CprE 450/550x
Distributed Systems and Middleware

Distributed Systems: Goals and a Taxonomy

Yong Guan
3216 Coover
Tel: (515) 294-8378
Email: guan@ee.iastate.edu
January 29, 2004

Readings for Today's Lecture

- Chapter 1 of "Distributed Systems: Principles and Paradigms"

Four Goals of Distributed Systems

- Making it easy for users to access remote resources and to share them with others in a controlled way.
- Hiding the fact that the processes and resources are physically distributed across multiple computers (Transparency)
- Offering services according to standard rules (in terms of both syntax and semantics) (Openness)
- Making it easy to be extended in terms of size, geography distribution, and management. (Scalability)

Transparency in a Distributed System

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

Different forms of transparency in a distributed system.

5

Openness

- Interface Definition Language (IDL)
- Interoperability
- Portability
- Separation policy from mechanism

6

Scalability

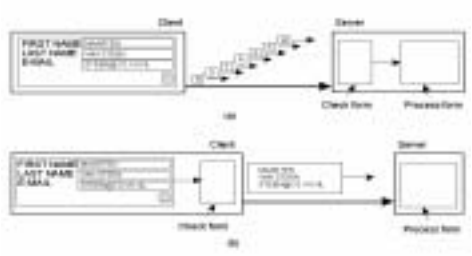
- Three dimensions:
  - **Size**
  - **Geography distribution**
  - **Management**

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

Examples of scalability limitations.

7

Scaling Techniques (1)



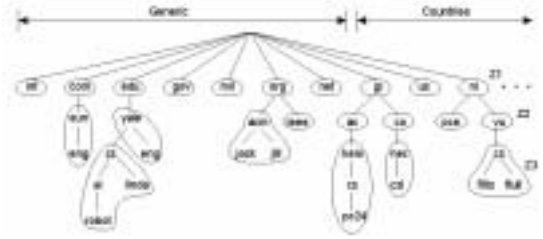
The difference between letting:

a) a server or

b) a client check forms as they are being filled

8

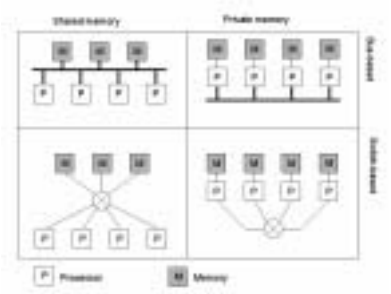
Scaling Techniques (2)



An example of dividing the DNS name space into zones.

9

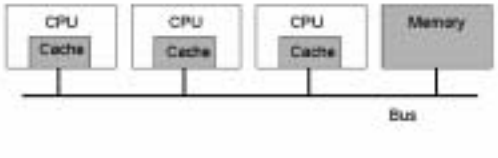
Classification of Distributed Systems: from hardware perspective



Different basic organizations and memories in distributed computer systems

10

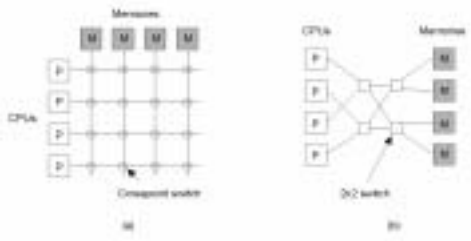
Multiprocessors (1)



A bus-based multiprocessor.

11

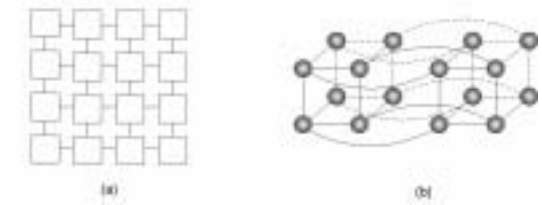
Multiprocessors (2)



a) A crossbar switch
b) An omega switching network

12

Homogeneous Multicomputer Systems



a) Grid
b) Hypercube

Cluster of Workstations (COWs)

13

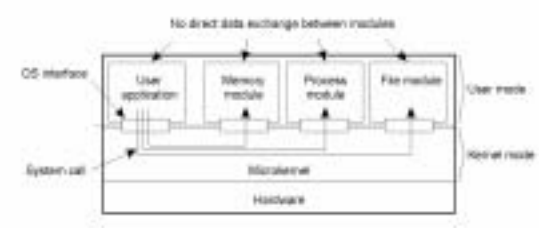
Software Concepts

System	Description	Main Goal
DOS	Tightly-coupled operating system for multi-processors and homogeneous multicomputers	Hide and manage hardware resources
NOS	Loosely-coupled operating system for heterogeneous multicomputers (LAN and WAN)	Offer local services to remote clients
Middleware	Additional layer atop of NOS implementing general-purpose services	Provide distribution transparency

- An overview of
- DOS (Distributed Operating Systems)
- NOS (Network Operating Systems)
- Middleware

14

Uniprocessor Operating Systems



- Separating applications from operating system code through
- a microkernel.

