Modern computing for physics

J.Pazzini, A.Renzi
Padova University

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Physics of Data
AA 2024-2025

INTRODUCTION

Modern computing for physics

J.Pazzini
Padova University

Physics of Data
AA 2024-2025

Università
degli Studi
di Padova

Quite a puzzling name for a course...

Why another course about *computing* for Physic(ist)s?

Computing has become central to most aspects of Physics nowadays:

- Numerical computations
- Simulations
- Machine learning applications
- you name it...

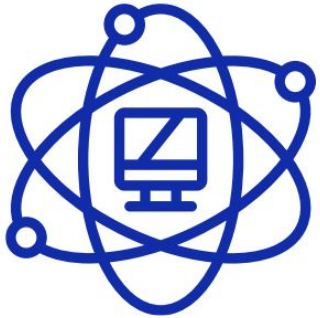
Regardless of we liking it or not, we do have to deal with the fact that significant advancements in almost every field in Physics are currently related to our ability of addressing it with computing.

... this is one of the main reason why ***Physics of Data*** came to life, after all!

The POV of the scientific community

CERN SCHOOL OF COMPUTING

BRIDGING SCIENCE & COMPUTING



- The unprecedented technological evolution in computing has profited directly to several scientific research projects, in particular in high energy physics
  - Computing is today the main strategy for many sciences to boost their research productivity

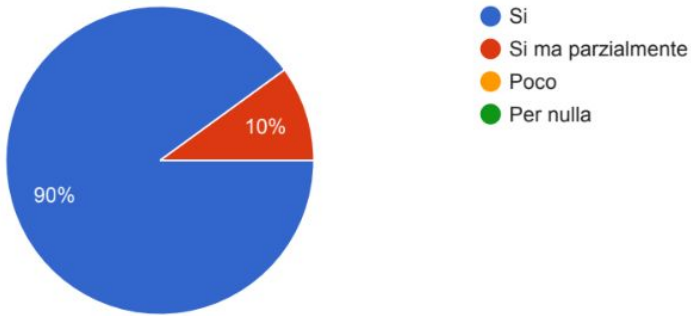
- It is nowadays essential that:
  - Scientists master computing technologies as the main tool for their research
  - Computer scientists understand the scientific domain of the investigation to deliver computing services that meet the needs of the research project

The POV of (some of) the companies providing internships for PoD

Crédit Agricole Italia
U-Hopper srl
Nozomi Networks
Porini
XSOR Capital
Radarmeteo Srl
E4 Analytics
M31 srl
221e

I laureati in fisica con competenze in Analisi Dati, Big Data, Machine Learning sono di interesse per la struttura in cui lavora

10 risposte



The POV of (some of) the companies providing internships for PoD

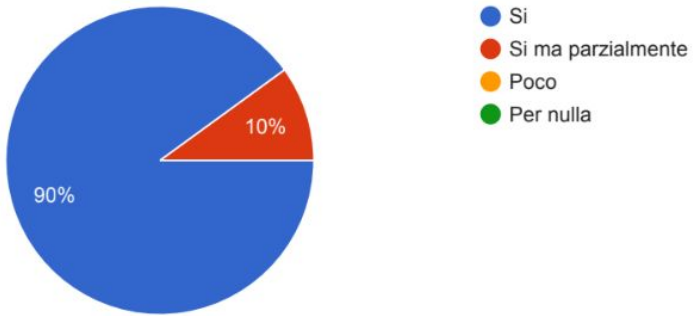
Quali sono a suo avviso i punti di forza di questo corso di studi?

9 risposte

- Viene fornito un metodo applicabile a tutti gli ambiti di business
- Trasversalità, interdisciplinarietà
- formazione eccellente su metodologie teoriche e pratiche (e.g. uso di TensorFlow)
- La grande capacità di affrontare sfide nuove
- Students seems very well prepared and show strong problem solving skills. They also combine good team working skills and solid capabilities to work independently
- Le tematiche affrontate e l'approccio che garantisce agli studenti di fare esperienze dirette

I laureati in fisica con competenze in Analisi Dati, Big Data, Machine Learning sono di interesse per la struttura in cui lavora

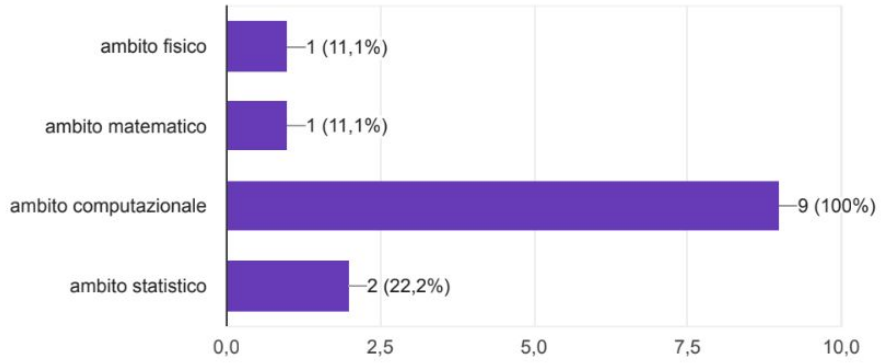
10 risposte



The POV of (some of) the companies providing internships for PoD

Quali competenze curricolari pensa potrebbero/dovrebbero essere potenziate? (indicare anche più opzioni)

9 risposte



Ha suggerimenti?

