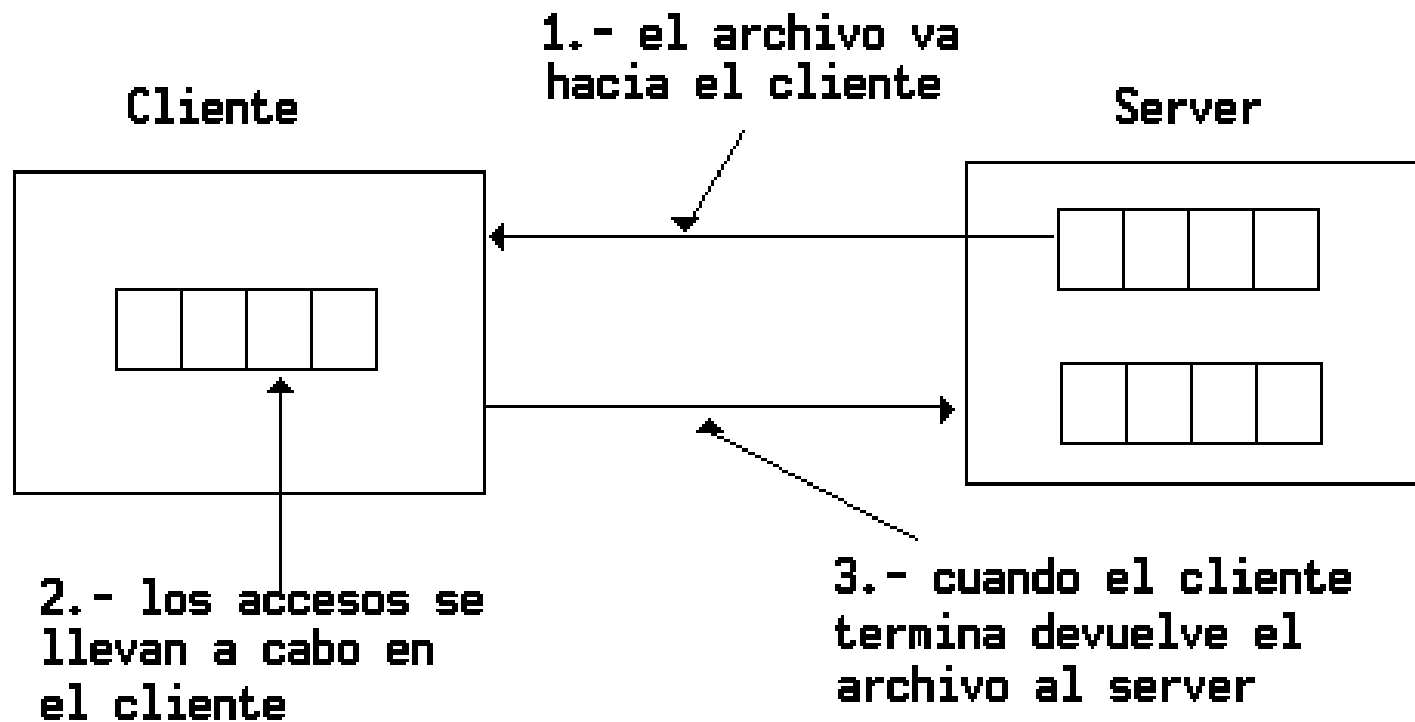


File System Distribuido - FSD

Objetivos

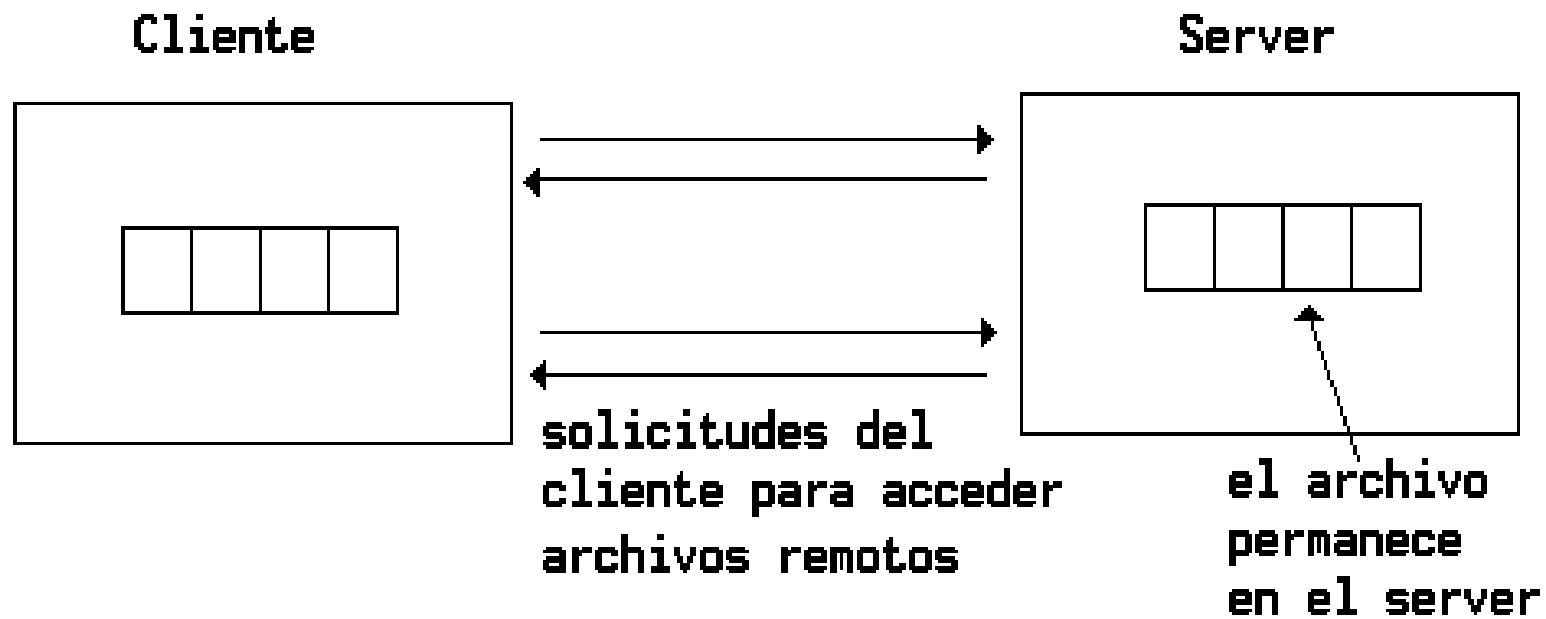
- Transparencia de acceso
- Transparencia de ubicación
- Transparencia de movilidad
- Transparencia de prestación
- Actualización concurrente
- Seguridad (Réplicas, Tolerancia a Fallas, LCA)
- Consistencia

