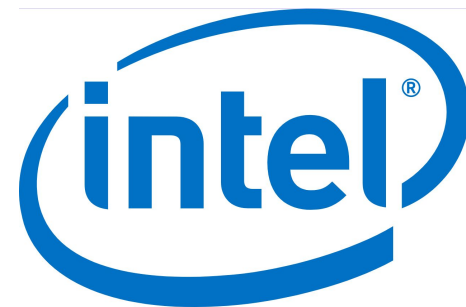


Development of a heterogeneous and portable CMS reconstruction with SYCL/OneAPI:

Andrea Bocci, Laura Cappelli, Luca Ferragina, Francesco Giacomini, Matti Kortelainen, Vincenzo Innocente, **Felice Pantaleo**, Aurora Perego, Wahid Redjeb, Marco Rovere
CERN Experimental Physics Department

For the Patatrack R&D

Work has been sponsored by:



LHC

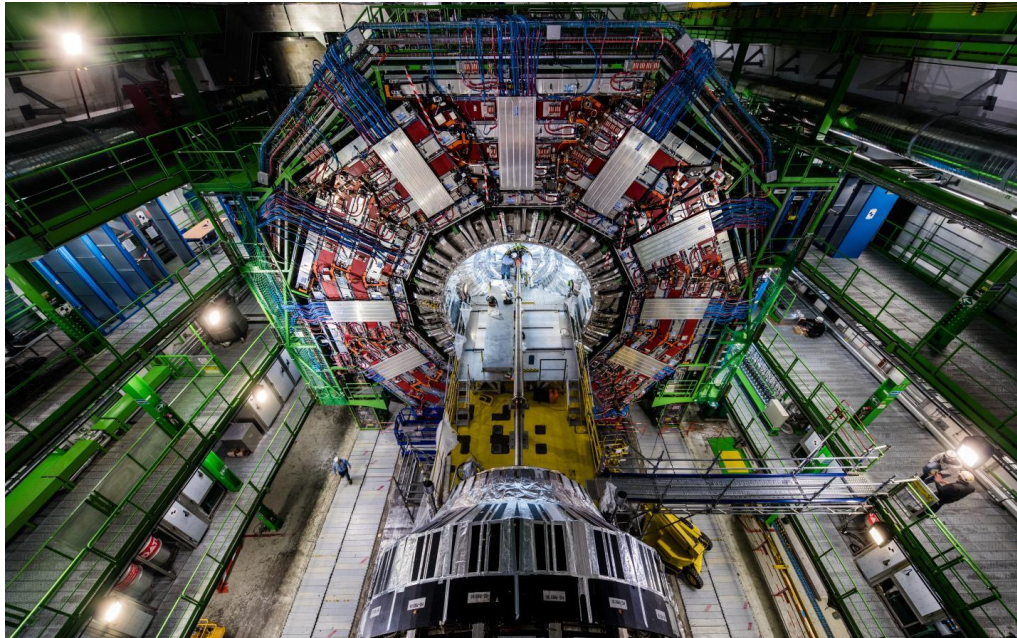




CMS Detector



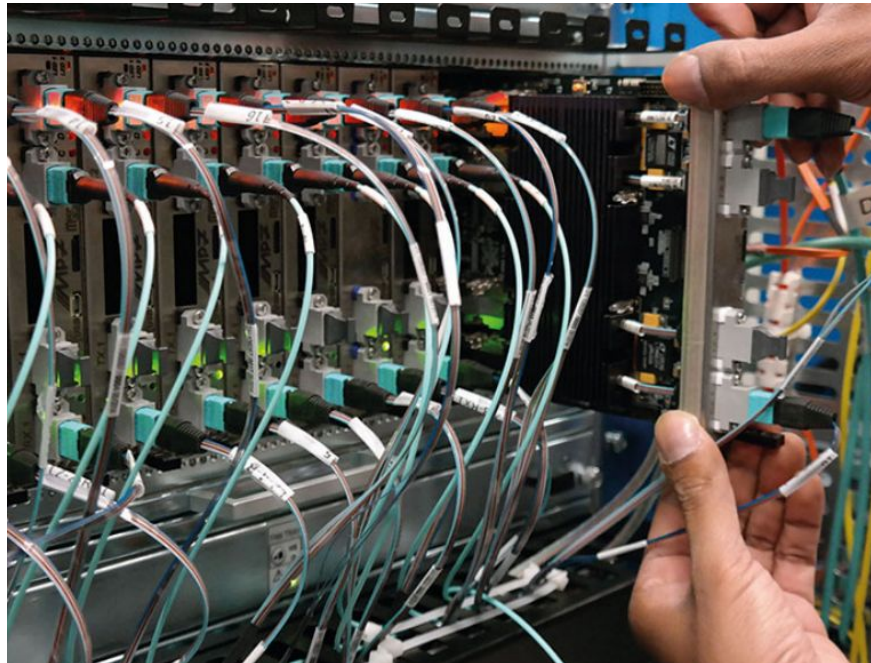
- Output 40MHz event rate
- Few MB per event



