



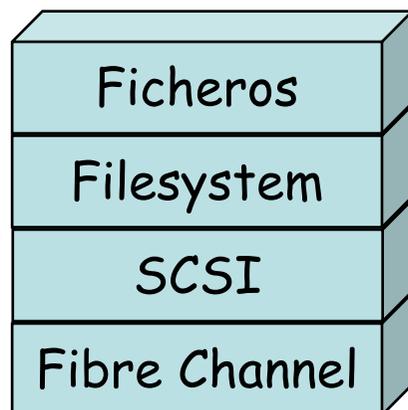
**SAN  $\neq$  NAS**



# SAN ≠ NAS

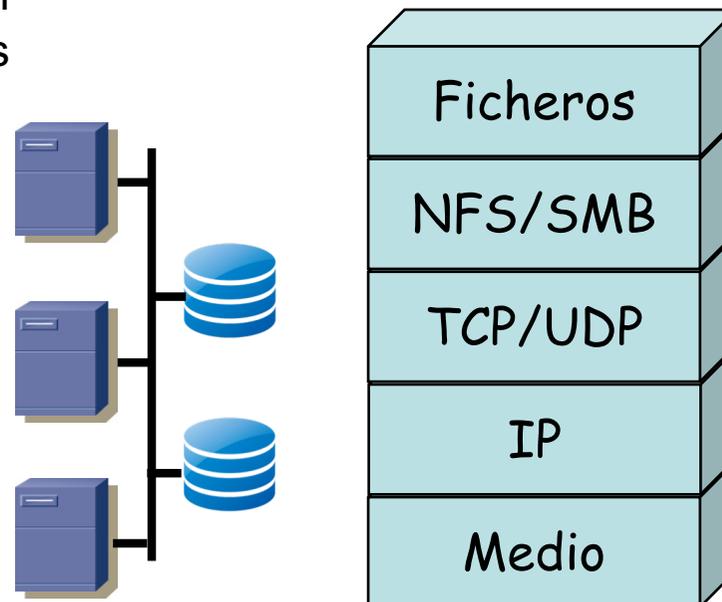
## En una SAN

- Se accede de forma serie a bloques de disco
- Normalmente mediante comandos SCSI-3
- Los protocolos están optimizados para baja latencia y nulas pérdidas
- La solución de transporte más habitual es Fibre Channel
- Acceso de varios servidores al mismo volumen requiere sistemas de ficheros especiales



## En una solución NAS

- Se accede a ficheros
- Se suele transportar sobre una tecnología LAN (o LAN + IP)
- Los protocolos no garantizan baja latencia ni nulas pérdidas (su recuperación aumenta la latencia)
- NFS, SMB/CIFS, AFP, etc



# SAN vs NAS

- Rendimiento haciendo backups
  - Mayor en SANs, generalmente más rápida
  - En algunos casos se pueden mover datos del disco de un servidor a una cabina sin intervención de la CPU
  - En sistemas de ficheros con gran cantidad de ellos es más eficiente una copia del dispositivo *raw*
- Bases de datos
  - Algunas requieren un acceso al disco a nivel de bloques por rendimiento e integridad



