Distributed Systems
Introduction

[2] **Lectures (1)**

Tomasz Jordan Kruk (T.Kruk@ia.pw.edu.pl)

Office hours: room 530, Tuesday 18.15-19.00

Slides available *after* lectures on http://studia.elka.pw.edu.pl

Books:

1. *Distributed Systems. Principles and Paradigms.* Andrew S. Tanenbaum, Maarten van Steen. Prentice Hall 2002. (available in Polish)
2. *Distributed Systems. Concepts and Design. fourth edition.* G. Colouris, J. Dollimore, T. Kindberg. Addison Wesley 2005. (available in Polish)
3. *Reliable Distributed Systems. Technologies, Web Services, and Applications.* Kenneth P. Birman. Springer 2005.

[3] **Lectures (2)**

1. Introduction
2. Communication (I)
3. Communication (II)
4. Synchronization (I)
5. Synchronization (II)
6. Consistency and replication
7. Fault tolerance
8. File systems
9. Naming
10. Peer-to-peer systems
11. Web services

12. Security

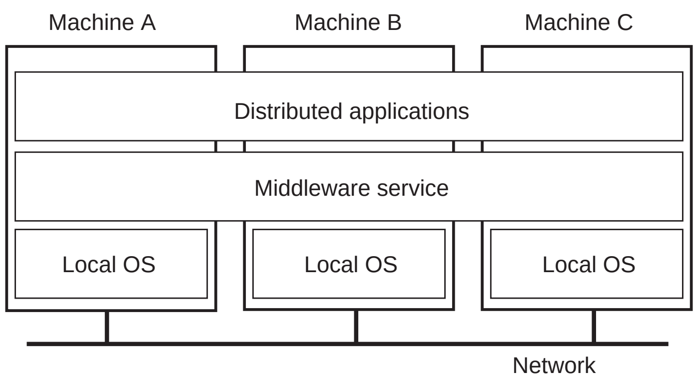
[4] **Definition of a Distributed System (1)**

Distributed system
Collection of independent computers that appears to its users as a single coherent system.

Goals:

- connecting users and resources,
- transparency,
- openness = to offer services according to standard rules that describe the syntax and semantics of those services (e.g. POSIX for OS),
- scalability.

[5] **Definition of a Distributed System (2)**



A distributed system organized as middleware. Note that the middleware layer extends over multiple machines.

[6] **Transparency in a Distributed System**

Different forms of **transparency** in a distributed system.

Transparency	Description
access	hide differencies 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 is replicated
concurrency	hide that a resource may be shared by several competitive users
failure	hide the failure and recovery of a resource
persitence	hide whether a resource is in memory or on disk

[7] **Degree of Transparency**

- some attempts to blindly hide all distribution aspects not always a good idea,
- a trade-off between a high degree of transparency and the performance.

The goal not to be achieved: **parallelism transparency**.

Parallelism transparency
Transparency level with which a distributed system is supposed to appear to the users as a traditional uniprocessor timesharing system.

[8] **Openness**

Completeness and neutrality of specifications as important factors for interoperability and portability of distributed solutions.

completeness all necessary to make an implementation as it has been specified,

neutrality specification do not prescribe what an implementation should look like.

Interoperability
The extent by which two implementations of systems from different manufactures can cooperate.

Portability
To what extent an application developed for A can be executed without modification on some B which implements the same interfaces as A.

[9] **Scalability Problems**

Three different dimensions of the system scalability:

- scalable with respect to its size,
- geographically scalable systems (users and resources may lie apart),
- system administratively scalable.

