



# High-Performance Mathematics

Objectives, Program and Organization

Progetto Speciale per la Didattica 2023/24

**Fabio Durastante** (Lo)

March 27, 2024





# Table of Contents

## 1 High-Performance Mathematics: why?

- ▶ High-Performance Mathematics: why?
- ▶ Yes, but why?
- ▶ High-Performance Mathematics: what?
- ▶ High-Performance Mathematics: how?
- ▶ High-Performance Mathematics: when?



# Scientific computing

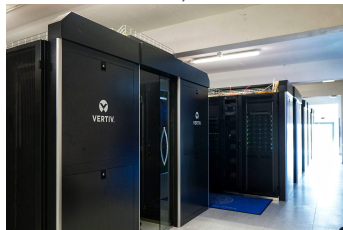
## 1 High-Performance Mathematics: why?

“**Computational science** (also **scientific computing** or **scientific computation** (SC)) is a rapidly growing multidisciplinary field that uses advanced computing capabilities to *understand and solve complex problems*. It is an area of science which spans many disciplines, but at its core it involves the development of *models and simulations to understand natural systems*.”

Wikipedia



Leonardo, CINECA



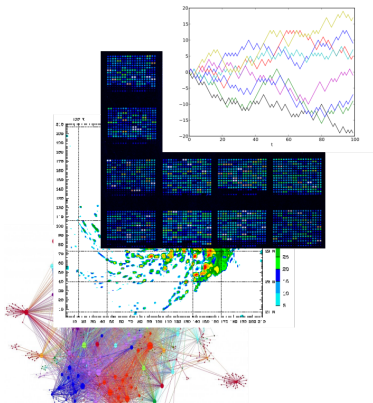
Green Data Center, UNIPI



# What are the applications?

## 1 High-Performance Mathematics: why?

- Computational finance,
- Computational biology,
- Simulation of complex systems,
- Network analysis
- Multi-physics simulations,
- Weather and climate models,
- Artificial Intelligence,
- ...

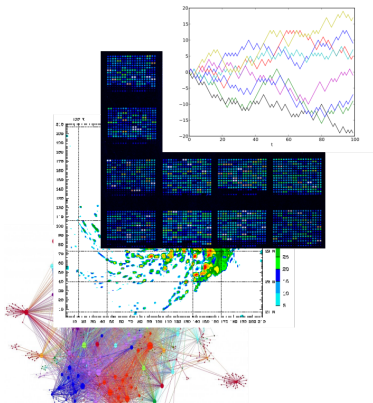




# What are the applications?

## 1 High-Performance Mathematics: why?

- Computational finance,
- Computational biology,
- Simulation of complex systems,
- Network analysis
- Multi-physics simulations,
- Weather and climate models,
- Artificial Intelligence,
- ...



Why the need for **parallelism**?



# Moore's law

## 1 High-Performance Mathematics: why?

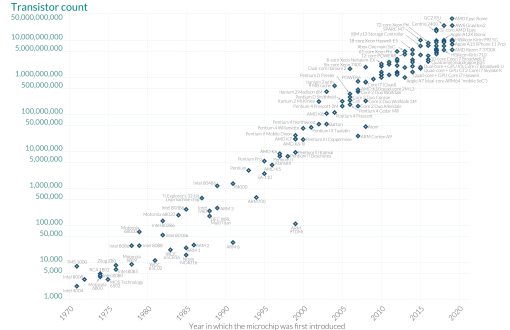


“The complexity for minimum component costs has increased at a rate of **roughly a factor of two per year**. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.”

G. Moore, 1975

### Moore's Law: The number of transistors on microchips has doubled every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



Data source: Wikipedia (wikipedia.org/wiki/Transistor\_count)  
Our World in Data.org – Research and data to make progress against the world's largest problems. Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.



# Moore's law

## 1 High-Performance Mathematics: why?

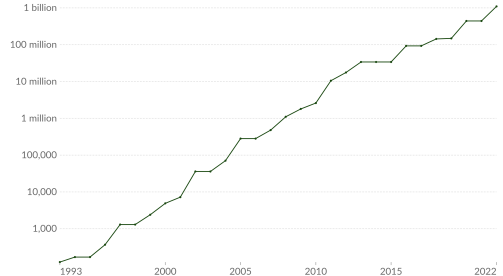


“The complexity for minimum component costs has increased at a rate of **roughly a factor of two per year**. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.”

