Introducing:
Free/Open Source Software
Distributed Systems
Real Time Systems
and Multimedia Systems

Rahmat M. Samik-Ibrahim

VauLSMorg

Ver. 2013-05-14 rev 02

References 1 (Cut&Pasted)

- Beberg, Distributed Systems: Computation With a Million Friends (and a few Foes), EE380, Stanford University, 2008.
- FSF, What is Free Software, www.gnu.org, accessed on 12-May-2013.
- Haridi, Introduction to Distributed Systems
- Jayaswal, Administerting Data Centers, Wiley, 2001.

References 2 (Cut&Pasted)

- Microchip, Concepts of Real Time Systems, WebSeminar, accessed on 10-May-2013.
- OSI, The Open Source Definition, www.opensource.org, accessed on 12-May-2013.
- Silberschatz, et.al., Operating Systems Concepts, 9th Ed, 2012.
- –, Wikipedia: The Free Encyclopedia, Wikipedia Foundation, accessed on 12-13 May-2013.

Topics

- Free/Open Source Software
- Distributed Systems
- Real Time Systems
- Multimedia Systems

Topic #1
Free/Open Source Software

- **Topic #1: Free/Open Source Software**
- Topic #2: Distributed Systems
- Topic #3: Real Time Systems
- Topic #4: Multimedia Systems

Sub-Agenda

- Introduction
- License
- Free Software
- Open Source Software
- Free Software vs. Open Source Software
- Popular F/OSS Licenses
- Success Stories

Guess: are these following free?

- Microsoft Windows
- Microsoft Internet Explorer
- Mozilla Firefox
- GNU/Linux
- LibreOffice
- Public Domain
- Shareware
- Freeware

Is it Free Software IF you can:

- use for non-commercial usage only.
- download it from the internet for free.
- have the source code of the software
- charge a fee to development
- charge a fee to distribute
- use it for illegal activities

License

- A license agreement is needed for using a software (do-s and don't-s):
  - Can you use it?
  - Can you copy and redistribute it?
  - Can you modify it?
  - Can you transfer the license?
  - Is it viral? - allows derivative works only when licensed identically to the original

Free Software

- about liberty
- FREE like in ”free speech”
- not like FREE in ”free gift”

A matter of the users' freedom to **run, copy, distribute, study, change** and **improve** the software.

Four Essential Freedoms

0) To run the program, for any purpose.

1) To study how the program works, and adapt it to your needs (needs source code).

2) To redistribute copies/ help neighbor.

3) To improve the program, and release the improvements to the public, so that the whole community benefits (needs source code).

The Open Source Definition (1)

- Free Redistribution
- Source Code
- Derived Works
- Integrity of The Author's Source Code
- No Discrimination Against Persons or Groups

The Open Source Definition (2)

- No Discrimination Against Fields of Endeavor
- Distribution of License
- License Must Not Be Specific to a Product
- License Must Not Restrict Other Software
- License Must Be Technology-Neutral

Free = Open Source ?

- ”All religions are one!?”
- Free Software (FSF, 1985)
  - four-point definition
- Open Source Software (OSI, 1998)
  - ten-point definition
- guarantee a certain set of freedoms.
- but, different words convey different ideas

Free Software vs. Open Source

- Open Source Software
  - a development methodology
  - non-free software is a sub-optimal solution.
- Free Software: a social movement
  - non-free software is a social problem
  - Free software is the solution!

Popular F/OSS Licenses

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- Common Development and Distribution License
- GNU General Public License (GPL)
- MIT license
- GNU Library or "Lesser" General Public License (LGPL)
- Microsoft Public License
- Mozilla Public License 2.0
- Nokia Open Source License
- PHP License 3.0

Success Stories (1)

- Red Hat
  - Founded in 1993
  - Product: Red Hat Enterprise Linux
  - F/OSS for the enterprise community
  - The Fedora Project sponsor
  - the first one-billion dollar FOSS company (2012)
  - Minority owners: Intel, Netscape (Mozilla), HP, IBM, Dell, Novell.

Success Stories (2)

- Ubuntu Linux/Canonical Ltd.
  - Founded in 2004
  - A fork of Debian with a “6 months release cycle”
  - Most popular Linux Distribution for desktop/laptop
  - A distribution for laymen
  - A US$ 30 million company (2009)

End of Topic #1
Free/Open Source Software

Topic #2
Distributed Systems

- Topic #1: Free/Open Source Software
- **Topic #2: Distributed Systems**
- Topic #3: Real Time Systems
- Topic #4: Multimedia Systems

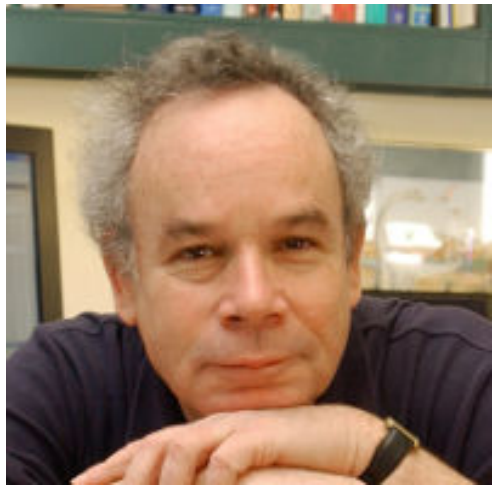
Sub-Agenda

- Introduction
- Definition
- Issues
- Distributed Operating Systems
- Network Model
- Robustness
- Design Issues
- Consolidation

A Distributed System is

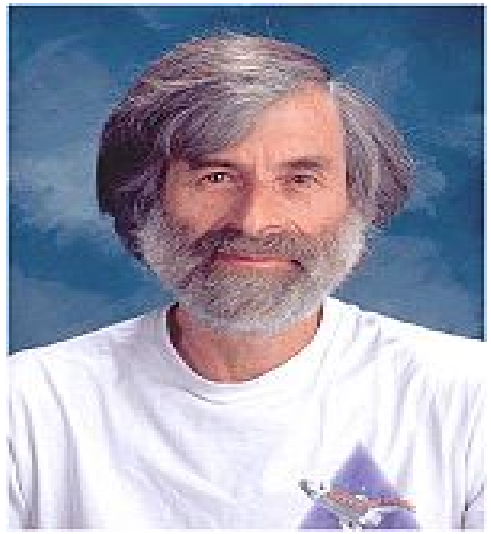
collection of loosely coupled processors interconnected by a communications network

– Silberschatz et. al.



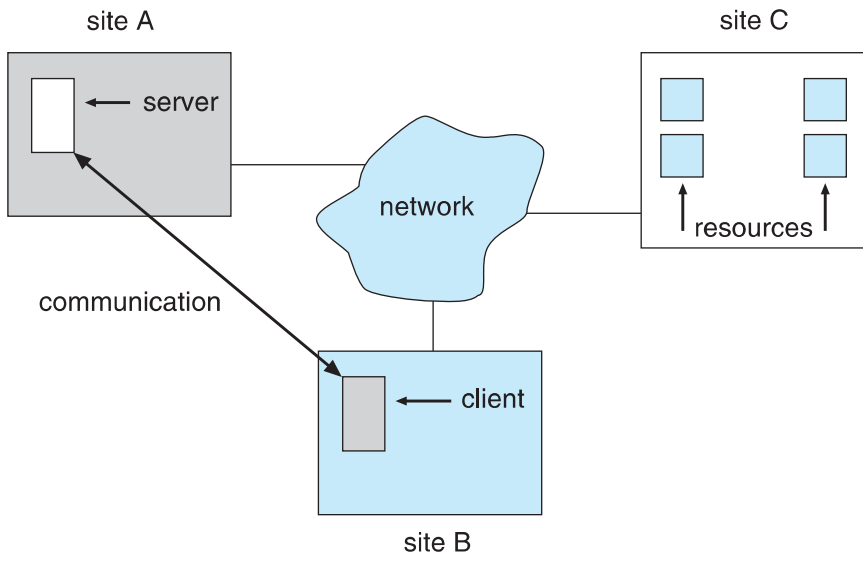
one in which the failure of a computer you didn't even know existed can render your own computer unusable.