Level 1 Trigger

Custom electronics and FPGAs

Output Today: 100kHz



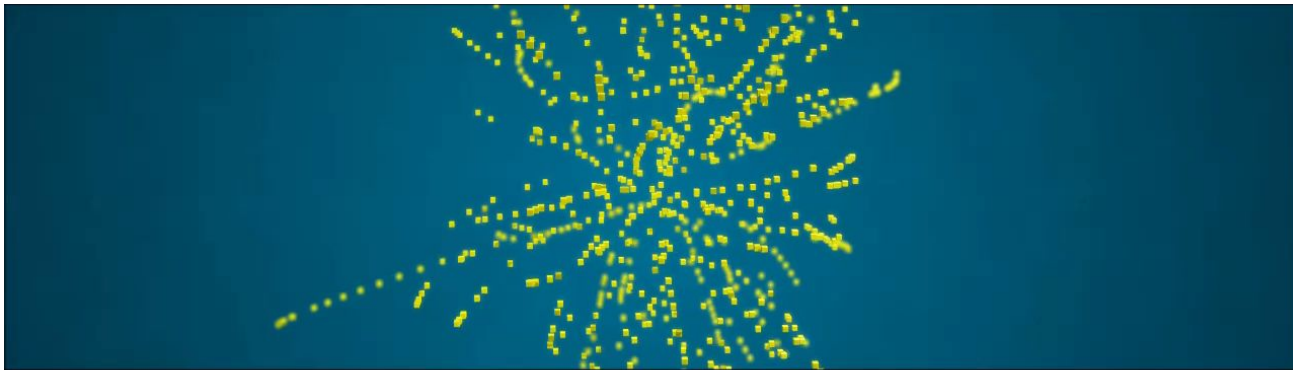
CMS High Level Trigger



Readout of the whole detector with full granularity, based on the CMS software, running on 200 nodes:

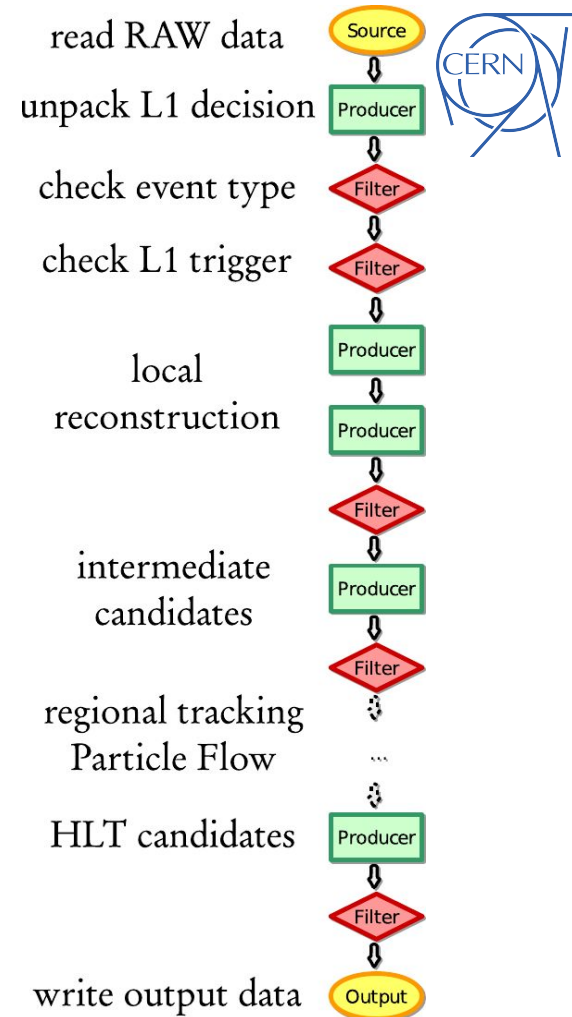
- ~26k CPU cores AMD Milan 7633
- 400 NVIDIA T4