15

Multiprocessor Operating Systems (1)

```
monitor Counter {
private:
 int count = 0;
public:
  int value() { return count;}
  void incr () { count = count + 1;}
  void decr() { count = count – 1;}
}
```

- A monitor to protect an integer against concurrent access.

16

Multiprocessor Operating Systems (2)

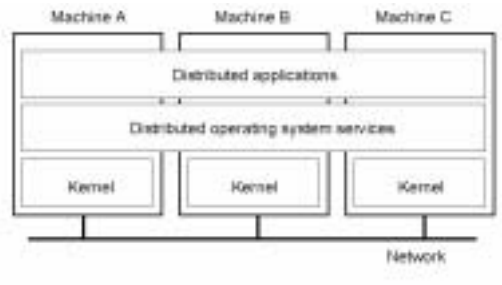
```
monitor Counter {
private:
  int count = 0;
  int blocked_procs = 0;
  condition unblocked;
public:
  int value () { return count;}
  void incr () {
      if (blocked_procs == 0)
        count = count + 1;
      else
        signal (unblocked);
  }

void decr() {
if (count ==0) {
  blocked_procs = blocked_procs + 1;
  wait (unblocked);
  blocked_procs = blocked_procs – 1;
  }
 else
  count = count – 1;
 }
}
```

A monitor to protect an integer against concurrent access, but blocking a process.

Multicomputer Operating Systems (1)

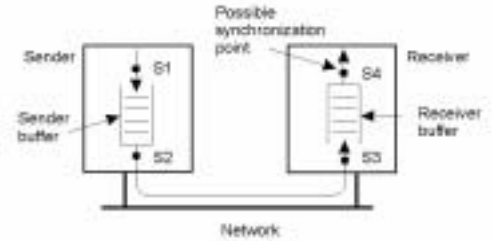
17



- General structure of a multicomputer operating system

Multicomputer Operating Systems (2)

18



- Alternatives for blocking and buffering in message passing.

Multicomputer Operating Systems (3)

19

Synchronization point	Send buffer	Reliable comm. guaranteed?
Block sender until buffer not full	Yes	Not necessary
Block sender until message sent	No	Not necessary
Block sender until message received	No	Necessary
Block sender until message delivered	No	Necessary

- Relation between blocking, buffering, and reliable communications.

Distributed Shared Memory Systems (1)

20

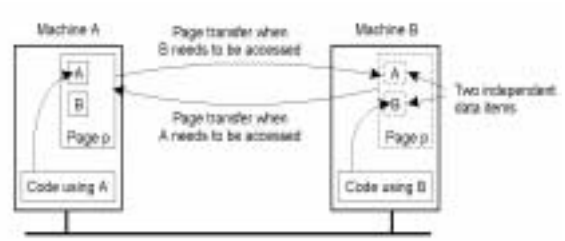
a) Pages of address space distributed among four machines

b) Situation after CPU 1 references page 10

c) Situation if page 10 is read only and replication is used

Distributed Shared Memory Systems (2)

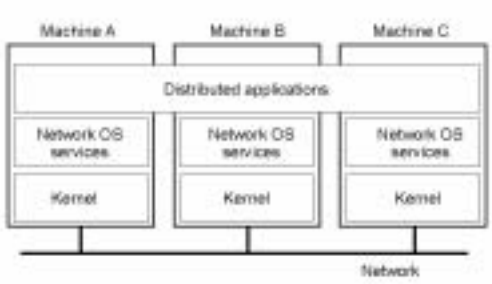
21



- False sharing of a page between two independent processes.