– Leslie Lamport.

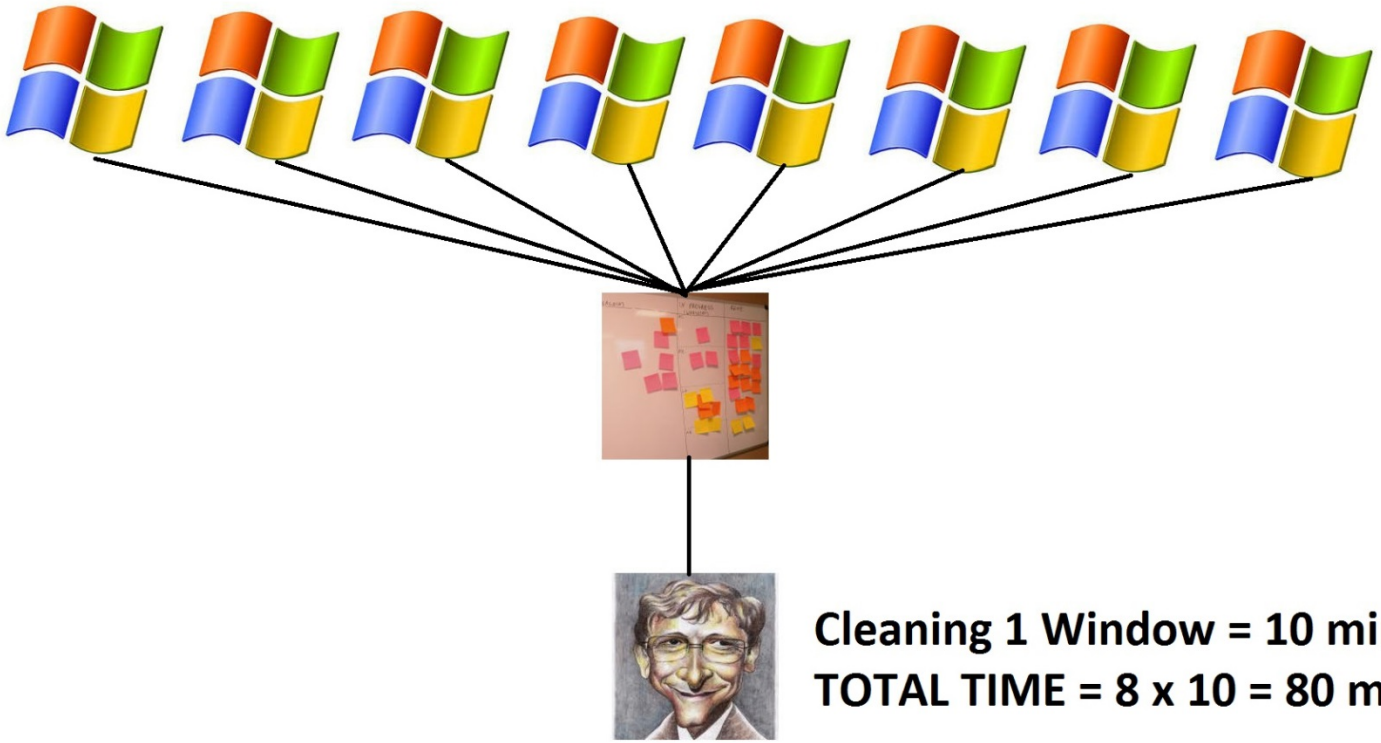


A Distributed System is

- a set of **nodes**, connected by a **network**, which appear to its users as **a single coherent system**. – Haridi
- Clients & Servers

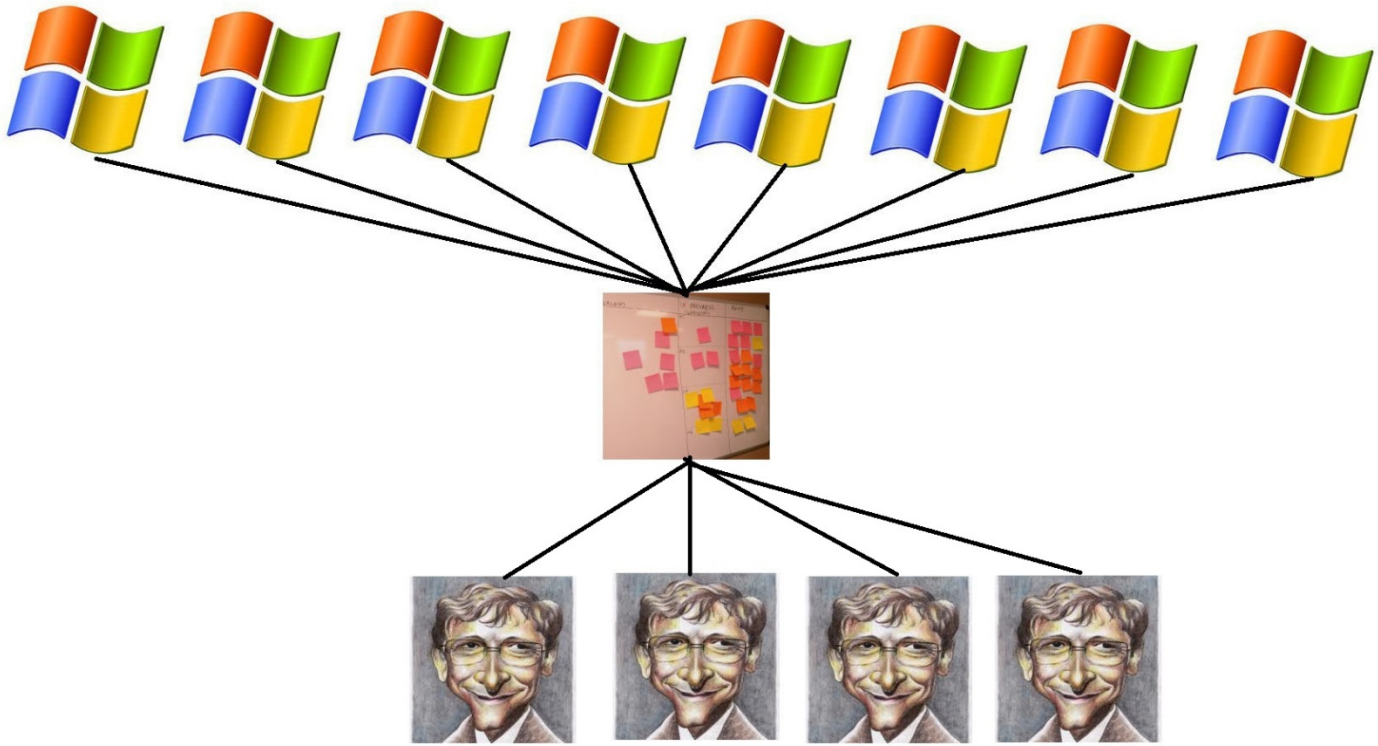


Cleaning 8 Windows



Cleaning 1 Window = 10 minutes
TOTAL TIME = 8 x 10 = 80 minutes

Cleaning By 4



TOTAL TIME = 8 /4 * 10 = 20 minutes

How about this?

1024

1024

1024

10 minutes?

1024

2048

5 minutes?

Why Distributed Systems?