Modelos



- Modelo Upload/Download.

Modelos



• - Modelo Remote Access.

Servicio de Directorios

- Especifica los servicios a los clientes
 - Primitivas
 - Parámetros
 - Acciones

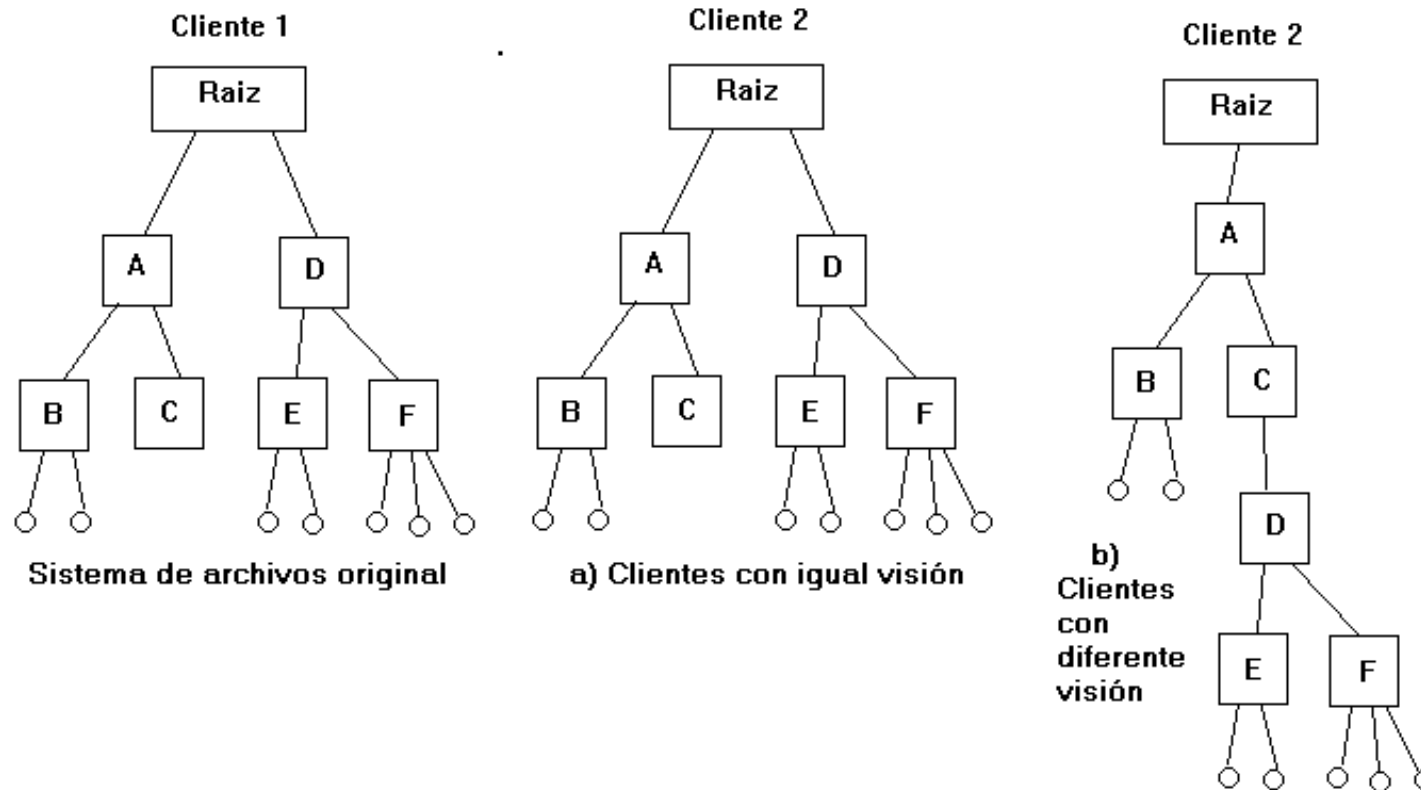
Despachador de Archivos

- Brinda los servicios de archivos
- Hace que se vea al sistema de archivos como único y no distribuido
- Ej.: VFS – v-nodes

Interfaz de Directorio

- Se necesita un alfabeto y sintáxis de nombres (archivos y directorios)
- No todos los usuarios tendrán la misma visión del árbol de directorios

Interfaz de Directorio



Transparencia de Nombres

- - Nombre máquina + path (ruta de acceso) Ej: /maquina:/path/file1.
- - Montaje de archivos remotos en la jerarquía local.
- - Espacio de nombres con la misma apariencia en todas las máquinas.