Output event rate: few kHz



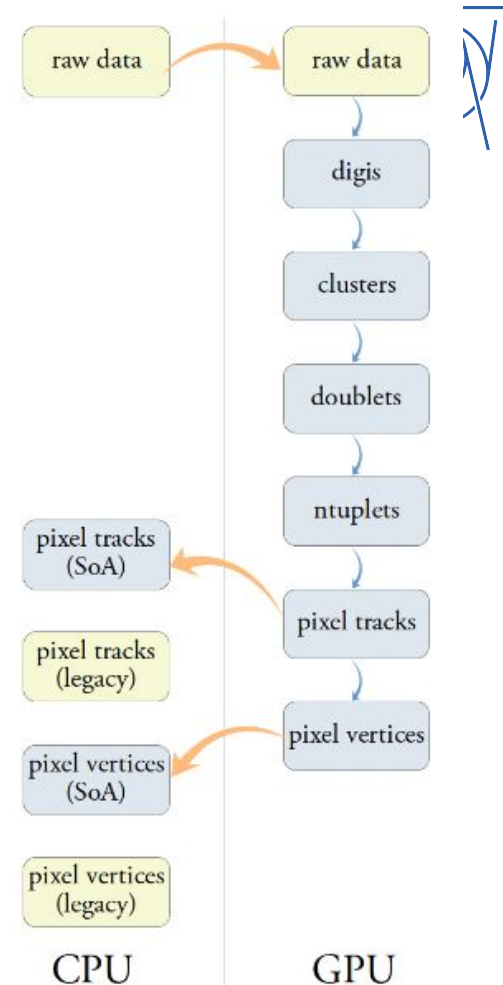
CMSSW

- Running the online and offline reconstruction
- Thousands of C++ modules configured via python.
- Scheduling based on runtime resolution of dependency graph (producers, consumers, filters, output...)
- Each instance of a producer defines one or more tbb tasks → tbb will assign a CPU thread to them
- Some producers can be executed asynchronously on GPUs



CMSSW

- Running the online and offline reconstruction
- Thousands of C++ modules configured via python.
- Scheduling based on runtime resolution of dependency graph (producers, consumers, filters, output...)
- Each instance of a producer defines one or more tbb tasks → tbb will assign a CPU thread to them
- Some producers can be executed asynchronously on GPUs



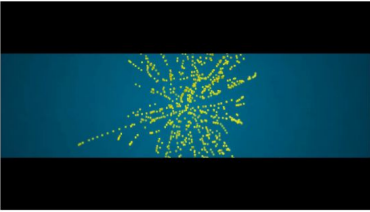
3 years ago in this workshop...



...CMS committed to offload 30% of the online reconstruction to GPUs for Run 3

Heterogeneous Run3 HLT Farm

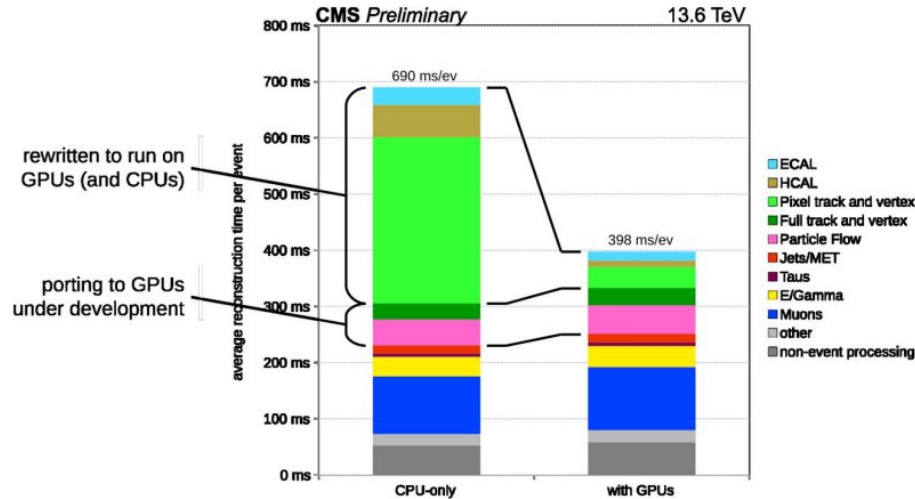
- We would like to exercise a Heterogeneous HLT farm together with a online/offline heterogeneous CMSSW well before Run4
- 30% of the HLT reconstruction algorithms seem like a good target
- What does CMS gain in the short term?
 - Better physics performance
 - Reconstruction able to run on Supercomputers
 - Expertise in Heterogeneous computing