- Resource Sharing
  - Most everything is now connected
- Computation Speedup
  - Spreading the load
- Reliability
  - One is down, others can take over
- Communication

Central vs. Distributed

- SLA
- “Reliable”
- Homogeneous
- Trusted
- Cost=Money

- Untrusted systems
- Unstable
- Heterogeneous
- Distrusted
- Cost=Time

Issues

- Improve Scalability
  - Concurency: web/mail server
- Reliability
  - Tandem System
- Inherent Distribution
  -

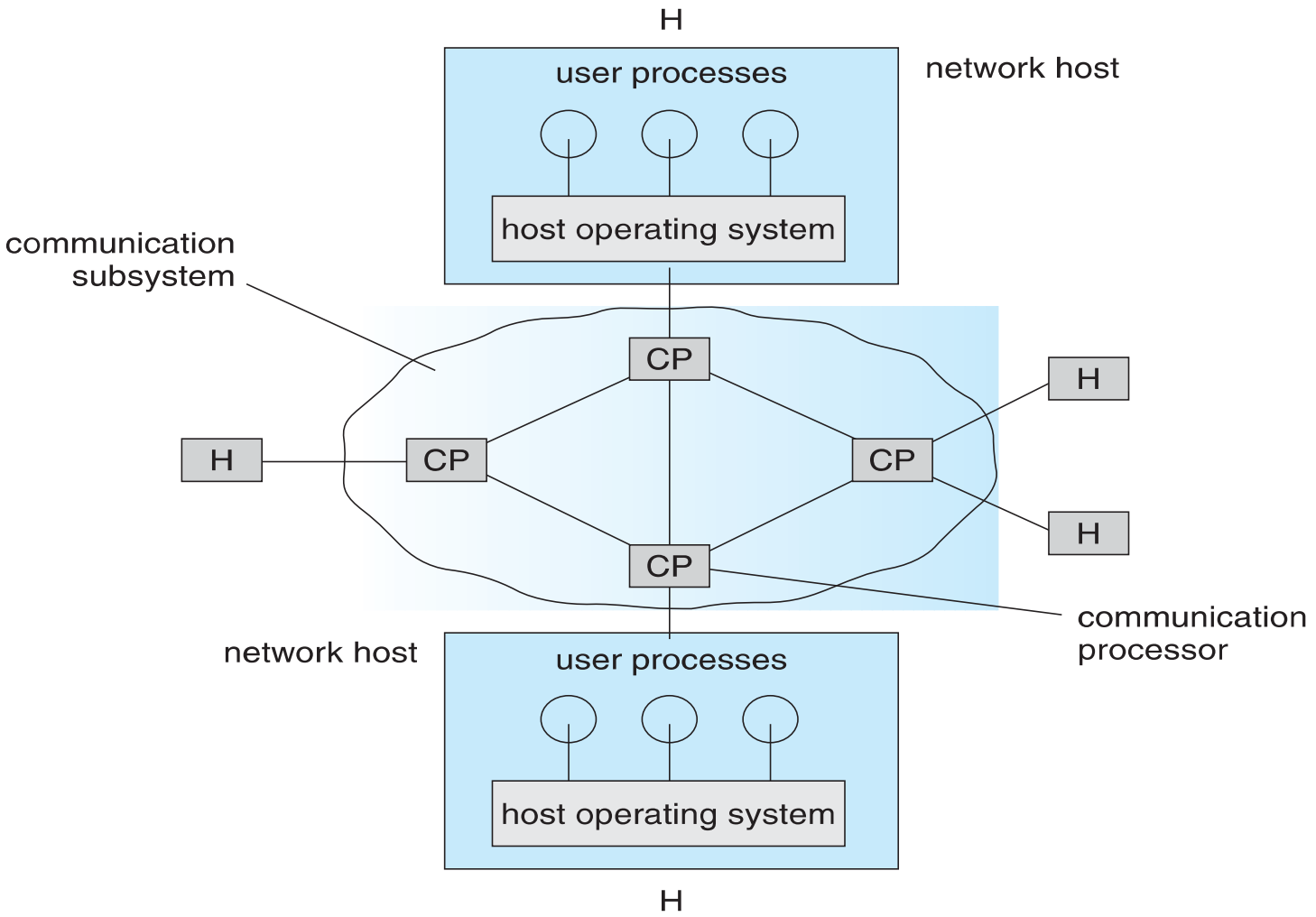
Types of Distributed OS

- Network Operating Systems
  - Users are aware of multiplicity of machines (more difficult)
- Distributed Operating Systems
  - Users are not aware of multiplicity of machines

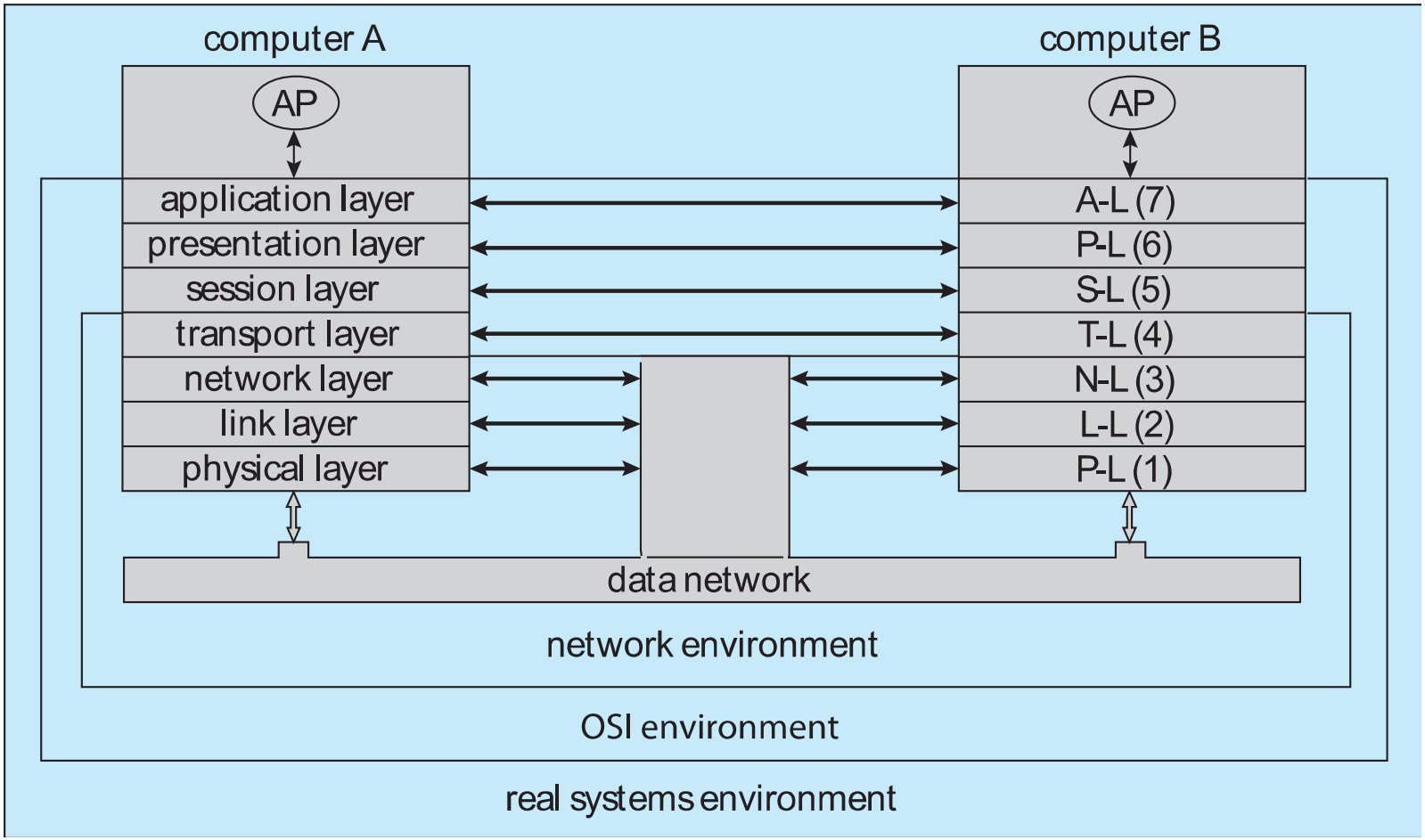
Distributed Operating Systems

- Data Migration
- Computation Migration
- Process Migration
  - Load balancing
  - Computation speedup
  - HW/SW preferences
  - Remote Data Access

