Distributed Systems

Fred Long
based on slides originally written by Clive King
and modified by Mike Tedd and Edel Sherratt

- One of the main aims of a network is to allow data that physically resides on one machine to be visible on other machines
- Distributed file systems attempt to provide a mechanism for consistent and reliable access to data stored at one location across a network of a variety of architectures

- The *Network File System* (NFS) developed by Sun gives a remote file system the appearance of being locally mounted
- The NFS protocol was designed to be portable across different machines, operating systems, network architectures, and transport protocols
- NFS portability is achieved by using RPC primitives on top of an *eXternal Data Representation* (XDR)
- This solution allows NFS to be implemented on a wide range of systems, e.g., Unix, VMS, CM VMS, MS-DOS, etc.

- NFS assumes a hierarchical file system, with directories at all but the bottom level
- Each entry in a directory has a string name
- Different operating systems may have restrictions on the depth of the tree or on the names used, as well as using different syntax to represent a “pathname”
- A “file system” is viewed as a tree on a single server (usually a single disc or physical partition) with a specified “root” location

- The NFS protocol was intended to be as *stateless* as possible
- That is, a server should not need to maintain any protocol state information about any of its clients in order to function correctly
- Stateless servers have a distinct advantage over stateful servers in the event of failure
  - a client need only retry a request until the server responds
  - a client need not be aware that the server failed, or that the network temporarily went down
- In contrast, the client of a stateful server will need to either detect a server failure and rebuild the server’s state when it comes back up, or cause client operations to fail

- While NFS is the most widely used distributed file system, a number of others have emerged:
  - *Remote File System* (RFS) developed by AT&T, aims to provide the same services as NFS but is stateful and attempts to provide full Unix system semantics such as file and record locking which NFS cannot do
  - NFS does not allow for replication of file systems to provide higher availability should one server fail
  - A number of research file systems exist that do provide replication, such as the *Andrew File System* developed at Carnegie Mellon University

Directory Services

- Directory services provide the basis for distributed programming and interhost communication
- On a small network, it is feasible for all information relating to other hosts and networks to be stored and retrieved from local tables that reside on each host
- An administrator may update each host when any of the information changes

- As a network increases in size above a few tens of nodes, it is clearly not feasible for an administrator to take responsibility for keeping the tables on each node up to date
- *Network Information Service* (NIS+) is an attempt to provide a repository of network names and attributes that addresses the problems of management and resource location in heterogeneous distributed systems

- A specific design goal of NIS+ is that it will work with a range of operating system platforms
- NIS+ allows a client, such as an application program, to look up information on network resources and access the resources in a location independent way
- The server side of NIS+ is a repository which acts as a central location for network information

- The *Domain Name Service* (DNS) is a distributed database and protocol for mapping IP addresses to hostnames over a potentially heterogeneous network
- The fundamental idea of DNS is a hierarchical name space, with the hierarchy roughly corresponding to organisational structure, and names using "." as the character to mark the boundary between hierarchy levels

- DNS has three major components:
  - *Domain Name Space and Resource Records* (RR): specifications for a tree structured name space and data associated with the names
  - *Name Servers*: server programs which hold information about the domain tree's structure and set information
  - *Resolvers*: programs that extract information from name servers in response to client requests

CORBA

- Everything is becoming object oriented, but support for OO breaks down when faced with distribution
- The *Common Object Request Broker Architecture* (CORBA) provides a standard for the distribution of objects
- CORBA is an *ad hoc* standard being developed by the Object Management Group (OMG), a mainly industrial-based committee founded by SunSoft in 1989

- CORBA lays down the standards to be adopted by an *Object request Broker* (ORB)
- An ORB provides the integration and communication mechanisms for objects
- CORBA goes on to define *services* which may be supported, e.g., naming, events, properties, security, transactions, etc.
- In writing a CORBA compliant application, one defines the data structures in DDL, the interfaces in IDL, and the methods in an OOP (currently, C++, Ada, Java and SmallTalk are supported)