High Energy Physics

Traineeships in Advanced Computing for High Energy Physics (TAC-HEP)

GPU programming module

Week 1 : Introduction to GPUs and heterogeneous computing

Lecture 2 - September 12th 2024

What we learnt in the previous lecture

- Hardware accelerators can be used in combination with CPUs to executing specific tasks more efficiently
- There are many types of hardware accelerators :
  - GPUs / FPGAs / ASICs / VPUs / TPUs etc.
  - The choice of hardware accelerator is based on the task at hand
- The GPU has thousands of cores and therefore can provide massive parallelization

Overview of today's lecture

- GPU vs CPU & GPU vs FPGA
- Heterogeneous computing
- The computing challenges in HEP
- GPU applications in HEP

Why GPUs?

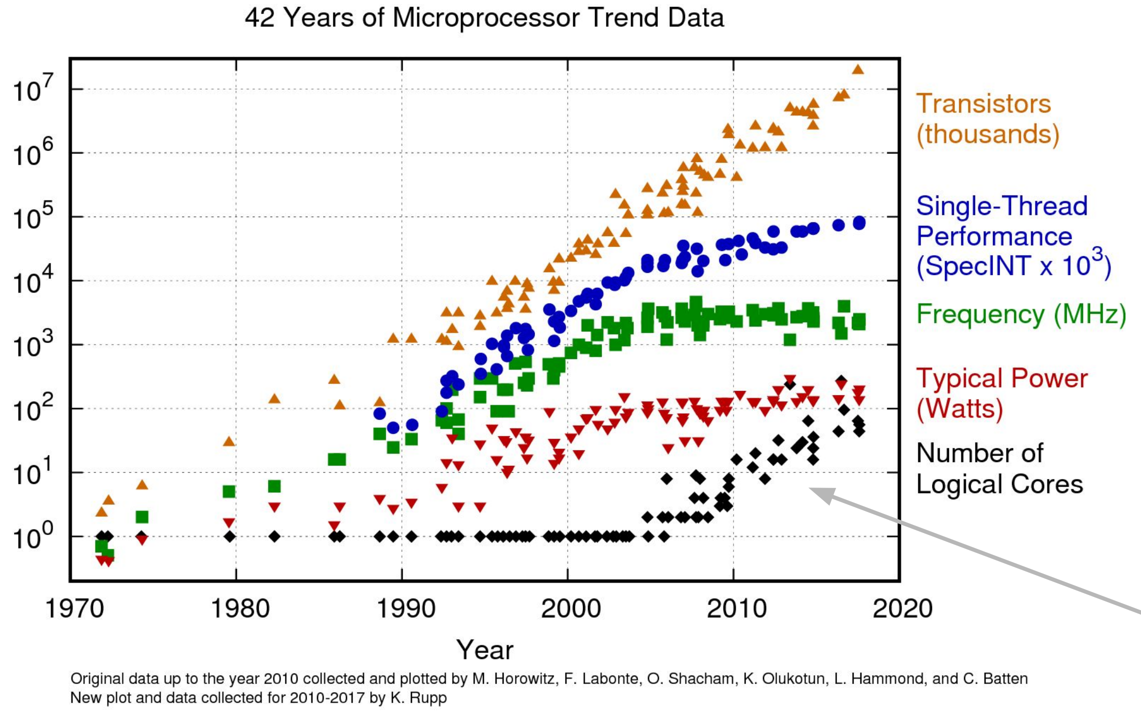


Image source [1]

- Moore's law states that the number of transistors in a dense IC doubles every ~2 years.
- Since ~2010 there seems to be a plateauing in single-thread performance
- Gains expected through exploiting parallelization

Number of cores has started to increase

Performance comparison of CPUs and GPUs (1)

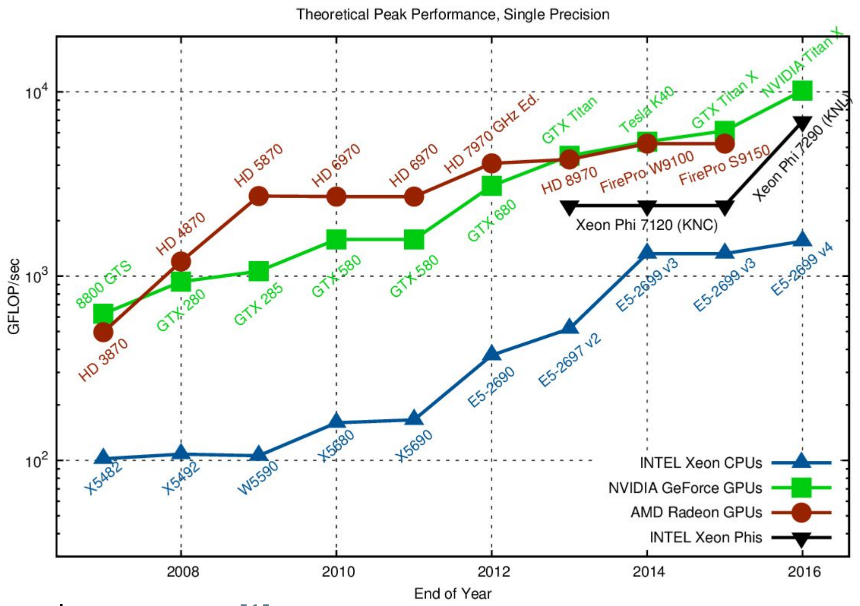


Image source: [1]