5 risposte

- Oltre alle componenti modellistiche bisognerebbe prevedere anche argomenti di "data management" per far comprendere le parti propedeutiche all'ingegnerizzazione ed automazione dei modelli
- Far lavorare gli studenti su dataset veri (ovvero sporchi, brutti e cattivi).
- Students seem quite well prepared, even in comparison with peers from other Universities
- Non in particolare
- Per aumentare le possibilita' degli studenti di essere inseriti direttamente in azienda, suggerirei di proporre anche dei corsi focalizzati sulla messa in produzione e management di modelli di machine learning (i.e. MLOps).
- È debole la conoscenza sull'utilizzo di librerie di ML/DL standard, e di Python 'avanzato' (non notebook, con utilizzo di classi etc.).

The POV of (some of) the companies providing internships for PoD

Quali sono a suo avviso i punti di debolezza di questo corso di studi?

6 risposte

- Leggermente carente degli aspetti più IT necessari all'automazione
- Lontano da problemi veri di ML/DL e manca preparazione sulla parte di preprocessing per dataset reali.
- software engineering skills
- Il "dimensionamento dei problemi". Solitamente i problemi studiati in questo corso non sono gli stessi che si trovano nella realtà aziendale
- È carente nell'esperienza specifica su GPU Computing e Orchestrazione di Container
- Non lo conosco al punto da ravvisare punti di debolezza - vedi suggerimenti.

Why *Modern Computing for Physics* then?

Modern Computing for Physics – **MCP** – is the (not very inventive) course name mean to represent what could possibly be considered as a “container” for recent topics and technologies that have a strong impact in the daily life of physicists, engineers, and developers.

Several topics have been on the plate for this first iteration of the course:

- Advanced uses of containerization
- Container orchestration
- NoSQL data management solutions
- Continuous Integration / Continuous Development
- DevOps/MLOps techniques
- System design of live-data pipelines
- **Parallel computing w/ accelerators**
- …

We decided to focus on **1**, single, yet very vast and impactful topic, and *dive deeper into it…*

But all feedback is (and will always be) welcome, to extend and improve this course over the years

Accelerators

We live in the era of the **accelerated computing**

Using specialized **hardware accelerator devices** to speed up computation and perform task more **efficiently** than general purpose CPUs is now the norm

Accelerators comes in many different forms and flavors:

- GPUs / FPGAs / ASICs / TPUs / ...

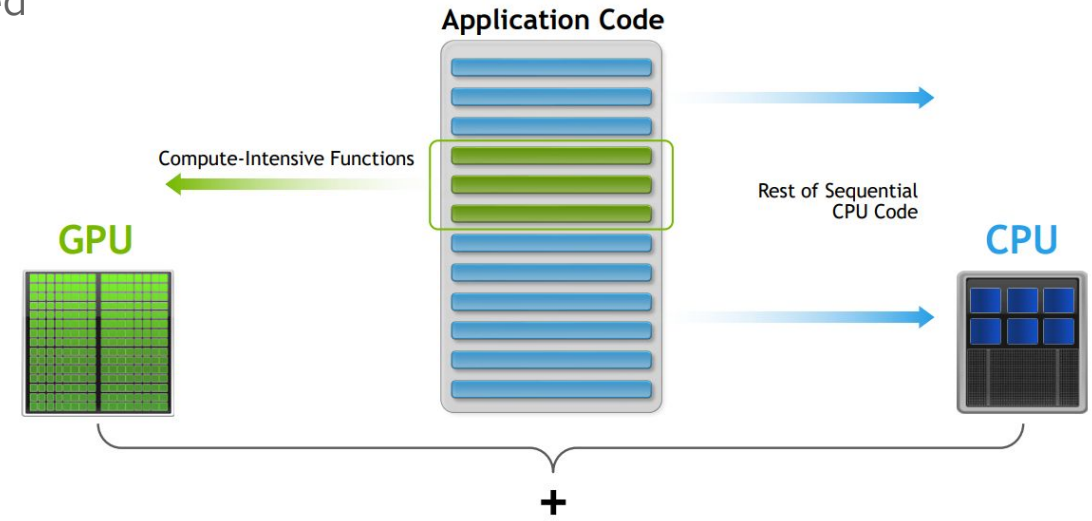
If some of these HW are still more suited for niche applications, **GPUs** are now almost ubiquitous, and have been extensively found fertile ground in a wide range of applications

