

Comparación entre los principales servidores Web

• APACHE

Apache es el servidor web mas popular, se trata de un software libre y de código abierto y cuenta entre sus principales virtudes lo de disponer de una gran estabilidad y robustez.

Apache está diseñado para ser un Servidor Web potente y flexible que pueda funcionar en la mas amplia variedad de plataformas y entornos.

Apache se usa principalmente para el envío de paginas dinámicas y estáticas en la WWW y tiene incorporado en su soporte una amplia gama de lenguajes de programación web como Perl , PHP y Python.

Además es muy popular en el conjunto de aplicaciones XAMP.

Dada su alta capacidad de configurabilidad y modularidad permite ampliar fácilmente tanto sus capacidades como la creación de logs con los que aumentar el control sobre lo que sucede en el servidor, es ampliamente aceptado en la red y puede conectarse directamente a una base de datos.

Al tratarse de software de código abierto, programadores de todo el mundo contribuyen de forma constante a su mejoría, lo cual permite a Apache actualizarse constantemente.

Apache tiene muchas características positivas, pero su gran deficiencia es el rendimiento, Apache por si mismo no es el más ligero ni el más rápido, cosa que es sumamente importante en la web actual.

• NGINX

Nginx es un servidor web ligero, fácil de instalar, de software libre y código abierto, por lo que es capaz de funcionar en multitud de sistemas.

Está construido mediante una arquitectura asíncrona que deja muy poca huella en la web y consume muy poco recursos, haciéndolo ideal para manejar múltiples y cambiantes activas paginas web.

Esta característica de ser asíncrono (como Lighttpd también) le permite una gran escalabilidad y velocidad.

Una desventaja de Nginx es que no tiene módulos para servir contenido dinámico como PHP, Python , Ruby etc. Y para servir este contenido utiliza herramientas externas.

Nginx se puede utilizar junto a Apache y en efecto algunas empresas solo usan Nginx para servir contenido estático y Apache para el contenido dinámico.

Este servidor se puede utilizar como cache permitiendo de mejorar la eficiencia de las aplicaciones sin tocar la programación de las mismas y puede funcionar también como balanceador de carga distribuyendo el trafico entro varios servidores, permitiendo mayor escalabilidad.

• **CHEROKEE**

Cherokee Web Server es un servidor de código abierto y multiplataforma que surgió en 2001 de la mano de un desarrollador español.

La motivación del proyecto fue construir un servidor más nuevo que el servidor NCSA HTTPd y no tan grande y pesado como Apache.

Este servidor tiene una excelente interfaz web desde la cual se puede configurar el servidor web, configurar los virtual hosts y incluso optimizar el servidor web para funcionar mejor. Esta característica grafica es algo que no tiene ningún servidor web de forma predeterminada, sería necesario instalar un panel de control.

Cherokee es un servidor muy eficiente, veloz, escalable y modular; Dado su reducido núcleo es muy apto para su uso en sistemas empujados.

Además incluye herramientas y utilidades integradas que permite transmitir streaming de audio y video de forma optimizada adaptándose automáticamente a la velocidad de la conexión del visitante.

Otra peculiaridad de este servidor es que puede ser actualizado en caliente sin necesidad de parar o reiniciar el servicio, una característica imprescindible en cualquier servidor que ofrezca un servicio pero que no ofrece ningún otro servidor web.

• **LIGHTTPD**

Lighttpd es también un servidor de software libre, rápido, seguro y flexible especialmente recomendado a servidores con mucha carga y donde la velocidad sea una cuestión prioritaria, ya que Lighttpd es muy ligero y hace muy poco uso tanto de la RAM como de la CPU.

Función con múltiples plataforma, es muy flexible y tiene una arquitectura asíncrona.

Lighttpd sigue ser menos popular de Nginx porque el primero funciona como un "single process with a single thread and non-blocking" y el segundo trabaja como un proceso maestro que delega su trabajo a los procesos de trabajo.

• **NODE.JS**

Node.js es un intérprete Javascript cuya meta es permitir a un programador construir aplicaciones altamente escalables y escribir código que manejen montones de conexiones simultáneas en una única máquina física.

Esta tecnología es basada en el motor de Javascript V8 de Google y el lenguaje de programación es el Javascript con lo que se utilizaría el mismo lenguaje en el servidor que en el cliente.

Node.js está orientada a eventos y no bloqueo por esto es muy bueno por manejo de solicitudes simultáneas. Es un tipo de servidor más veloz que otros, en efecto utilizando un servidor web tradicional, como pueda ser Apache, cada vez que solicitas un recurso web, éste crea un hilo separado, o invoca un nuevo proceso, para atender dicha solicitud. A pesar de que Apache responde rápidamente a las solicitudes, y limpia el proceso una vez que ha terminado, este sistema puede necesitar un montón de recursos. Una aplicación web con muchas visitas, puede tener serios problemas de rendimiento.

Node.js, por el contrario, utiliza un sistema de E/S asíncrono y basado en eventos y callbacks de funciones. Es decir, todo lo realiza en un único hilo: se queda escuchando a ciertos eventos que ocurran en nuestra aplicación, y cuando ocurren, responde en consecuencia, de modo asíncrono, es decir, un evento no bloquea a otro. Dicho de otro modo, crea miles de subprocesos y trata a sus outputs como streams. Esto supone una mayor velocidad de respuesta. Además node.js tiene la posibilidad de hacer extensiones escritas en C/C++ para optimizar ciertas acciones que consuman muchos recursos. Este servidor web es menos eficiente con el envío de imágenes o archivos CSS muy pesados y va a funcionar igual o un poco peor que otros servidores del mercado.

Existen varios casos donde no tiene sentido utilizar Node.js. Por ejemplo, cuando nuestra aplicación necesite muchísima CPU, y tenga pocos procesos de E/S, como podría ser, codificando vídeo. Para eso, casi mejor utilizar otros lenguajes como C o C++. Otro ejemplo donde no vamos a ver una gran ventaja utilizando Node.js, es si estamos programando una aplicación clásica contra base de datos: alta, baja, edición y modificación de registros. Aquí, Node.js no te va a dar más ventajas de las que te pueda dar PHP, o Ruby, ya que trabajarás con un único canal de E/S.

Donde si es adecuado Node.js, es en aplicaciones de una sola página, con mucho AJAX, o cuando necesitas hacer muchas cosas al mismo tiempo, sobre todo muchas operaciones E/S (acceso a ficheros, bases de datos,...) a la vez.

También obtendrás una gran ventaja en velocidad y rendimiento, con aplicaciones en tiempo real, que necesitan mantener una conexión persistente entre el navegador y el servidor (juegos online, chats, herramientas de colaboración, etc).

• **TOMCAT**

Tomcat es un servidor web con soporte para Servlets y Java Server Pages, en efecto funciona como contenedor de servlets y JSP's y incluye un compilador Jasper.

Tomcat al principio no era concebido como un servidor de aplicaciones y se presentaba en combinación con el servidor web Apache, pero hoy en día Tomcat puede funcionar como servidor de aplicaciones por si solo.

Apache por si solo es incapaz de ejecutar el contenido dinámico de algunas paginas y ahí es donde entra a trabajar Tomcat quien facilita la ejecución de estos servlets o JSP .

Tomcat es particularmente usado en entornos de alto nivel de trafico y funciona en cualquier dispositivo que cuente con la maquina virtual de JAVA.