upna

Universidad Pública de Navarra  
Nafarroako Unibertsitate Publikoa

**Redes de Nueva Generación**  
*Área de Ingeniería Telemática*

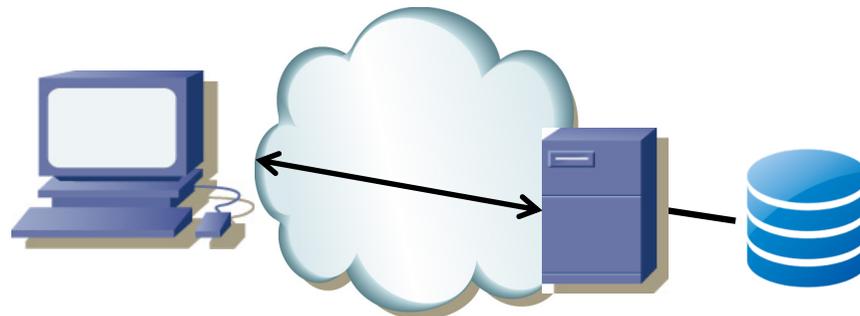


# NAS



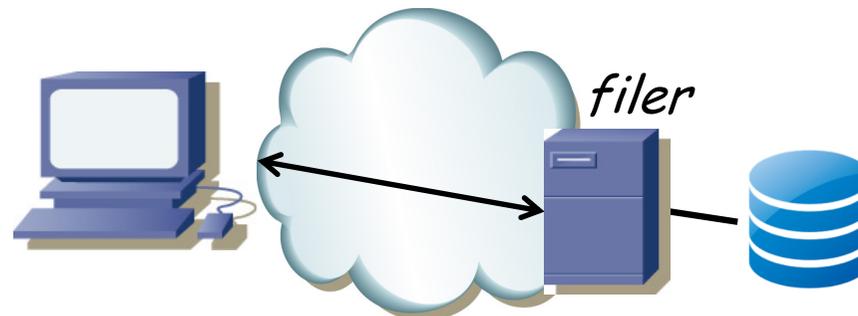
# Acceso a ficheros

- Solución cliente-servidor, hoy en día sobre TCP/IP
- Integrado con el sistema operativo en el cliente
- Eso permite que las aplicaciones vean el sistema de ficheros remoto como si fuera local
- Ejemplos de protocolos: SMB, NFS, AFP
- Otras alternativas menos frecuentemente integradas en el cliente: FTP, HTTP (WebDav), SSH/SFTP
- El protocolo da acceso a ficheros y directorios, no a bloques



# Acceso a ficheros

- Inicialmente el servidor es un hardware de propósito general que comparte parte de su disco
- Protocolos más habituales
  - SMB
  - NFS

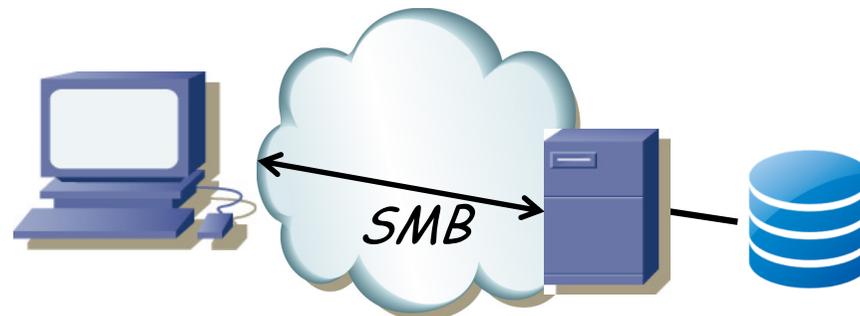




# SMB



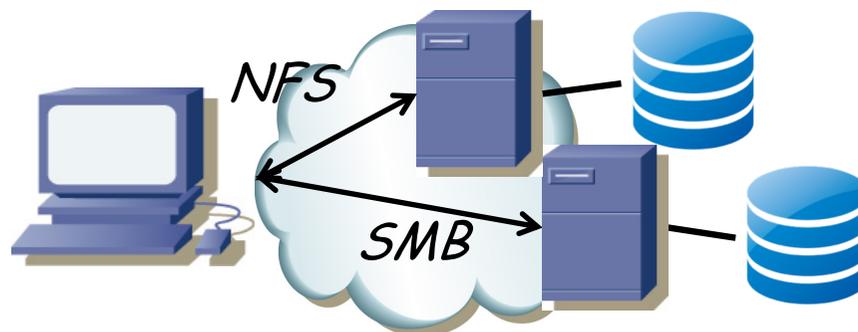
- *Server Message Block*
- Desarrollado por IBM y Microsoft
- Principalmente en sistemas Windows
- Empleado para resolución de nombre, navegar recursos compartidos, compartición de ficheros, acceso a impresoras, autenticación, etc
- Sobre TCP, UDP y otros
- SMB, CIFS (*Common Internet File System*), SMB2, SMB3, ...



