

Servidor

Virginia Padilla

Universidad Nacional Experimental de Guayana

virginiapadillas@gmail.com

8 de mayo de 2025

Contenido

1 Servidor

- Problemas generales de diseño
 - Servidores Concurrentes versus Iterativos
 - Ponerse en contacto con un servidor: puntos finales
 - Interrupción de un Servidor
 - Servidores sin estados versus servidores con estado
- Cluster de Servidores
 - Cluster de Área Local
 - Cluster de Área Amplia
 - Caso de Estudio: PlanetLab

Servidores: Organización General

Modelo básico

Un proceso que implementa un servicio específico en nombre de una colección de clientes. Espera una solicitud entrante de un cliente y se asegura de que se atiende la solicitud, después espera la próxima solicitud entrante.

Servidores Concurrentes

Dos tipos básicos

- Servidor iterativo: el servidor maneja la solicitud antes de asistir a la siguiente solicitud.
- Servidor concurrente: utiliza un despachador, que recoge un mensaje entrante (solicitud) que luego se pasa a un hilo/proceso separado.

Observación

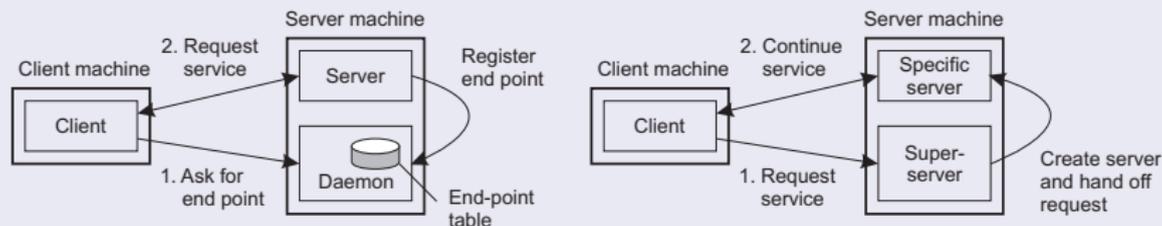
Los servidores concurrentes son la norma: pueden manejar fácilmente múltiples solicitudes, especialmente en presencia de operaciones de bloqueo (a discos u otros servidores)

Ponerse en contacto con un servidor

Observación: la mayoría de los servicios están vinculados a un puerto específico.

ftp-data	20	File Transfer [Default Data]
ftp	21	File Transfer [Control]
telnet	23	Telnet
smtp	25	Simple Mail Transfer
www	80	Web (HTTP)

Asignar dinámicamente un punto final



Comunicación fuera de banda

Problema

¿Es posible interrumpir un servidor una vez que ha aceptado (o procesando) una solicitud de servicio?

Comunicación fuera de banda

Problema

¿Es posible interrumpir un servidor una vez que ha aceptado (o procesando) una solicitud de servicio?

Solución 1: use un puerto separado para datos urgentes

- El servidor tiene un hilo/proceso separado para mensajes urgentes
- Mensaje urgente \Rightarrow la solicitud asociada se pone en espera
- Nota: requiere que el sistema operativo admita la programación basada en prioridades

Comunicación fuera de banda

Problema

¿Es posible interrumpir un servidor una vez que ha aceptado (o procesando) una solicitud de servicio?

Solución 1: use un puerto separado para datos urgentes

- El servidor tiene un hilo/proceso separado para mensajes urgentes
- Mensaje urgente \Rightarrow la solicitud asociada se pone en espera
- Nota: requiere que el sistema operativo admita la programación basada en prioridades

Solución 2: utilizar instalaciones de la capa de transporte

- Ejemplo: TCP permite mensajes urgentes en la misma conexión
- Los mensajes urgentes pueden capturarse utilizando técnicas de señalización del sistema operativo

Servidores y estado

Servidores sin estado

Nunca guarde información sobre el estado de un cliente después de haber manejado una solicitud:

- No registre si se ha abierto un archivo (ciérrelo después del acceso)
- No prometa invalidar el caché de un cliente
- No haga un seguimiento de sus clientes.

Servidores y estado

Servidores sin estado

Nunca guarde información sobre el estado de un cliente después de haber manejado una solicitud:

- No registre si se ha abierto un archivo (ciérrelo después del acceso)
- No prometa invalidar el caché de un cliente
- No haga un seguimiento de sus clientes.

