CPA – Parallel Computing

Degree in Computer Science

T1. Introduction to Parallel Computing

J. M. Alonso, P. Alonso, F. Alvarruiz, I. Blanquer,
J. Ibáñez, E. Ramos, J. E. Román

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València

Year 2024/25

Contents

Section 1

Introduction

- Parallel Computing Applications
- Supercomputing

Parallel Computing

What is parallel computing?

The computation performed on a parallel computer

- Can execute concurrently several instructions for the resolution of a single problem

As opposed to a sequential computer

- Follows the conventional (von Neumann) architecture
- Instructions are executed sequentially one after the other

Parallel computing requires

- Redesigning the algorithms
- Changing the data structures

$\longrightarrow$ parallel programming

Motivation

Why parallel computing?

There is a trend to include ever more parallelism in hardware

- Clock frequency limited by the speed of light
- Larger integration scale unfeasible due to heating problems

Examples: Game consoles with more than 2000 cores
$\longrightarrow$ There is a lack of programmers capable of exploiting such hardware!

Sometimes, sequential computing does not cover the needs

- Large scale problems
- Real time constraints

Example: many matrix algorithms have a cost $O(n^3)$ requiring problem dimensions of $n = 10^6$ or even larger

Numerical Simulation

Simulation: emulate a physical system using a computer

In engineering:

- Virtual prototyping
- Cost and production cycle reduction

In science:

- Key for scientific progress
- Complex systems: complex geometry, multi-physics models, non-linearity

Often, simulation requires a large computation capacity

Simulation: Example

Simulation of physiological behaviour of organs, tissues and cells (VPH: *Virtual Physiological Human*)

- Long term vision: To simulate the complete physiological behaviour of the human body

Example: Alya-Red

- Is a computational simulator of a heart that models the electric, visco-elastic, fluid-dynamic and physic-chemical behaviour of a whole heart
- 1st award of divulgation scientific videos NSF 2012: `www.youtube.com/watch?v=tKD2hfF27rM`

Data processing

Large scientific challenges require acquiring, storing and processing large data volumes

- Disciplines such as High Energy Physics, Astrophysics or Biotechnology
- Data coming from radiotelescopes, particle colliders or human genome

Example: The Large Hadron Collider at CERN

Supercomputing

Parallel computing, High Performance Computing or Supercomputing, consist in the concurrent execution of different tasks by different processing units

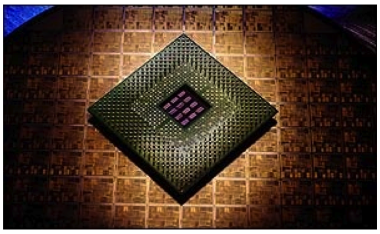
Concurrency may happen at different scales:

- In a single processor
- In a single computer
- On the Internet

Supercomputing

In a single processor

- Include several compute units in a single processor
- Simultaneous execution of multiple basic instructions



In a single computer

- Integration of different processors in a single computer
- Depending on the coupling of processors:
  - Shared memory
  - Distributed memory

Parallel Computers

Shared Memory

- All processors can access the whole memory
- Different latencies depending on the memory bank accessed (due to proximity)
- Scalability up to hundreds of processors
- High cost; high performance in fine-grain parallelism

Distributed Memory

- Each processor has only access to its local memory block
- Information is explicitly exchanged through messages
- Higher scalability (thousands of processors)
- Reduced cost; worse performance in the fine grain

TOP500

The TOP500 list records the 500 most powerful computers in the world
http://www.top500.org

- Updated twice a year
- Computers are classified according to their sustained performance, measured in floating-point operations per second (Flop/s)

#	Centre	System	Manuf.	Cores	RMax
1	ORNL (USA)	Frontier - AMD EPYC 64C 2GHz + AMD Instinct MI250X, Slingshot-11	HPE	8.730.112	1102,0
2	ARGON N.L. (USA)	Aurora -INTEL EXASCALE COMPUTE BLADE, XEON CPU MAX 9470 52C 2.4GHZ, INTEL DATA CENTER GPU MAX, SLINGSHOT-11	HPE	9.264.128	1012,0
3	MICROSOFT (USA)	Eagle - MICROSOFT NDV5, XEON PLATINUM 8480C 48C 2GHZ, NVIDIA H100, NVIDIA INFINIBAND NDR	HPE	2.073.600	561,2
4	RIKEN (Japan)	Fugaku - A64FX 48C 2.2GHz, Tofu interconnect D	Fujitsu	7.630.848	442,01
5	EuroHPC/CSC (Finland)	LUMI - AMD EPYC 64C 2GHz + AMD Instinct MI250X, Slingshot-11	HPE	1.110.144	151,9