FLOPS : **Flo**ating **P**oint **O**perations per **S**econd :

- Measure of computing performance useful in fields that require floating point calculations (such as HEP)
- Can be measured for simple systems i.e. computers with exactly 1 CPU as well as complex systems such as **H**igh **P**erformance **C**omputing (HPC) centers :

$$\text{FLOPS} = \text{racks} \times \frac{\text{nodes}}{\text{rack}} \times \frac{\text{sockets}}{\text{node}} \times \frac{\text{cores}}{\text{socket}} \times \frac{\text{cycles}}{\text{second}} \times \frac{\text{FLOPs}}{\text{cycle}}$$

- Tesla T4 GPU Single Precision Performance (FP32) : 8.1 TFLOPS [2]

Performance comparison of CPUs and GPUs (2)

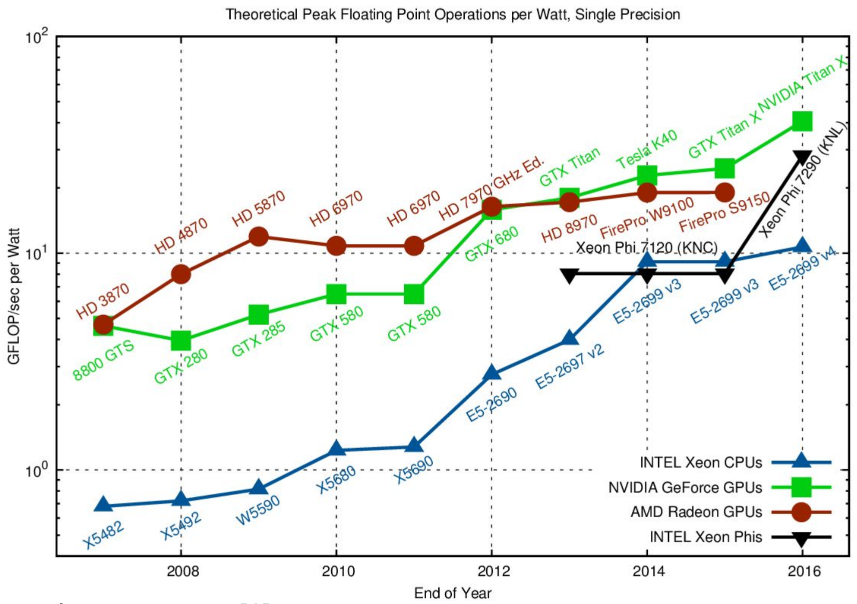


Image source: [1]

An important metric is the **FLOPS per Watt** :

- Rate of floating point operations performed per watt of energy consumed

Important metric since power consumption is a limiting factor today in both usage and manufacturing of hardware :

- Energy usage is an important constraint on the maximum computational capabilities that can be achieved
- Peak performance is constrained by the amount of power it can draw and the amount of heat it can dissipate

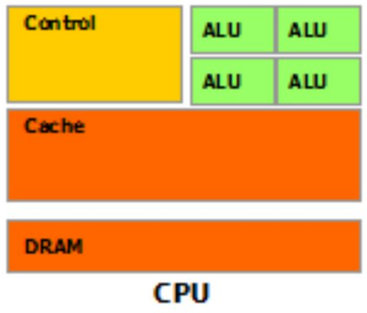
CPU vs GPU - overview of main differences

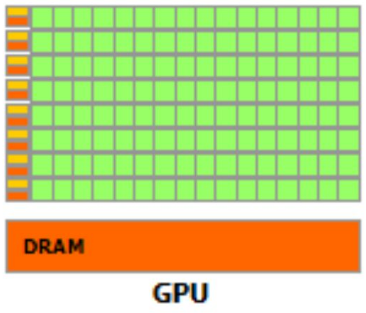
CPU

- ~O(10) powerful cores
- Low latency
- Serial processing
- Complex operations
- Higher clock speeds

GPUs

- ~O(1000) of less powerful cores
- High throughput
- Parallel processing
- Simple operations
- Better per-watt performance