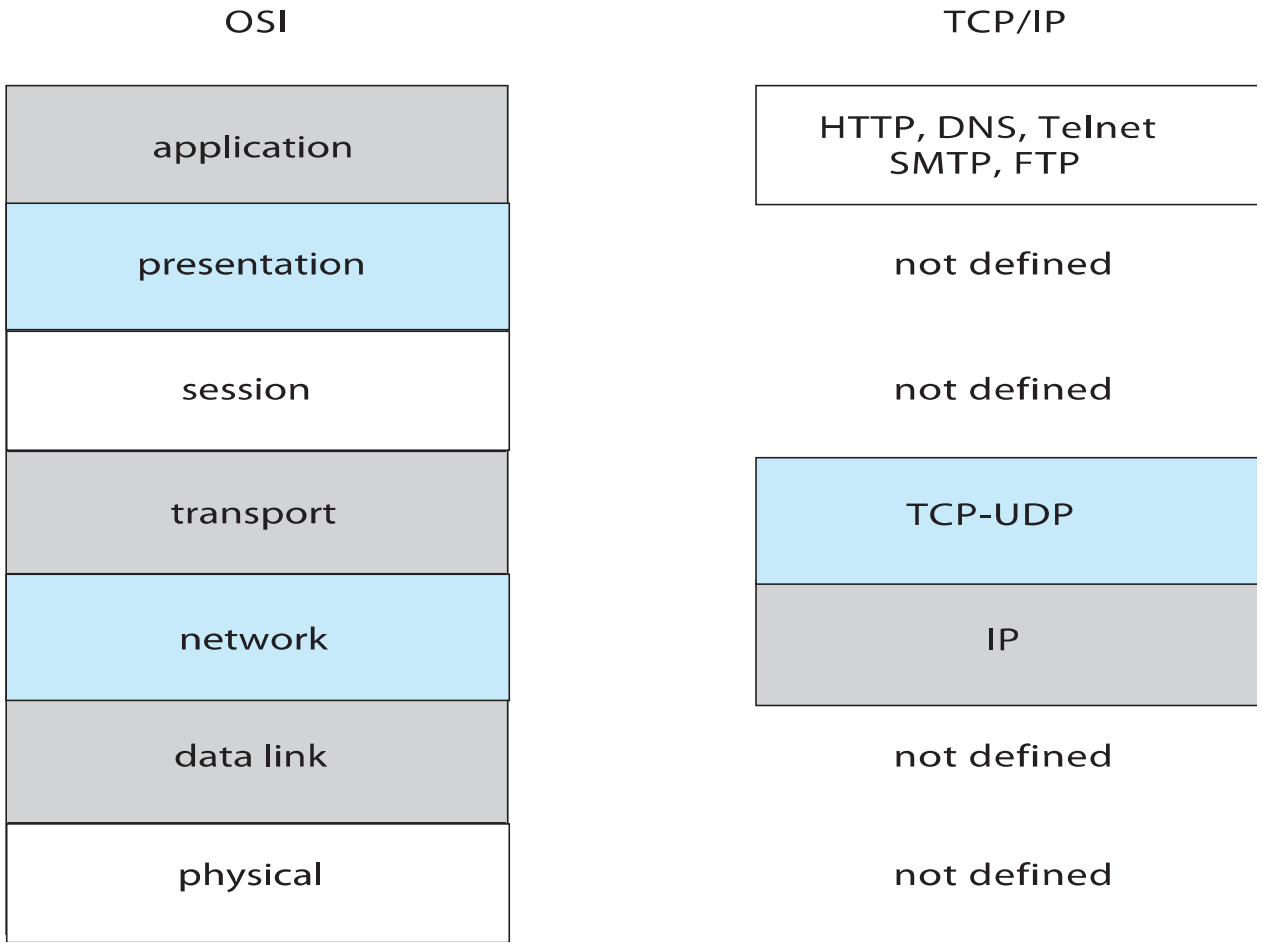
Wide Area Network



ISO Network Model



TCP/IP vs OSI ISO model



Robustness

- Failure Detection
  - Heatbeat protocol: I am UP/Are you UP?
  - Failure Types: host, link, loss
- Reconfiguration
  - Notify host/link/loss failure
  - Reconfigure to avoid host/link/loss

The Two Generals' Problem

- How can G1 synchronize with G2?
- How can G1 be sure that G2 has received the message?
- How can G2 be sure that G1 knows that G2 has received the message?

Design Issue

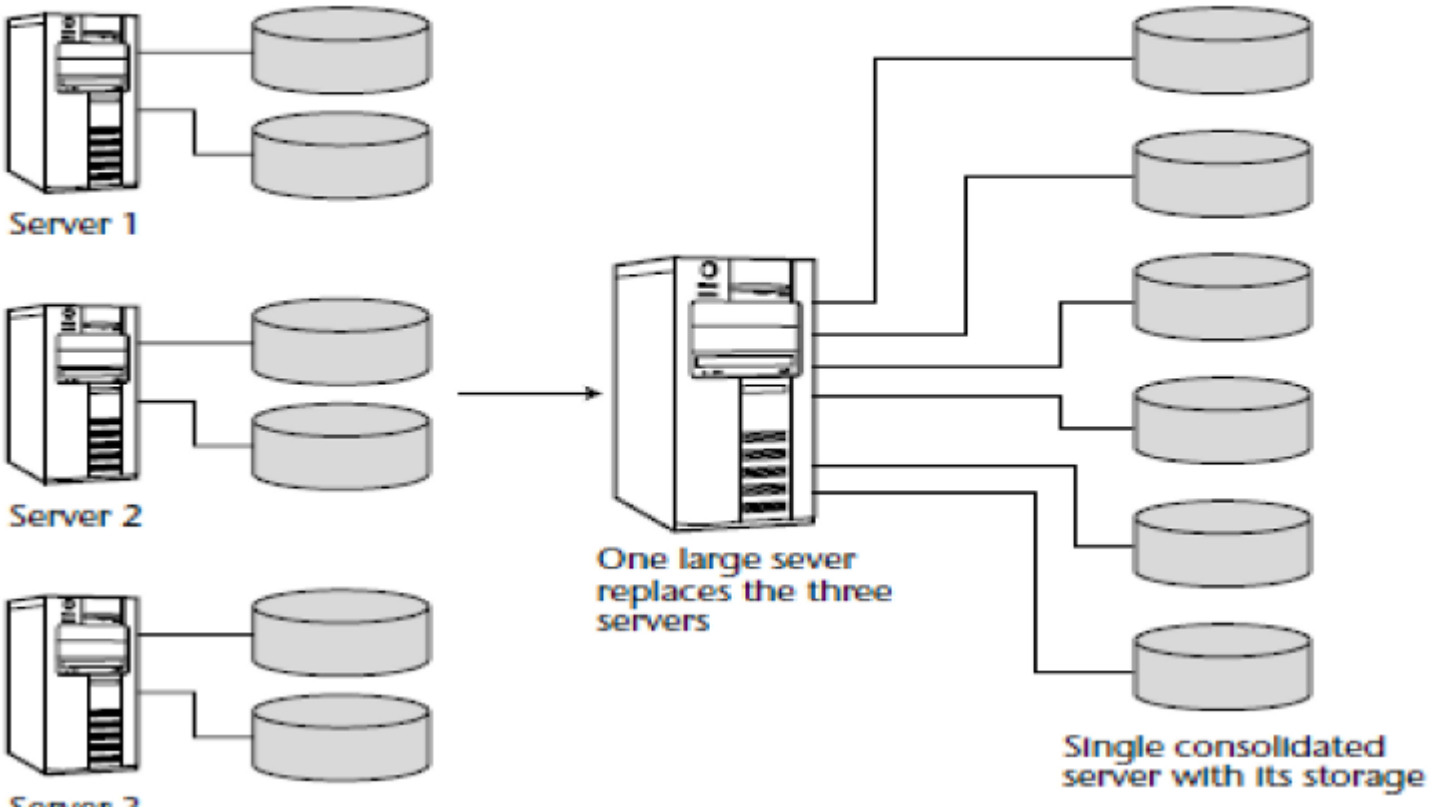
- **Transparency** – the distributed system should appear as a conventional, centralized system to the user
- **Fault tolerance** – the distributed system should continue to function in the face of failure
- **Scalability** – as demands increase, the system should easily accept the addition of new resources to accommodate the increased demand
  - Consider Hadoop open source programming framework for processing large datasets in distributed environments (based on Google search indexing)
- **Clusters** – a collection of semi-autonomous machines that acts as a single system