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

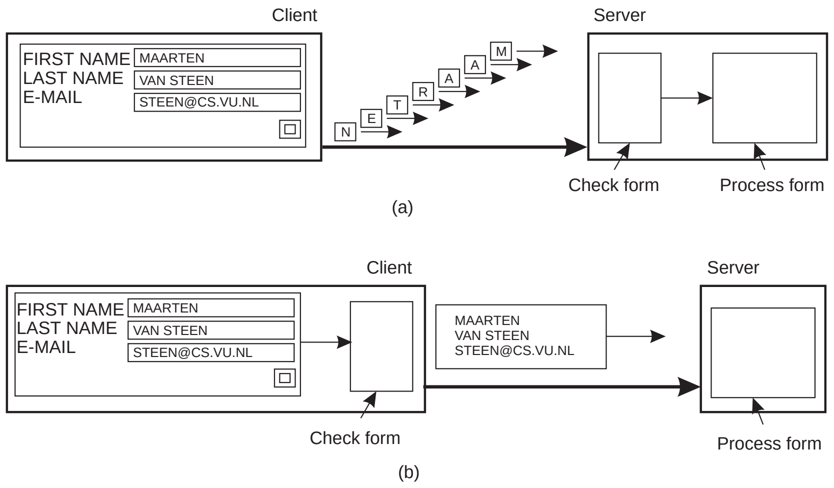
[10] **Decentralized Algorithms**

1. No machine has complete information about the system state.
2. Machines make decisions based only on local information.
3. Failure of one machine does not ruin the algorithm.
4. There is no implicit assumption that a global clock exists.

[11] **Scaling Techniques (1)**

- asynchronous communication (to hide communication latencies),
- distribution (splitting into smaller parts and spreading),
- replication (to increase availability and to balance the load),
- caching (as a special form of replication).

[12] **Scaling Techniques (2)**



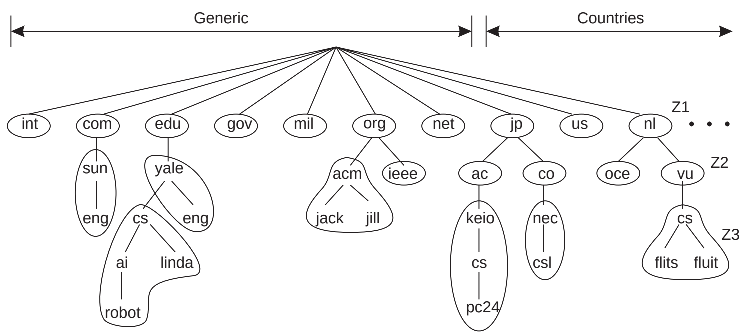
A difference between letting:

a. a server or

b. a client

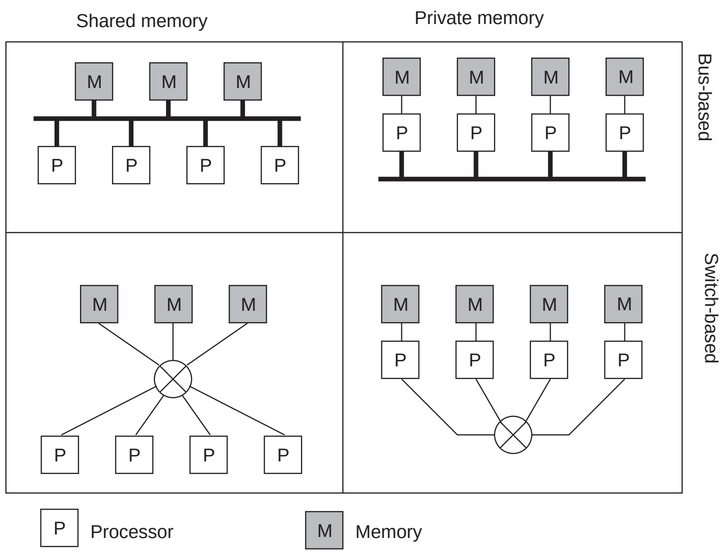
check forms as they are being filled.

[13] **Scaling Techniques (3)**



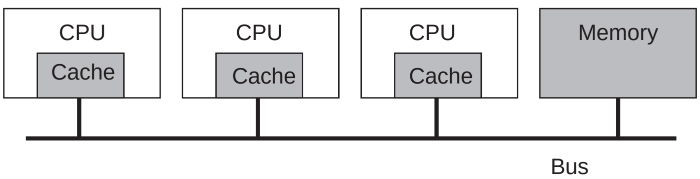
An example of dividing the DNS name space into zones.

[14] **Hardware Concepts**



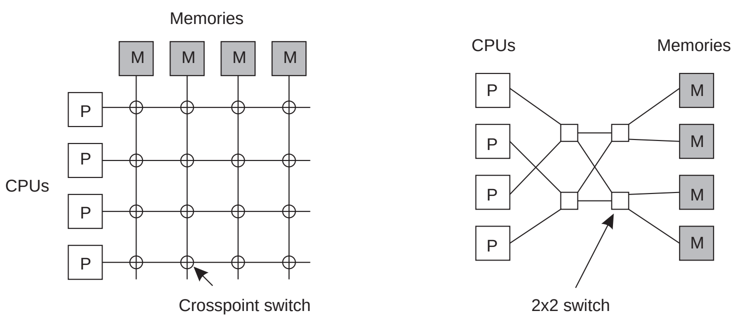
Different basic organizations and memories in distributed computer systems.