Heterogeneous computing

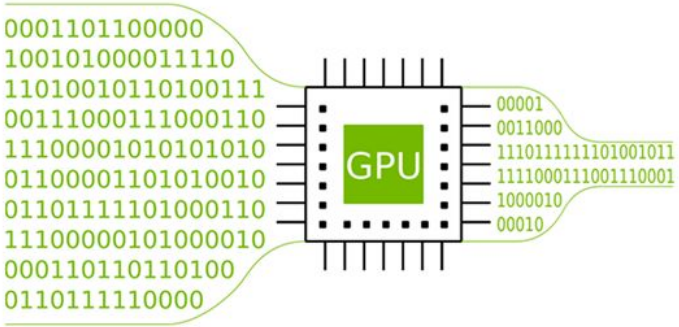
Accelerating data processing is now a key feature of most applications, both scientific and not, due to several factors:

- Large amounts of data
- Complex computations
- Continuously streaming data

Most often, applications have both heavily parallel and inherently sequential components, deeply intertwined



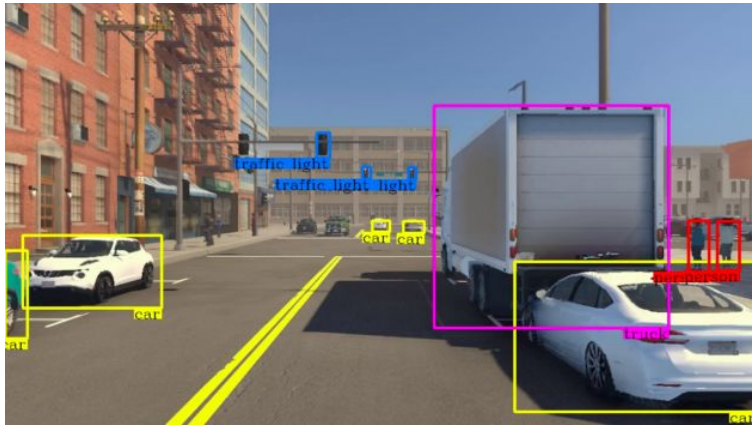
Accelerators in the industry...