Consistency

- Is locally cached copy of the data consistent with the master copy?
- **Client-initiated approach**
  - Client initiates a validity check
  - Server checks whether the local data are consistent with the master copy
- **Server-initiated approach**
  - Server records, for each client, the (parts of) files it caches
  - When server detects a potential inconsistency, it must react

Data Center Consolidation (1)

Storage consolidation by consolidating servers



Data Center Consolidation (2)

Various servers linked to a single storage subsystem

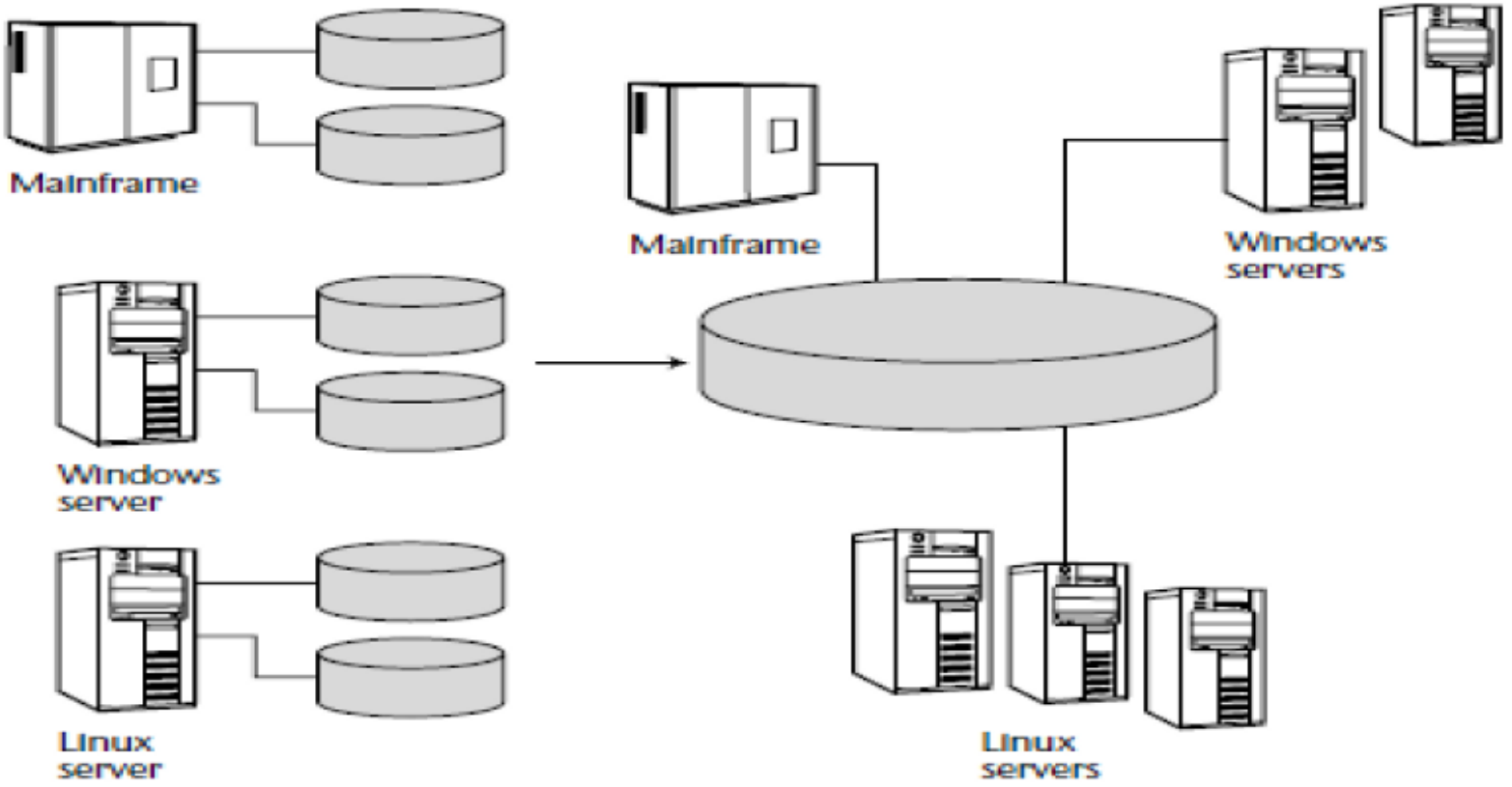


Figure 9-2 Consolidation to a direct-attached single storage pool.

Data Center Consolidation (3)

Consolidation with storage area network (SAN)

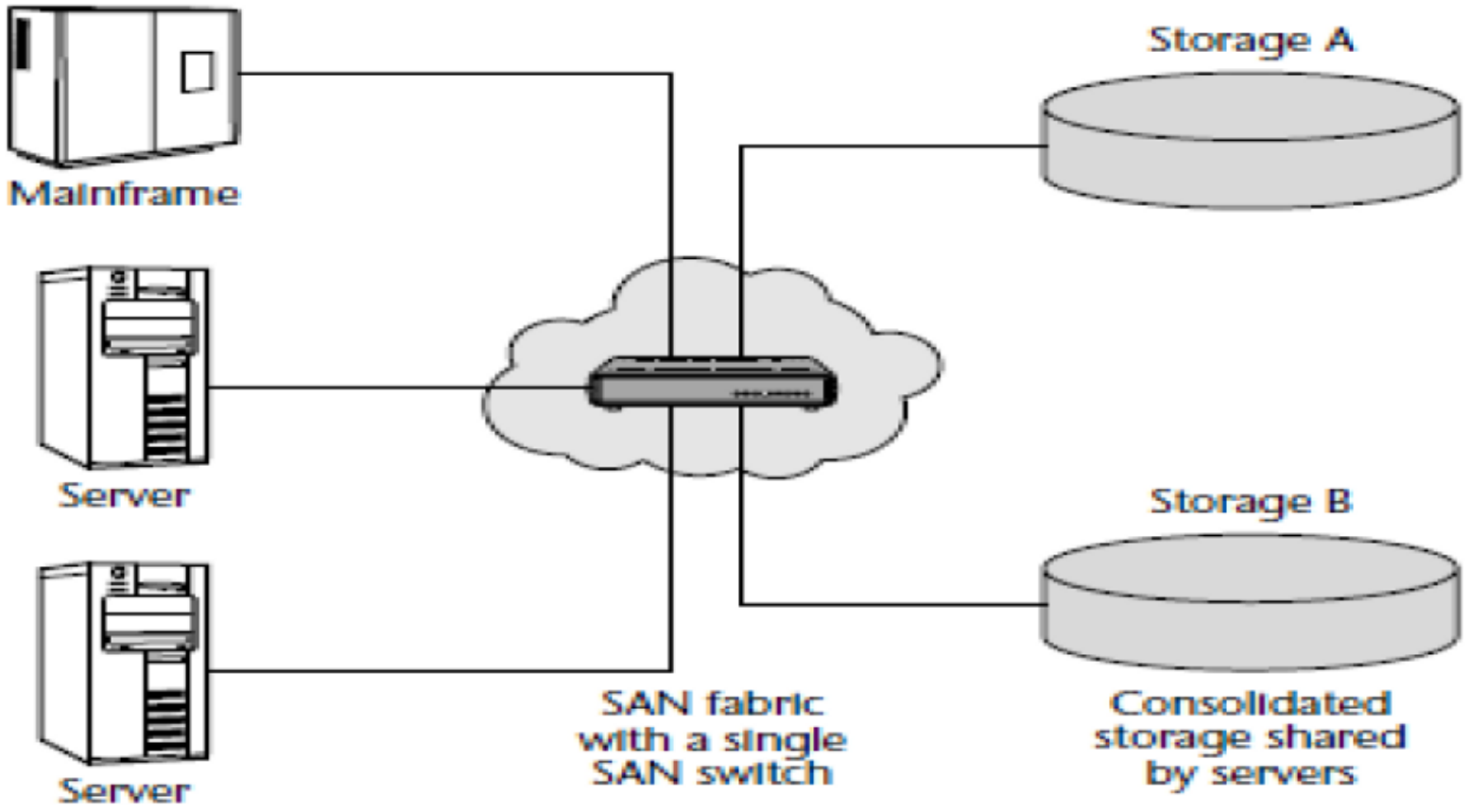


Figure 9-3 Consolidation of storage to a SAN-attached single pool.