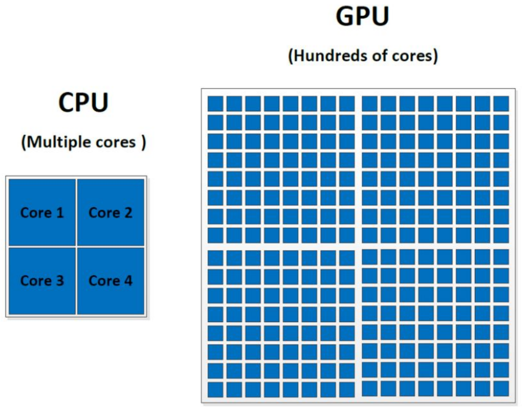


Image source : Nvidia

The Field Programmable Gate Array (FPGA)

- Semiconductor device that is based around a matrix of configurable logic blocks (CLBs)
    - Logic blocks consist of few logic cells
- CLBs connected via programmable interconnects
- Can be reprogrammed to desired application or functionality requirements after manufacturing
- Has a fixed latency
- Has its own I/O → Does not require a computer to run on
- Programmable using a Hardware Description Language (HDL)

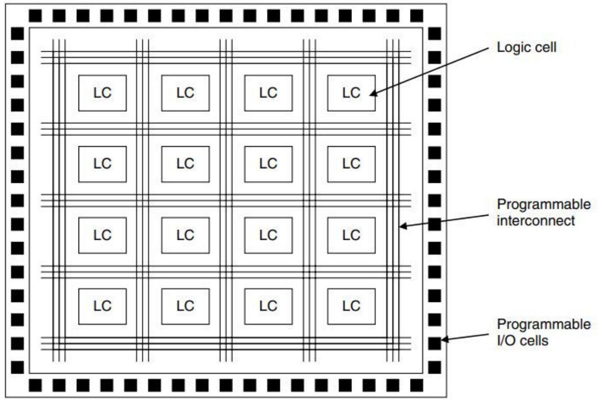


Image source [1]

CPU vs FPGA - Pros & Cons

FPGA

Pros

- Re-configurable circuitry
- Lower latency
- More power efficient
- Interconnect is not a bottleneck → high bandwidth

Cons

- Programmable using hardware description languages (VHDL etc.) → are more difficult to learn/use
- Usually not backward/forward compatible
- Larger cost

GPUs

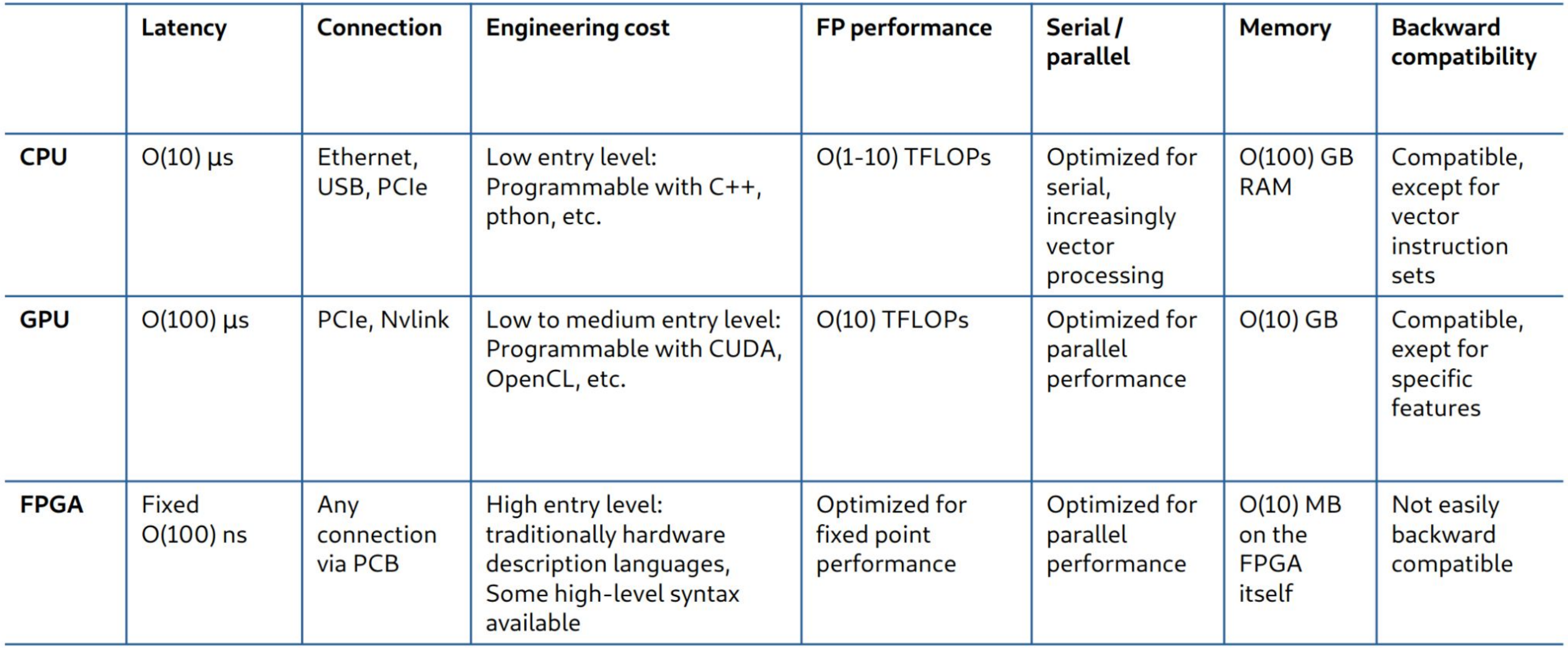
Cons

- Not reconfigurable
- Higher latency
- Less power efficiency
- Interconnect is NVLink or PCIe → data transferred can be a bottleneck