[15] **Multiprocessors (1)**



A bus-based multiprocessor.

[16] **Multiprocessors (2)**



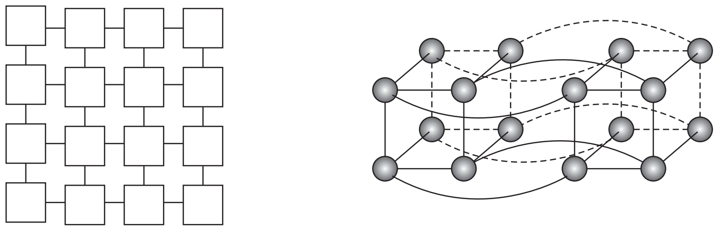
(a) (b)

a. a crossbar switch,

b. an omega switching network (2^k inputs and a like outputs; $\log_2 N$ stages, each having $N/2$ exchange elements at each stage),

NUMA - NonUniform Memory Access - hierarchical systems.

[17] **Homogeneous Multicomputer Systems**



(a) (b)

a. grid,

b. hypercube.

Examples: Massively Parallel Processors (MPPs), Clusters of Workstations (COWs).

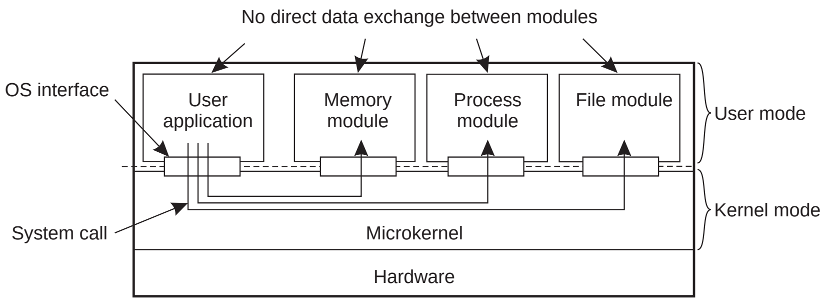
[18] **Software Concepts**

System	Description	Main goal
DOS	Tightly-coupled operating system for homogeneous multicomputers	Hide and manage 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

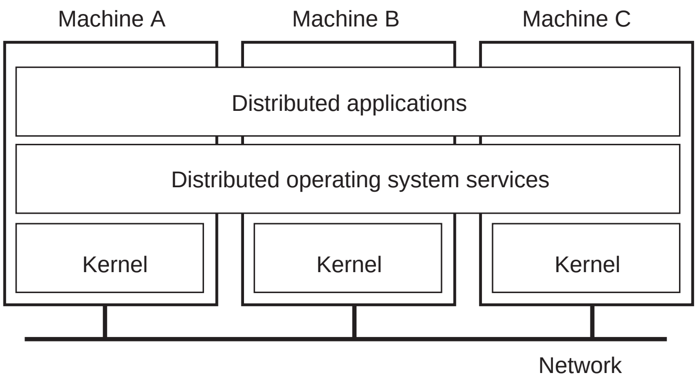
DOS distributed operating system.

NOS network operating system.

[19] **Uniprocessor Operating Systems**

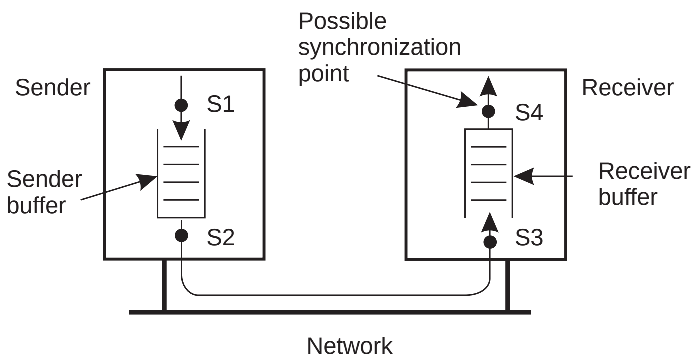


[20] **Multicomputer Operating Systems (1)**



General structure of a multicomputer operating system.