Data Center Consolidation (4)

Consolidation with network attached storage (NAS)

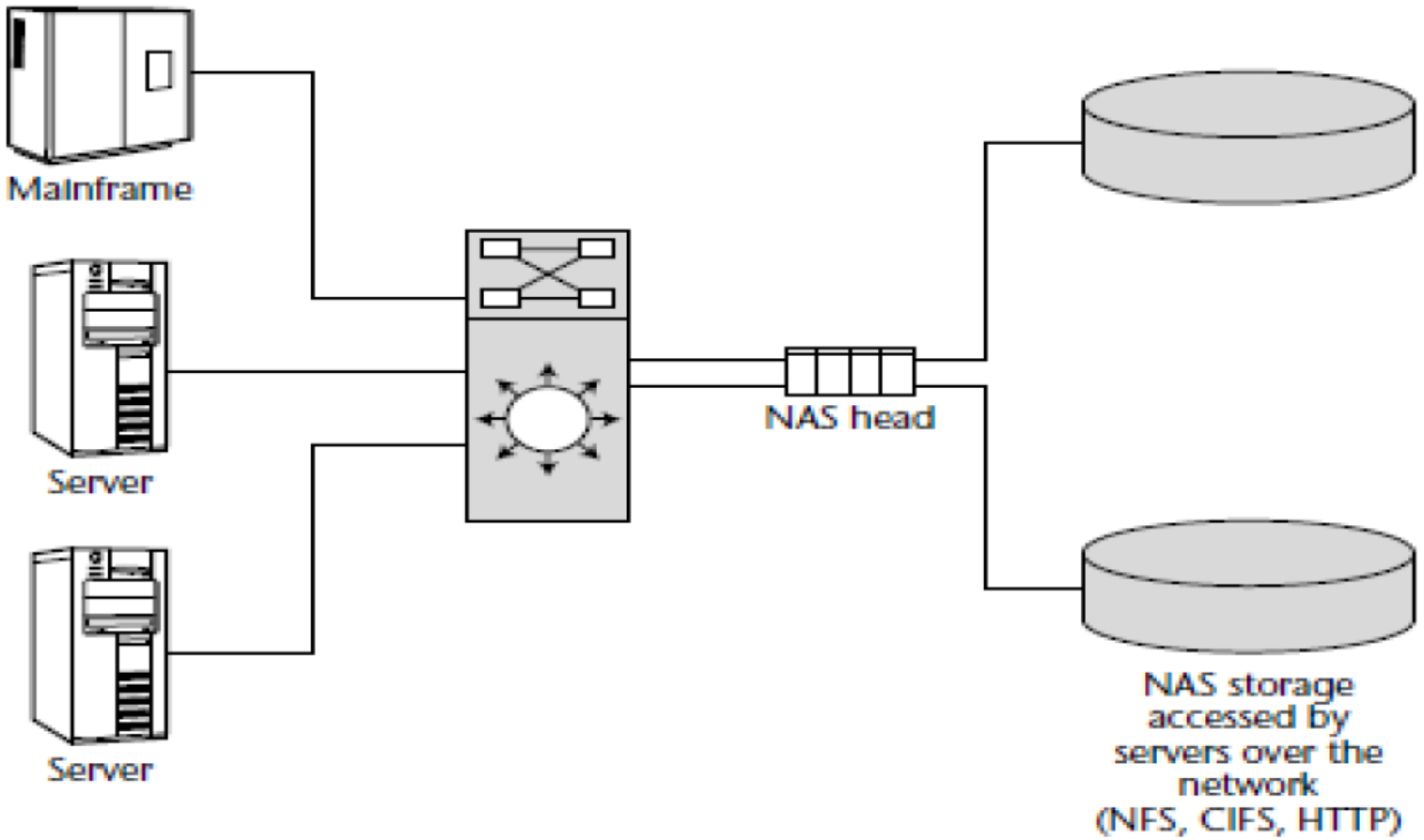
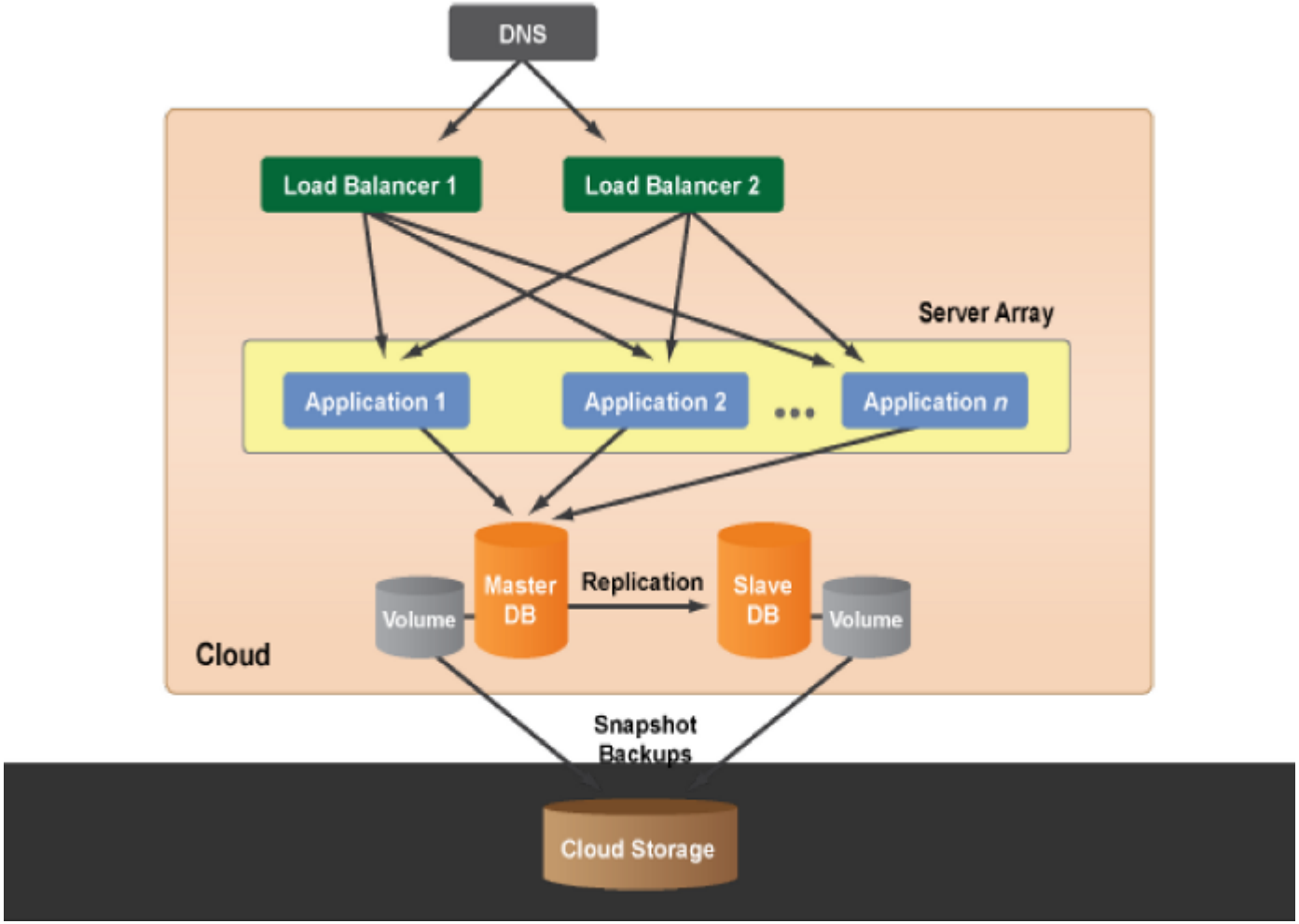


Figure 9-4 Consolidated storage accessed by server using network protocols such as NFS, CIFS, and HTTP.

