewalls 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 patterns

  - Bandwidth management and traffic shaping, in order to smooth down peaks in network usage
- Deploying web cache techniques
- Using different domain names to serve different (static and dynamic) content by separate web servers, e.g.:
  - http://images.example.com
  - http://www.example.com
- Using different domain names or computers to separate big files from small and medium-sized files; the idea is to be able to fully cache small and medium-sized files and to efficiently serve big or huge (over 10 - 1000 MB) files by using different settings
- Using many internet servers (programs) per computer, each one bound to its own network card and IP address
- Using many internet servers (computers) that are grouped together behind a load balancer so that they act or are seen as one big web server
- Adding more hardware resources (i.e. RAM, disks) to each computer
- Tuning OS parameters for hardware capabilities and usage
- Using more efficient computer programs for web servers, etc.
- Using other workarounds, especially if dynamic content is involved

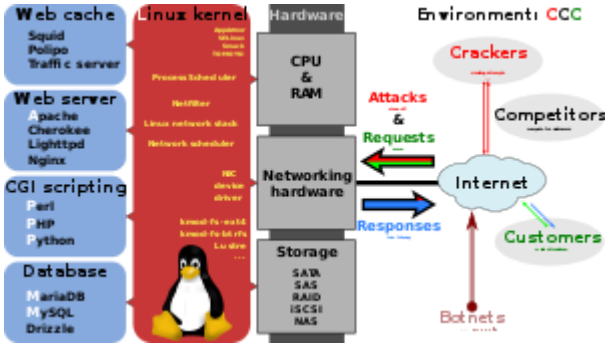
Market share

July 2018

Below are the latest statistics of the *market share of all sites* of the top web servers on the Internet by W3Techs Usage of Web Servers for Websites (https://w3techs.com/technologies/overview/web_server/all).

Product	Vendor	Percent
Apache	Apache	45.9%
nginx	NGINX, Inc.	39.0%
IIS	Microsoft	9.5%
LiteSpeed Web Server	LiteSpeed Technologies	3.4%
GWS	Google	1.0%

All other web servers are used by less than 1% of the websites.



The LAMP (software bundle) (here additionally with Squid), composed entirely of free and open-source software, is a high performance and high-availability heavy duty solution for a hostile environment

February 2017

Below are the latest statistics of the *market share of all sites* of the top web servers on the Internet by Netcraft February 2017 Web Server Survey (https://news.netcraft.com/archives/2017/02/27/february-2017-web-server-survey.html).

Product	Vendor	January 2017	Percent	February 2017	Percent	Change	Chart color
IIS	Microsoft	821,905,283	45.66%	773,552,454	43.16%	-2.50	red
Apache	Apache	387,211,503	21.51%	374,297,080	20.89%	-0.63	black
nginx	NGINX, Inc.	317,398,317	17.63%	348,025,788	19.42%	1.79	green
GWS	Google	17,933,762	1.00%	18,438,702	1.03%	0.03	blue

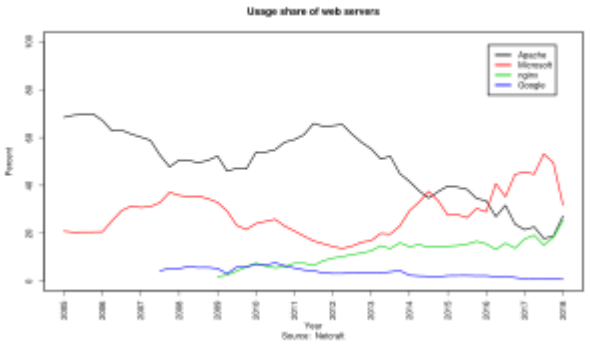


Chart:
Market share of all sites of major web servers 2005-2018

February 2016

Below are the latest statistics of the *market share of all sites* of the top web servers on the Internet by Netcraft February 2016 Web Server Survey (http://news.netcraft.com/archives/2016/02/22/february-2016-web-server-survey.html).

Product	Vendor	January 2016	Percent	February 2016	Percent	Change	Chart color
Apache	Apache	304,271,061	33.56%	306,292,557	32.80%	0.76	black
IIS	Microsoft	262,471,886	28.95%	278,593,041	29.83%	0.88	red
nginx	NGINX, Inc.	141,443,630	15.60%	137,459,391	16.61%	-0.88	green
GWS	Google	20,799,087	2.29%	20,640,058	2.21%	-0.08	blue

Apache, IIS and Nginx are the most used web servers on the Internet.

See also

- Server (computing)
- Application server
- Comparison of web server software
- HTTP compression
- Open source web application
- SSI, CGI, SCGI, FastCGI, PHP, Java Servlet, JavaServer Pages, ASP, ASP.NET, SAPI
- Variant object
- Virtual hosting

- Web hosting service
- Web container
- Web proxy
- Web service

References

1. "What is web server?'" (http://www.webdevelopersnotes.com/basics/what_is_web_server.php). *webdevelopersnotes*. 2010-11-23. Retrieved 2010-11-23.

External links

- RFC 2616, the Request for Comments document that defines the HTTP 1.1 protocol