Pros

- Programmable using high level languages (CUDA, OpenCL, portability libraries etc.)
- Backward & forward compatibility
- Cheaper

CPUs vs GPUs vs FPGAs comparison

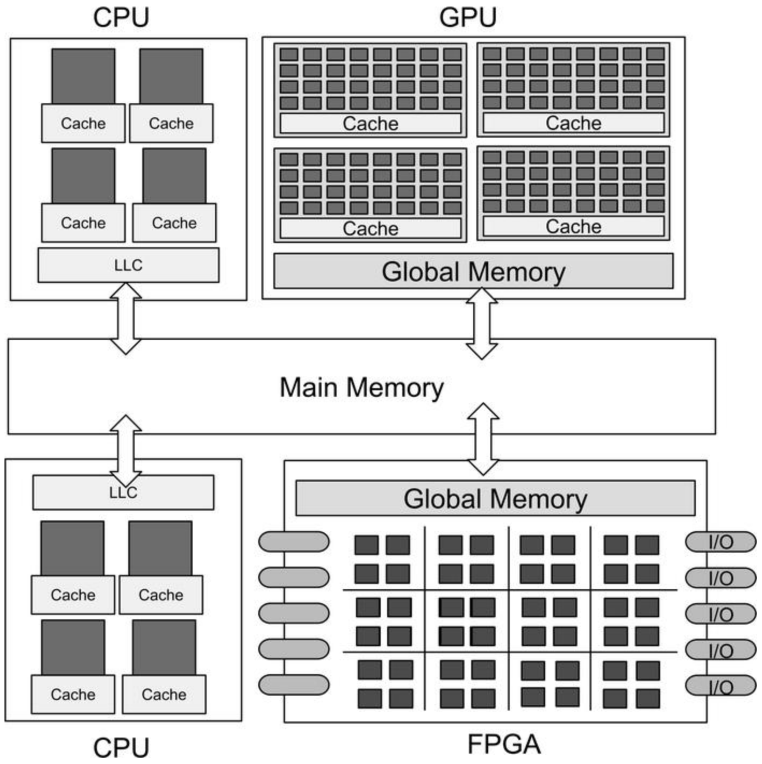


	Latency	Connection	Engineering cost	FP performance	Serial / parallel	Memory	Backward compatibility
CPU	O(10) µs	Ethernet, USB, PCIe	Low entry level: Programmable with C++, pthon, etc.	O(1-10) TFLOPs	Optimized for serial, increasingly vector processing	O(100) GB RAM	Compatible, except for vector instruction sets
GPU	O(100) µs	PCIe, Nvlink	Low to medium entry level: Programmable with CUDA, OpenCL, etc.	O(10) TFLOPs	Optimized for parallel performance	O(10) GB	Compatible, exept for specific features
FPGA	Fixed O(100) ns	Any connection via PCB	High entry level: traditionally hardware description languages, Some high-level syntax available	Optimized for fixed point performance	Optimized for parallel performance	O(10) MB on the FPGA itself	Not easily backward compatible

Source : https://arxiv.org/pdf/2003.11491.pdf

Heterogeneous computing

- Heterogeneous computing involves using multiple different types of processors to accomplish a task
- A heterogeneous system can consist of :
  - Different types of CPUs (i.e. combine compute powerful with less compute powerful but more power efficient CPU cores)
  - Hardware accelerators



Images source : https://www.routledgehandbooks.com/doi/10.1201/9780429399602-3

How are hardware accelerators used?