Consecuencias

- Los clientes y servidores son completamente independientes.
- Se reducen las inconsistencias de estado debido a bloqueos del cliente o servidor
- Posible pérdida de rendimiento porque, por ejemplo, un servidor no puede anticipar comportamiento del cliente (ejm: captación previa de bloques de archivos)

Servidores y estado

Servidores sin estado

Nunca guarde información sobre el estado de un cliente después de haber manejado una solicitud:

- No registre si se ha abierto un archivo (ciérrelo después del acceso)
- No prometa invalidar el caché de un cliente
- No haga un seguimiento de sus clientes.

Consecuencias

- Los clientes y servidores son completamente independientes.
- Se reducen las inconsistencias de estado debido a bloqueos del cliente o servidor
- Posible pérdida de rendimiento porque, por ejemplo, un servidor no puede anticipar comportamiento del cliente (ejm: captación previa de bloques de archivos)

Pregunta

¿La comunicación orientada a la conexión encaja en un diseño sin estado?

Servidores y estado

Servidores con estado

Realiza un seguimiento del estado de sus clientes:

- Registre que se ha abierto un archivo, de modo que se pueda realizar la captación previa
- Sabe qué datos ha guardado en caché un cliente y permite que los clientes se mantengan locales copias de datos compartidos

Servidores y estado

Servidores con estado

Realiza un seguimiento del estado de sus clientes:

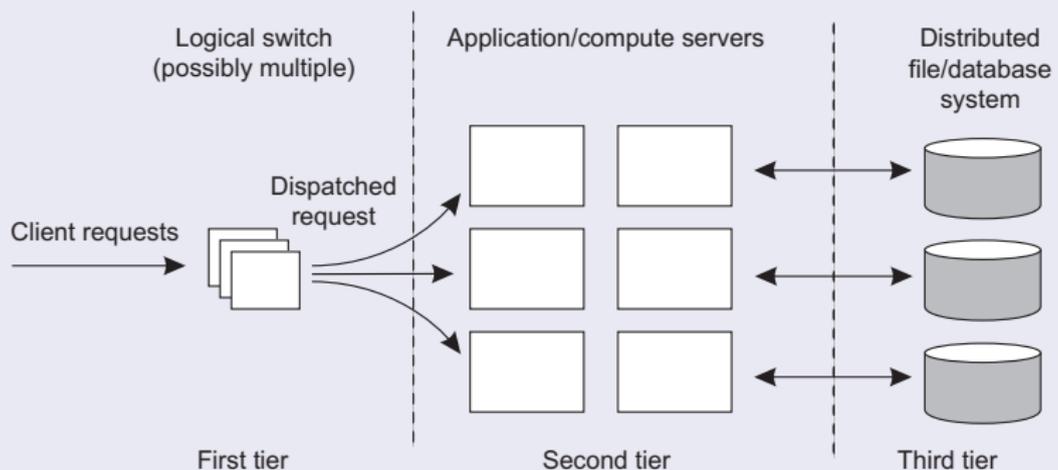
- Registre que se ha abierto un archivo, de modo que se pueda realizar la captación previa
- Sabe qué datos ha guardado en caché un cliente y permite que los clientes se mantengan locales copias de datos compartidos

Observación

El rendimiento de los servidores con estado puede ser extremadamente alto, siempre que los clientes pueden guardar copias locales. Como resultado, la fiabilidad no es un problema.

Cluster de Área Local: Tres niveles diferentes

Organización común



Elemento Crucial

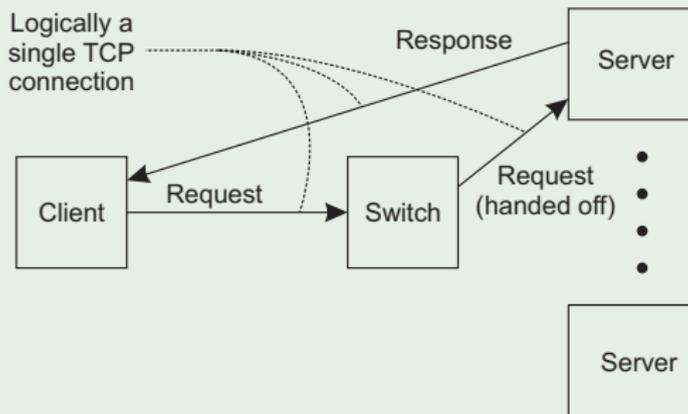
El primer nivel es generalmente responsable de pasar las solicitudes a un servidor: ejm, solicitud de despacho

Manejo de solicitudes

Observación

Hacer que el primer nivel maneje todas las comunicaciones desde/hacia el clúster puede conducir a una embotellamiento.

