

# CSCE 455/855

## Distributed Operating Systems

### Introduction to Distributed Systems

Steve Goddard  
*goddard@cse.unl.edu*

<http://www.cse.unl.edu/~goddard/Courses/CSCE855>

1

#### What is meant by Distributed Computing?

- ◆ Working definition:
  - » "... a collection of independent computers that appear to the users of the system as a single system." [p. 2]
- ◆ Sometimes referred to as "Client-Server Computing"
- ◆ Defining characteristics
  - » individual computers are independent
    - ◆ i.e. do not share memory, CPU
  - » transparency
    - ◆ achieved at user or programmer viewpoint
  - » all achieved through a layer of software

2

## **Distributed System Advantages**

---

- ◆ More reliable than time-sharing systems
  - » Another definition by Leslie Lamport: a distributed system is “one in which the failure of a computer you didn't even know existed can render your own computer unusable”
- ◆ Resource sharing (printers, cpu's)
- ◆ Cost-effective
- ◆ Performance advantage (utilize idle CPUs)
- ◆ Incremental growth (network allowing)

3

## **Distributed System Disadvantages**

---

- ◆ Security problems
- ◆ Software
  - » load distribution difficult
  - » reliability needs to be built in
  - » control information is distributed
- ◆ Communications overhead

4

## **Centralized and Distributed Systems**

- ◆ Centralized/Distributed
  - » process management
  - » resource management
  - » memory management
  - » file management
  - » protection
- ◆ Distributed system
  - » distributed management of otherwise centralized facilities
  - » communication overhead
  - » distributed control mechanisms
  - » interconnecting heterogeneous systems
- ◆ Major problems for distributed systems not faced in centralized systems
  - » lack of global state information
  - » communication & transmission delays

5

## **Networked Operating Systems**

- ◆ Transparency not very high
  - » user must know location of services
  - » file servers provide some degree of transparency
- ◆ Mounting file systems
  - » mount remote files (directories) on local machine
  - » access accomplished by sending a message
    - ◆ system (transparently) distinguishes between local and remote files
  - » files mounted *onto* individual machines
    - ◆ therefore possible to mount a file at different places on to machines
    - ◆ or one machine may mount files that another doesn't

6

## **Distributed System Goals**

- ◆ Control resources on the network
  - » make effective use of disk space, cpu cycles
  - » synchronize events
- ◆ Provide a convenient virtual computer
  - » easy way to communicate, access resources
  - » naming, sockets, RPC
- ◆ Make access to distributed resources transparent to users
  - » want user to see one machine
  - » OS provides glue that makes a number of machines seem as one

7

## **Distributed System Goals (cont.)**

- ◆ Protect system resources
  - » unauthorized and wayward users pose additional problems
  - » provide easy access yet provide enough security
- ◆ Provide secure, reliable, fast communication
  - » the basis for all distributed facilities

8

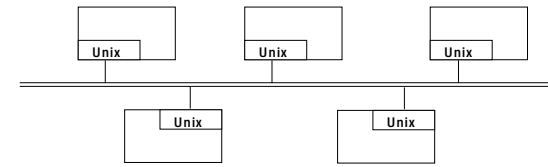
## The Continuum From Networked to Distributed Operating Systems

- ◆ Transparency (user's perspective)
  - » Networked:
    - ❖ user must know where services are located
    - ❖ calvin:/home/goddard/dissertation/thesis.tex
    - ❖ must log into different machines (*rlogin, telnet*)
    - ❖ explicitly transfer files between machines
  - » Distributed:
    - ❖ users are unaware of where services are located
    - ❖ /home/goddard/dissertation/thesis.tex
    - ❖ different machines accessed without user knowledge
    - ❖ transferring between different machines irrelevant (from user's perspective)

9

## Distributed System Architecture

- ◆ Monolithic kernel
  - » each computer has copy of OS
  - » Unix, etc.

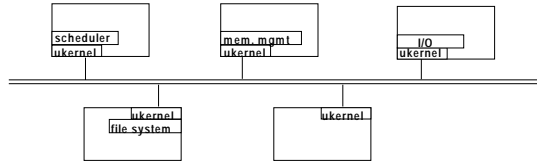


10

## Distributed System Architecture (con't)

### ◆ Microkernel

- » kernel provides minimal services
  - ❖ interprocess communication
  - ❖ primary memory management
  - ❖ process management
  - ❖ I/O
- » everything else done through server
  - ❖ microkernel sends requests to servers



11

## Distributed System Architecture (con't)

### ◆ Microkernel trade-offs

- » servers can support heterogeneous systems
- » flexibility: can interchange servers
- » performance can suffer
  - ❖ communication overhead

12

## Transparency

---

- ◆ Location transparency
  - » users don't need to know where resources are located
- ◆ Migration transparency
  - » files, processes move without impacting users
  - » mounting files is not transparent
    - ◆ if file moves, mounting system ill not see it
- ◆ Replication transparency
  - » system can make multiple copies of files
    - ◆ for increased reliability, availabilitiy

13

## Transparency

---

- ◆ Concurrency transparency
  - » resources can be accessed simultaneously
- ◆ Parallelism transparency
  - » system uses parallelism without user intervention...  
some day

14

## Achieving Transparency

- ◆ Operating system responsible for making distributed system appear as one
  - » request a service
    - ❖ determine where service is located
    - ❖ translate request into appropriate form
    - ❖ send a message
    - ❖ receive results
    - ❖ translate results into form needed by user
  - » receive a request
    - ❖ service set-up (make service available)
    - ❖ receive message
    - ❖ process service
    - ❖ send results

15

## Other Design Issues

- ◆ Flexibility
  - » accommodating heterogeneous hardware and operating systems
- ◆ Reliability
  - » availability, fault tolerance

16

## **Other Design Issues**

---

- ◆ Performance

- » communication efficiency is key
- » distributed goals may reduce performance: fault tolerance through replication

- ◆ Scalability

- » depends critically on software
  - ❖ often seductive to use a centralized algorithm to coordinate distributed systems
  - ❖ distributed algorithms harder to understand, more scalable

17