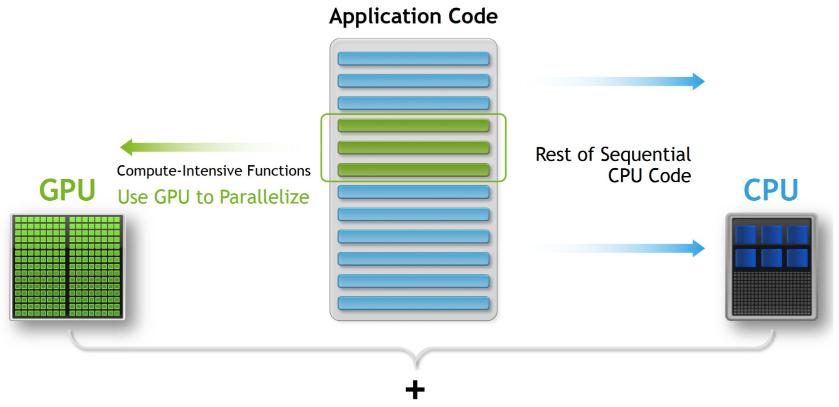


*Image source Nvidia

- In accelerated computing we take the compute intensive parts of the application code and parallelize that for execution on i.e. a GPU
  - Typically integer or floating point mathematical operations
- The remainder of the code (usually the vast majority) remains on the CPU
  - The part of code that remains on the CPU is ideally serial code
- Data between the the CPU and the accelerator has to be transferred:
  - This is performed via interconnect i.e. PCIe (*Peripheral Component Interconnect Express*), NVLink, Ethernet etc.

Challenges in heterogeneous computing

- **Data transmission**
  - Data has to be transferred from host (CPU) to device (GPU/accelerator)
  - Costly → Interconnect can be a bandwidth bottleneck

- **Portability**
  - HPC centers used by experiments will have different types of GPUs / different accelerators / different types of CPUs
  - Software should be portable and able to run on different platforms with minimal changes

- **Reproducibility**
  - Different computing sites/HPCs will have different architectures
  - Should determine how to manage cases where instruction sets produce results that are not bitwise reproducible

Challenges in heterogeneous computing

- **Data transmission**
  - Data has to be transferred from host (CPU) to device (GPU/accelerator)
  - Costly → Interconnect can be a bandwidth bottleneck

- Minimize data transfers
- Use accelerator friendly data structures

- **Portability**
  - HPC centers used by experiments will have different types of GPUs / different accelerators / different types of CPUs
  - Software should be portable and able to run on different platforms with minimal changes

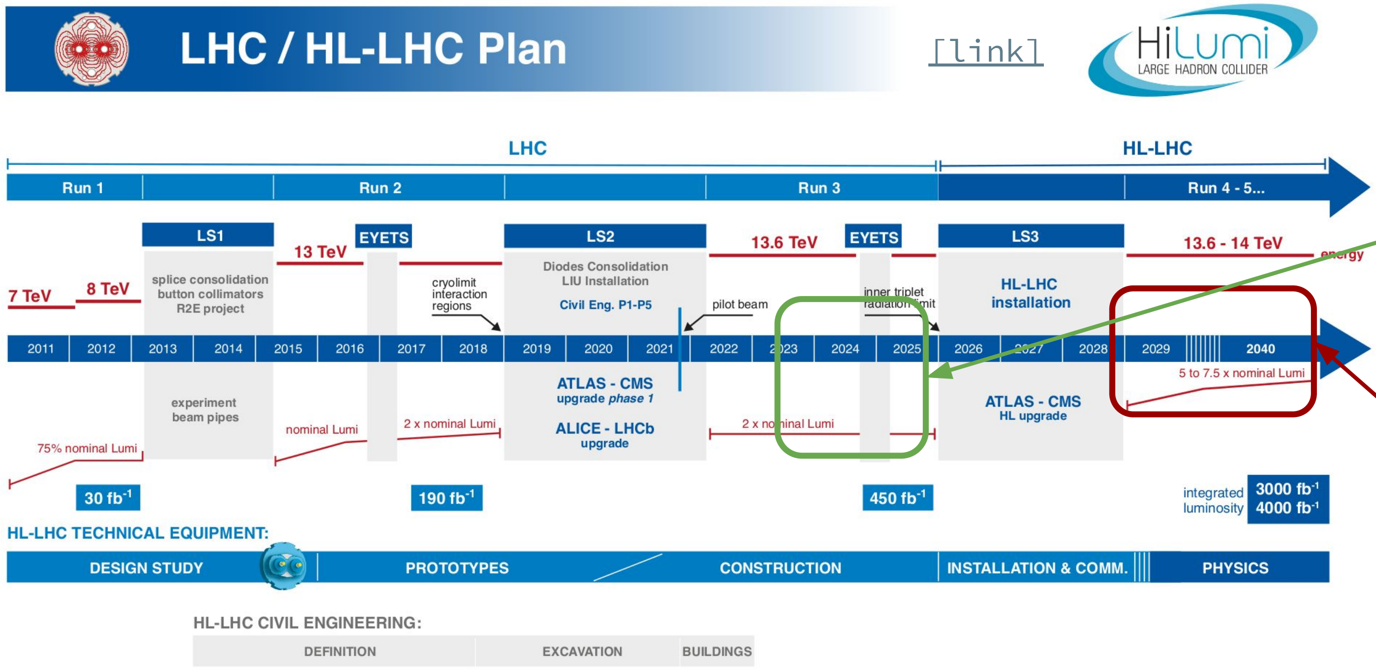
- Use of portability libraries

- **Reproducibility**
  - Different computing sites/HPCs will have different architectures
  - Should determine how to manage cases where instruction sets produce results that are not bitwise reproducible

- Consider the resolution requirements of task

Computing challenges in HEP

Computing at the HL-HLC (1)



We are here !