RMax in PFlop/s

Spanish Supercomputing Network (RES)

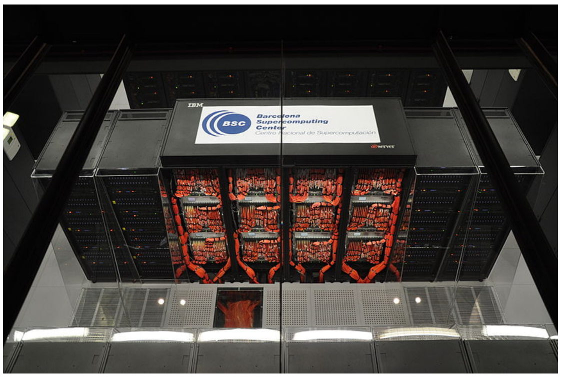
It currently comprises the following centres:

- Barcelona Supercomputing Centre (BSC)
- U. Politécnica Madrid (CeSViMa)
- Universidad de Cantabria (IFCA)
- I. Astrofísica de Canarias
- Universidad de Málaga
- Universitat de València
- Universidad de Zaragoza (BIFI)
- U. Las Palmas de Gran Canaria

Spanish Supercomputing Network (RES)

MareNostrum 5 currently has:

- General Partition with 6408 nodes 2x Intel Shappire Rapids 8480+ 56 cores 2 GHz
- Fast Partition with 1120 nodes 2x Intel Shappire Rapids 8460Y+ 40 cores 2.3 GHz 4x NVIDIA Hopper 64 GB HBM
- Maximum Performance: 305.5 PFlop/s
- Infiniband Network NDR200
- 2615 TB main memory, 652 PB disk



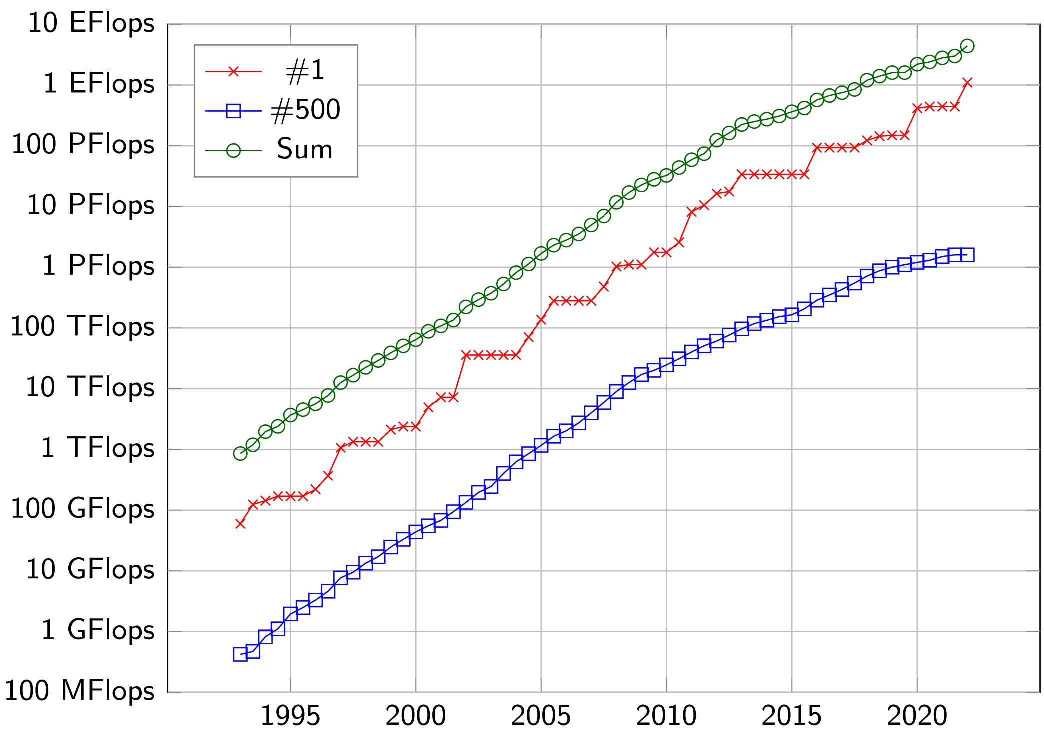
Other Centres

Regional Supercomputing Centres:

- CESGA: Centro de Supercomputación de Galicia
  - FinisTerrae II, 320 nodes, 7712 Haswell cores, 213 TFlop/s
- SCAYLE: Supercomputación Castilla y León
  - Several machines
- CénitS: Centro Extremeño de iNvestigación, Innovación Tecnológica y Supercomputación
  - Lusitania II, 2x40 Intel Xeon E5-2660v3 10C, 2x168 Intel E5-2670 8C
- CSUC: Consorci de Serveis Universitaris de Catalunya

UPV: (Rigel) 155 nodes, with 2 Intel Xeon processors each; 10 GbE network; performance 50 TFlop/s

Top500: Evolution of the Performance



Source: `http://www.top500.org`

Section 2

Parallel Computing Models

- Parallel Computing Architectures
- Parallel Programming Methodologies

Flynn Taxonomy

SISD	SIMD
Single Instruction, Single Data	Single Instruction, Multiple Data

MISD	MIMD
Multiple Instruction, Single Data	Multiple Instruction, Multiple Data

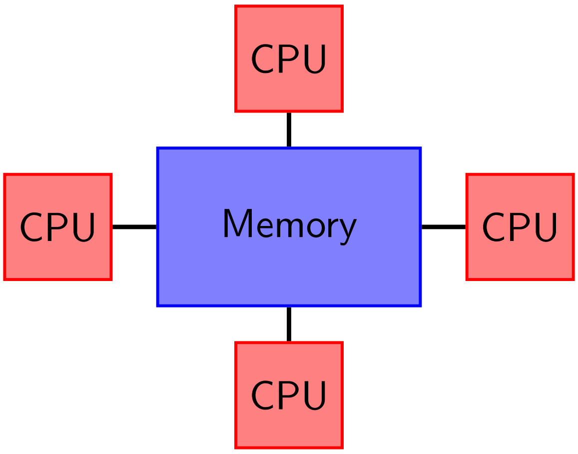
SISD Sequential computer

SIMD Vector computers (NEC, Fujitsu, Cray), processors with vector extensions (SSE3, Altivec, AVX-512)

MIMD Multiprocessors, clusters, multi-core

Shared Memory Architectures

Single address space for all the processors



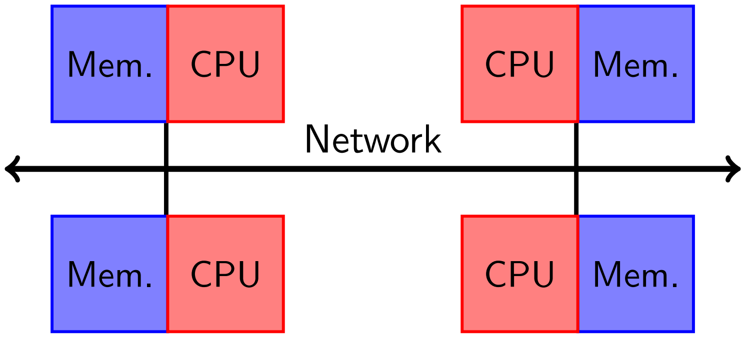
Types:

- UMA: Uniform Memory Access
- NUMA: Non-Uniform Memory Access
- cc-NUMA: Cache Coherent NUMA

Advantages: easy programming; Disadvantages: scalability, price

Distributed Memory Architectures

It requires a communication network to let processors access data outside the local space