[21] **Multicomputer Operating Systems (2)**



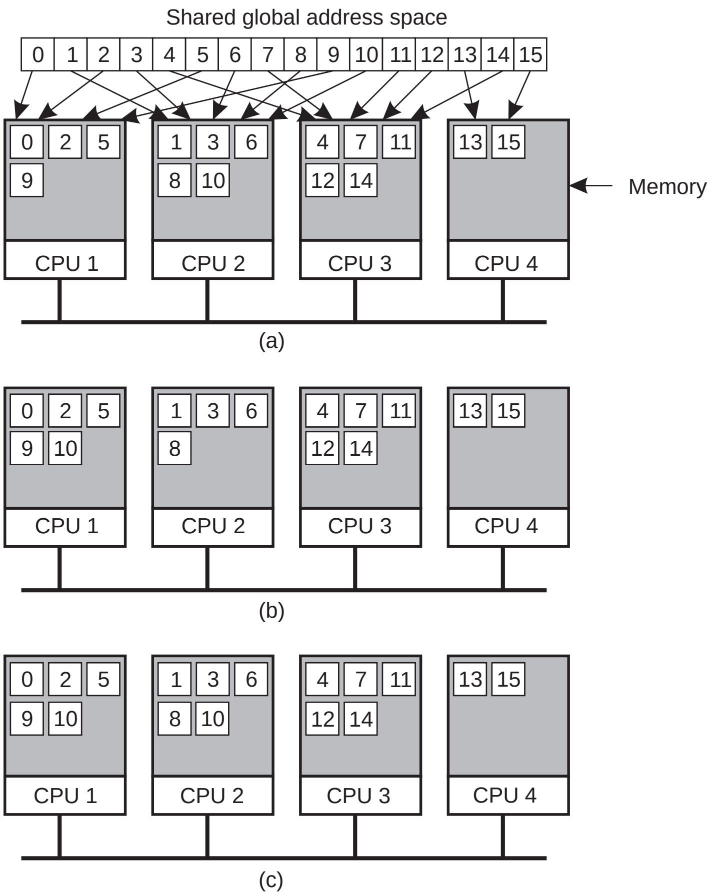
Alternatives for blocking and buffering in message passing.

[22] **Multicomputer Operating Systems (3)**

Synchronization point	**Sent buffer**	**Reliable communication**
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

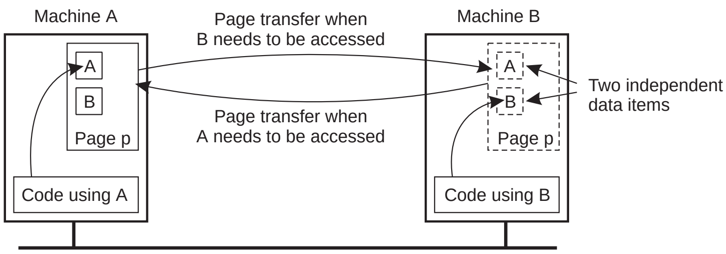
Relation between blocking, buffering, and reliable communications.

[23] **Distributed Shared Memory Systems (1)**



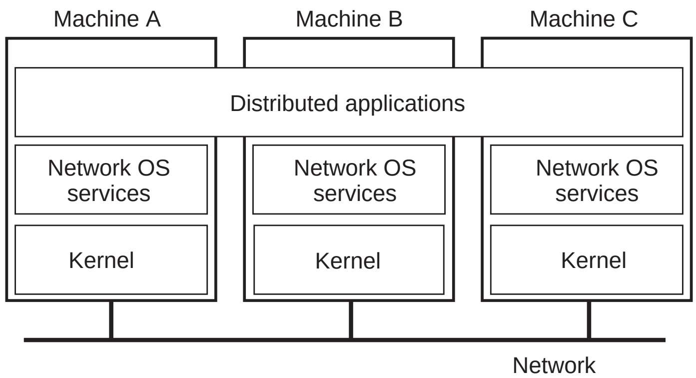
- Pages of address space distributed among four machines,
- Situation after CPU 1 references page 10,
- Situation if page 10 is read only and replication is used.

