Modelos de ejecución en servidores web

En esta práctica investigaremos sobre varios modelos de ejecución que son utilizados por servidores web. Para cada uno de ellos explicaremos su definición de modelo, que servidores lo utilizan, ventajas e inconvenientes y casos en los que se recomienda su uso.

Modelo de ejecución basado en procesos

Definición de modelo:

Este es un modelo en el que el proceso principal se encuentra escuchando las peticiones. Una vez llega una petición, el proceso hace un fork y crea una copia exacta en la que se atiende la petición, mientras el proceso principal continúa atendiendo nuevas peticiones que puedan llegar.

Podríamos decir que el modelo basado en procesos es como un restaurante en el que hay una persona atendiendo en la entrada. Esta persona, cada vez que viene un cliente, llama a un empleado nuevo para atender a cada cliente, lo que hace que sean necesarios muchos empleados para todos los clientes que vayan entrando.

Ejemplo de servidor:

Un servidor que utilizaba el modelo basado en procesos era Apache en su modelo Prefork.

Ventajas e inconvenientes:

Las principales ventajas de este modelo son su simplicidad gracias a la facilidad de entendimiento de su arquitectura. Además, al crear procesos independientes ofrece un aislamiento en caso de fallos.

La gran desventaja es el bajo rendimiento que proporciona, crear y eliminar los procesos son tareas de gran carga para el sistema operativo y consumen una gran cantidad de tiempo.

Casos de uso recomendados:

Este modelo es recomendable en casos en los que vayan a haber una cantidad baja de peticiones con la que no afecte casi al rendimiento.

Modelo de ejecución basado en hilos

Definición de modelo:

El modelo basado en hilos, al igual que el de procesos, tiene un proceso principal escuchando peticiones. Cuando llegan peticiones, el proceso crea hilos que comparten recursos para atender la petición mientras el proceso principal escucha nuevas peticiones.

Este modelo sigue la analogía anterior del restaurante, donde el encargado de recibir a los clientes llama a un empleado para el cliente que entra, con la diferencia de que los diversos empleados pueden atender a los clientes de otros.

Ejemplo de servidor:

Un servidor de ejemplo que usa el modelo de hilo es Apache, que contiene esta arquitectura en su servidor.

Ventajas e inconvenientes:

La ventaja más grande, a diferencia del modelo de procesos, es el mayor rendimiento que ofrece al manejar solicitudes de forma paralela. La capacidad de gestionar múltiples solicitudes simultáneamente permite al servidor manejar un mayor número de conexiones concurrentes sin perder rendimiento.

Por otro lado, este modelo introduce un nivel de complejidad superior al de procesos, debido al manejo de comunicación y sincronización entre hilos.

Casos de uso recomendados:

El principal caso de uso sería para servidores web que necesitan atender a varios clientes al mismo tiempo.

Modelo de ejecución dirigido por eventos

Definición de modelo:

Se basa en sockets no bloqueantes, es decir, en espacios de memoria compartida entre procesos de dos aplicaciones en un cliente y un servidor. Además, consta de lectura y escritura asíncrona y bidireccional entre sockets. Este modelo espera a que sucedan eventos para ejecutar una función.

Para entenderlo mejor podemos imaginar que somos el organizador de una fiesta. Cuando alguien llega a la fiesta, le ponemos un gorro; cuando alguien pide una bebida, se la servimos o cuando suena la música, cambiamos las luces.

Ejemplo de servidor:

Un ejemplo de servidor basado en ejecución dirigido por eventos es Node.js. Utiliza un bucle de eventos asíncrono y no bloqueante para gestionar múltiples operaciones simultáneamente con un único hilo.

Ventajas e inconvenientes:

Este modelo ofrece ventajas como componentes desacoplados que pueden escalar de forma independiente o el bajo acoplamiento entre los servicios que permite a los desarrolladores añadir, modificar o eliminar componentes sin afectar a otras partes del sistema.

También presenta algunas dificultades como el poco control sobre la infraestructura, lo que dificulta el monitoreo y la depuración del backend de las operaciones.

Casos de uso recomendados:

Es recomendable usar este modelo para aplicaciones que requieren procesamiento en tiempo real, escalabilidad y acoplamiento mínimo.

Modelo de ejecución implementado en el Kernel

Definición de modelo:

En este modelo se utiliza un espacio de trabajo que pertenece al sistema operativo y no en el área de trabajo. Las operaciones de los procesos están en un servidor que se ejecuta en el modo Kernel.

Por ejemplo, en una oficina donde hay muchos empleados, cada vez que un empleado necesita realizar alguna acción, como sacar una fotocopia, no lo hace el mismo. Tiene que pedirlo al jefe de la oficina, que es el único que tiene acceso a la fotocopiadora.

Ejemplo de servidor:

Un ejemplo de servidor es el servidor de archivos UNIX o Linux.

Ventajas e inconvenientes:

El modelo presenta ventajas como un menor consumo de memoria, CPU y espacio en disco gracias a los unikernels. Además, presenta una mayor seguridad ya que tiene menos código y se reducen las vulnerabilidades.

También presenta desventajas como la dependencia del Kernel, si un servicio falla en el Kernel, puede afectar a todo el sistema operativo dejándolo inservible.

Casos de uso recomendados:

El mejor de los casos para usarlo es en sistemas donde los componentes del sistema operativo no cambian mucho y la estabilidad del código del kernel esté probada y sea segura.

Conclusión

Para una aplicación web con alta concurrencia, el modelo de ejecución basado en hilos es el mejor que podemos usar. Este modelo nos permite que múltiples hilos se ejecuten de manera concurrente dentro de un mismo proceso compartiendo memoria y recursos. Este modelo ofrece un buen equilibrio entre rendimiento, escalabilidad y simplicidad para aplicación web altamente concurrentes. Frente a otros modelos diferentes, este ofrece una programación más natural y estructurada facilitando el desarrollo y mantenimiento del código.