Compression/decompression

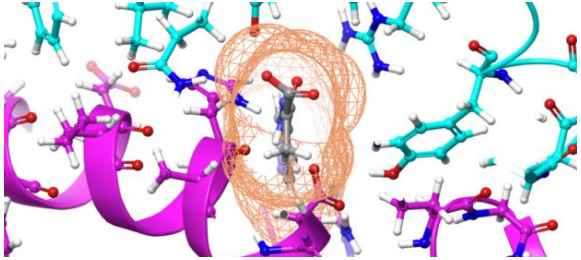
Trading and fraud detection in finance

AI and image generation models

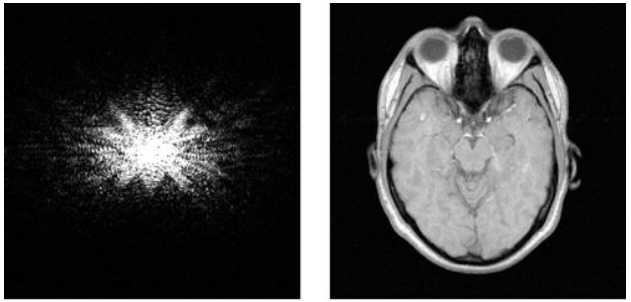


Pattern/object recognition for automatic vehicles / industrial premises

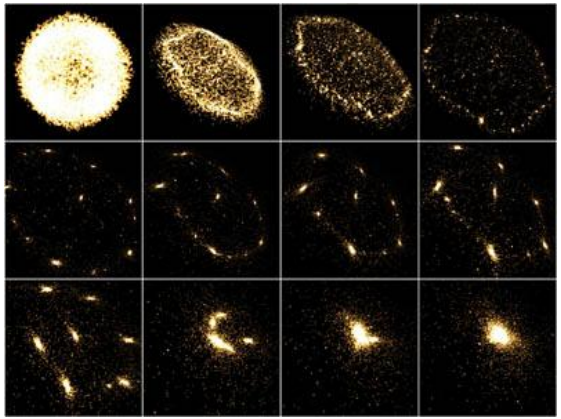
...and in Physics and Science



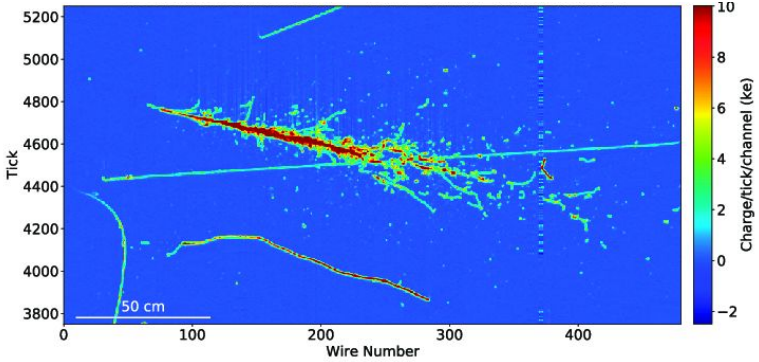
Molecular dynamics



Medical imaging



Many bodies system simulations



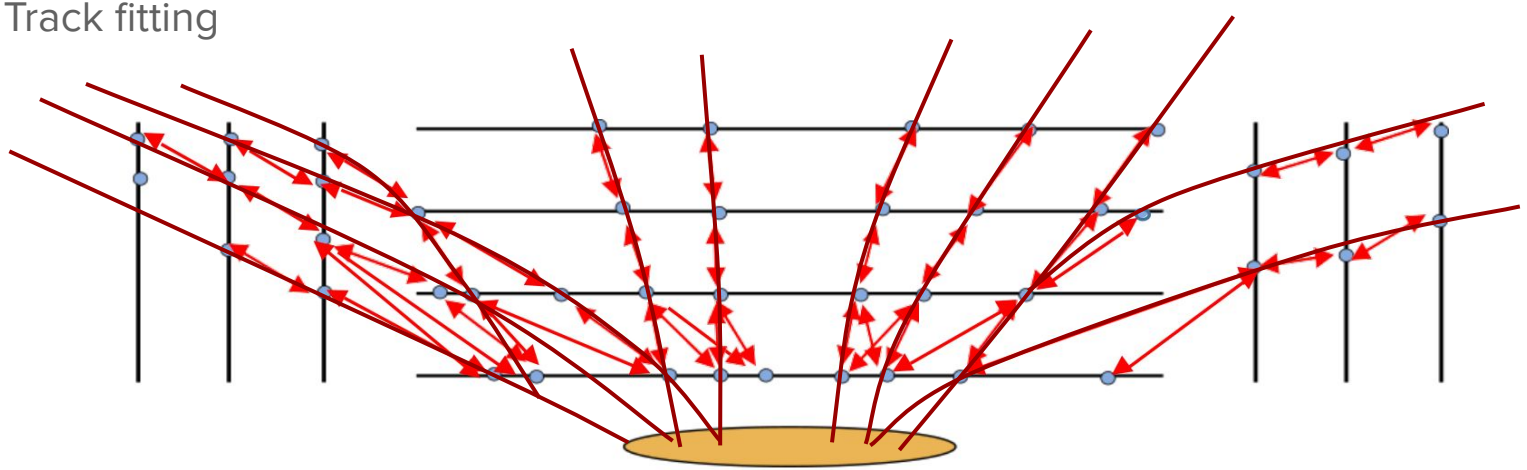
Real time data collection and Trigger

An example from HEP

A “thin slice” of the processes running at the trigger level of a High Energy Physics experiment:

Processing of RAW data to perform track reconstruction

1. Raw data decoding
2. Hit clustering
3. Pattern recognition
4. Track fitting

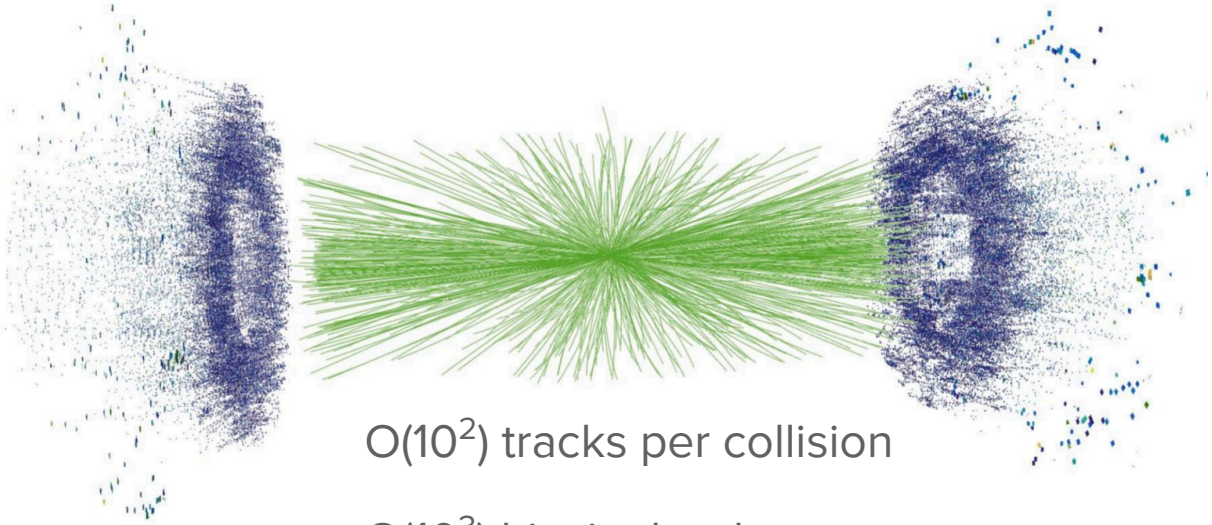


An example from HEP

A “thin slice” of the processes running at the trigger level of a High Energy Physics experiment:

