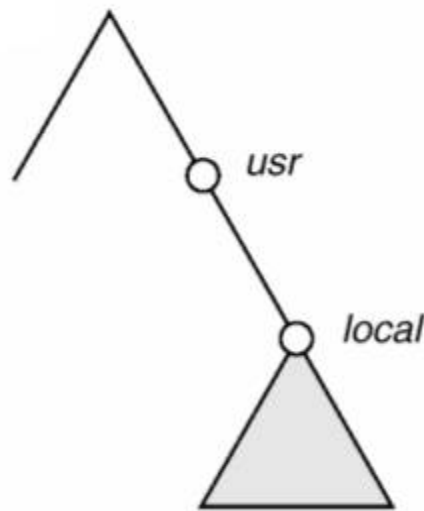


# NFS

Sun's Network File System – a stateless system

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.

# NFS

No need for dedicated servers.

Remote directories (or entire file systems) 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)

# 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
```

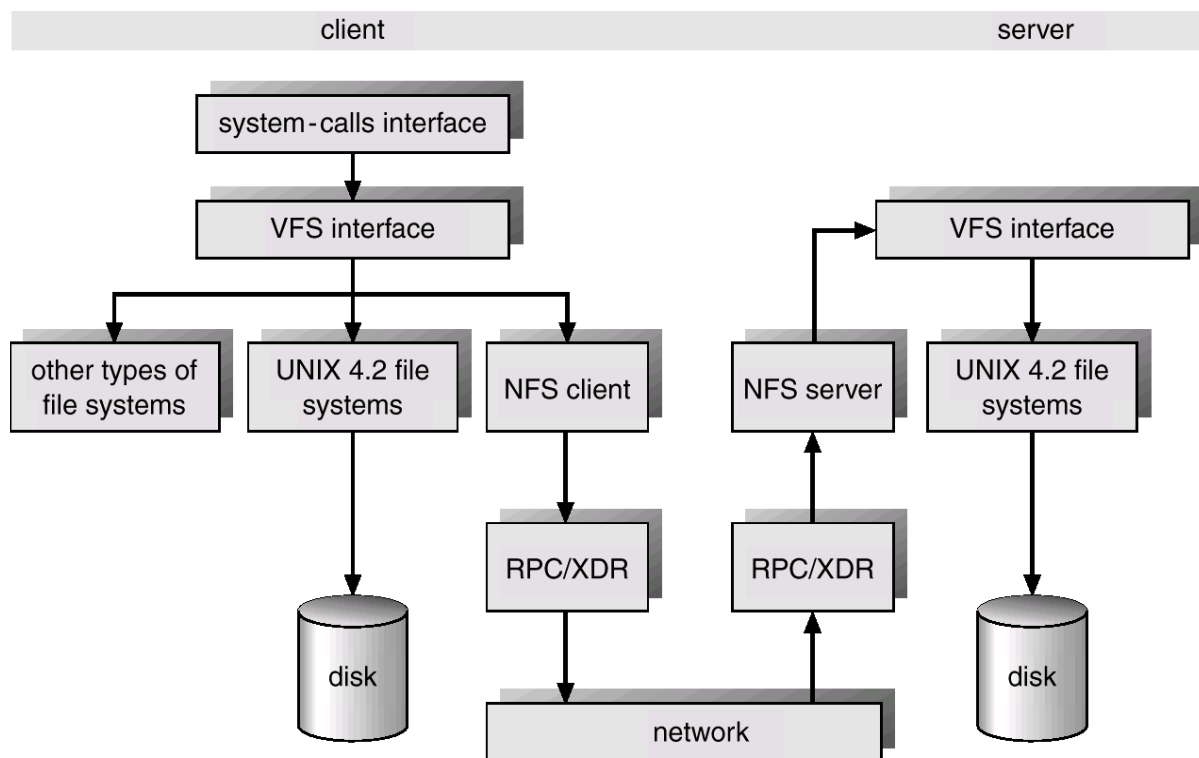
## Automounter continued

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 protocol

- 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 a pain.

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.

# AFS

Originally called the Andrew File System

– now known as 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 and location transparent (not like NFS) 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.

Files are grouped into volumes.

A volume is commonly the files of a particular workstation.

The volumes are the things that are referenced in the location database.

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

## 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.

When the 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 NFS.

This causes more worry than it should.

# Two Phase Commit Protocol

With distributed systems we want to ensure that if something goes wrong at one site we don't end up with inconsistent data.

A “transaction” is some event that to be seen as completely successful or not done at all (atomic).

We need stable storage – usually replicated on several devices – can be done with two copies.

- make the change to one (check for success)
- make the change to the other (check for success)
- if ever the copies disagree copy the original data from the second back to the first

2PC – transaction coordinator and all sites involved in a transaction have stable storage logs.

All transactions can be undone and redone safely.

Log entries and messages

- <prepare> - started the protocol, sent to all sites
- <ready> - recorded and returned if ok, <no> if not ok
- <commit> - if all reply in time, sent to all sites
- <abort> - sent if something went wrong

# Distributed Systems

A distributed system is ...

"one on which I cannot get any work done because some machine I have never heard of has crashed".

- Loosely-coupled
- network connection
- could be different OSs, or different parts of the OS
- processes must communicate via messages

What advantages does a network of sites offer?

- More work can get done
- Ability to share devices, programs and data
- Greater reliability
- Easier to expand

# Network & Distributed Operating Systems

## Network OS

- Communications layer on top of a normal OS.
- Possibly different OS.
- User is aware of different machines.
- Some can copy files across the network but not share them. e.g. ftp

In this case the file location is explicitly known.

- Others can share files but the location is still part of the name.

## Distributed OS

Aim to have the system look like one machine.

There is no difference (except speed) between accessing local and remote resources (location transparency).

The Distributed OS can move resources and processes (migration transparency).

# Before next time

## Read from the textbook

16.6 – An Example: AFS

17.3.1 – The Two-Phase Commit Protocol

15.1.1 – Advantages of Distributed Systems

15.1.2 – Types of Distributed Operating Systems