Network Operating System (1)

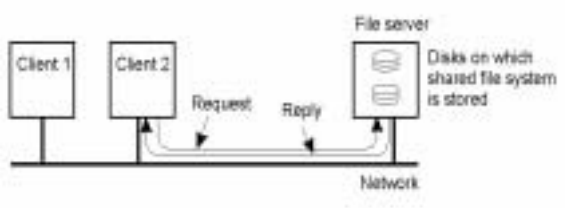
22



- General structure of a network operating system.

Network Operating System (2)

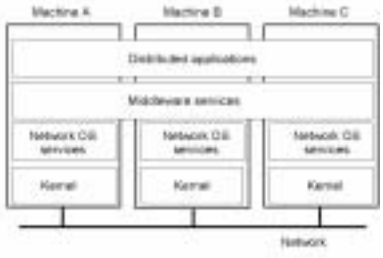
23



- Two clients and a server in a network operating system.

Middleware

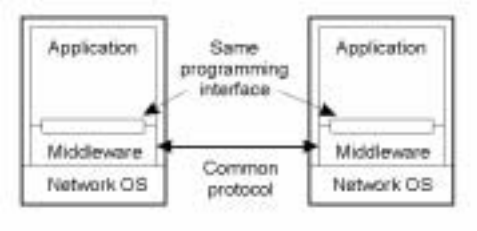
24



- General structure of a distributed system as middleware.

Middleware and Openness

25



- In an open middleware-based distributed system, the protocols used by each middleware layer should be the same, as well as the interfaces they offer to applications.

Comparison between Systems

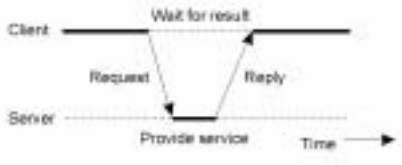
26

Item	Distributed OS		Network OS	Middleware-based OS
	Multiproc.	Multicomp.		
Degree of transparency	Very High	High	Low	High
Same OS on all nodes	Yes	Yes	No	No
Number of copies of OS	1	N	N	N
Basis for communication	Shared memory	Messages	Files	Model specific
Resource management	Global, central	Global, distributed	Per node	Per node
Scalability	No	Moderately	Yes	Varies
Openness	Closed	Closed	Open	Open

A comparison between multiprocessor operating systems, multicomputer operating systems, network operating systems, and middleware based distributed systems.

Clients and Servers Model

27



General interaction between a client and a server.

An Example Client and Server (1)

28

- The *header.h* file used by the client and server.

29

An Example Client and Server (2)

- A sample server.

30

An Example Client and Server (3)

- A client using the server to copy a file.

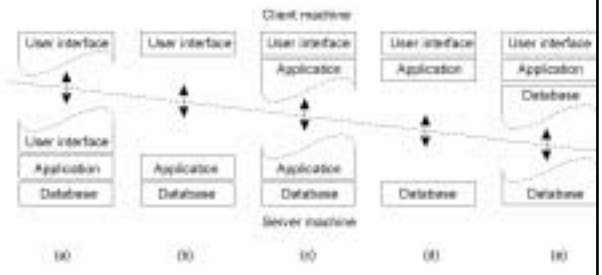
31

Processing Level

- The general organization of an Internet search engine into three different layers

32

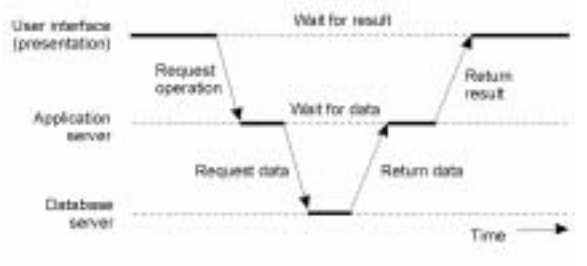
Multitiered Architectures (1)



Alternative client-server organizations (a) – (e).

33

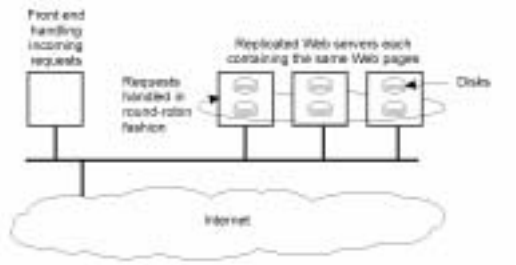
Multitiered Architectures (2)



An example of a server acting as a client.

34

Modern Architectures



- An example of horizontal distribution of a Web service.

35

Any Questions?

See you next time.