Data Center Consolidation (5)



End of Topic #2
Distributed Systems

Topic #3
Real Time Systems

- Topic #1: Free/Open Source Software
- Topic #2: Distributed Systems
- **Topic #3: Real Time Systems**
- Topic #4: Multimedia Systems

Sub-Agenda

- Introduction
- Characteristics
- Implementation
- Schedulling

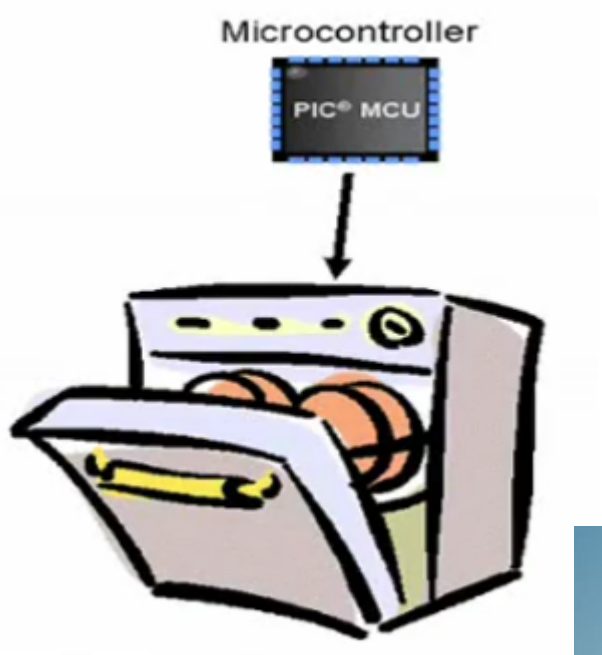
Real Time Systems (RTS)

- A **real-time system** requires that results be produced within a specified deadline period.
- An **embedded system** is a computing device that is part of a larger system (i.e., automobile, airliner).
- A **safety-critical system** is a real-time system with catastrophic results in case of failure.
- A **hard real-time system** guarantees that real-time tasks be completed within their required deadlines.
- A **soft real-time system** provides priority of real-time tasks over non real-time tasks.

RTS Characteristics

- Single purpose & Small size
- Inexpensively mass-produced
- Specific timing requirements
- Designed using system-on-a-chip (SOC) strategy
- SOC allows the CPU, memory, memory-management unit, and attached peripheral ports (i.e., USB) to be contained in a single integrated circuit
- Not all features found in a standard system

RTS Examples

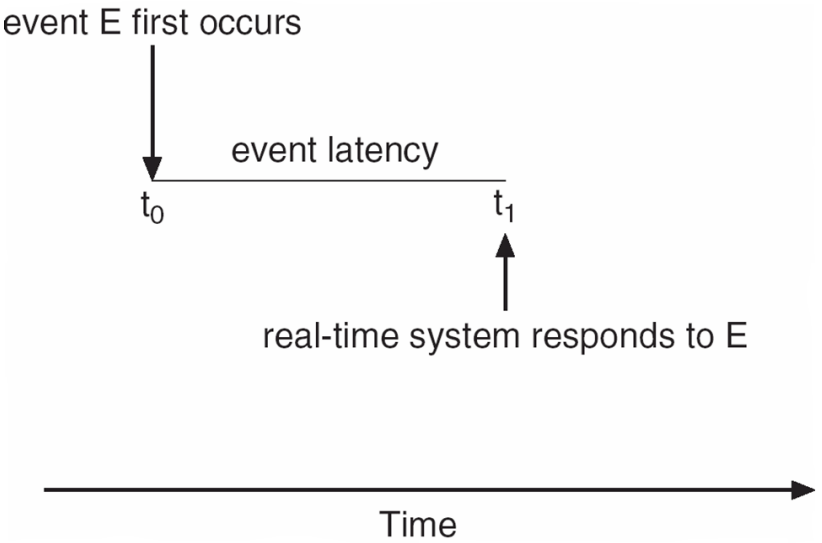


RTS Processes

- Tasks/Processes
  - Have priority
  - Have deadlines
  - Higher priority tasks PREEMPT lower ones (Priority Based Preemption)

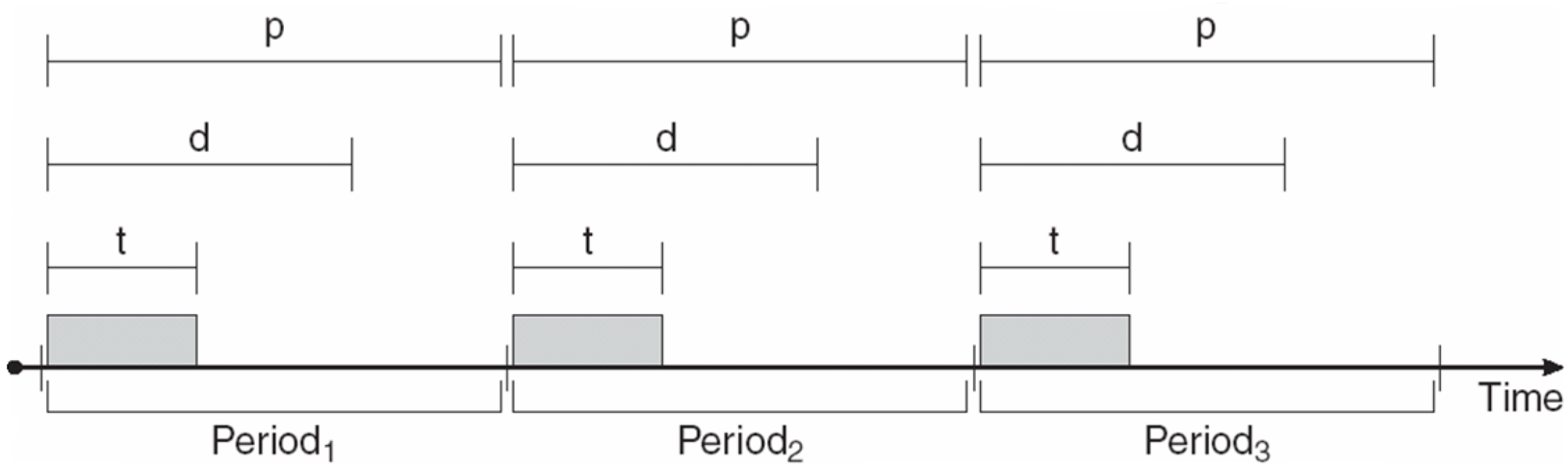
Implementing RTS

- RTS must provide:
  - **Preemptive**, **priority**-based scheduling
  - Preemptive **kernels**
  - **Latency** must be minimized