Processing of RAW data to perform track reconstruction

1. Raw data decoding
2. Hit clustering
3. Pattern recognition
4. Track fitting



$O(10^2)$ tracks per collision

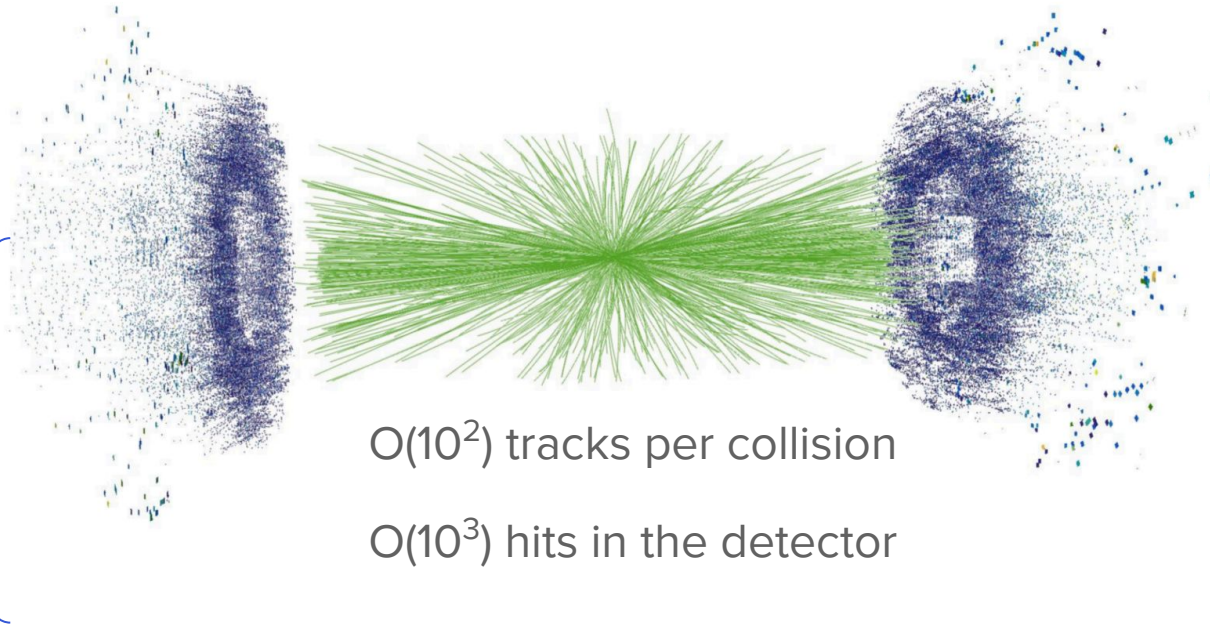
$O(10^3)$ hits in the detector

An example from HEP

A “thin slice” of the processes running at the trigger level of a High Energy Physics experiment:

Processing of RAW data to perform track reconstruction

1. Raw data decoding
2. Hit clustering
3. Pattern recognition
4. Track fitting

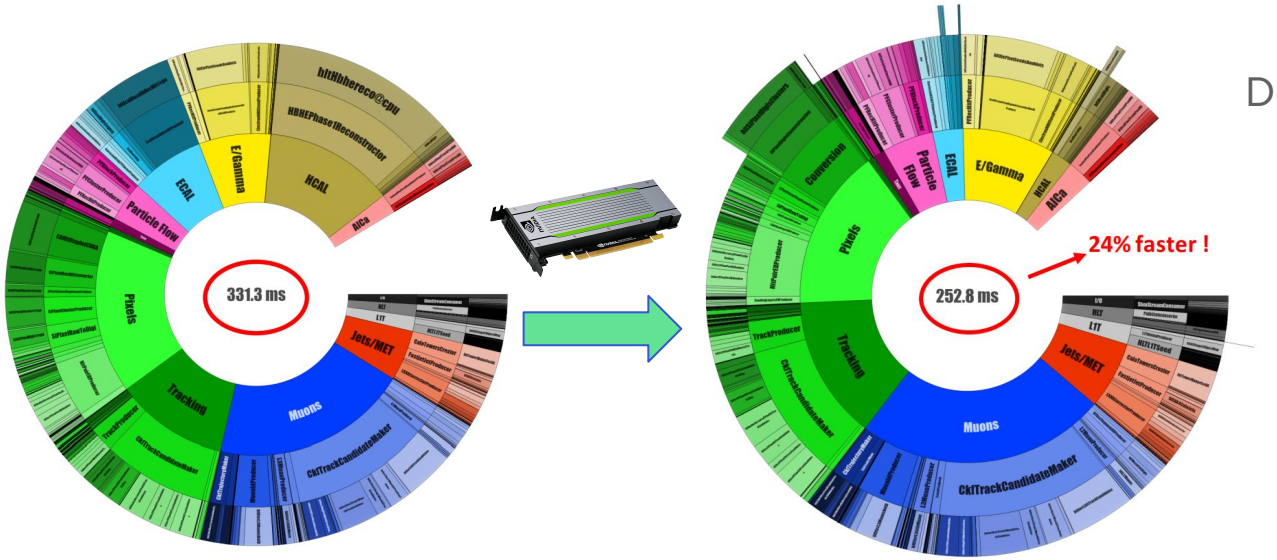


all this, with ~ 100kHz input rate

$O(10^2)$ tracks per collision

$O(10^3)$ hits in the detector

An example from HEP



Doesn't seem much... or is it?

