PARALLEL AND DISTRIBUTED COMPUTING

CLOUD COMPUTING AND INTERNET OF THINGS | ITE6300

JOHN PAUL M. MANUEL | ACLC COLLEGE OF TAYTAY

ACLC COLLEGE OF TAYTAY
(A Member of AMA Education System)

VERSION 01-250201-2425

INTRODUCTION TO PARALLEL AND DISTRIBUTED COMPUTING

In today's fast-paced digital world, computing power is crucial for solving complex problems efficiently. Parallel and distributed computing are two key approaches that enhance processing speed and scalability.

PARALLEL COMPUTING

Parallel computing refers to the simultaneous execution of multiple tasks using multiple processors within a single system.

- A problem is divided into smaller tasks.
- These tasks are processed simultaneously on multiple processors.
- All processors work together to solve a common problem.

💡 Example: Weather forecasting simulations use multiple processors to analyze massive data sets in real-time.

DISTRIBUTED COMPUTING

Distributed computing involves multiple computers, often located remotely, that work together to solve a computation problem.

- Each computer performs a portion of the task.
- Communication occurs through message passing.
- Designed for scalability and resource sharing.

💡 Example: Google Search relies on thousands of servers distributed globally to process search queries instantly.

PARALLEL AND DISTRIBUTED COMPUTING

SERIAL VS. PARALLEL COMPUTING

Feature	Serial Computing	Parallel Computing
Execution	Tasks run one after another on a single processor.	Tasks run simultaneously on multiple processors.
Speed	Slower due to sequential execution.	Faster due to concurrent execution.
Efficiency	Limited by processor capability.	Utilizes multiple processors efficiently.

💡 Example:

- Serial Computing: A single cashier handling a long queue of customers.
- Parallel Computing: Multiple cashiers handling different customers simultaneously.

WHY DO WE NEED PARALLEL COMPUTING?

- **Faster Processing:** Speeds up computation time significantly.
- **Large-Scale Problems:** Essential for handling complex simulations, AI, and data analytics.
- **Better Resource Utilization:** Utilizes multiple processors efficiently.

💡 Example: Rendering a high-quality 3D animation using multiple processors to generate frames in parallel.

PARALLEL AND DISTRIBUTED COMPUTING

PARALLEL VS. DISTRIBUTED COMPUTING

Feature	Parallel Computing	Distributed Computing
Number of Systems	Uses a single system with multiple processors.	Uses multiple independent computers.
Memory Usage	Shared or distributed memory.	Each system has its own memory.
Communication	Uses shared memory or interconnection.	Communicates through message passing.
Synchronization	All processors follow a global clock.	No global clock; uses synchronization algorithms.

- **Parallel Computing:** Running AI algorithms on GPUs.
- **Distributed Computing:** Cloud storage services like Google Drive, where files are distributed across multiple servers.

PARALLEL COMPUTING ARCHITECTURES

Parallel computing architectures are categorized based on instruction and data streams:

1. Single Instruction, Single Data (SISD)
2. Single Instruction, Multiple Data (SIMD)
3. Multiple Instruction, Single Data (MISD)
4. Multiple Instruction, Multiple Data (MIMD)

PARALLEL COMPUTING ARCHITECTURES

SINGLE INSTRUCTION SINGLE DATA (SISD)

The original Von Neumann Architecture that does not employ any kind of parallelism. The sequential processor takes data from a single address in memory and performs a single instruction on the data. All single processor systems are SISD.