[24] **Distributed Shared Memory Systems (2)**



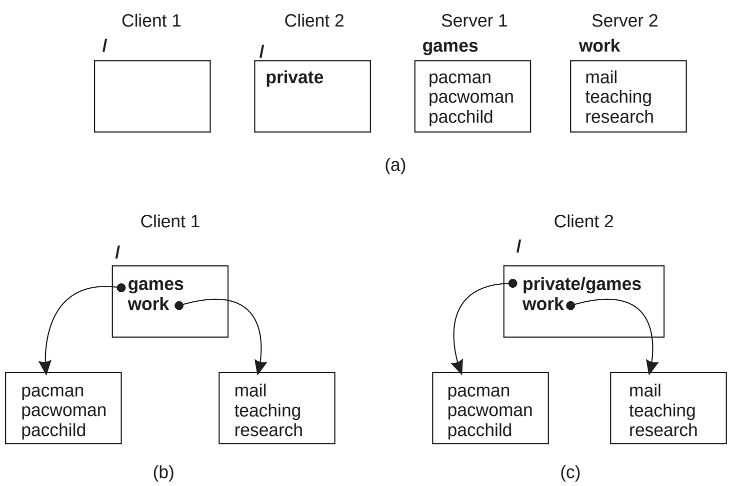
False sharing of a page between two independent processes.

[25] **Network Operating Systems (1)**



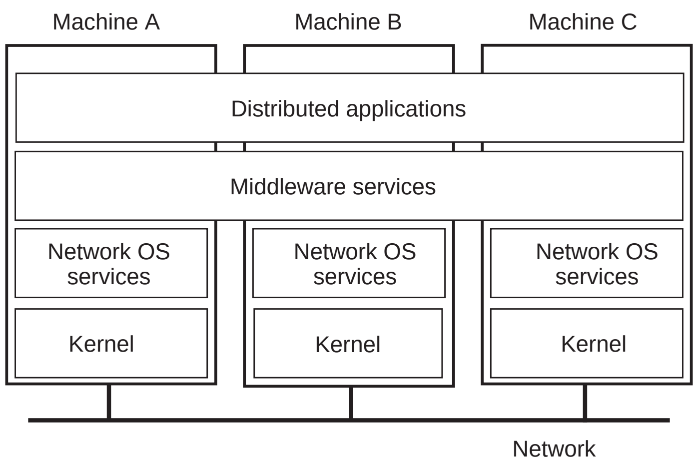
General structure of a network operating system.

[26] **Network Operating Systems (2)**



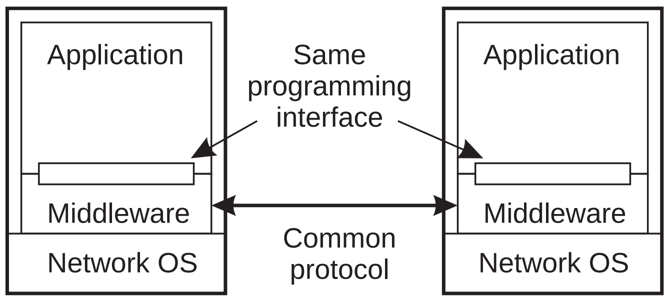
Different clients may mount the servers in different places.

[27] **Positioning Middleware**



General structure of a distributed system as middleware.

[28] **Middleware and Openness**



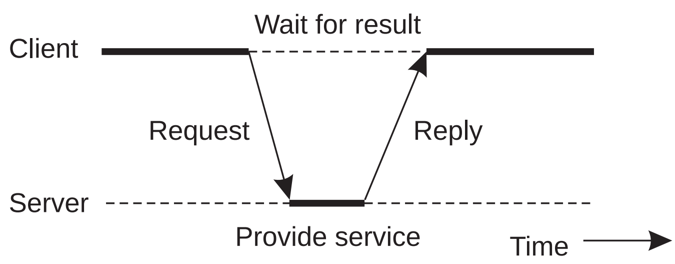
In an open middleware-based distributed system, the protocols used by each middleware layer should be the same, as well as the interfaces they offer to applications.

[29] **Comparison of Operating Systems Types**

Item	**DOS**	**NOS**	**middleware**
Degree of transparency	high	low	high
Same OS on all nodes?	yes	no	no
Basis for communication	messages	files	model specific
Resource management	global, distributed	per node	per node
Scalability	moderately	yes	varies
Openness	closed	open	open

A comparison between distributed OS, network OS, and middleware-based distributed systems.