- can process 24% more data
- can unburden other computing resources for other tasks
- can reduce the power consumption of the computing farm

And GPU-based acceleration can be integrated in all parts of the process, from data acquisition to the final result:

- Event generation
- Simulation
- Event reconstruction
- Event post-processing
- Data analysis

Sustainability through efficiency

Acceleration does not only necessarily map to a speed up of the computation
➜ doing the same, but in less time

But can also lead to **more efficient** processing
➜ doing the same with less resources

Low performance ⇒ **Long computing time** ⇒ **Waste**

- Having resources that are not used in the computation
- Powering resources at idle

Better performance ⇒ **Shorter computing time** ⇒ **Optimization**

- Maximizing the energy *efficiency*

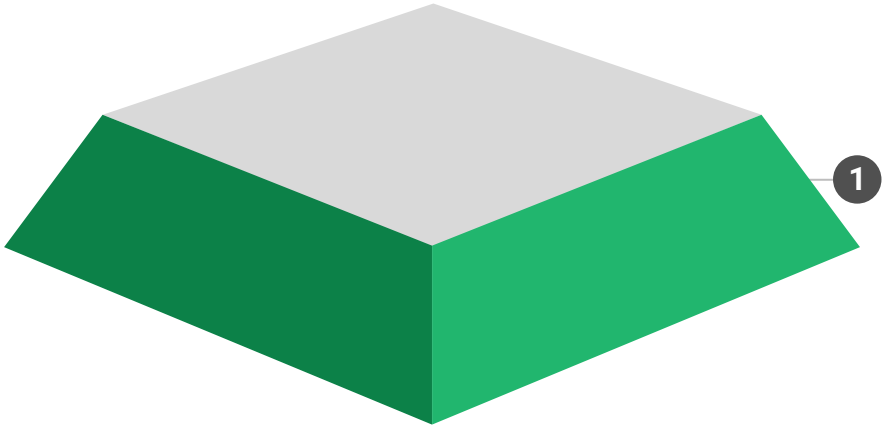
Parallel programming with GPUs

Being able to **exploit accelerators for our use-cases** is increasingly paramount, and does not always end with a “click” or an automated call

In some cases, invoking a library will be sufficient to gain an enormous advantage, but it’s not always the case… as being physicist, we typically like to try to find answers to new/yet unanswered questions

The goal of MCP is to get you some first-hand understanding on GPUs from the ground up: what they are, how to exploit them, how to accelerate computations

The menu



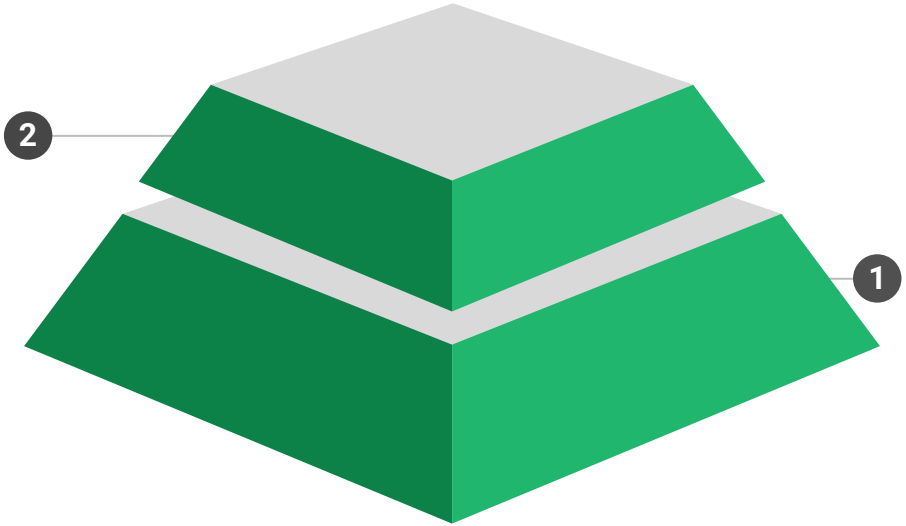
1 **Parallel processing/programming**

- Where and how parallel programming impacts physics computation
- Recap on concurrency and parallelization
- Figures of merits for GPU programming