Nombres de dos niveles

- a) El nombre binario indica el server y el nombre del archivo en ese servidor.
- b) El uso de un enlace de tipo simbólico como una entrada de directorio asociada a una cadena (server, nombre del archivo). El enlace en sí es solo el nombre de una ruta de acceso.
- c) Otra posibilidad es utilizar capacidades como nombres binarios. El buscar un nombre en ASCII da como resultado una capacidad que puede tener varias formas. Por ejemplo: puede tener el número físico o lógico de una máquina o la dirección en la red en el server apropiado así como un número que indique el archivo específico.

Semántica de archivos compartidos

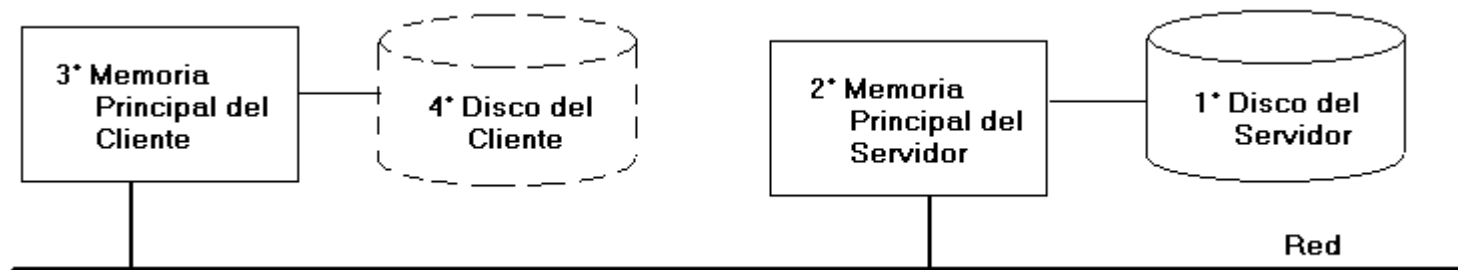
- UNIX.
- Sesión.
- Archivos inmutables.
- Transacciones.

Servers

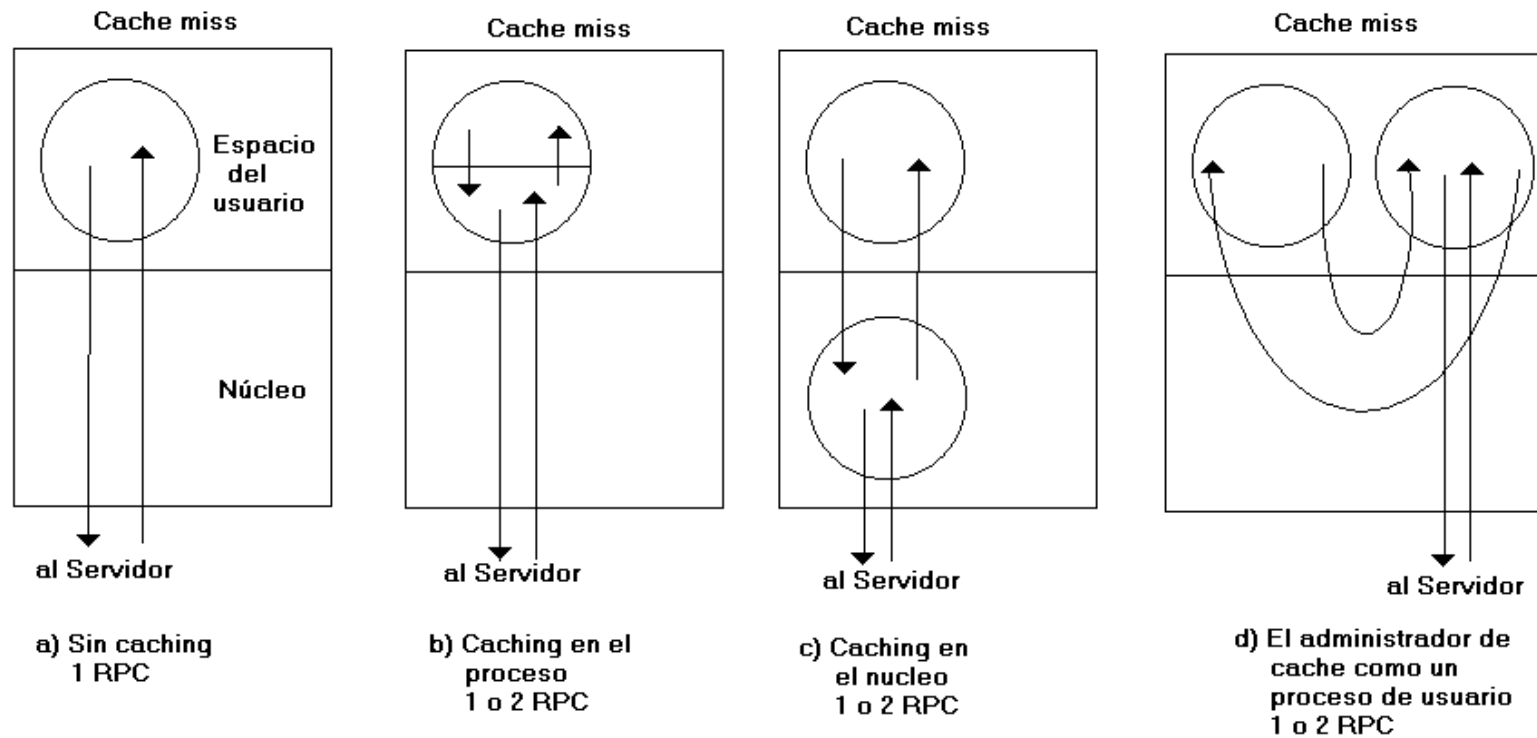
SERVER CON ESTADO	SERVER SIN ESTADO
Puede existir lectura adelantada.	No necesita llamadas OPEN/CLOSE.
Es posible la cerradura de archivos.	No existe límite para el número de archivos abiertos.
Mejor desempeño.	Tolerancia de fallas.
Fácilmente aplicable la idempotencia.	No se desperdicia espacio en el server con tablas de mantenimiento.
Mensajes de solicitud más cortos.	No existen problemas al fallar un cliente.
Cuando el servidor se cae se pierde toda la información y la recuperación queda a cargo de los clientes	Mensajes de solicitud más largos
Si el cliente se cae el servidor no sabe si eliminar o no las entradas abiertas inactivas	Si el cliente se cae no hay problema para el servidor