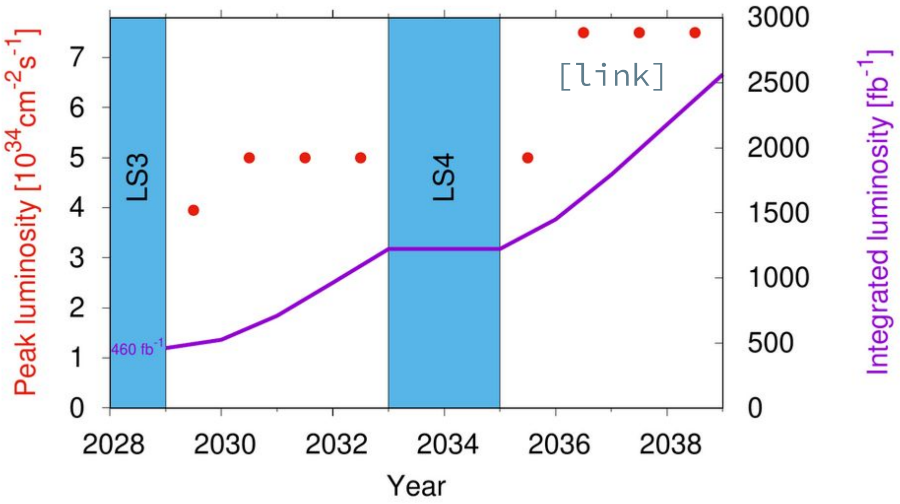
And plan to get here !

Computing at the HL-HLC (2)

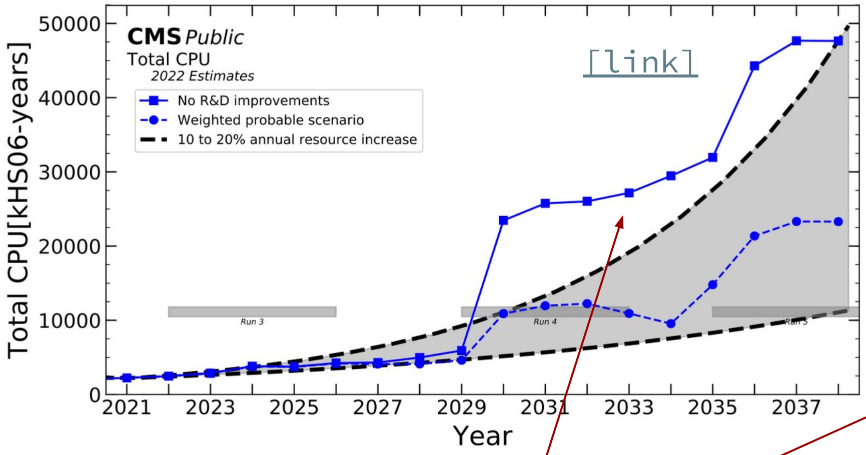
- During the **H**igh **L**uminosity **LHC** (HL-LHC) 2-3 times greater instantaneous luminosity compared to Run-2 (2016-2018)
- Up to ~200 pile-up interactions
- More events at a finer detector read-out granularity

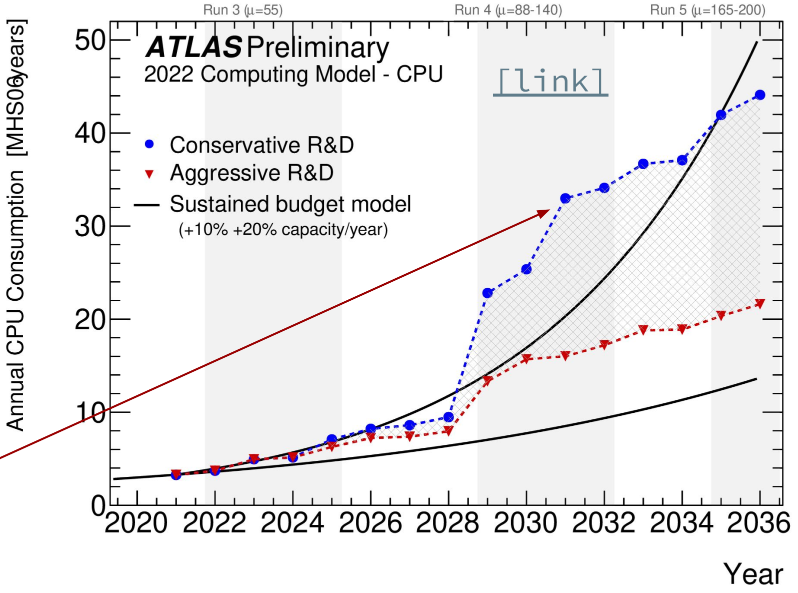
Required processing power larger by an order of magnitude with respect to what we have today!

