Introduction

- ❑ What is a Distributed System?
- ❑ Different types of distributed systems
- ❑ Definition(s) of distributed systems
- ❑ Examples of distributed systems
- ❑ Important issues within distributed systems

What is a Distributed System?

- ❑ A Distributed system is a computer system that uses computer communication.

Hardware:

- ❍ Computer networks
- ❍ Processors

Software:

- ❍ Computer Communication
- ❍ Processes

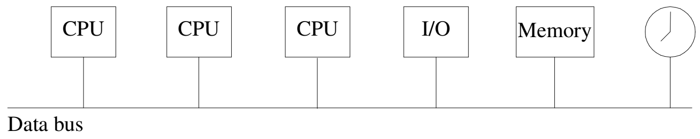
Applications

- ❍ Distributed programs
- ❍ Distributed algorithms

Different types of distributed systems

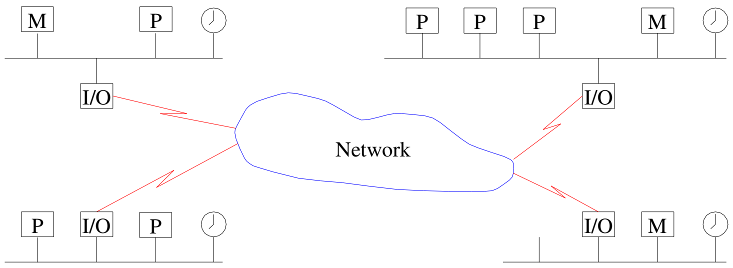
- ❑ There are two types of systems that are called Distributed Systems!
  - ❍ Hard Coupled Systems
  - ❍ Loosely Coupled Systems
    These are considered in this course.

Hard Coupled Systems



- ❑ Hard coupled systems have a common clock that allows the different processes to agree on a common time.
- ❑ Multiprocessor

Loosely Coupled Systems



- ❑ Loosely Coupled Systems do their communication through data networks.
  - — Thus they can **not** read each other clocks which will result in synchronization problems.
  - — and they can **not** read each other memories which result in different knowledge about the system state.

Multi-core processors

Lately there has been a lot work done on Multi-core processors

- ❑ Can be
  - — Hard coupled

    or
  - — Loosely coupled

Parallel computing

Loosely Coupled Systems	Hard Coupled Systems
Interaction via messages	Interaction through shared memory
Decentralized control	Centralized control
The global state **not** known	The global state known
Distributed system	Multiprocessor system

Why have a Distributed System?

- ❍ computing requirements
  - increased computer power
  - high availability
  - high reliability
- ❍ organizational requirements
  - geographical adaptation
    - locally
    - globally
    - mobile computing
  - Modularization:
    - easy to support
    - easy to enlarge
    - sharing of resources

A Modularized Architecture

- simple system design, specialized processors/processes
- simple installation of units (step wise)
- simpler support
- increased availability
  - A system using redundancy. It continues to work although one, two, or more units fail.
  - A *n-resilient* system continues to work if at most *n* units fail at the same time.
  - This can be used to achieve a system with high availability and/or reliability.

Here units might be: processors, communication links, data repositories, other devices ...

⇒ Fault Tolerance

Definition of Distributed System (1)

(Tanenbaum)

“A distributed system is a collection of independent computers that appears to it users as a single coherent system”

Definition of Distributed System (2)

(Ezhilchelvan)

1. There are multiple occurrence of processors and other general units.
2. The system is loosely coupled,
   i.e. all communication among the processes is made by sending messages and the processes are not sharing a common memory.
3. The exist common control in some sense within the system,
   i.e. there is dynamic interaction among the processes.
4. The system is transparent.,
   i.e. a user should be able to request a resource without knowing its physical location.
5. The system components should be autonomous,
   i.e. no Master-Slave architecture.

Definition of Distributed System (3)

(LeLann)

1. Multiple occurrence of system and user processes.
2. Modular architecture,
   i.e. not a fixed number of processors.
3. All communication among the processes is made by sending messages and the processes are not sharing a common memory.
4. The exist some common control within the system,
   i.e. there is dynamic interaction among the processes.
5. The time it takes from sending to receiving a message is varying and always >0.
   This leads to:
   - The messages will cause a time cost that is big compared with other costs within the system. Subsequently it will be desirable to minimize the number of messages when optimizing the system.
   - There can be no knowledge of the systems global state by ant unit within the system.

