Lecture Topics: 12/3

- Networked and distributed systems
- Distributed file systems
  - NFS
  - AFS
  - WWW

Network Terms

- In a network, some resources are *local* and some are *remote*
- A machine providing a resource is called a *server*; a machine using a resource is called a *client*
- Two schemes for providing networked resources:
  - *Network* OS
  - *Distributed* OS

Network Operating Systems

- To use a remote resource, a user must either
  - log into remote machine, or
  - transfer data from remote machine and use it locally
- Either way, the user is aware that multiple computers are involved
- User must be aware of which computers provide which resources

Distributed Operating Systems

- To use a remote resource, the user accesses it as if it were local
- The operating system keeps track of the details
- The user need never know the network is involved (until something breaks)
- "A distributed system is one in which the failure of a machine I've never heard of can prevent me from doing my work." --Lamport

Why Distributed Systems?

- Resource Sharing
  - Only one printer, share it over the network
- Computation Speedup
  - Parallel computation on multiple computers
- Reliability
  - One node fails, but the service goes on
- Communication
  - Sharing information between people and processes that are far apart

Distributed File Systems

- A motivating example for distributed systems
- Store (all) files on a *file server* rather than the local hard disk
- Before, all file operations involved local OS and local disk
- Now, they involve local OS, network, remote OS, and remote disk

Why Distributed File Systems?

- Sharing programs everyone uses
  - Save on disk space
  - Make upgrades easier
- Reliability
- Uniformity of function in clusters
- Diskless workstations

Naming

- Goals: *location transparency* and *location independence*
- Name is combination of location and filename
- Name is in a part of the local tree that tells the OS the file's true location
- Single global namespace; separate service for mapping names to locations

Performance

- Now that files are coming from far away, they make take a long time to get to the local machine
  - Disk is already so slow we may not notice a difference
  - As always, we can use caching
    - memory, in the usual file cache
    - local disk
  - Cache updates: write-back vs. write-through

Security

- How does the server know who the client is?
  - Bad clients may lie and get access to files they shouldn't
- How does the client know who the server is?
  - If a bad server masquerades as the good one, it may serve up false files
- File transfer "in the clear" over the network

Sun NFS

- Both a specification and an implementation
- Remote files are just part of the tree of all files accessible from a given machine
- Servers are *stateless* - they do not "remember" anything from one request to another
- Statelessness is supposed to allow them to support a large client population

CMU's AFS

- Originally a research project; eventually a commercial product
- All computers in the world running AFS are interconnected and can see each other's files (mostly)
- Servers are *stateful* - they do "remember" information from request to request
- Statefulness is supposed to allow them to support a large client population

The World Wide Web

- Can think of the web as a huge, mostly read-only file system
- Naming has neither location transparency nor location independence
  - The name reflects the file's location
  - If the file moves, the name changes
- Performance aided by caching
- Global security is hard because administration is also distributed