# Report su GPU course

### **Special thanks to**

- Lecturers:
  - Andrea Bocci (CERN), Felice Pantaleo (CERN), Francesco Visconti (INAF).
- Tutors:
  - Lorenzo Capriotti, Adriano Di Florio, Tommaso Diotalevi, Aurora Perego, Giorgio Pizzati.
- Technical support:
  - Gioacchino Vino and Adriano Di Florio.
- ReCas HPC for granting access to the virtual nodes and GPUs

#### **Few numbers**

#### First course about the porting on GPUs of code and algorithms

In 19 Jun 2023, 11:00 → 21 Jun 2023, 20:00 Europe/Rome

🚹 Alexis Pompili (Istituto Nazionale di Fisica Nucleare) , Simone Gennai (MIB)

Description

Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing



- Attendance theoretical lessons (exercise):
  - 75 (45) day one

- 50 (30) day two
- 35 (10) day three
- Very tight schedule, there was not much to assimilate theoretical concepts before exercises
  - ReCas granted access to the gpu nodes also for the week after the course, to let people have more time to finish their exercises

# **Based on the survey we launched after the course**

- Overall satisfaction of the course (19 answers in total)
  - All the scores are between 3.5 and 4.0 out of 5

## C/C++ part





**ALEXIS & SIMONE 11/07/2023** 

## **Python part**







**ALEXIS & SIMONE 11/07/2023** 

### Alpaka part





**ALEXIS & SIMONE 11/07/2023** 

# **Based on the survey we launched after the course**

• Feedback for future courses

#### Which advice would you give to improve these introductory courses?

- Improve the content of the slides for an easier use in offline-mode
- Increase the length of the course
  - Increase the part dedicated to exercises
  - it could be maybe better to create separate courses for the three topics (C++/Cuda, Python/GPU, Portability tool).
    - Adding a session dedicated to basic principles of code parallelization and how to match this to GPU hardware using pseudocode might be beneficial
    - Improve theoretical part and let example to do later or after the course.
- Better define the pre-requisite to follow the course or add a first section with more general information about parallelization to ease the attending of the more specific part
- I'd like to see more of a focus on the hands on session, with the task of delivering a small (group) project in order to better understand what we are working with.

# What would you like to see in an advanced course being programmed for this autumn?

• yes

# What would you like to see in an advanced course being programmed for this autumn?

- More on Python and backend-agnostic computing
- Something about Julia and Julia on GPU
- How to actually parallelize a code
- real life HEP use cases of heterogeneous computing and performance benchmarks and optimization.
  - How to implement a real physics use case, with more details on the step-by-step exercise
- More about Alpaka
- Add a ML based use case
- Benchmarks. Software support for heterogeneous computing. FPGA vs. GPU comparison.
  FPGA programming.

#### Summary

• Overall it has been a positive experience with a profund impact in the Spoke2 community

- There is a clear request for future courses but with some improvements in the structure of the lessons.
  - We will try to implement the improvements within the constraint of the length of the course



#### Which advice would you give to improve these introductory courses?

More on Python and backend-agnostic computing

The three courses were great, just being very peaky I would like to point out some details: - The C++ course could the improve a bit the contents of the slide to make it more didactical. How Fellice presented was very good, but I could not see all that knowledge well summaresed/represented on the slides, so I would get lost a bit when looking at them. On the other hand, they contained all the important infomation so they were very helpful when doing the exercises. The drawings and concepts introduced here were very clear and helped me to follow the next courses. The Python-cuda course was nice as well. When programming in python is very difficult for me to think in terms of lower level programming, Francesco did a good job giving the message that cupy could be used similarly to numpy, but there is the possiblity to improve performance if needed. - Last presentation was amazing, maybe the theoretical part would have been better if explained in two parts. I would have been also good if Andrea had prepare some more specific exercises. In any case I'm very happy to have attended, my brain could not stop thinking about the exercises you proposed during the evening, which is fun.

#### Increase the part dedicated to exercises

I liked a lot the idea of making exercises in a breakout room, it helped me a lot to understand. On the other hand I found it difficult to finish the python exercise session because it wasn't in a breakout and I couldn't interact a lot with other students. The C++ theoretical part was not so easy to follow in the way it was presented (the slides were not so explanatory).

it could be maybe better to create separate courses for the three topics (C++/Cuda, Python/GPU, Portability tool).

Adding a session dedicated to basic principles of code parallelization and how to match this to GPU hardware using pseudocode might be beneficial

Less topics or more days, in my honest opinion it was done too much in three days. Furthermore, give the possibility to have an entire course in C++ or Python, especially the last part was too little

Improve theoretical part and let example to do later or after the course.

I would suggest the teachers to go through the different rooms more often

approfondirei maggiormente l'introduzione teorica sui concetti che differenziano la programmazione per GPU rispetto a CPU, con esempi anche banali ma partendo più dai concetti base, con esempi di codice C++ associati, oppure specificare quali siano le competenze richieste come prerequisiti. Dedicare come nella parte python anche una parte di apprendimento legata direttamente agli esercizi/esempi

Maybe spending more time on the theoretical lessons. Also, since lessons are online probably adding more questions to the audience here and there would help keeping the attention.

# What would you like to see in an advanced course being programmed for this autumn?

Something about Julia and Julia on GPU

How to actually parallelize a code

real life HEP use cases of heterogeneous computing and performance benchmarks and optimization. More about Alpaka

Maybe to implement something more elaborated using the tools teached here.

yes

More CUDA-Python part

A better and gradual introduction to the various topics treated, whatever they are.

2 working examples (a close-to-real physics analysis and an ML tool implementation) with focus on architectural aspects (and not on data preparation).

GPU parallelization applied to more physic problems

maybe some more step by step exercise.

Benchmarks. Software support for hetrogeneous computing. FPGA vs. GPU comparison. FPGA programming.

GPU in julia

panoramica sui diversi tool esistenti e per quali scopi sono più indicati

I'd like to see more of a focus on the hands on session, with the task of delivering a small (group) project in order to better understand what we are working with.