Ocultamiento (caching)

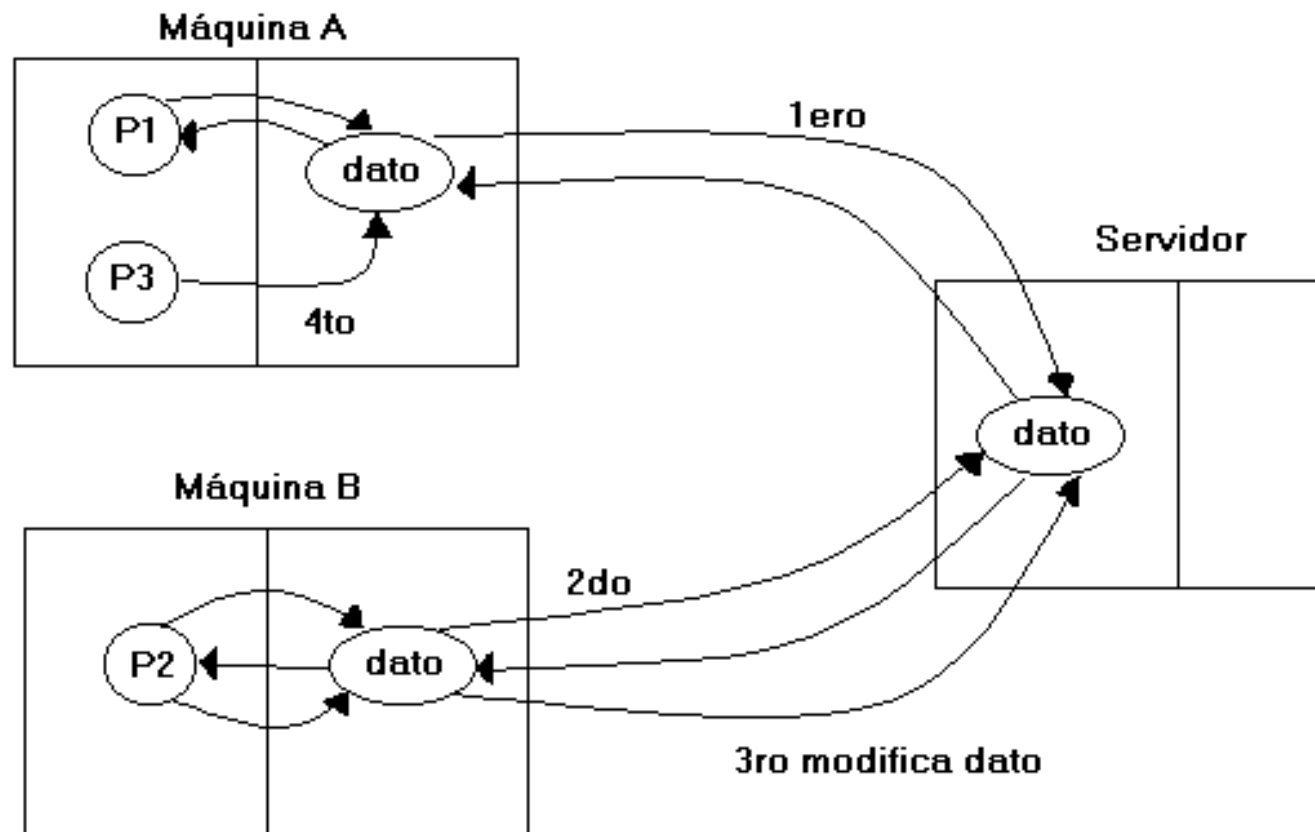
- El disco servidor.
- La memoria del servidor.
- El disco cliente (si existe).
- La memoria cliente.



Caching



Consistencia



Consistencia

Método	Comentarios
Escritura a través del caché	Funciona pero no afecta el tráfico de escritura
Escritura demorada	Mejor desempeño pero es posible que la semántica sea ambigua
Escritura al cierre	Concuerda con la semántica de sesión
Control centralizado	Semántica UNIX, pero no es robusto y es poco escalable
Algoritmos para administrar el ocultamiento del cliente	

Implementaciones

- RFS (Remote File System)
- NFS (Network File System)
- AFS (Andrew File System)

RFS

- Estado en Cliente y Servidor
- Todos los pedidos por medio de RPC
- Caída de Servidor afecta Cliente
- Lecturas adelantadas
- Control de acceso centralizado
- Cantidad máxima de archivos en operación