- Can we keep up by only using conventional CPUs and homogeneous architectures???



Estimated CPU requirements for CMS & ATLAS



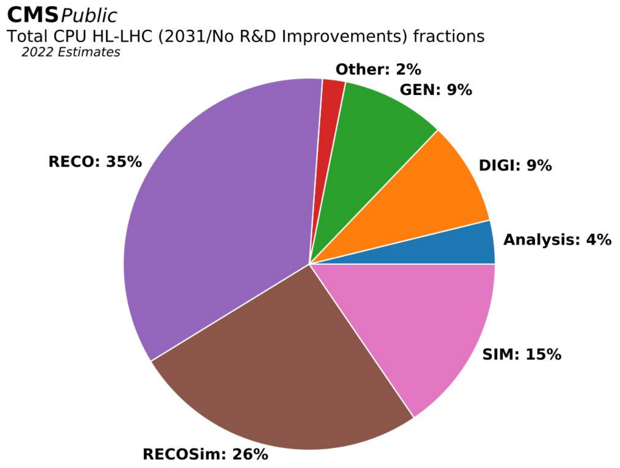


Without R&D and without taking advantage of the industry hardware evolution the we will not be able to cope with the computing demands !!

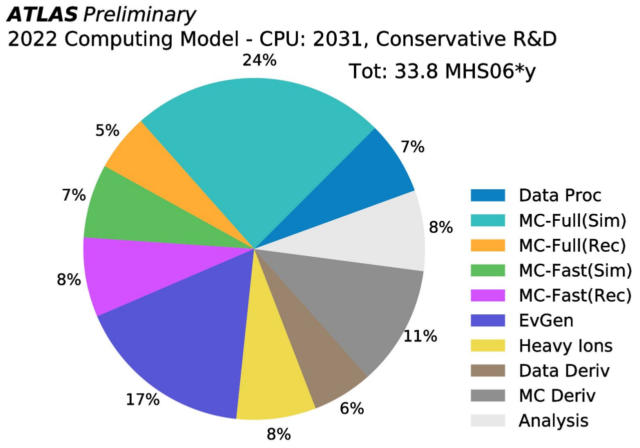
Computing needs in HEP

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

[link]



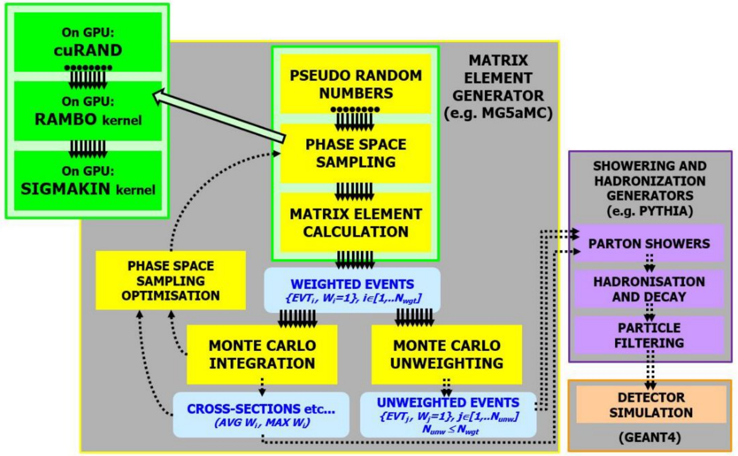
[link]



How can GPUs help?

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

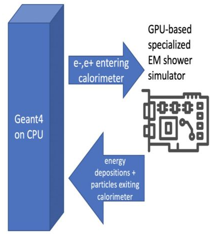
- **GPU enabled event generator i.e. Madgraph [1]**


On GPU: cuRAND
On GPU: RAMBO kernel
On GPU: SIGMAKIN kernel
PSEUDO RANDOM NUMBERS
PHASE SPACE SAMPLING
MATRIX ELEMENT CALCULATION
MATRIX ELEMENT GENERATOR (e.g. MG5aMC)
PHASE SPACE SAMPLING OPTIMISATION
WEIGHTED EVENTS {EVT_i, W_i=1}, i∈[1,..N_wgt]
MONTE CARLO INTEGRATION
MONTE CARLO UNWEIGHTING
CROSS-SECTIONS etc... (AVG W_i, MAX W_i)
UNWEIGHTED EVENTS {EVT_j, W_j=1}, j∈[1,..N_unw] N_unw ≤ N_wgt
SHOWERING AND HADRONIZATION GENERATORS (e.g. PYTHIA)
PARTON SHOWERS
HADRONISATION AND DECAY
PARTICLE FILTERING
DETECTOR SIMULATION
(GEANT4)

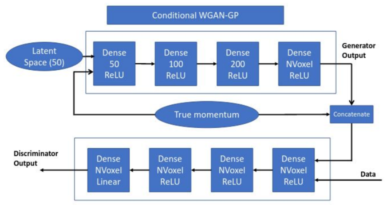
How can GPUs help?