G. Moore, 1975

### Computational capacity of the fastest supercomputers

The number of floating-point operations<sup>1</sup> carried out per second by the fastest supercomputer in any given year. This is expressed in gigaFLOPS, equivalent to  $10^9$  floating-point operations per second.



Source: TOP500 Supercomputer Database (2023)

OurWorldInData.org/technological-change • CC BY

1. Floating-point operation: A floating-point operation (FLOP) is a type of computer operation. One FLOP is equivalent to one addition, subtraction, multiplication, or division of two decimal numbers.

Computers *should* reach the physical limits of Moore's Law at some point in the 2020s...exponential functions saturates physical capabilities!



## Parallel computing: why?

1 High-Performance Mathematics: why?

- We are hitting the wall of single processor transistor count/computing capabilities,





## Parallel computing: why?

### 1 High-Performance Mathematics: why?

- We are hitting the wall of single processor transistor count/computing capabilities,
- Some applications needs more memory than the one that could be available on a single machine,



## Parallel computing: why?

### 1 High-Performance Mathematics: why?

- We are hitting the wall of single processor transistor count/computing capabilities,
- Some applications needs more memory than the one that could be available on a single machine,
- Optimization of sequential algorithms can bring us only to a certain extent



## Parallel computing: why?

### 1 High-Performance Mathematics: why?

- We are hitting the wall of single processor transistor count/computing capabilities,
- Some applications needs more memory than the one that could be available on a single machine,
- Optimization of sequential algorithms can bring us only to a certain extent

*“διαίρει καὶ βασιλεύει”*  
(diáirei kái basíleue)



## Parallel computing: why?

### 1 High-Performance Mathematics: why?

- We are hitting the wall of single processor transistor count/computing capabilities,
- Some applications needs more memory than the one that could be available on a single machine,
- Optimization of sequential algorithms can bring us only to a certain extent

*“διαίρει καὶ βασιλεύει”*

(diáirei kái basíleue)

Dividi et Impera



# Parallel computing: why?

## 1 High-Performance Mathematics: why?

- We are hitting the wall of single processor transistor count/computing capabilities,
- Some applications needs more memory than the one that could be available on a single machine,
- Optimization of sequential algorithms can bring us only to a certain extent

*“διαίρει καὶ βασιλεύει”*

(diáirei kái basíleue)

Dividi et Impera

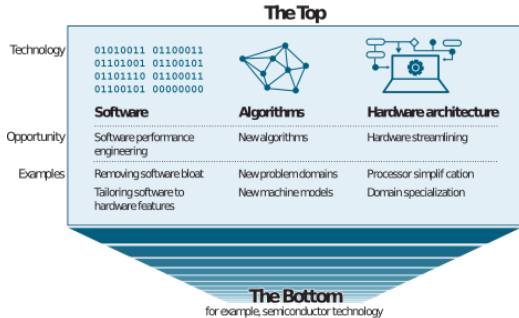
Therefore, we need

- **Algorithms** that can work in parallel,
- A **communications protocol** for parallel computation integrated with our programming languages,
- **Parallel machines** that can actually run this code.



# The philosophy behind the effort

## 1 High-Performance Mathematics: why?



“As miniaturization wanes, the silicon-fabrication improvements at the Bottom will no longer provide the predictable, broad-based gains in computer performance that society has enjoyed for more than 50 years. Software performance engineering, development of algorithms, and hardware streamlining at the Top can continue to make computer applications faster in the post-Moore era.”

C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, D. Sanchez, and T. B. Schardl, “There’s plenty of room at the Top: What will drive computer performance after Moore’s law?”, *Science* (2020)



# Table of Contents

2 Yes, but why?

- ▶ High-Performance Mathematics: why?
- ▶ **Yes, but why?**
- ▶ High-Performance Mathematics: what?
- ▶ High-Performance Mathematics: how?
- ▶ High-Performance Mathematics: when?