The menu

General Purpose GPUs 2

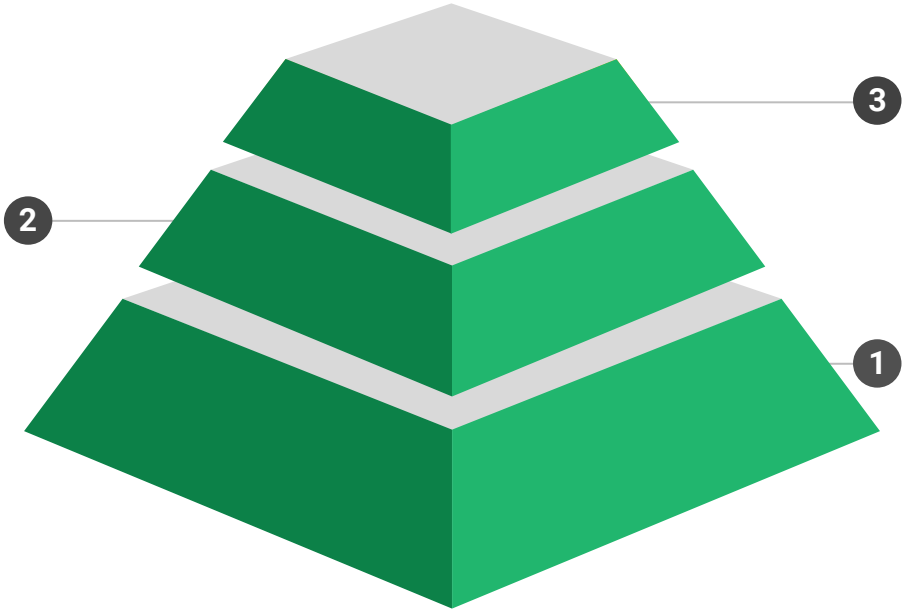
- Understanding the hardware to be able exploit it
- Differences between CPU vs GPU
- Handling memory and thread execution



1 Parallel processing/programming

- Where and how parallel programming impacts physics computation
- Recap on concurrency and parallelization
- Figures of merits for GPU programming

The menu



Basics of CUDA

- How to create and launch threads for vector/matrix processing on GPU
- Low-level language (C/C++) to understand the basics
- Porting CUDA to Python

General Purpose GPUs

- Understanding the hardware to be able exploit it
- Differences between CPU vs GPU
- Handling memory and thread execution

Parallel processing/programming

- Where and how parallel programming impacts physics computation
- Recap on concurrency and parallelization
- Figures of merits for GPU programming

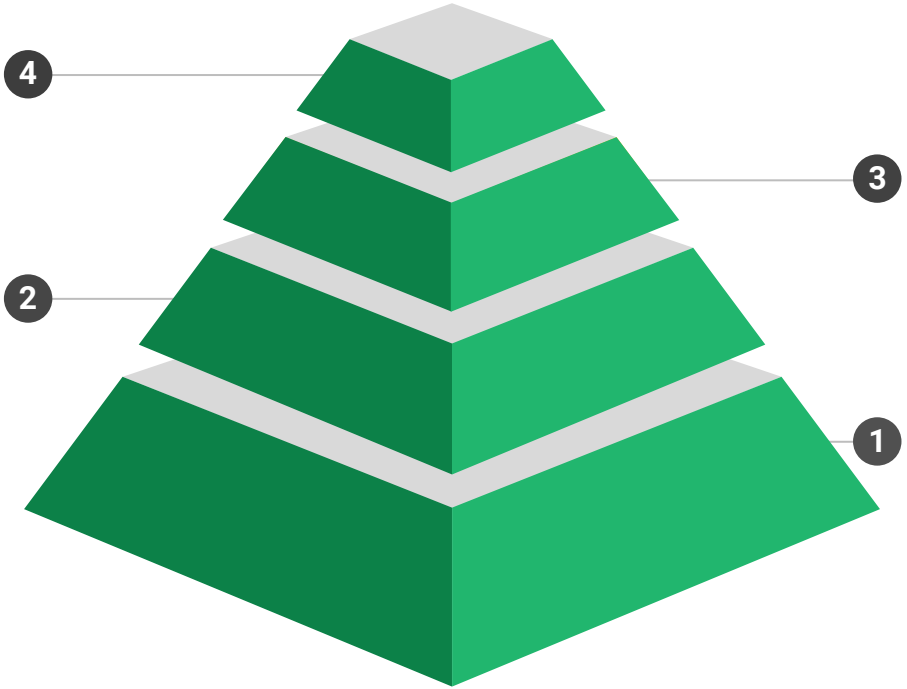
The menu

Directive-based programming

- Parallel programming with OpenACC
- Interoperability with CUDA and libraries

General Purpose GPUs

- Understanding the hardware to be able exploit it
- Differences between CPU vs GPU
- Handling memory and thread execution



Basics of CUDA

- How to create and launch threads for vector/matrix processing on GPU
- Low-level language (C/C++) to understand the basics
- Porting CUDA to Python

