

Introduction to Distributed Systems

- Distributed Systems
 - Network OSs vs. distributed OSs
 - Research and design issues
 - *Reading: Coulouris, chapters 1, 2, and 3*
-

Distributed vs. Centralized Systems

- Advantages of Distributed Systems:
 - Reliability.
 - Sharing of resources.
 - Aggregate computing power.
 - Openness/Scalability
 - Disadvantages of distributed systems:
 - Security.
 - Physical distribution of resources vs. demand.
 - Computing power per node is limited.
-

Distributed Operating Systems

- Requirements:
 - Provide user with convenient virtual computer.
 - Hide distribution of resources.
 - Mechanisms for protecting resources.
 - Secure communication.

- Definition

Distributed OS looks to user like ordinary centralized OS, but runs on multiple, independent CPUs.

- Use of multiple processors is invisible.
 - User views system as virtual uniprocessor.
-

Distributed vs. Networked OS

- Transparency:
 - How aware are users of the fact that multiple computers are being used?
 - Network OS:
 - Users are aware where resources are located
 - Network OS is built on top of centralized OS.
 - Handles interfacing and coordination between local OSs.
 - Distributed OS:
 - Designed to control and optimize operations and resources in distributed system.
-

Network OSs

- Definition:

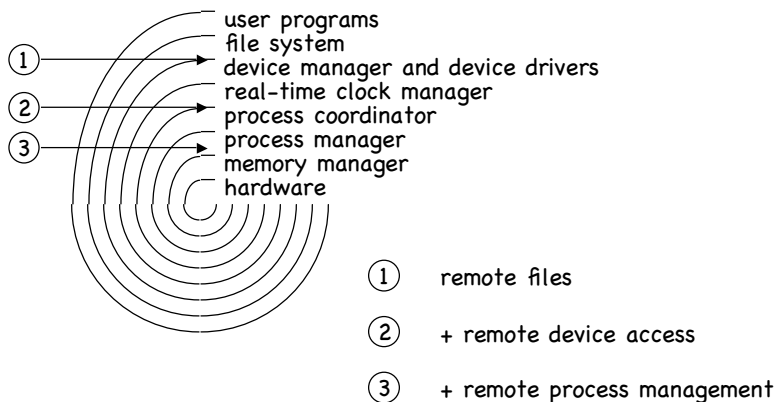
A network OS is a collection of OSs of computers connected through a network incorporating modules to provide access to remote resources.

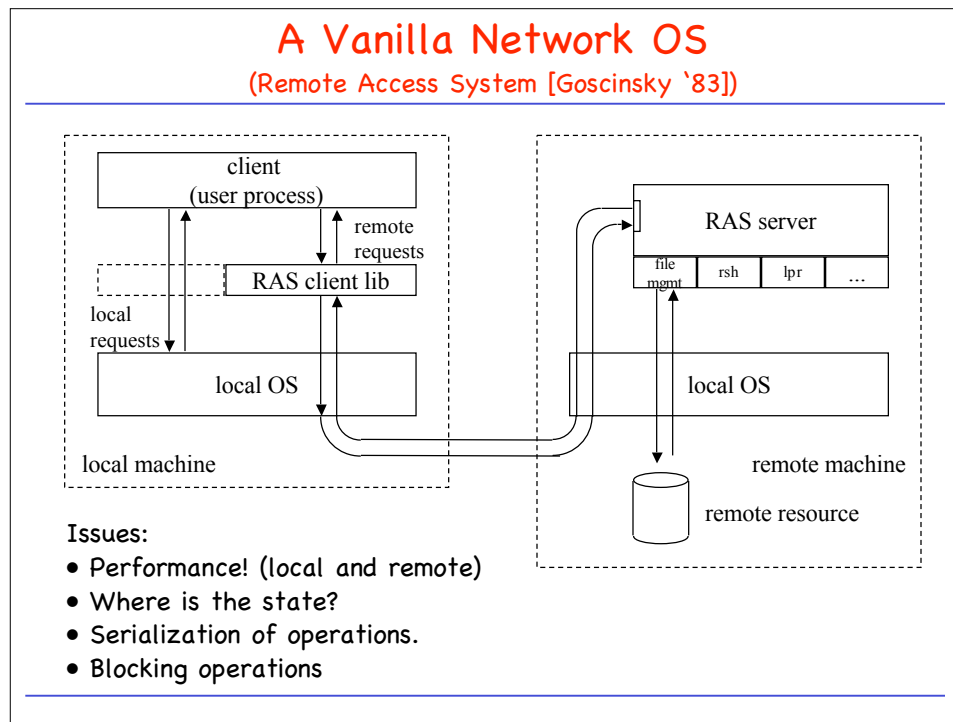
- Characteristics:

- Each computer has private OS.
 - User works on his own machine and remotely logs in to other computers.
 - Users are aware of location of files.
 - Limited fault tolerance.
-

IPC and Transparency

The insider's view of a centralized OS.
 (Roughly patterned after XINU [Comer 1984])





Research and Design Issues

- Communication model
- Paradigms for process interaction
- Transparency
- Heterogeneity
- Autonomy and/or interdependence
- Reliable distributed computing
- Replication

Communication Model

- ISO/OSI Model
 - Physical
 - Datalink
 - Network
 - Transport
 - Session
 - Presentation
 - Application
- An alternative, e.g. *Functional*, Model
 - Physical
 - same as ISO/OSI
 - Datagram
 - connectionless service between source and destination process
 - location of services
 - Transport
 - reliable transport between client and server
 - "transaction level"
 - Binding
 - location of resources within the server
 - logical connection between client and server
 - User
 - request semantics

Process Interaction: Client/Server

Server: A subsystem that provides a particular type of service to *a priori* unknown clients.

- Control functionally distributed among the various servers in the system.
- Control of *individual* resources is centralized in a server. (localized?)
- Problems:
 - Reliability/Availability
 - Scalability
 - Replication?

Process Interaction: Pipe Model

Pipe: Communication facility to transfer data between processes in FIFO order. Can be used for synchronization purposes.

- Named/unnamed pipes
 - Pipes for secure IPC
 - Pipes across network?
 - Multicast pipes?
-

Process Interaction: RPC Model

- Allows a process to call a procedure on a remote computer.
 - Looks like a local procedure.
 - Blocking characteristics?
 - *More about this later.*
-

Transparency

Transparency: Make the network invisible to user/applications.

Various degrees of transparency:

- Access Transparency
 - Location Transparency
 - Name Transparency
 - Data Transparency
 - Execution Transparency
 - Performance Transparency
-

Autonomy and Interdependence

- Disadvantage generated by interdependence:
 - cannot work stand-alone
 - globally controlled
 - difficult to identify source of authority and responsibility
 - what about mutual suspicion?
 - Reasons for autonomy:
 - policy freedom
 - robustness
 - cooperation between mutually suspicious users
-