Software & Hardware Concept

Operating System

CACS 251

PRESENTORS

Jiban Niraula | Samir Adhikari | Kiran Tamang
Grishma Shrestha | Pragya Ojha
BCA 4th Sem

Table of contents

01 Hardware Concept

02 Software Concept

Hardware Concept 01

In a distributed operating system, hardware consists of multiple interconnected nodes, each with its own CPU, memory, and storage, communicating via networks. Tasks are distributed across nodes, enhancing performance and reliability.

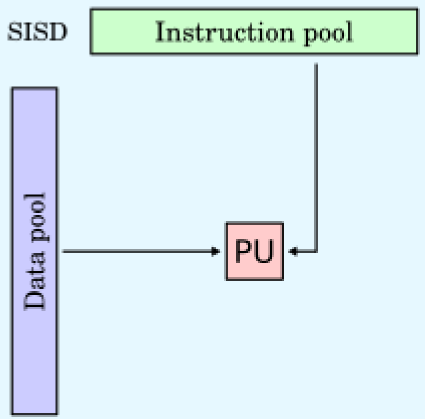
Multiple-CPU systems are classified using Flynn's taxonomy:

- SISD (single CPU)
- SIMD (parallel data processing)
- MISD (fault-tolerant systems)
- MIMD (independent tasks on multiple CPUs)

Single Instruction, Single Data (SISD)

A single processor executes a single instruction stream on a single data stream. Used in basic personal computers with one core.

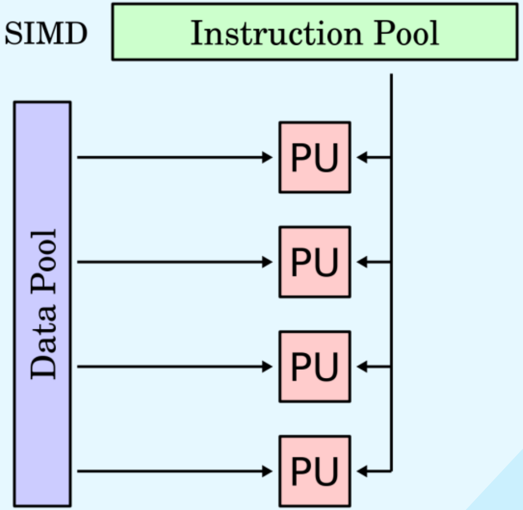
Example: Traditional uniprocessor systems.



Single Instruction, Multiple Data (SIMD)

Multiple processing units execute the same instruction on different data streams simultaneously. Used in Parallel data processing in graphics rendering and scientific simulations.

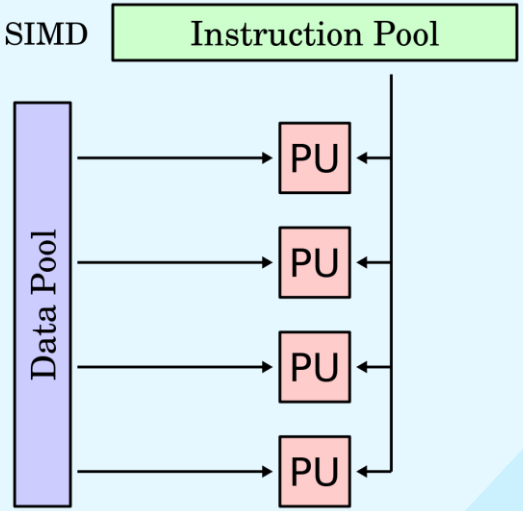
Example: Vector processors, GPUs (Graphics Processing Units).



Multiple Instruction, Single Data (MISD)

Multiple processors execute different instructions on the same data stream. Used in specialized systems requiring high reliability.

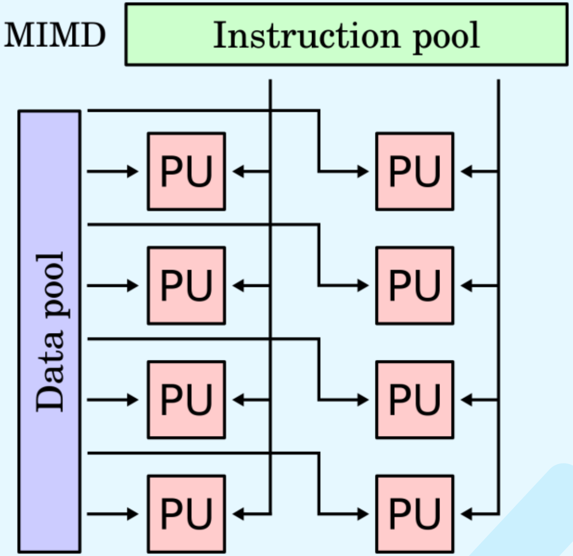
Example: Rarely used in practice; sometimes referenced in fault-tolerant systems (e.g., redundant computations).



Multiple Instruction, Multiple Data (MIMD)

Multiple processors execute different instructions on different data streams simultaneously. Used in modern supercomputers and general-purpose multi-core CPUs.

Example: Multi-core processors, distributed systems, and cloud servers.



Software Concept 02

In a distributed operating system, software manages multiple nodes, ensuring coordination, communication, and resource sharing. Key components include the kernel for hardware control and middleware for services like authentication and synchronization. Processes communicate via message passing or RPC, with algorithms ensuring consistency and fault tolerance. The system focuses on transparency, load balancing, and reliability across the network.

Nodes can be:

- Tightly coupled (sharing memory and resources closely)
- Loosely coupled (independent systems communicating via networks).

Tightly Coupled Software:

- Multiple processors or nodes share a common memory and resources.
- Often found in a single physical unit or closely interconnected system.
- Communication is direct and fast, typically through shared memory.
- Efficient for parallel processing tasks.
- Limited scalability.

Loosely Coupled Software:

- Nodes are independent, each with its own memory and resources.
- Communication occurs over a network.
- Allows for greater scalability and flexibility.
- Ideal for distributed systems.
- More complex communication and synchronization.

Important Questions

Thank You

Do you have any questions?