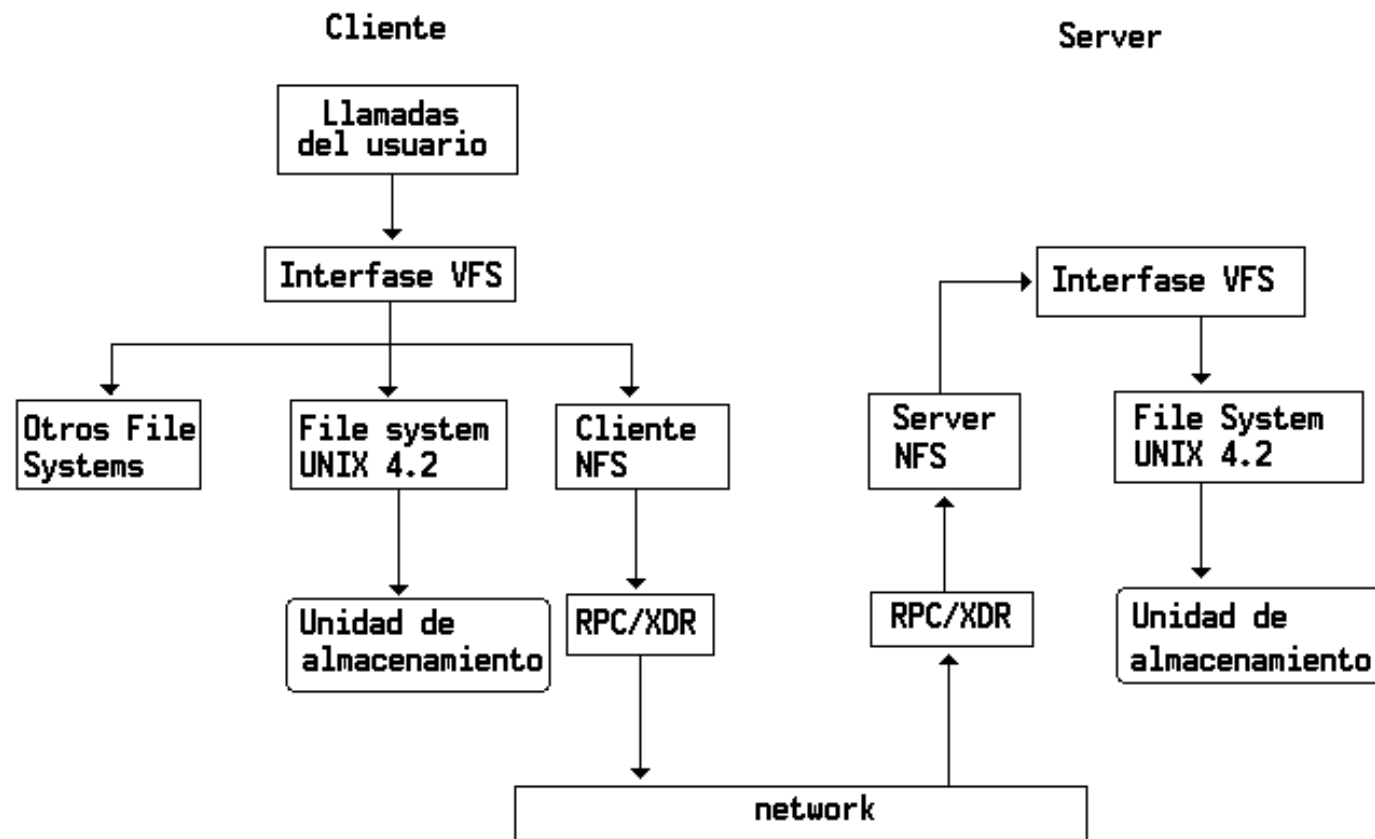
NFS (Mount Protocol)

- El servidor al recibir un pedido de montaje verifica dicha lista y en caso de tener el permiso adecuado retorna un **file handle** al cliente el cual es la unidad componente de todo pedido.
- El file handle contiene toda la información que el server necesita para distinguir un archivo individual que se guarda.
- En términos de UNIX, el file handle consiste de un identificador del archivo dentro del sistema y un número de i-nodo para identificar exactamente al directorio montado dentro del FS exportado.
- El server mantiene también una lista de las máquinas clientes y sus correspondientes montajes para uso administrativo.

NFS Protocol

- Este protocolo provee un conjunto de RPC's para realizar las operaciones remotas correspondientes a:
 - ☐ • Lectura de un conjunto de entradas de directorio.
 - ☐ • Lectura y escritura de archivos.
 - ☐ • Búsqueda de un archivo dentro de un directorio.
 - ☐ • Acceso a los atributos de un archivo.
 - ☐ • Manipulación de links y directorios.

NFS Protocol



NFS

- Server
 - Exporta Sistemas de Archivos
 - R/W según req. Del cliente
 - No lleva Estado
 - Se comunica por RPC
 - No hace cache
- Client
 - Monta archivos y directorios compartidos
 - R/W archivos que solicita al NFS Server
 - Lleva estado
 - Se comunica por RPC
 - Hace cache

NFS

- Server
- Exporta según
/etc/exports (nfsd)
- Regista lo que exporta
en /etc/xtab
- Con /etc/rpc.mountd
entrega puntero
(handler)
- Client
- Pide montaje por
/etc/mount o /etc/fstab
- Con el handler crea un
inode –v que apunta a
inode-r

NFS

v-node

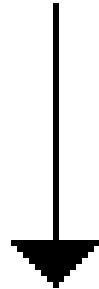
id. FS – i-nodo – nro i-nodo local

AFS

- Típico caso de ocultamiento (sesión)
- Espacio único de nombres
- El cliente por medio del Venus solicita al servidor (Vice) por medio del protocolo Virtue un archivo
- Cada cliente tiene su servidor, pero este consulta a varios otros