## Yes, but why?

2 Yes, but why?

- 😊 It is **good fun!**
- 🎓 It can be spent on the **programming/project part** of “Calcolo Scientifico”, “Laboratorio Computazionale” and “Metodi Numerici per le PDE” **exams**.
- 📖 It teaches you to use various computational resources of the Department of Mathematics, resources that can be used to carry out *thesis projects* even in disciplines *other than Numerical Analysis*.
- 🎓 If you end up liking it, it can give you the opportunity to write **theses in computational mathematics** that are **very close to applications** and **current research**.





# Table of Contents

3 High-Performance Mathematics: what?

- ▶ High-Performance Mathematics: why?
- ▶ Yes, but why?
- ▶ **High-Performance Mathematics: what?**
- ▶ High-Performance Mathematics: how?
- ▶ High-Performance Mathematics: when?



## Parallel computing: what?

3 High-Performance Mathematics: what?

What are the **topics** we will cover in these lectures?

1. Abstract models of parallel machines and their operation,



# Parallel computing: what?

3 High-Performance Mathematics: what?

What are the **topics** we will cover in these lectures?

1. Abstract models of parallel machines and their operation,
2. **Extension** and **improvement** of the **parallel computer prototype** assembled in the previous edition,



# Parallel computing: what?

3 High-Performance Mathematics: what?

What are the **topics** we will cover in these lectures?

1. Abstract models of parallel machines and their operation,
2. **Extension** and **improvement** of the **parallel computer prototype** assembled in the previous edition,
3. Laboratory introduction to **programming techniques** for the implementation of **parallel algorithms**,



# Parallel computing: what?

3 High-Performance Mathematics: what?

What are the **topics** we will cover in these lectures?

1. Abstract models of parallel machines and their operation,
2. **Extension** and **improvement** of the **parallel computer prototype** assembled in the previous edition,
3. Laboratory introduction to **programming techniques** for the implementation of **parallel algorithms**,
  - MPI,
  - OpenMP,
  - NVIDIA/CUDA,



# Parallel computing: what?

3 High-Performance Mathematics: what?

What are the **topics** we will cover in these lectures?

1. Abstract models of parallel machines and their operation,
2. **Extension** and **improvement** of the **parallel computer prototype** assembled in the previous edition,
3. Laboratory introduction to **programming techniques** for the implementation of **parallel algorithms**,
  - MPI,
  - OpenMP,
  - NVIDIA/CUDA,
4. **Use** of what has been discussed for the **solution of selected applied mathematics problems**:



# Parallel computing: what?

3 High-Performance Mathematics: what?

What are the **topics** we will cover in these lectures?

1. Abstract models of parallel machines and their operation,
2. **Extension** and **improvement** of the **parallel computer prototype** assembled in the previous edition,
3. Laboratory introduction to **programming techniques** for the implementation of **parallel algorithms**,
  - MPI,
  - OpenMP,
  - NVIDIA/CUDA,
4. **Use** of what has been discussed for the **solution of selected applied mathematics problems**:
  - Solution of large and sparse linear systems,
  - Numerical solution of PDE/ODE,
  - Training and inference with neural networks.



# Table of Contents

## 4 High-Performance Mathematics: how?

- ▶ High-Performance Mathematics: why?
- ▶ Yes, but why?
- ▶ High-Performance Mathematics: what?
- ▶ **High-Performance Mathematics: how?**
- ▶ High-Performance Mathematics: when?





# Parallel computing: how?

4 High-Performance Mathematics: how?

The idea is to

- organize **one meeting per week** lasting a **couple of hours**,
- have **practical programming sessions** in which **problem solving** is tackled,

employing:



The steffe cluster.



The NVIDIA T1000 GPUs in Aula 3



# Table of Contents

5 High-Performance Mathematics: when?

- ▶ High-Performance Mathematics: why?
- ▶ Yes, but why?
- ▶ High-Performance Mathematics: what?
- ▶ High-Performance Mathematics: how?
- ▶ **High-Performance Mathematics: when?**



## Parallel computing: when?

5 High-Performance Mathematics: when?



The **QR code** leads to a framadate to **schedule the next lecture**.