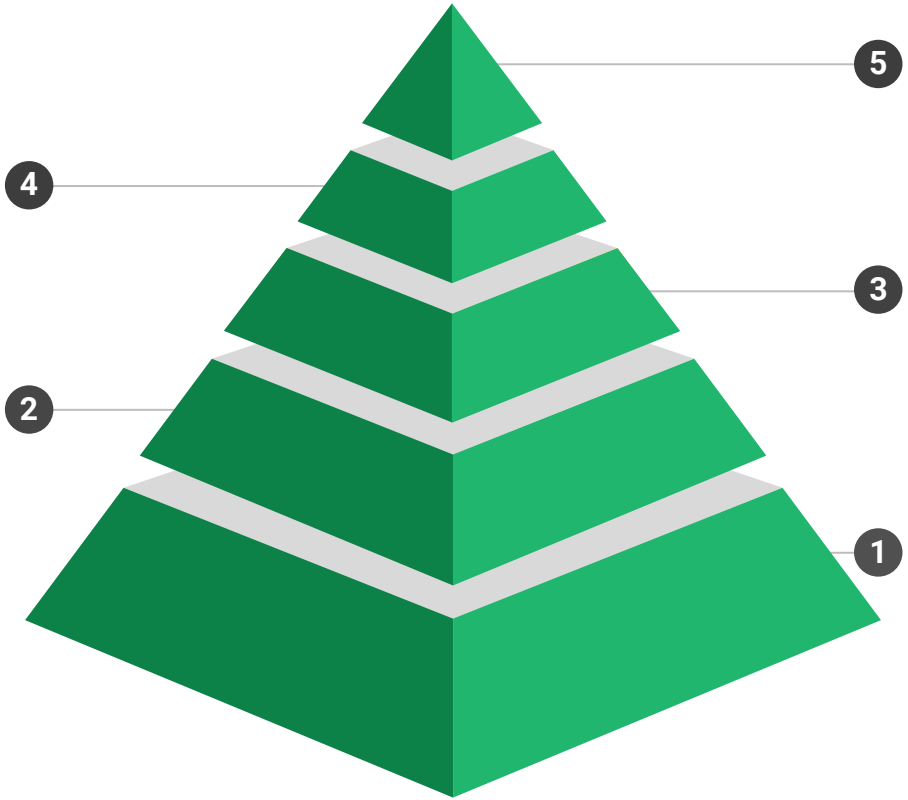
Parallel processing/programming

- Where and how parallel programming impacts physics computation
- Recap on concurrency and parallelization
- Figures of merits for GPU programming

The menu

Directive-based programming
- Parallel programming with OpenACC
- Interoperability with CUDA and libraries

General Purpose GPUs
- Understanding the hardware to be able exploit it
- Differences between CPU vs GPU
- Handling memory and thread execution



Heterogeneous computing cluster
- Programming using Message-Passing Interface
- Computation and communication on a cluster
- CUDA-aware MPI

Basics of CUDA
- How to create and launch threads for vector/matrix processing on GPU
- Low-level language (C/C++) to understand the basics
- Porting CUDA to Python

Parallel processing/programming
- Where and how parallel programming impacts physics computation
- Recap on concurrency and parallelization
- Figures of merits for GPU programming

Hands-on

Classes intended as a fair (~50-50) combination of:

- frontal classes to illustrate topics and discuss tools
- lab/exercise sessions to **practice GPU programming**

Small boards equipped with GPU:

- to use them in class for lab sessions
- and be used at home to carry on exercises on your own/in groups

Remote machines w/ GPU will be made accessible:

- to run lab and exercises on the the more advanced parts of the course

cloudveneto

Caveats and Prerequisites

- **Familiarity with Linux/Unix operating systems** (using the shell) is going to help enormously
  - Although accessing remote servers/machines via IDEs or editors (e.g. VSCode) is a feasible alternative

- A **basic level of Python** will be taken for granted
  - For Physics of Data students, what discussed in LCP and MAPD classes is absolutely sufficient

- **Basis of C/C++** are going to be required for learning CUDA
  - Write / compile / execute C/C++ programs
  - Variables and pointers
  - Loops and functions

➔ If required, a brief recap on the basics of C/C++ will be provided in class

Final exam

Development of a **final project concerning the application of GPU-accelerated parallel programming to a Physics or Machine Learning topic**

Can be performed using the JetsonNano board, CloudVeneto VMs, or private GPUs

Can be developed using **any of the tools and techniques described in the course**, but should at least integrate one minor part (kernel) of **basic CUDA code**

The examination will proceed through the **evaluation of the code** produced during and an oral discussion

Administrative stuff

When:

- Wed 8:30-10:30 (again!)
- Fri 10:30-12:30

Where:

- LabWiFi*

Moodle:

- https://stem.elearning.unipd.it/course/view.php?id=9692

Contacts:

- jacopo.pazzini@unipd.it , alessandro.renzi@unipd.it
  Room 134 Room 171

Material:

- Slides on Moodle
- Code on GitHub
- Book ➜ “Programming Massively Parallel Processors”, David B. Kirk, Wen-mei W. Hwu

Administrative stuff

When:

- Wed 8:30-10:30 (a
- Fri 10:30-12:30

Where:

- LabWiFi*

Moodle:

- https://stem.elearninc

Contacts:

- jacopo.pazzini@unipc
  Room 134

Material:

- Slides on Moodle
- Code on GitHub
- Book ➜ “Programming Massively Parallel Processors”, David B. Kirk, Wen-mei W. Hwu

For this course we needed a room with plugs…

Seems easy enough, but there are only a few of these, and the rooms’ availability was conflicting with PoD schedules

In agreement with Prof. Zanetti and Prof. Seno it was decided to use this room, but I explicitly asked:

1. **That this was an exception (emergency solution for this semester only)**
2. **That no other lectures will be placed in this room, that should still be considered as a “study-room” for PoD students**

BOTTOMLINE… THANKS FOR YOUR PATIENCE

A very quick survey...