Common usage

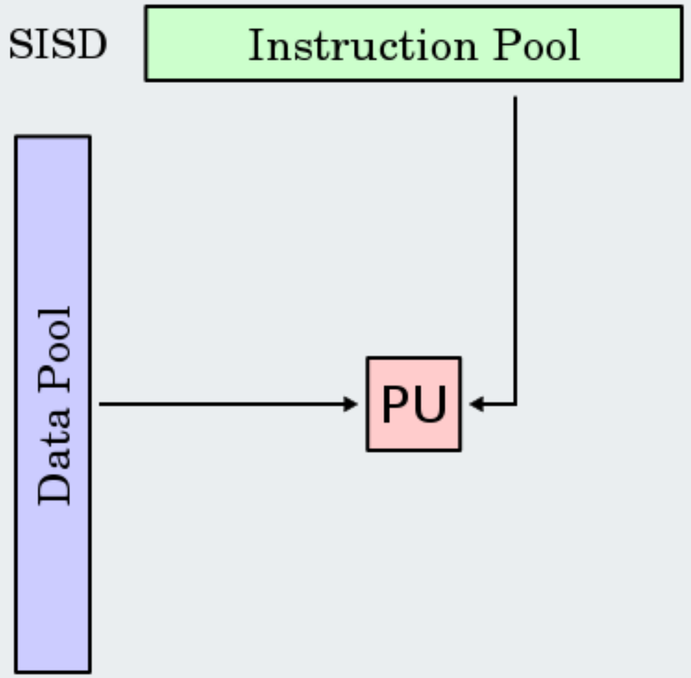
- Older Computers
- Microcontrollers

Advantages

- Low power requirements as only a single core
- Simpler architecture than others therefore cheaper and easier to manufacture

Disadvantages

- Speed of the system limited due to it being a single core



PARALLEL COMPUTING ARCHITECTURES

SINGLE INSTRUCTION MULTIPLE DATA (SIMD)

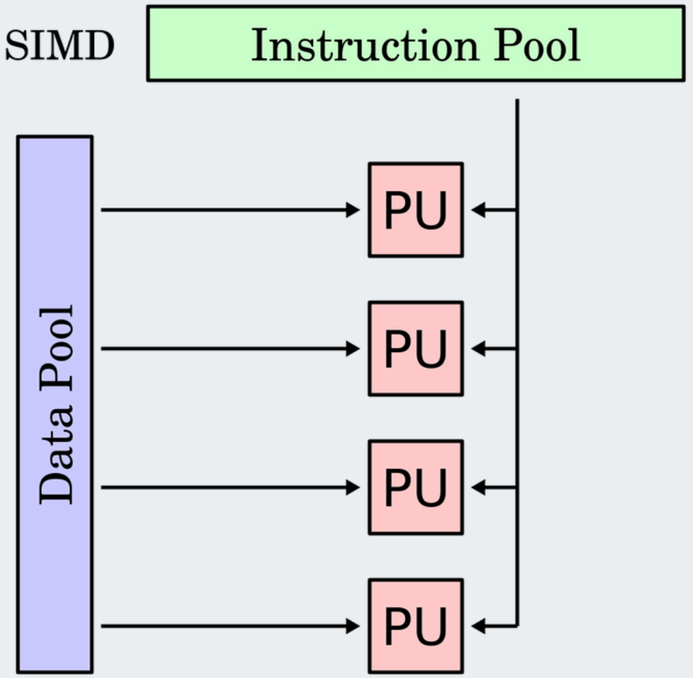
A single instruction is executed on multiple different data streams. These instructions can be performed sequentially, taking advantage of pipelining, or in parallel using multiple processors. Modern GPUs, containing Vector processors and array processors, are commonly SIMD systems.

Common usage

- Graphics Processing Units when performing vector and array operations.
- Scientific processing

Advantages

- Very efficient where you need to perform the same instruction on large amounts of data.



PARALLEL COMPUTING ARCHITECTURES

MULTIPLE INSTRUCTION SINGLE DATA (MISD)