Una solución: transferencia TCP

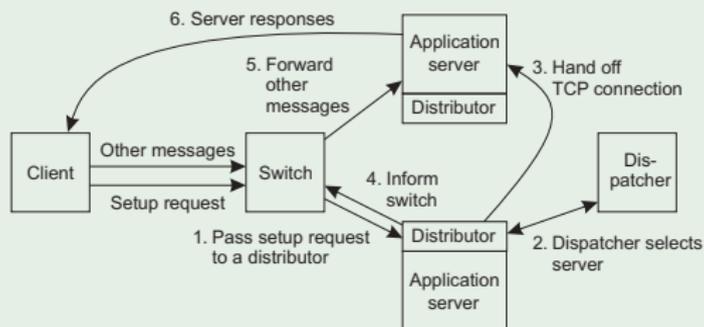


Clúster de servidores

El extremo frontal puede sobrecargarse fácilmente: son necesarias medidas especiales

- Conmutación de la capa de transporte: el front-end simplemente pasa la solicitud TCP a uno de los servidores, teniendo en cuenta algunas métricas de rendimiento.
- Distribución consciente del contenido: el front-end lee el contenido de la solicitud y luego selecciona el mejor servidor.

Combinando dos soluciones



Cuando los servidores se extienden por Internet

Observación

La distribución de servidores a través de Internet puede presentar problemas administrativos. Estos pueden evitarse en gran medida mediante el uso de centros de datos desde un sola nube/proveedor.

Solicitud de despacho: si la localidad es importante

Enfoque común: usar DNS:

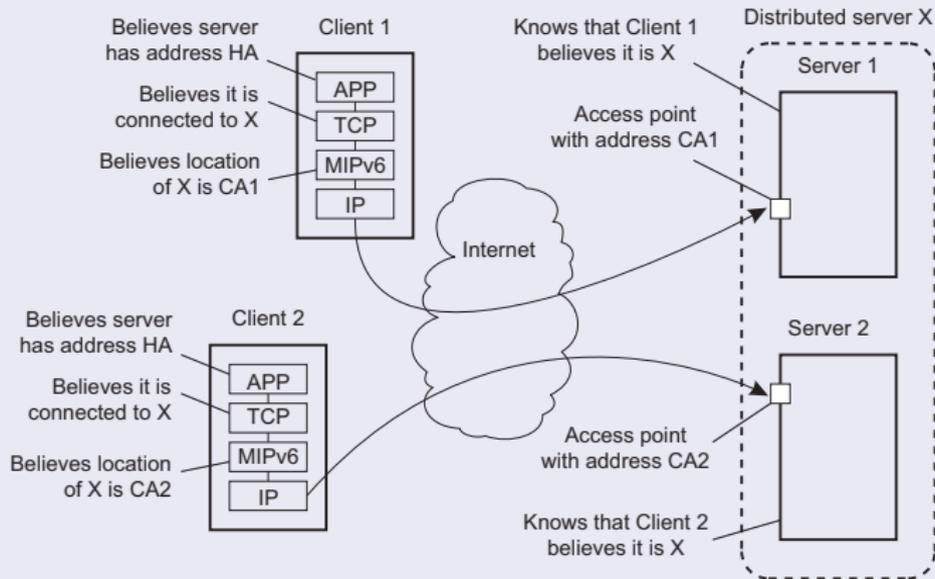
- 1 El cliente busca un servicio específico a través de DNS: la dirección IP del cliente es parte de solicitud
- 2 El servidor DNS realiza un seguimiento de los servidores de réplica para el servicio solicitado, y devuelve la dirección de la mayoría del servidor local.

Transparencia en el Cliente

Para que el cliente no se dé cuenta de la distribución, deje que el sistema de resolución de DNS actúe en nombre del cliente. El problema es que el resultado para un cliente real puede estar lejos de ser local.