3 years ago in this workshop...

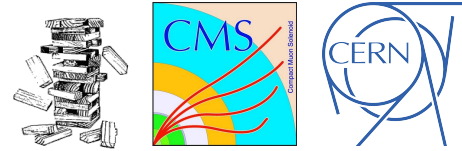


...CMS committed to offload 30% of the online reconstruction to GPUs for Run 3



We achieved more than 40%

Performance portability



Performance portability in code is becoming increasingly important:

- accelerators, such as GPUs, become ubiquitous in HPC/Data Centers
- accelerators are being integrated into Trigger farms for experiments such as the CMS experiment
- Redesigning algorithms and data structures to be well-suited for these accelerators can improve time-to-solution, energy-to-solution, and cost-to-solution.

- Maintaining and testing more than one codebase might not be the most sustainable solution in the medium/long term
- In the long term other accelerators might appear

Investigating sustainable ways to achieve performance portability, such as using programming models that abstract away hardware-specific details, is crucial.

This will enable the CMS experiment to make the most of all available computing resources and potential firepower, especially as the demands for computational power continue to grow.

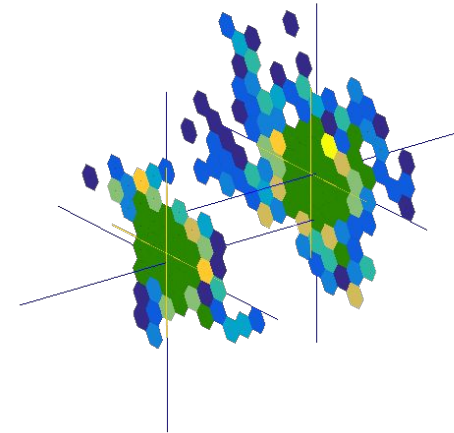
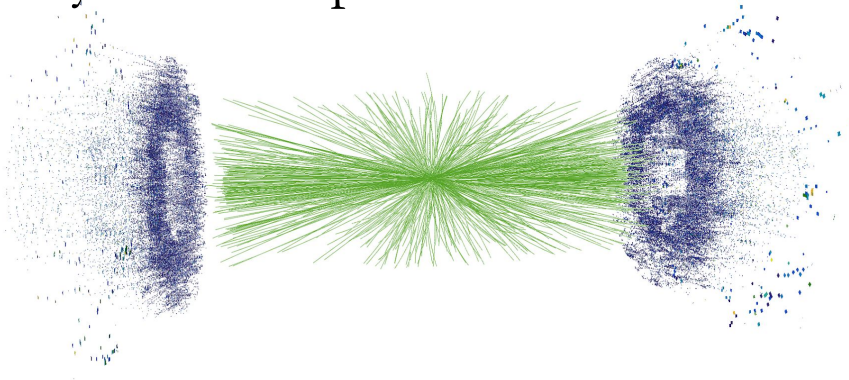
Exploring SYCL



Performance portability layers like SYCL and Alpaka have been tested using two workflows already implemented to CUDA:

- CLUE: CMS Phase-2 HGCal clustering algorithm
- Patatrack Pixel Reconstruction

Very different patterns wrt HPC codes



Methods - A success story

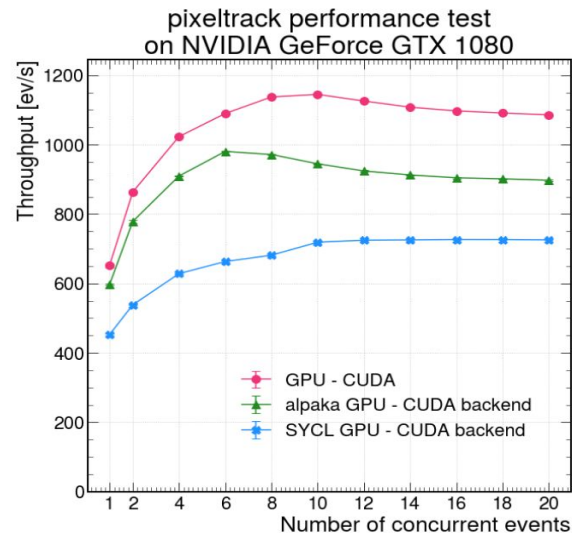
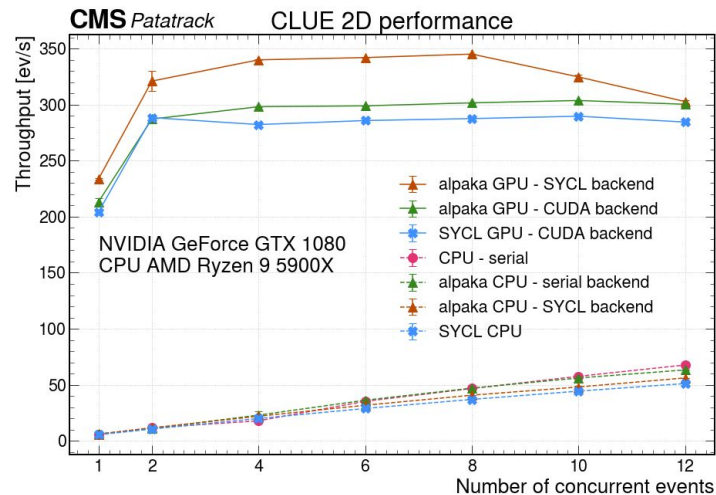


- [Ported the full Pixel and HGCal reconstruction from Alpaka to native SYCL](#)
- Implemented a SYCL-based Caching Allocator
- Implemented a SYCL backend for Alpaka
- Written a guide for converting [Alpaka to SYCL](#)
- Submitted a dozen bug reports/reproducers to OneAPI/clang
- Biweekly meetings with Intel experts with bidirectional feedbacks
- A week-long Patatrack SYCL Hackathon with daily interactions with openlab & Intel engineers



Results

- Performance Portability achieved for CLUE
NVIDIA GPU, AMD GPUs, CPU, Parallel CPU (TBB)
Intel GPUs and FPGAs emulator (SYCL backend)
- Default offline configuration (4 streams, 4 threads)
 - CLUE: ~15x throughput using GPUs
 - CLUE3D: ~10x throughput using GPUs
- Pixel Reconstruction much more complex than CLUE
 - Contains many atomic operations and group primitives that are for the moment slower than CUDA counterparts
 - More synchronizations in the code
 - Reproducers sent to Intel engineers
 - some already fixed!



Conclusion

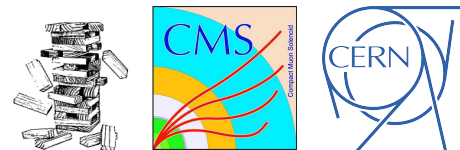


Heterogeneous computing provides the means to tackle the new computational challenges, but requires to write and maintain different source code for each device

- Established a successful collaboration with Intel for performance portability
- SYCL allows to write a single source code to be executed on multiple devices
- CMS has implemented support for SYCL for two expensive reconstruction algorithms and in the alpaka intermediate performance portability layer
 - This allows CMS to run on AMD, Intel, NVIDIA GPUs and possibly Intel FPGAs with a single code base
 - **CMS heterogeneous reconstruction will be completely portable by the end of 2023**
- Negligible loss of performance observed through SYCL with embarrassingly parallel problems like the CLUE clustering algorithm
- SYCL standard is frequently updated

The CMS Patatrack R&D will continue to work with CERN openlab and its partners to improve performance and its portability aiming an efficient data-taking at HL-LHC

Backup





Total execution time of CLUE on Intel Xeon Silver 4114 + NVIDIA A10

