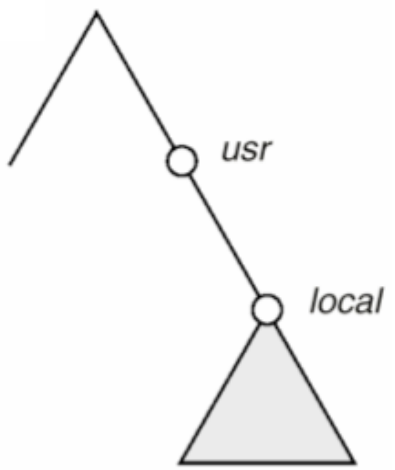
Distributed File Systems

Distributed File Systems

- Ch15.8 NFS
- Ch19.6 Distributed File Systems

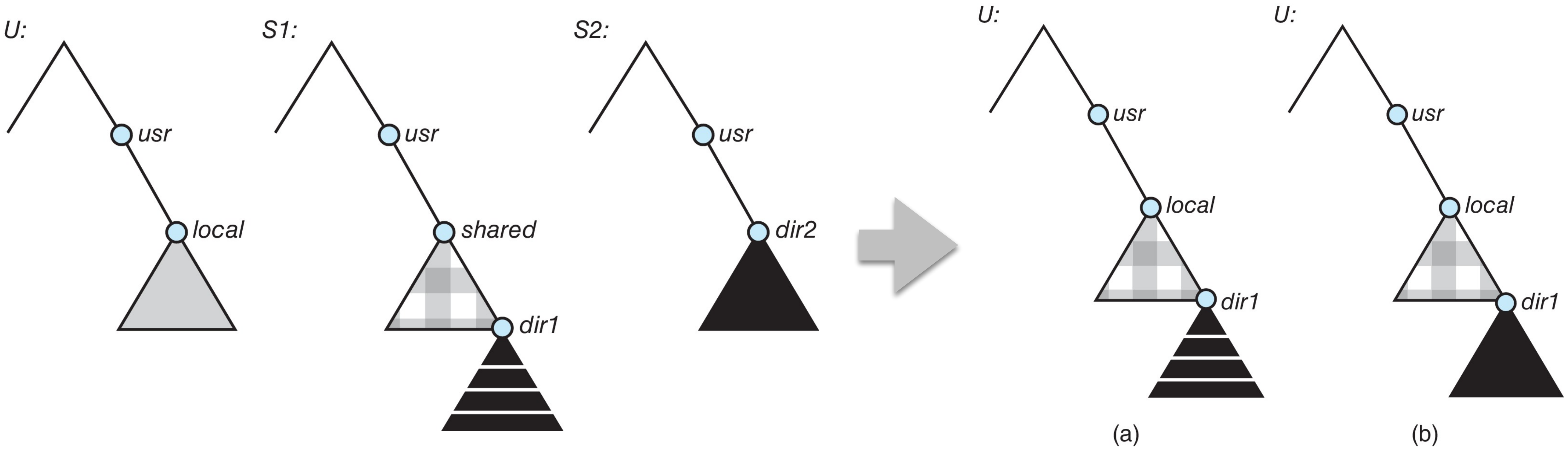
NFS

- Was Sun's Network File System (NFS) – a stateless system (up until Version 4)
- Based on the UNIX method of *mounting* disk devices within a file directory tree.
- Another disk can be mounted at /usr/local for example. The inode contains a bit indicating a device is mounted there (a table holds the required information).
- In NFS remote file directories can be mounted on a local directory structure.



```
mount serverX:/export/home/bob /home/bob
```

Example: Mounting in NFS



Three Independent File Systems

Mounts

Cascading Mounts

Looking for a Particular File

- Accessing `/usr/local/bin/dir1/filename` on machine 1
- `/usr/local` might be a directory on machine 2
- `/usr/local/bin` is within that directory on machine 2
- `/usr/local/bin/dir1` might be a directory on machine 3.
- Once a mount point to a remote file is crossed then accessing the rest of the path is expensive.
- Each directory has to be read remotely.
- Actually it must be checked locally first in case there is another mount point to a different directory on another server.
- Each machine maintains a directory name cache to speed up the lookup.

NFS

- No need for dedicated servers.
- Remote directories (or entire devices) can be mounted anywhere in the local directory tree
- Works with heterogeneous environment
  - RPC and XDR - external data representation

Mount protocol

- Mount servers on each machine
  - export table `/etc/exports`
- The full pathname of the directory to be exported
- The client machines that will have access to the exported directory
- Any access restrictions
- Request comes to mount a directory from this machine
  - returns a file handle to this directory (file-system:inode)
- Server maintains a list of which machines have mounted one of its directories.