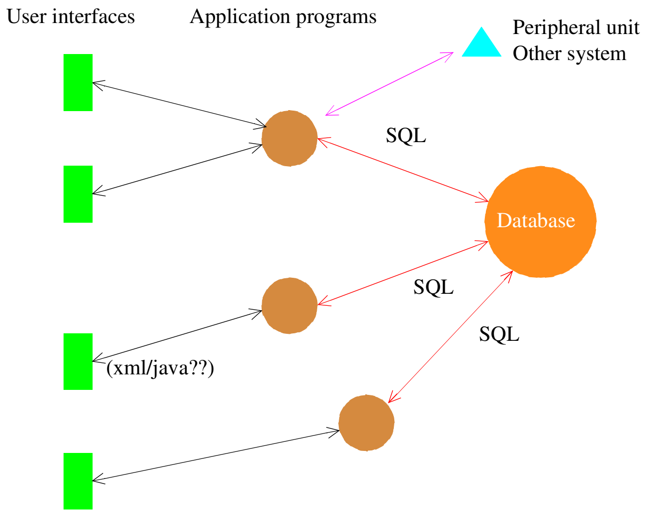
Definition of Distributed System (4)

“If your program does not work because a computer or program that you have never known the existences of is not working properly, then you have a distributed system”

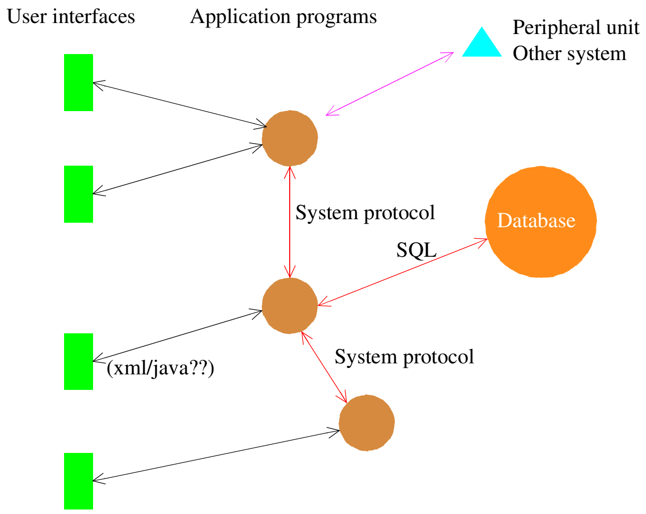
Examples of distributed systems

- □ office computers that share file systems, printers. E-mail ...
- □ air ticket booking systems
- □ bank systems
- □ real-time system for manufacturing
- □ telecommunication systems
- □ e-mail systems
- □ conference systems
- □ database systems
- □ World Wide Web
- □ multimedia systems
- □ Web Services
- □ **more or less every system of some size!**

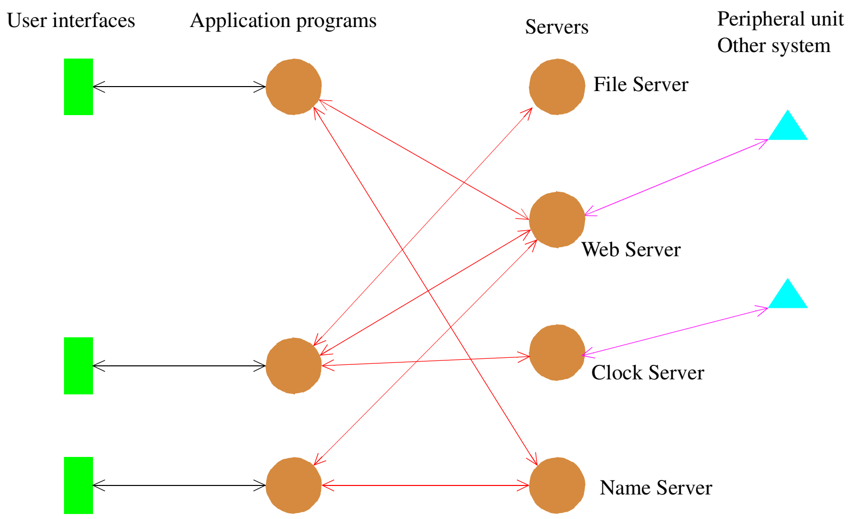
Typical computer system



A more distributed solution



A modularized solution



User interfaces

Application programs

Servers

Peripheral unit
Other system

File Server

Web Server

Clock Server

Name Server

Important Qualities

- Availability
- Reliability
- Scalability
- Openness
- Performance
- Transparency

Pitfalls

These are some typical assumptions that makes a distributed system to fail:

1. The network is reliable.
2. The network is secure.
3. The network is homogenous.
4. The topology does not change
5. Latency is zero.
6. Bandwidth is infinite.
7. Transport cost zero.
8. There is one administrator.

Important issues within distributed systems

Course Content

- Computer communication and networks
- Inter process communication
- Inter process synchronization
- Distributed algorithms
- Reliable systems
- Systems with high availability
  Fault Tolerant Systems
- Distributed operating systems
  - Distributed file systems
- Security