



Parallelism Basics

Some Basic Parallel Computing Terminology

Sequential Program

Concurrent Program

To cooperate, the processes need _____ and _____.

Two major mechanisms to achieve these are _____ and _____.

Hardware to Run Concurrent Processes

- single processor
- multiprocessor
- multicomputer
- networked systems

Computers may be classified using _____:

- *single instruction, single data (SISD)*
- *single instruction, multiple data (SIMD)*
- *multiple instruction, multiple data (MIMD)*

Some Hardware Examples

- SISD: Legacy single-core processors (e.g., early Intel Pentium systems).
- SIMD: CPU vector extensions (e.g., Advanced Vector Extensions 512 [AVX-512], ARM Scalable Vector Extension [SVE]).
- SIMD / Accelerators: Modern Graphics Processing Units (GPUs) like the NVIDIA H100/B200 or AMD Instinct MI300X, which execute identical instructions across broad vector lanes (often managed via Single Instruction, Multiple Threads [SIMT]).
- MIMD (Shared Memory): Modern consumer laptops, desktops, and workstations featuring multi-core CPUs (e.g., Intel Core Ultra, AMD Ryzen/Threadripper, Apple M-series) where each core can execute independent instruction streams.
- MIMD (Distributed Memory / Cluster): `noreaster.teresco.org`: Dual Intel Xeon Processor E5-2630 v4 (10 cores per socket, 20 cores per node, 2.2GHz).
- MIMD / Heterogeneous (Supercomputer): Stampede3 at the Texas Advanced Computing Center (TACC): A heterogeneous cluster comprising 1,858 compute nodes with over 140,000 CPU cores (including Intel Xeon CPU Max with High-Bandwidth Memory [HBM] and Ice Lake nodes), alongside NVIDIA H100 and Intel Data Center GPU Max acceleration nodes.
- MIMD (Historical Benchmark): Accelerated Strategic Computing Initiative (ASCI) Red, Sandia National Laboratories: 4,600+ nodes, each with 2 Intel Pentium II Xeon processors; reached 1.33 TeraFLOPS (Floating-Point Operations Per Second) in 1997 as the world's first TeraOp machine.
- See <https://www.top500.org/>.

The primary programming model we will use in this class is called *single program, multiple data (SPMD)*

Key point: From smartphones and tablets to high-performance scientific clusters and exascale supercomputers, contemporary computing hardware relies fundamentally on parallel processing!

How to Achieve Parallelism

We need to determine where concurrency is possible, then _____ the work accordingly. There are two major approaches to this problem.

- *task parallelism*

- *data parallelism*

- This is easiest if a compiler can do this for you – take your sequential program and extract the concurrency automatically. This is sometimes possible, especially with fixed-size array computations.
- If the compiler can't do it, it is possible to give “hints” to the compiler to tell it what is safe to parallelize.
- But often, the parallelization must be done explicitly: the programmer has to create the threads or processes, assign work to them, and manage necessary communication.
- AI assistance is sure to be helpful in dealing with the syntactic details, but getting concurrency right is hard and will take a human understanding to guide an AI to a good solution and to verify that it will behave correctly.