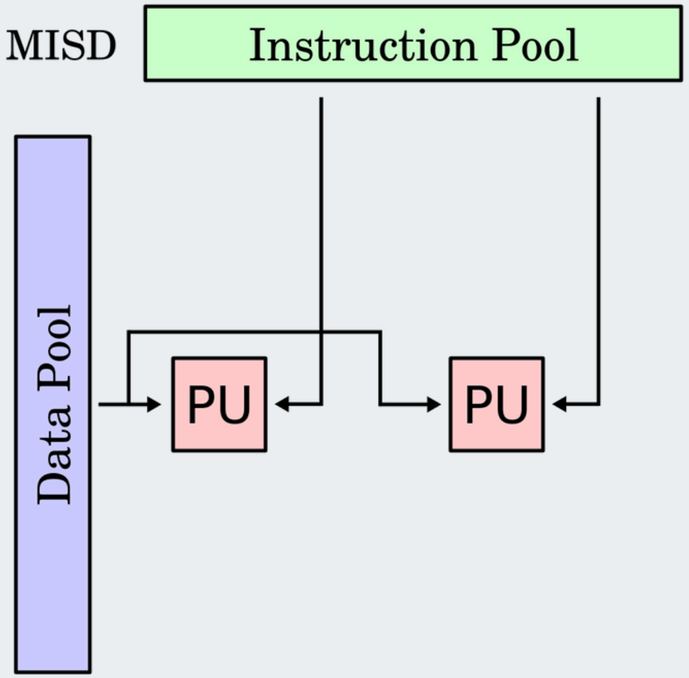
In this architecture multiple instructions are performed on a single data stream. An uncommon type commonly used for fault tolerance. Different systems perform operations on the data and all the results must agree. Used on flight control systems where fault detection is critical.

Common Usage

- Not used commercially.
- Some specific use systems (space flight control)

Advantages

- Excellent for situation where fault tolerance is critical



PARALLEL COMPUTING ARCHITECTURES

MULTIPLE INSTRUCTION MULTIPLE DATA (MIMD)

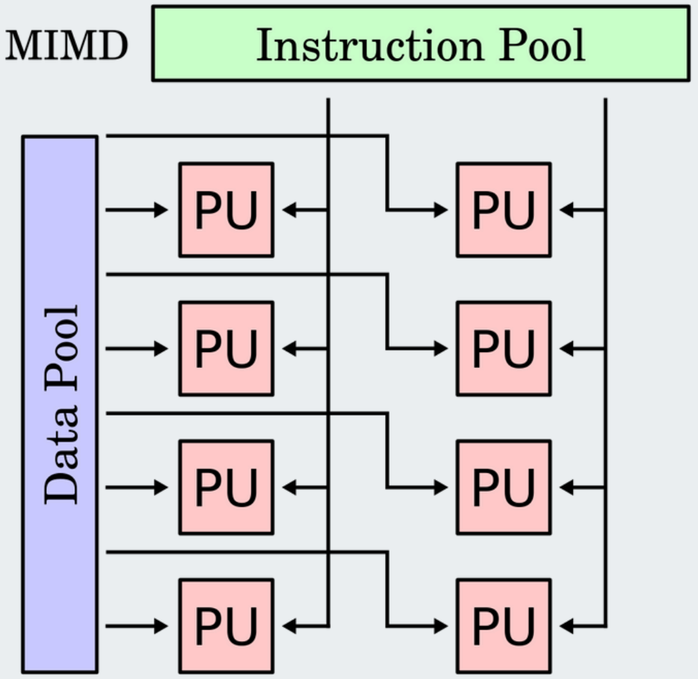
Multiple autonomous processors perform operations on difference pieces of data, either independently or as part of shared memory space.

Common Usage

- Most modern desktop / laptop / mobile processors are MIMD processors.

Advantages

- Great for situations where you need to perform a variety of processor and data intensive tasks (such as video editing, game rendering)



PARALLEL AND DISTRIBUTED COMPUTING

SHARED VS. DISTRIBUTED MEMORY IN MIMD SYSTEMS

Type	Shared Memory MIMD	Distributed Memory MIMD
Memory	Single shared memory for all processors.	Each processor has its own local memory.
Communication	Uses global memory for inter-processor communication.	Uses message passing for communication.
Speed	Faster for small-scale applications.	More scalable but requires synchronization.
Example	Multi-threaded applications on a single machine.	Cloud computing systems like AWS, Google Cloud.

- **Shared Memory MIMD:** High-performance gaming PCs using multiple cores.
- **Distributed Memory MIMD:** Supercomputers analyzing climate data across multiple locations.