AFS

VICE

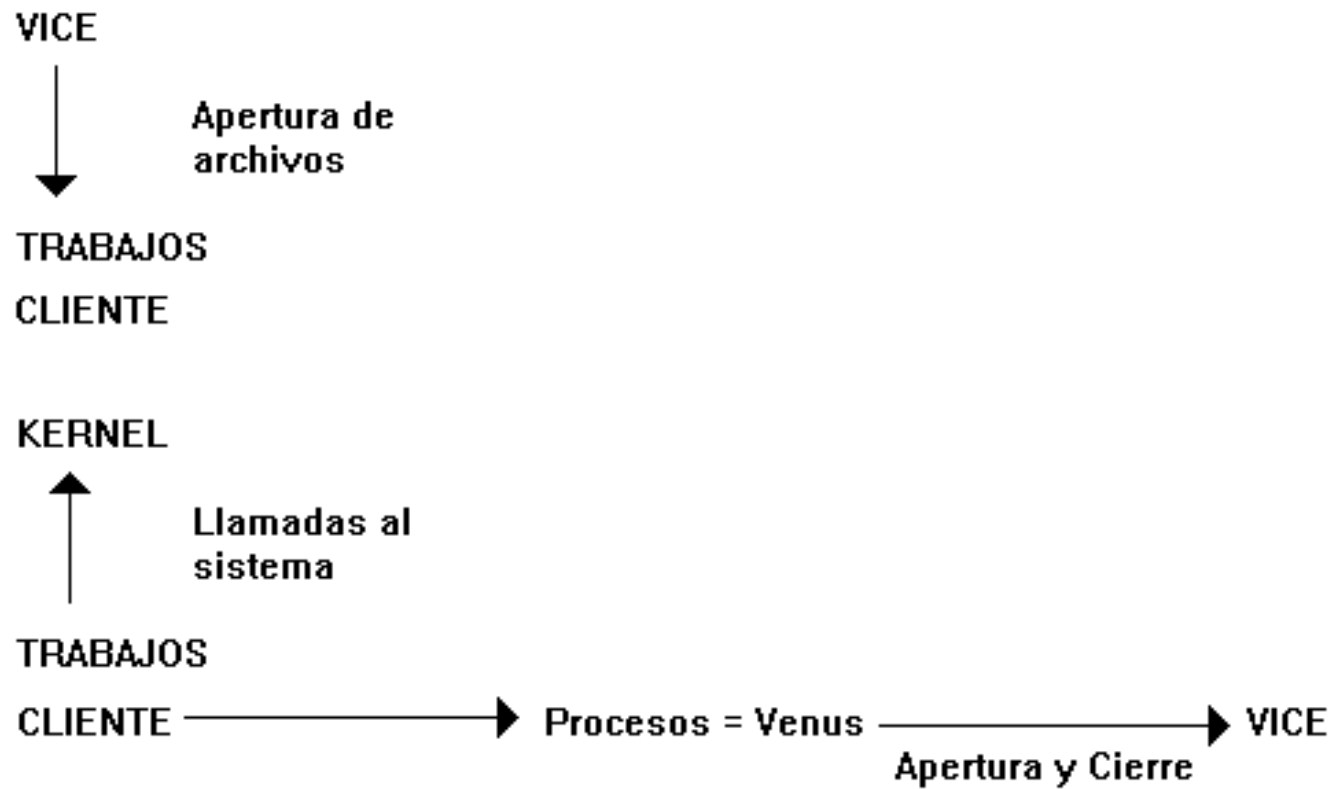


Apertura de
archivos

TRABAJOS

CLIENTE

AFS



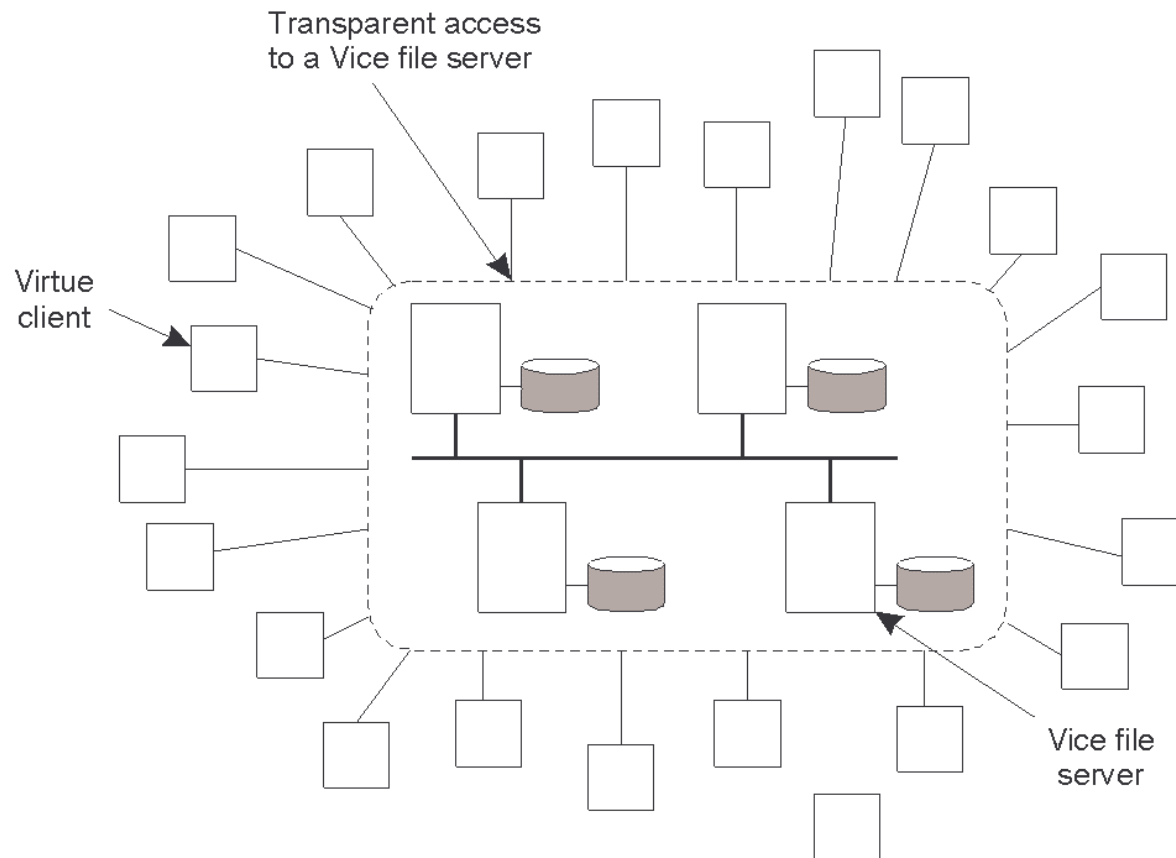
AFS

- Entre Venus y Vice manejan una estructura de fids, que tiene la forma de:
- nro. de volumen (identificación única)
- nro. de v-node (identifica archivo en volumen)
- Nro único local (fid)