# NFS

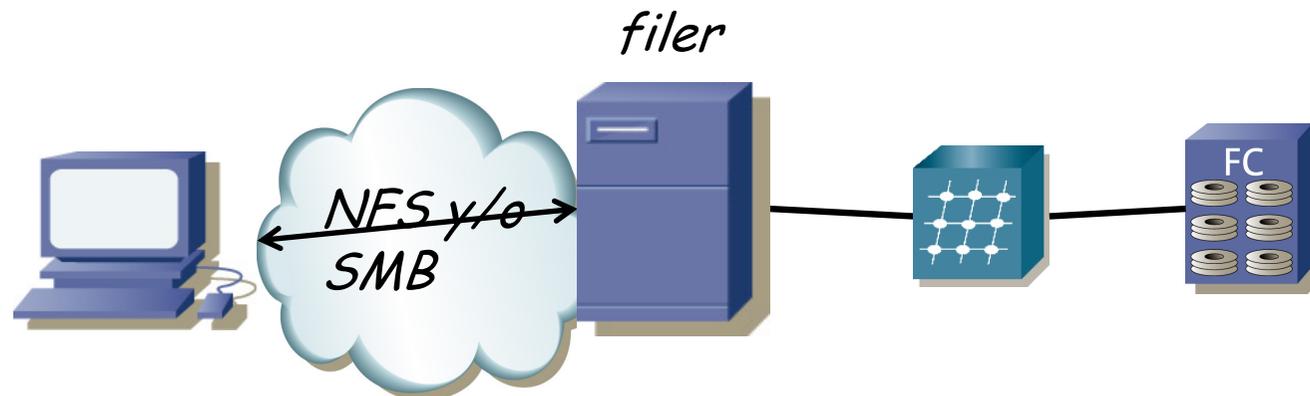


- *Network File System*
- Principalmente en sistemas UNIX (desarrollado por Sun Microsystems)
- Emplea los protocolos XDR (External Data Representation, RFC 1832) y RPC (Remote Procedure Call, RFC 1831)
- Muy común para centralizar los *home directories*
- mountd, nfsd
- NFSv2 (RFC 1094) sin estado (UDP), tamaños de 32 bits
- NFSv3 (RFC 1813) sin estado (UDP aunque también TCP), tamaños de 64 bits
- NFSv4 (RFC 3530) con estado (TCP), seguridad
- NFSv4.1 (RFC 5661) introduce *parallel NFS* (pNFS)



# NAS

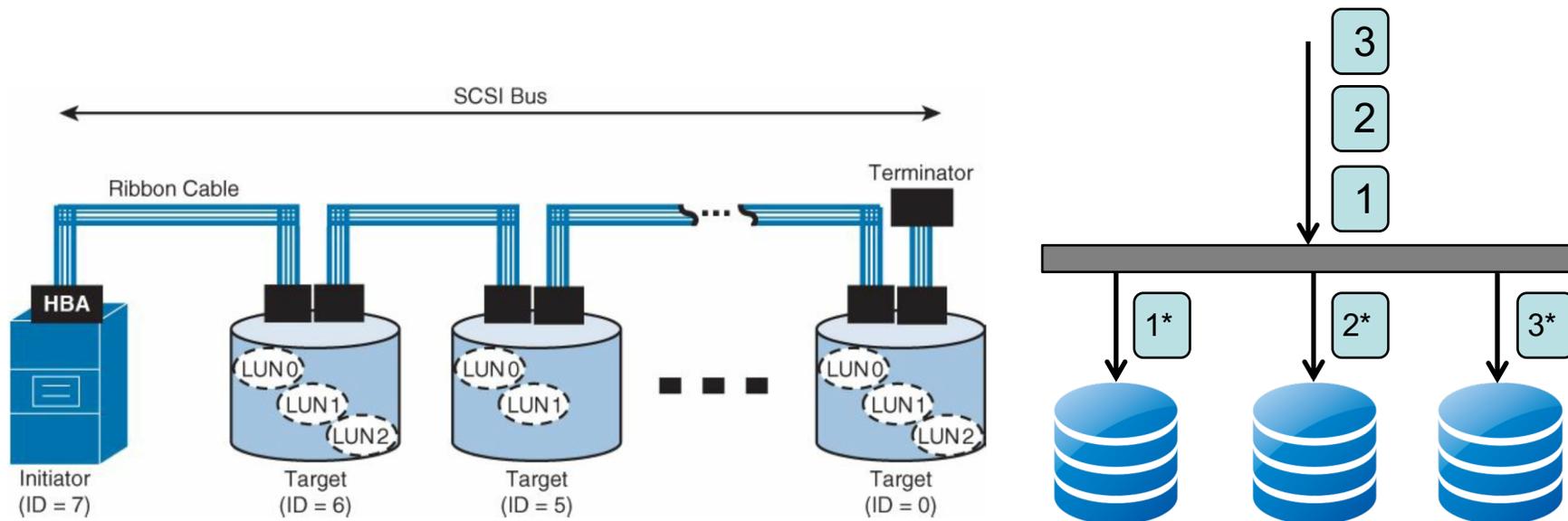
- NAS = *Network Attached Storage*
- Inicialmente el servidor es un hardware de propósito general
- Hoy en día hardware dedicado (*appliances*) para esta tarea
- Optimizado para ello
  - Simplifica la gestión
  - RAIDs, discos *hot-swappable*
  - Capaces de compartir los mismos recursos mediante varios protocolos simultáneamente (SMB + NFS)
  - Mejoras de rendimiento en la implementación del soft servidor
- El disco en vez de ser local puede estar en una SAN
- SMB Direct (empleando RDMA), NFS over RDMA