NFS

- NFS Path-Name Translation
  - Performed by breaking the path into component names and performing a separate NFS lookup call for every pair of component name and directory vnode
  - To make lookup faster, a directory name lookup cache on the client's side holds the vnodes for remote directory names

NFS Remote Operations

- Nearly one-to-one correspondence between regular UNIX system calls and the NFS protocol RPCs (except opening and closing files)

- NFS adheres to the remote-service paradigm, but employs buffering and caching techniques for the sake of performance

- File-blocks cache – when a file is opened, the kernel checks with the remote server whether to fetch or revalidate the cached attributes

    - Cached file blocks are used only if the corresponding cached attributes are up to date

- File-attribute cache – the attribute cache is updated whenever new attributes arrive from the server

- Clients do not free delayed-write blocks until the server confirms that the data have been written to disk

Automounter

- Client maintains a list of the directories which are mounted from other systems.
- ***Automounter*** mounts and unmounts remote directories on demand.
- Uses maps (files containing links between the mount point and the actual directory)
- e.g. Setting up a shared namespace for `/home`

  First an entry is made in `auto_master` (master configuration file) which associates the mount point `/home` to a map called `auto_home`:

  ```
  */home auto_home
  ```
- `auto_home` is a map that associates user names to home directories on their respective servers:

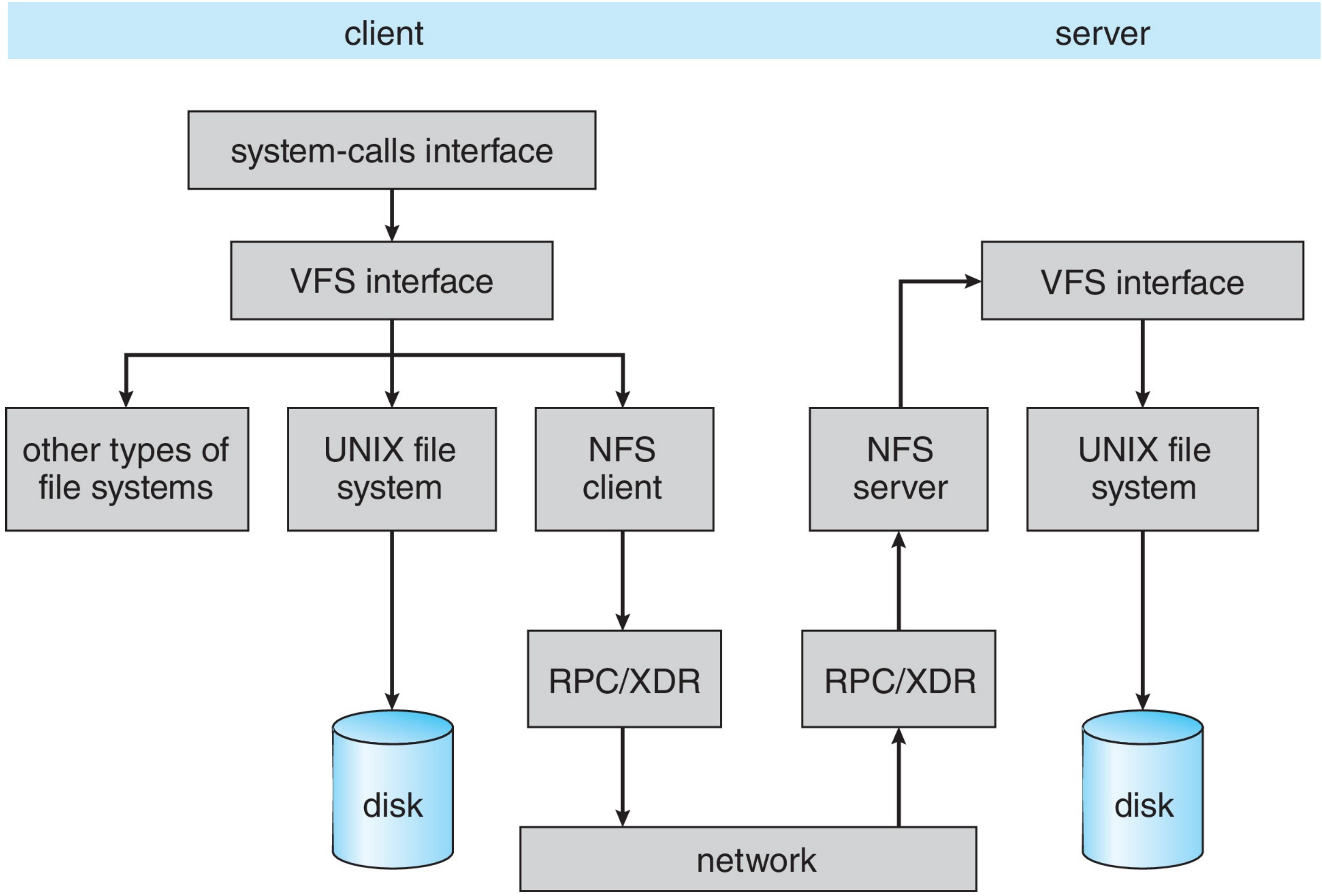
```
*sally server1:/export/home/sally
*greg server1:/export/home/greg
*tom server2:/export/home/tom
*grace server3:/export/home/grace
```