Servidores distribuidos con direcciones IPv6 estables

Transparencia a través de IP móvil



Servidores distribuidos: detalles de direccionamiento

Esencia: los clientes que tienen MobileIPv6 pueden configurar de forma transparente una conexión a cualquier peer

- El cliente C configura la conexión a la dirección HA particular de IPv6
- HA es mantenida por un agente local (nivel de red), que entrega conexión a una care-of address registrada CA.
- C puede aplicar la optimización de ruta reenviando directamente los paquetes a dirección CA (es decir, sin la transferencia a través del agente local).

Servidores distribuidos: detalles de direccionamiento

Esencia: los clientes que tienen MobileIPv6 pueden configurar de forma transparente una conexión a cualquier peer

- El cliente C configura la conexión a la dirección HA particular de IPv6
- HA es mantenida por un agente local (nivel de red), que entrega conexión a una care-of address registrada CA.
- C puede aplicar la optimización de ruta reenviando directamente los paquetes a dirección CA (es decir, sin la transferencia a través del agente local).

Sistemas distribuidos colaborativos

El servidor de origen mantiene una dirección particular, pero deja las conexiones a la dirección del servidor colaborador \Rightarrow el servidor de origen y los nodos se presentan como un servidor

Ejemplo: PlanetLab

Esencia

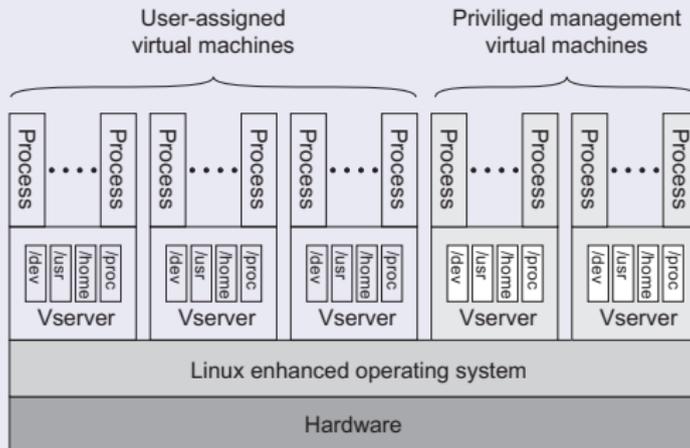
Diferentes organizaciones contribuyen con máquinas, que posteriormente comparten por diversos experimentos.

Problema

Necesitamos asegurarnos de que las diferentes aplicaciones distribuidas no se metan en el camino de otras \Rightarrow virtualización

PlanetLab Organización básica

Visión general



Vserver

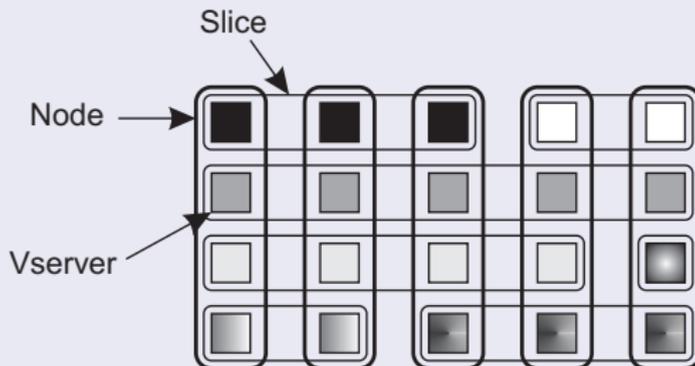
Entorno independiente y protegido con sus propias bibliotecas, versiones de servidor, y así. A las aplicaciones distribuidas se les asigna una colección de servidores V distribuido en múltiples máquinas

PlanetLab VServers y cortes

Esencia

- Cada Vserver funciona en su propio entorno (cf. chroot).
- Las mejoras de Linux incluyen el ajuste adecuado de los ID de proceso (por ejemplo, init con ID 0).
- Dos procesos en diferentes Vservidores pueden tener la misma ID de usuario, pero no implica el mismo usuario.

La separación conduce a cortes (rodajas)



References



Van Steen M , Tanenbaum, A (2017)

Distributed Systems

Pearson Education, Inc.



Coulouris G, Dollimore J, Kindberg T, Blair G (2011)

Distributed Systems: Concepts and Design

Addison-Wesley Publishing Company.



Veríssimo, P. and Rodrigues, L. (2012)

Distributed Systems for System Architects

Springer.



Limoncelli T, Strata R, Hogan C. (2014)

The Practice of Cloud System Administration: Designing and Operating Large Distributed Systems

Addison Wesley.

Fin