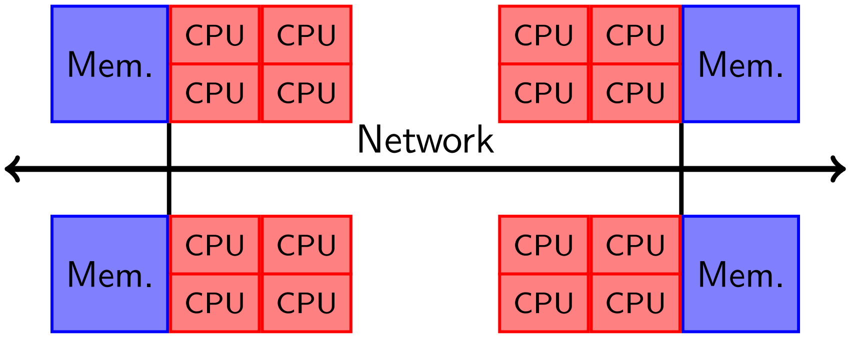
Features

- There is no global memory concept
- Independent processors (no coherence)
- The programmer explicitly exchanges data

Advantages: scalability, cost; Drawbacks: programming effort

Hybrid Architectures: *Distributed-Shared Memory*

Combination of the two models



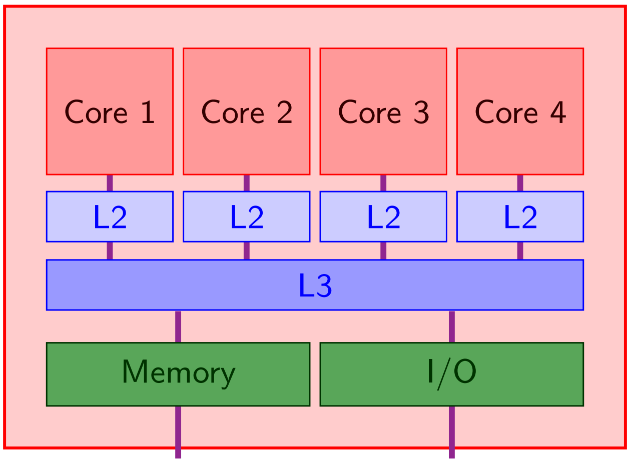
Features

- Each node is a multiprocessor (e.g. cc-NUMA)
- Communication to move data from one node to another

Current supercomputers often follow this model (also with multicores and GPUs).

Multi-Core Processors

Multiple cores: current trend in processor design



Features

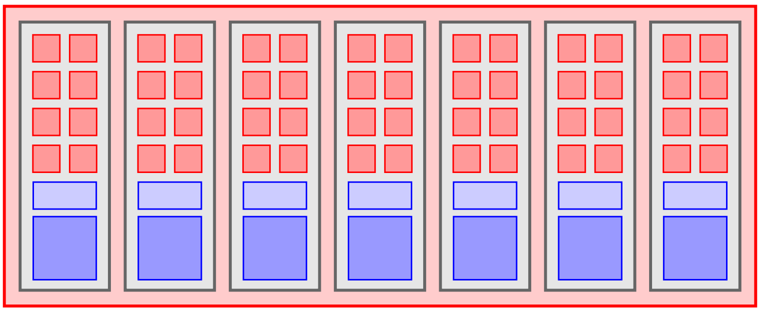
- Symmetric (or not) multiprocessing on a single chip
- Several cache levels in the same chip

Advantages: cost; Drawbacks: low efficiency (bandwidth)

Many-Core Processors

Massively parallel, with a large number of simple cores

- Graphic Processor Units (GPU)



Features

- Many cores (in the order of thousands)
- Light threads, very quick context switch

Advantages: cost, performance; Drawbacks: difficult programming

Clusters

A *cluster* is simply a set of PCs or workstations connected in a network to run parallel programs

- The first one using PCs and Linux dates back to 1994 (Beowulf)

Currently, a commercial cluster typically comprises:

- Rack structure
- A set of nodes
  - Typically, two multi-core processor and 1 disk
  - Optionally, 1 high-end GPU
  - Compact format: 1U, 2U, *blades*
- Network infrastructure
  - Typically, two networks: ethernet and low-latency network
  - Low latency networks: Infiniband, Myrinet, Quadrics, ...
  - Components: adapter, *switches*, cables
- A *front-end* node

Parallel Programming Methodologies

Different methodologies can be applied:

- Automatic parallelization (parallelizing compiler)
- Semi-automatic parallelization (compiler directives)
- New programming languages
- Extensions of conventional languages
- Libraries of subroutines or functions (API)

Examples

- OpenMP is based on directives
- CUDA C is an extension of C
- MPI is based on libraries