

Introduction to parallel computing

Julián Jiménez Cárdenas

Scientific Computing II

Fundación Universitaria Konrad Lorenz

Parallelization architectures

Parallelization architectures refer to the different ways in which tasks or processes can be executed concurrently to improve computational performance.

These architectures are essential for taking advantage of modern multi-core processors and distributed computing environments. Here we will show some common parallelization architectures.

Shared memory architecture

- In this architecture, multiple processors or threads share a common memory space.
- Each processor can access and update data stored in the shared memory.
- However, managing data consistency and synchronization between processors is crucial to prevent data conflicts.

Distributed memory architecture

- In contrast to shared memory, distributed memory architecture involves multiple processors or nodes, each with its own local memory.
- Communication between processors occurs explicitly through message passing. This architecture is often used in cluster computing.

SIMD (Single Instruction, Multiple Data)

- SIMD architecture processes multiple data elements in parallel using a single instruction.
- It is commonly used in vector processing units or GPUs (Graphics Processing Units) to accelerate tasks like image and video processing.

MIMD (Multiple Instruction, Multiple Data)

- MIMD architecture supports multiple processors, each with its own instruction stream and data stream.
- Each processor can execute different instructions and operate on different data.
- This architecture is typical in multi-core CPUs and distributed computing environments.

Task Parallelism

- Task parallelism involves dividing a program into multiple tasks or threads that can execute independently.
- Task parallelism is commonly used in parallel programming frameworks like OpenMP.

Data Parallelism

- Data parallelism involves splitting a dataset into smaller chunks and processing each chunk simultaneously on different processors.
- This approach is often used in parallel computing to speed up operations like matrix multiplication and data analysis.

Pipeline Architecture

- In pipeline architecture, multiple stages of a computation are executed in parallel, with each stage processing a portion of the data.
- This architecture is commonly used in scenarios where data flows through a sequence of processing steps, such as in hardware design and streaming data processing.

Hybrid Parallelism

- Hybrid parallelism combines multiple parallelization techniques within the same application.
- For example, a program might use both task parallelism and data parallelism to optimize different parts of the computation.

Cluster Computing

- Cluster computing involves connecting multiple independent computers (nodes) into a network to work together on a single task.
- Each node can perform parallel processing, and tasks are distributed among the nodes.
- High-performance computing clusters are commonly used in scientific and research applications.

Cloud Computing

- Cloud computing platforms provide scalable and distributed resources for parallel processing.
- Users can rent virtual machines or containers with varying degrees of parallelism to execute their applications.

Let's go to Python