Categories of computer systems

Primary issue: data accessing and generation

Types:

• Conventional sequential machines (mainframes, minicomputers + network of terminals) - multitasking, multiusers.

• Conventional systems with special purpose components (specialized processors) - single specialized task.

• Multiprocessor systems - single task allowing parallel computation.

• Distributed systems (computers connected by a network) - different task, shared data.
Conventional sequential machines
Conventional systems with special purpose components

(Master and slave architecture)

• A special purpose unit (e.g. math processor) attached to the main bus
• Back-end system (additional separate machine, e.g. graphic terminal)
• Example: iDBP
  • file operations: positioning and manipulating a cursor in a file
  • used to implement relational database systems
  • add-on board or back-end system
Multiprocessor systems

Multiprocessor systems:
- Multiple processors
- Shared memory (single address space) vs. multiple private memories
- Centralized memory vs. distributed memory

Categories of parallelism:
- Single instruction stream, single data stream (SISD)
- Single instruction stream, multiple data streams (SIMD)
- Multiple instruction streams, single data stream (MISD)
- Multiple instruction streams, multiple data streams (MIMD)
Distributed computer systems

Issues:
- data location and security
- load distribution
- process migration
- fault tolerance

Types:
- homogeneous systems
- heterogeneous systems

Distributed file systems:
- Apollo DOMAIN (efficient data sharing)
- NFS (UNIX, DOS, system independence)
- AFS (scalability, cache at the file level)
- CFS (mobile systems)

Fault-tolerant networks:
- redundancy (static, dynamic)
- consistency (strong, weak)
- self-stabilizing networks
Definition of a Distributed System (1)

- A distributed system is:

  A collection of independent computers that appears to its users as a single coherent system.
Definition of a Distributed System (2)

A distributed system organized as middleware. Note that the middleware layer extends over multiple machines.
# Transparency in a Distributed System

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

Different forms of transparency in a distributed system.
The difference between letting:

a) a server or

b) a client check forms as they are being filled
Scaling Techniques (2)

An example of dividing the DNS name space into zones.
Hardware Concepts

Different basic organizations and memories in distributed computer systems
Multiprocessors (1)

- A bus-based multiprocessor.

![Diagram of a bus-based multiprocessor](image-url)
Multiprocessors (2)

(a) Crosspoint switch

(b) 2x2 switch
Homogeneous Multicomputer Systems

a) Grid
b) Hypercube