# Virtualización del almacenamiento

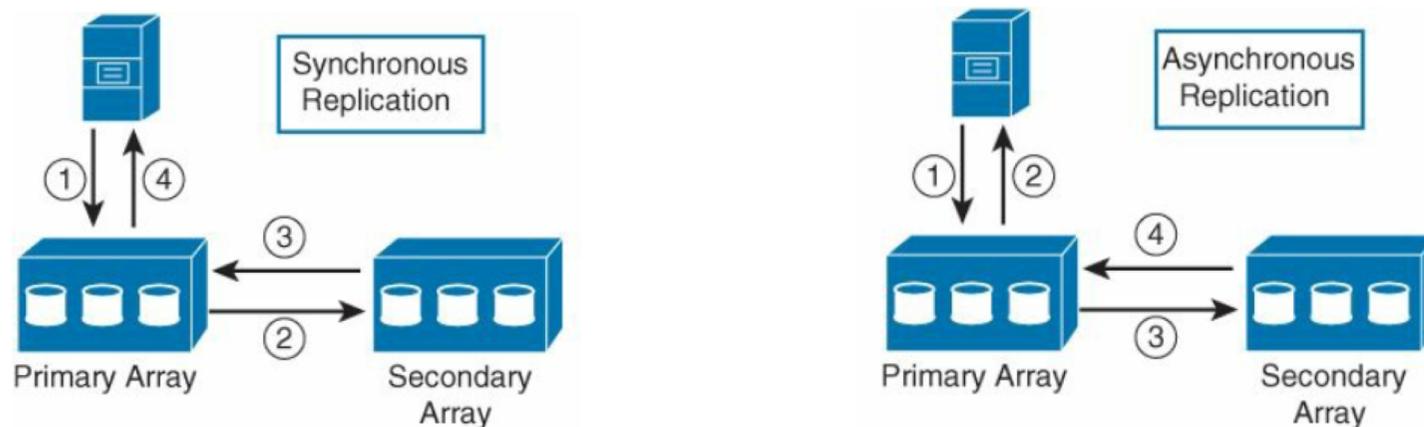
# Storage Virtualization

- Ya hemos visto varios casos:
  - Logical Units: Podemos segmentar un disco
  - RAID: Varios discos físicos se ven como una sola unidad



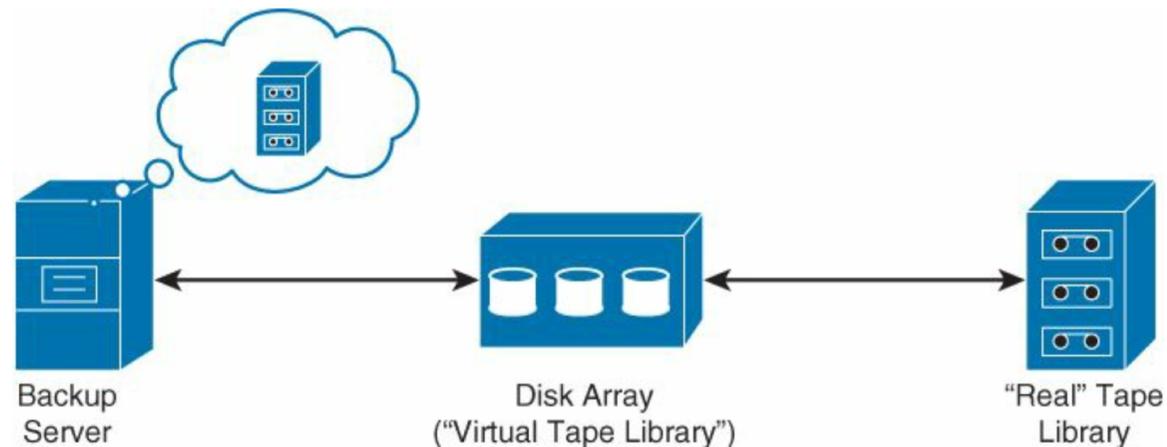
# Disk Array Virtualization

- *Partitioning*
  - Una cabina puede soportar subdividirse en dispositivos lógicos
  - Cada uno tendría asignados recursos de: discos, cache, memoria, puertos
  - Cada partición puede crear sus propias LUNs
- *Array-based data replication*
  - Múltiples cabinas pueden trabajar juntas en replicación
  - La *replicación síncrona* se basa en devolver confirmación de haber almacenado el dato cuando se ha escrito en las dos
  - La *replicación asíncrona* se basa en copiar después o periódicamente los datos (no bloquea la respuesta al usuario)