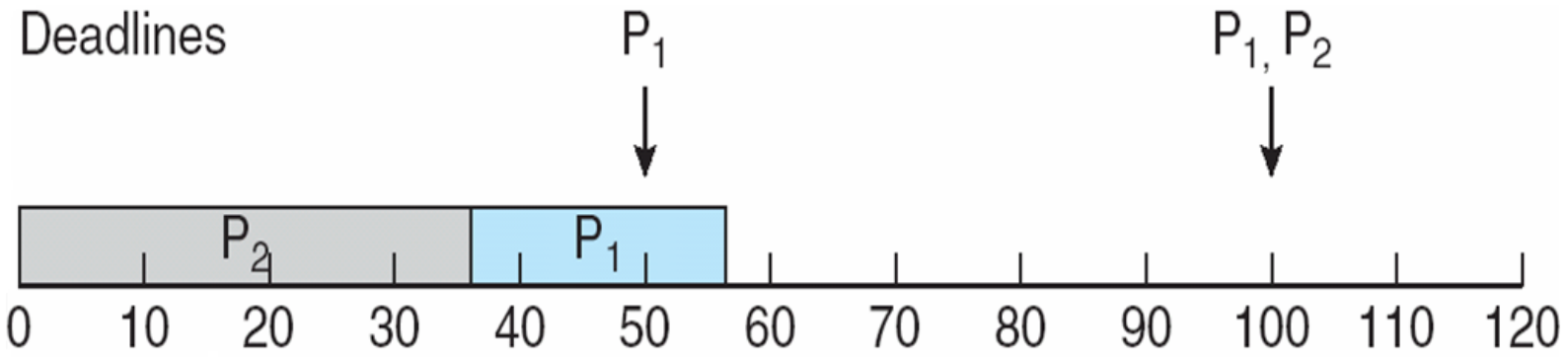
RTS CPU Schedulling

- Periodic processes require the CPU at specified intervals (periods)
- ***p*** is the duration of the period
- ***d*** is the deadline by when the process must be serviced
- ***t*** is the processing time

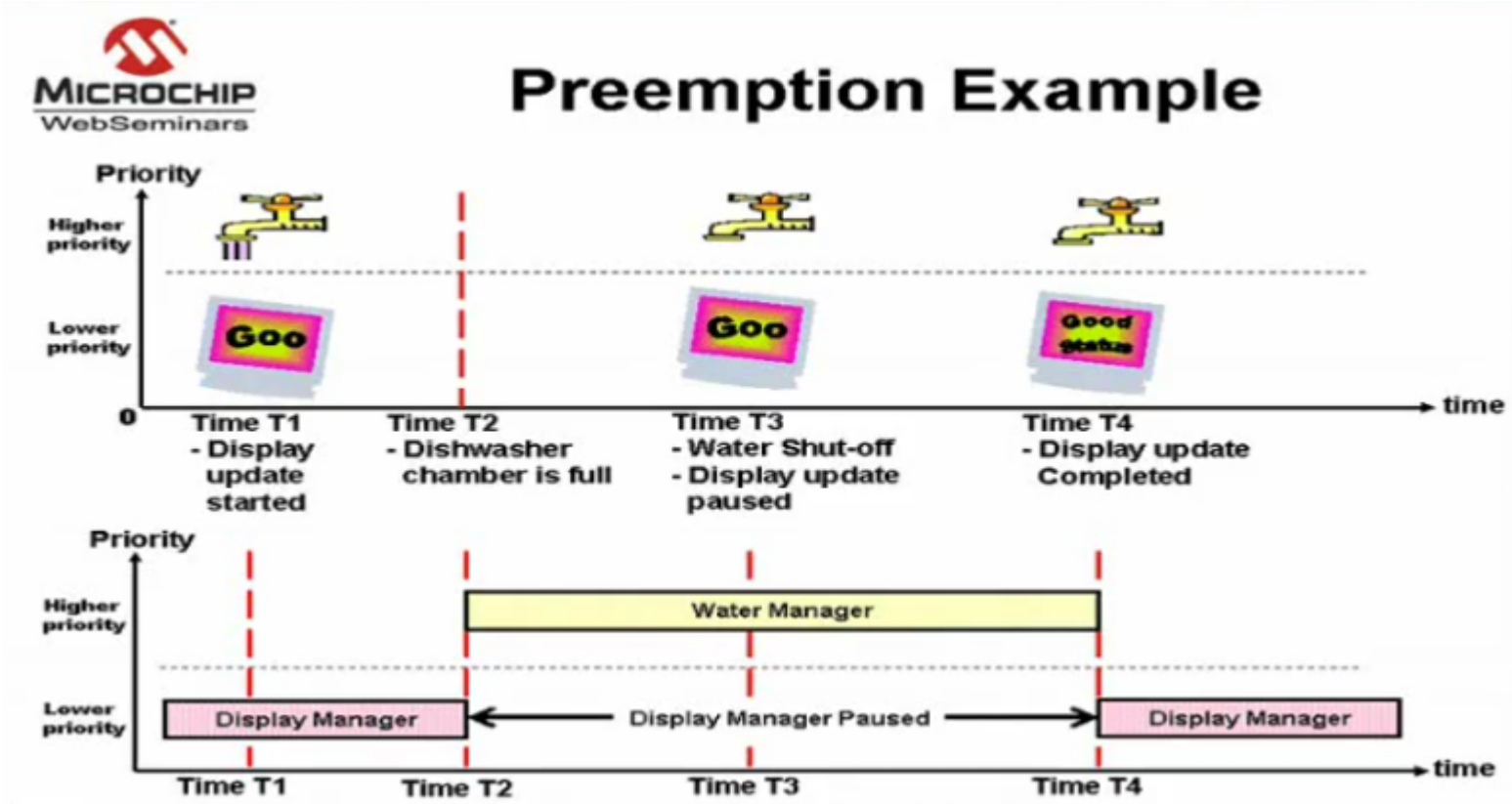


Priority Schedulling

- P2 has a higher priority than P1



Preemption Example



End of Topic #3
Real Time Systems

Topic #4
Multi Media Systems

- Topic #1: Free/Open Source Software
- Topic #2: Distributed Systems
- Topic #3: Real Time Systems
- **Topic #4: Multimedia Systems**

Sub-Agenda

- Introduction
- Delivery
- Characteristics
- Streaming
- Compression
- QoS
- Delivery Methods

Introduction

- Multimedia data includes
  - audio and video clips (i.e., MP3 and MPEG files)
  - live webcasts
- Multimedia data may be delivered to
  - desktop PC's
  - handheld devices

Media Delivery

- Data must be accessed with specific timing requirements. Eg.
  - video must be displayed at 24-30 frames per second. Multimedia video data must be delivered at a rate which guarantees 24-30 frames/second
  - Continuous-media data is data with specific rate requirements

Characteristics

- Multimedia files can be quite large
- Continuous media data may require very high data rates
- Multimedia applications may be sensitive to timing delays during playback of the media

Streaming

- Streaming is delivering a multimedia file from a server to a client - typically the deliver occurs over a network connection.
- There are two different types of streaming:
  - **Progressive download** - the client begins playback of the multimedia file as it is delivered. The file is ultimately stored on the client computer.
  - **Real-time streaming** - the multimedia file is delivered to - but not stored on - the client's computer.

Real Time Streaming

- There are two types of real-time streaming:
  - **Live streaming** - used to deliver a live event while it is occurring
  - **On-demand streaming** - used to deliver media streams such as movies, archived lectures, etc. The events are not delivered in real-time.

Compression

- Because of the size and rate requirements of multimedia systems, multimedia files are often compressed into a smaller form
- MPEG Compression:
  - MPEG-1 - 352 X 240 @ 30 frames/second
  - MPEG-2 - Used for compressing DVD and high-definition television (HDTV)
  - MPEG-4 - Used to transmit audio, video, and graphics. Can be delivered over slow connections (from 56 Kbps)

QoS

- Guaranteeing QoS has the following effects in a computer system:
  - CPU processing
  - Scheduling
  - File systems
  - Network protocols
- Levels:
  - **Best-effort service**
  - **Soft QoS**
  - **Hard QoS**

Delivery Methods

- Three general methods for delivering content from a server to a client across a network:
  - **Unicasting** - the server delivers the content to a single client.
  - **Broadcasting** - the server delivers the content to all clients, regardless whether they want the content or not.
  - **Multicasting** - the server delivers the content to a group of receivers who indicate they wish to receive the content

End of Topic #4
Multimedia Systems

EOF
Thank You!