AFS

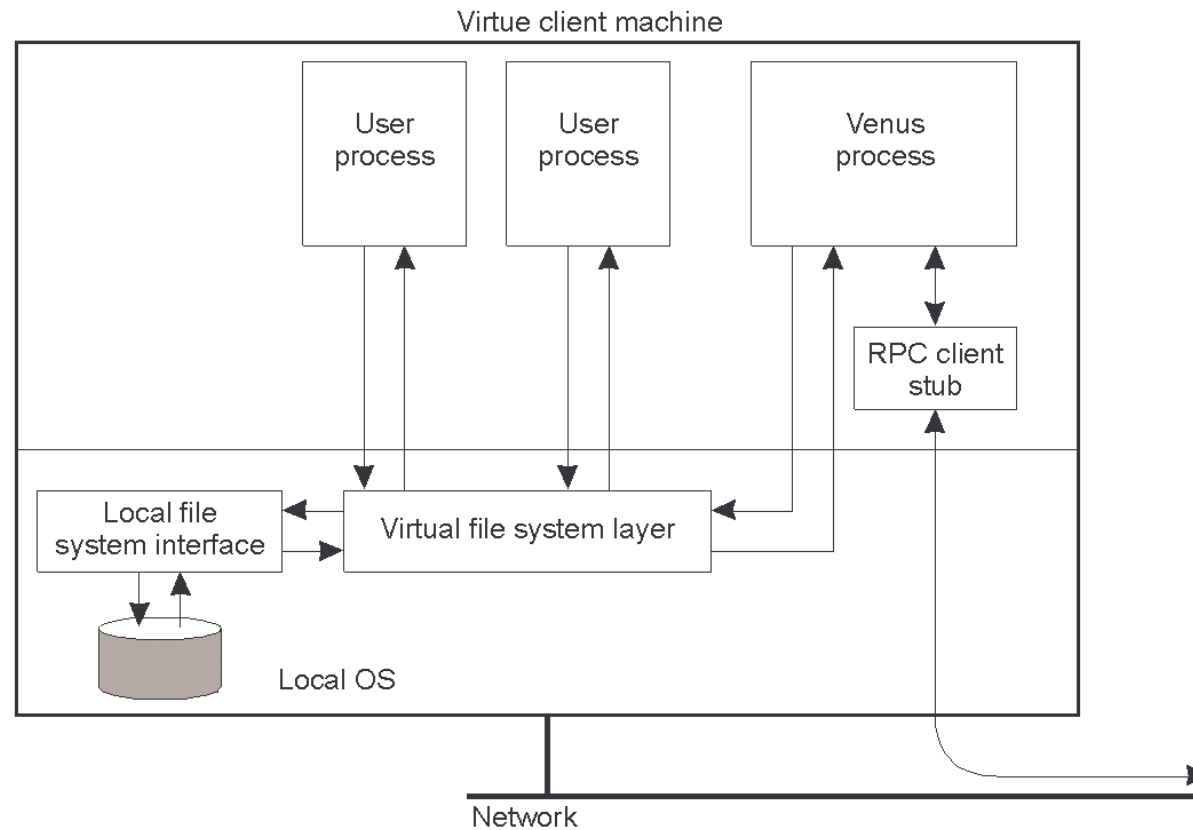
- Usa ocultamiento
- Venus pregunta a Vice si el archivo es válido
- Vice le envía una copia
- AFS asegura consistencia por 30 minutos