[30] **Clients and servers**

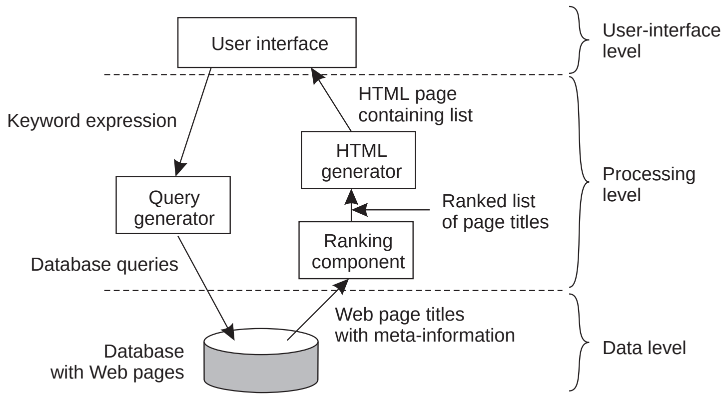


General interaction between a client and a server.

[31] **Application Layering**

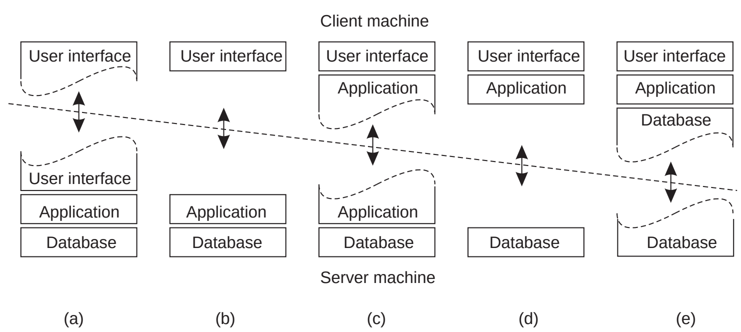
- The user-interface layer,
- The processing level,
- The data level.

[32] **Processing Level**



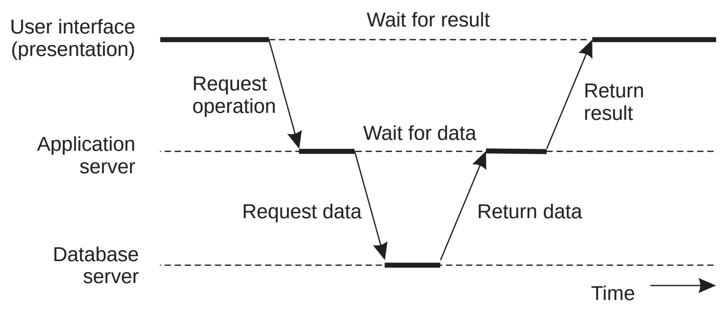
The general organization of an Internet search engine into three different layers.

[33] **Multitiered Architectures (1)**



Alternative client-server organizations.

[34] **Multitiered Architectures (2)**



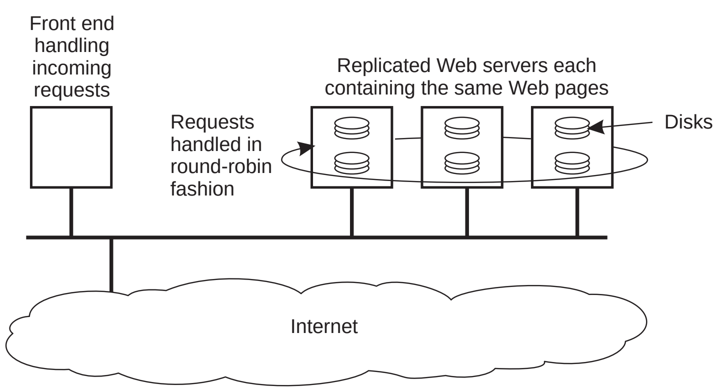
An example of a server acting as a client.

[35] **Modern Architectures (1)**

Vertical distribution
Achieved by placing logically different components on different machines.

Horizontal distribution
Client or server may be physically split up into logically equivalent parts, but each part is operating on its own share of the complete data set, thus balancing the load.

[36] **Modern Architectures (2)**



An example of horizontal distribution of a Web service.

[37] **Internet and Web Servers**

Date	**Computers**	**Web servers**
1979, Dec.	188	0
1989, July	130 000	0
1999, July	56 218 000	5 560 866
2003, Jan.	171 638 297	35 424 956
2007, Feb.	n.a.	108 810 358

active	deleted	TLD
.com	93,223,583	321,484,351
.net	13,796,853	34,991,342
.org	9,415,497	21,444,504
.info	7,882,564	10,915,879
.biz	2,227,815	2,358,538
.us	1,860,456	1,907,385
total	128,538,602	393,763,577

Number of worldwide registered domains (DomainTools, 2011, Feb.).