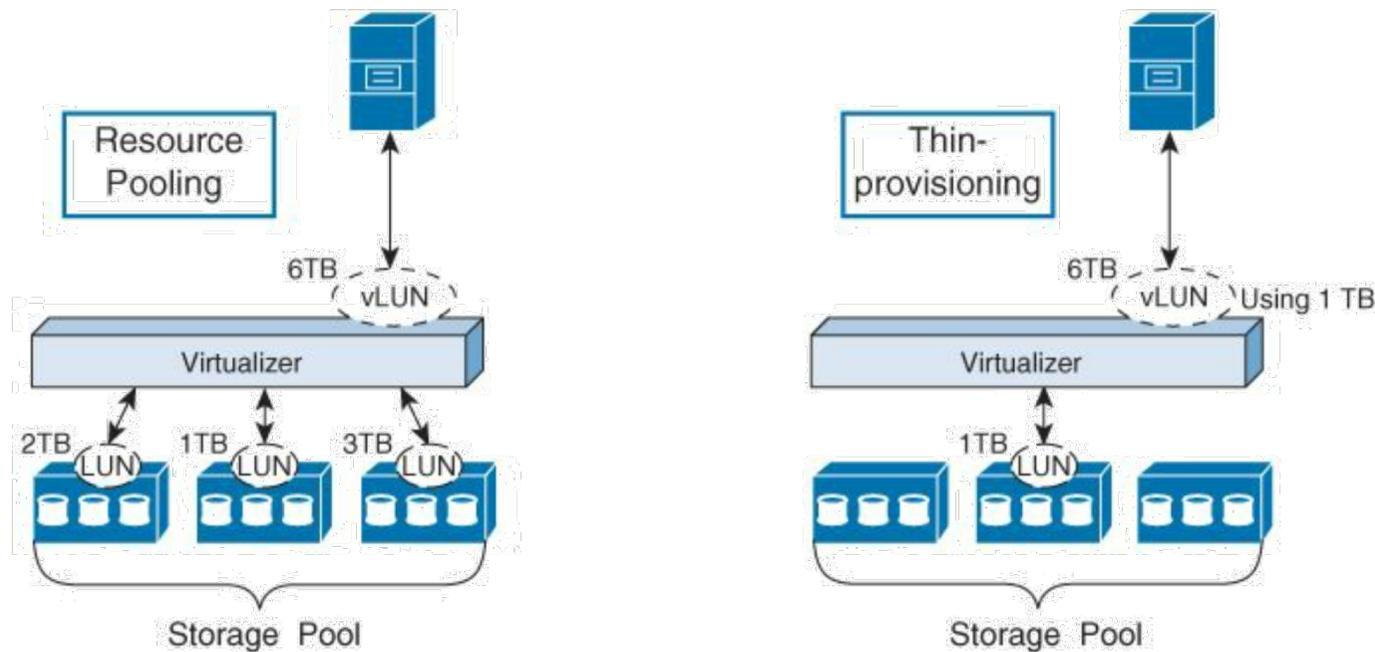
# Virtual Tape Library

- El servidor accede a la cabina como si fuera la biblioteca de cintas
- La cabina actúa como una cache
- *Deduplication*
  - No manda al almacenamiento una segunda vez algo que ya existe
  - Apunta simplemente una referencia
  - Si luego uno de los dos se modifica puede guardar solo las modificaciones
  - También en el escenario de cabina de discos independiente
  - Ahorra por ejemplo bastante con imágenes de OS
  - También puede hacer compresión



# Virtualización de LUNs

- Un virtualizador se interpone entre el servidor y la LUN
- Ofrece una vLUN al servidor
- Eso le permite modificar cómo la implementa sin alterar al servidor
- Puede agregar varias LUNs en una (*storage resource pooling*)
- Puede ofrecer una vLUN de mayor capacidad que la que realmente está empleando (*thin-provisioning*)
- Esto puede llevar a *over-subscription* y como tal funciona mientras todas las vLUNs no quieran usar toda la capacidad que anuncian



# Virtualización de LUNs

- Permite la migración de la LUN de una cabina a otra de forma transparente (*online migration*)
- Por ejemplo para cambiar a discos o un RAID más rápido
- El virtualizador puede dar la funcionalidad para la replicación entre dos cabinas, por ejemplo en DCs alejados
- Un virtualizador en cada DC puede estar ofreciendo la vLUN a los servidores de ese DC (*LUN extension*)