Coda



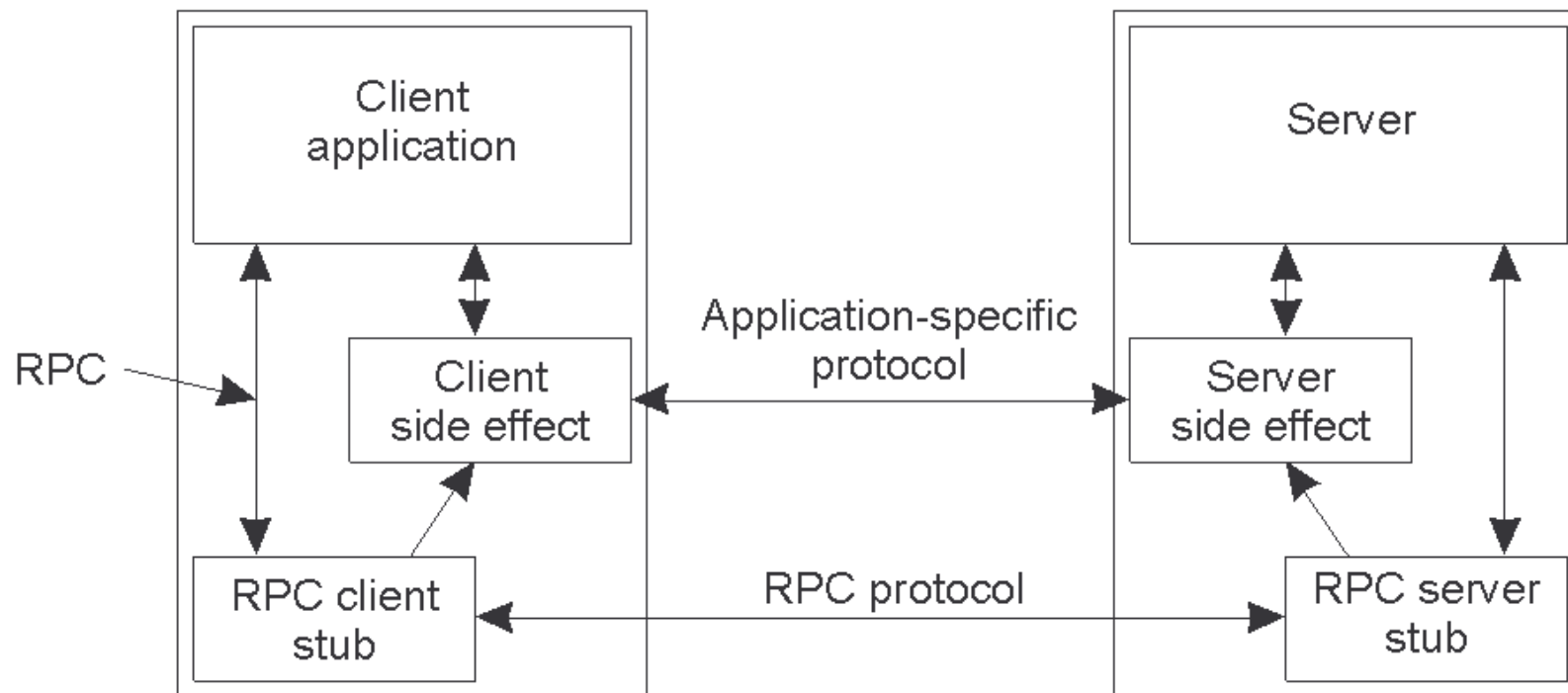
The overall organization of AFS

Coda



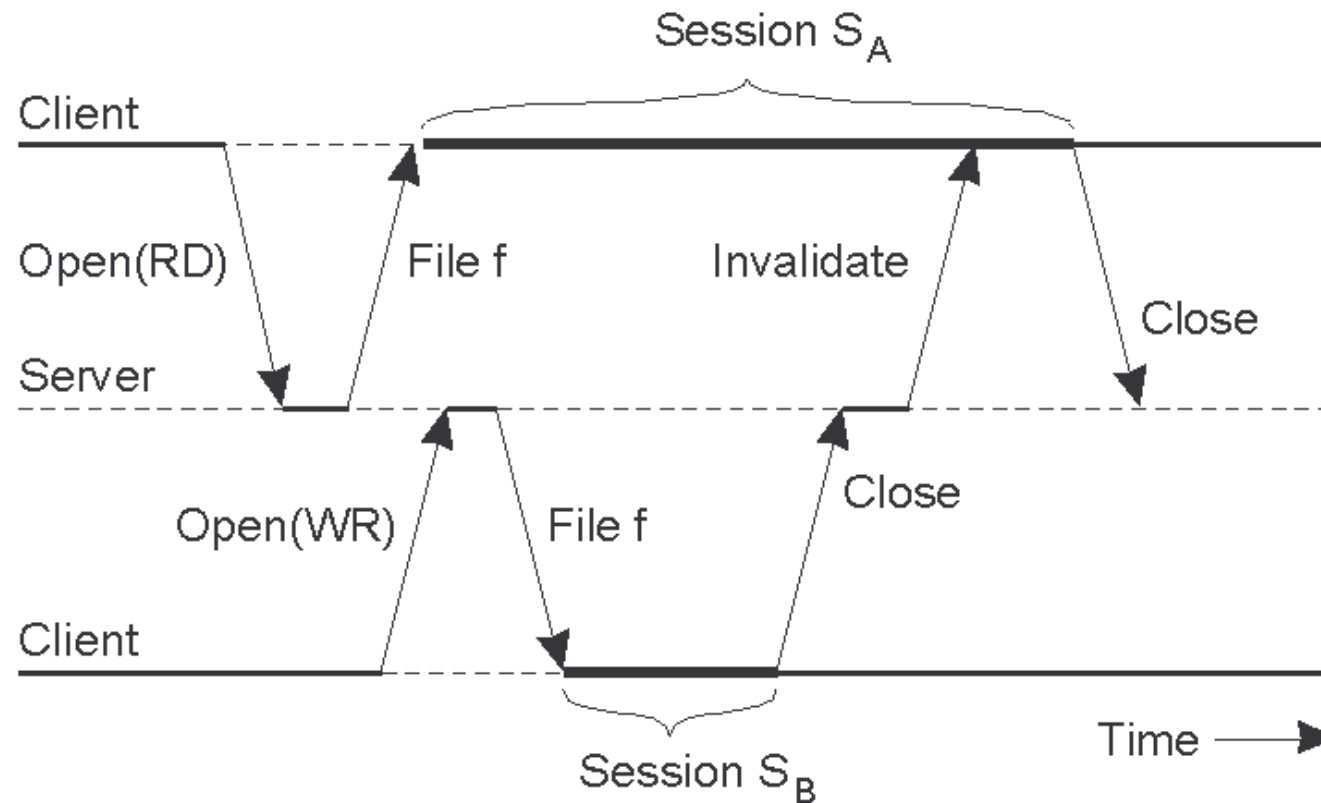
The internal organization of a Virtue workstation. ₃₁

Coda



Side effects in Coda's RPC2 system

Sharing Files in Coda



- a) Sending an invalidation message one at a time.
- b) Sending invalidation messages in parallel