Automounter

- When the *automount* command is invoked at system start up time it looks in `auto_master` and then `auto_home` and knows to set up `/home` as a directory of mount points. These mount points will become mounted file systems at the time they are referenced by users. Users can be added or deleted from the namespace by adding or subtracting them from the `auto_home` map. Any changes will be automatically implemented the next time the file system is mounted.
- The automounter uses a timeout (usually 5 minutes) to unmount a directory when unused.

NFS Architecture

- Normal system call
- **VFS** (virtual file system) determines local or remote
- **NFS** service layer makes the RPC to the remote machine
- Request gets pumped into VFS on the remote machine
- Carried out locally
- Then back goes the result



Problems with NFS

- Administration is difficult.
- As all sites can mount exported subtrees anywhere it can be difficult to maintain a uniform view of the directory structure.
- Moving a collection of files is complicated.
- All sites need to be notified as they all have location information stored in their maps.
- Can provide replicas of read-only file collections. But these suffer from difficult administration as well.
- Because of these problems it doesn’t scale well. Only used on medium size networks.
- Some of these problems have been addressed in the latest versions of NFS.

AFS

Originally called the Andrew File System (AFS)

- Local name space (like NFS)

Some files only appear on the local host. Usually machine specific files.

- Shared name space

`/afs` is the root of all shared files

- Identical on each client (not like NFS) and location transparent even over a WAN (SSI – single systems image)
- Files can be relocated without removing access (except for a brief time)
    - Copy them across.
    - Update the location servers.
    - The original location still handles old requests - shipping them to the new location.
    - Remove the originals.
- Scales easily (to thousands of machines)
- Uses Kerberos for authentication.

Login and be authenticated once for all network access.

Mutual authentication - clients **and** servers authenticate themselves.

AFS Implementation

- The Volume Location Database (VLDB) contains the location information and is usually held by several servers.
- Servers are dedicated; they are not also clients.
- Files are grouped into volumes.
  - A volume is commonly the files of a particular user or groups of users.
  - The volumes are the things that are referenced in the location database.
  - Volumes can be transparently migrated.
- Files identified by volume:vnode_number:uniquifier
  - vnode numbers can be reused, therefore to keep uniqueness there is the uniquifier (extra bits added on until unique)
- Client machines run a Cache Manager process
  - Finds where files are (from the VLDB).
  - Retrieves files from the host.
  - Uses caching rather than remote service
  - files are cached in large chunks (64K) - hopefully the whole file
  - this minimises network traffic and is usually more efficient at both client and server ends

AFS Shared Access to Files

- Files and directories are protected by access control lists not the simple Unix bit protection scheme.
  - This means that a directory may look as though it is not protected (e.g. is writable) when it actually is protected.
- Session semantics
- Changes in shared files are not seen until the file is closed (or synced).
- Callbacks
  - A callback in AFS is a promise from the server that the cached version of a file is up to date.
  - Before a file is changed the server breaks the callbacks.
  - Then when another process uses its cached version the Cache Manager detects the broken callback and refreshes its cached data from the server.
- So AFS is not as stateless as early versions of NFS.

Before Next Time

Networks and Distributed Systems

- Ch19.1 Advantages of Distributed Systems
- Ch19.4 Network and Distributed Operating Systems
- Ch21.6.2.7 Remote Procedure Calls
- B.6.2.7 Remote Procedure Calls
- Wikipedia – Two-Phase Commit Protocol