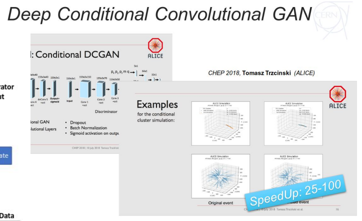
- Event generation
- Simulation
- Event reconstruction
- Event post-processing
- Data analysis

- **GPU based Geant4 application (i.e. AdePT) [1]**


Geant4 on CPU
e-,e+ entering calorimeter
GPU-based specialized EM shower simulator
energy depositions + particles exiting calorimeter

- **AI/ML enabled Fast Simulation (i.e. AltFast3 in ATLAS [1] , DC-GAN in ALICE [2])**




Deep Conditional Convolutional GAN
Conditional DCGAN
ALICE
CHEP 2018, Tomasz Trzcinski (ALICE)
Examples
for the conditional cluster simulation:
Original event
SpeedUp: 25-100

How can GPUs help?

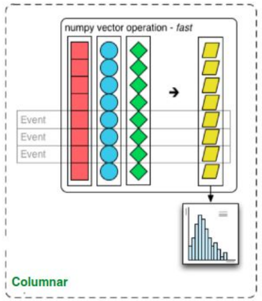
- Event generation
- Simulation
- Event reconstruction
- Event post-processing
- Data analysis

- **Track reconstruction, primary vertex reconstruction, raw data unpacking, clustering etc.**
- **Various efforts in different experiments (Patatrack track reconstruction, Allen project, ALICE TPC track reconstruction etc.)**

How can GPUs help?

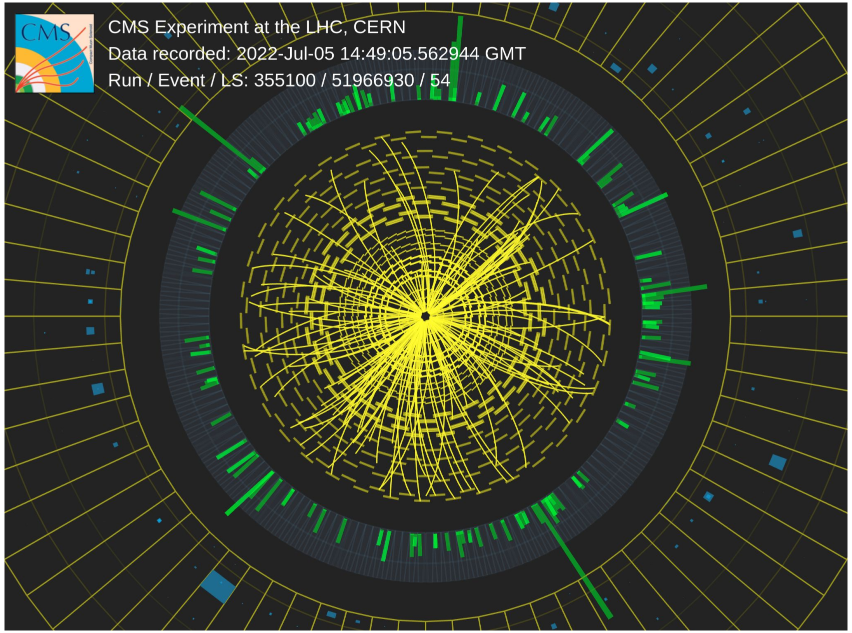
- Event generation
- Simulation
- Event reconstruction
- Event post-processing
- Data analysis

- **Training and inference of ML models**
- **Perform HEP analysis using the columinar analysis paradigm tools (i.e. coffea [1])**



Some of the current applications of GPUs in HEP

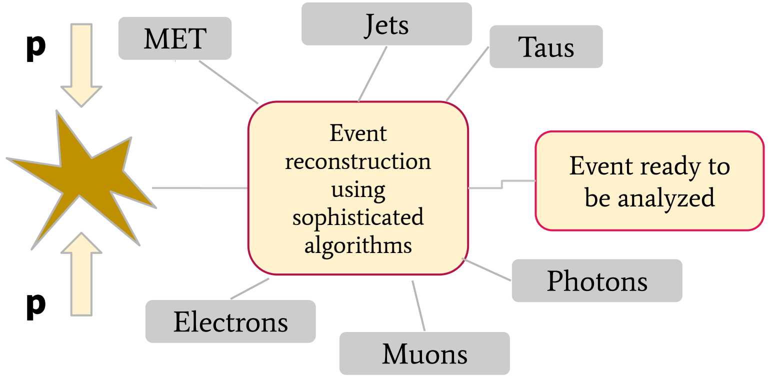
CMS (**C**ompact **M**uon **S**olenoid)

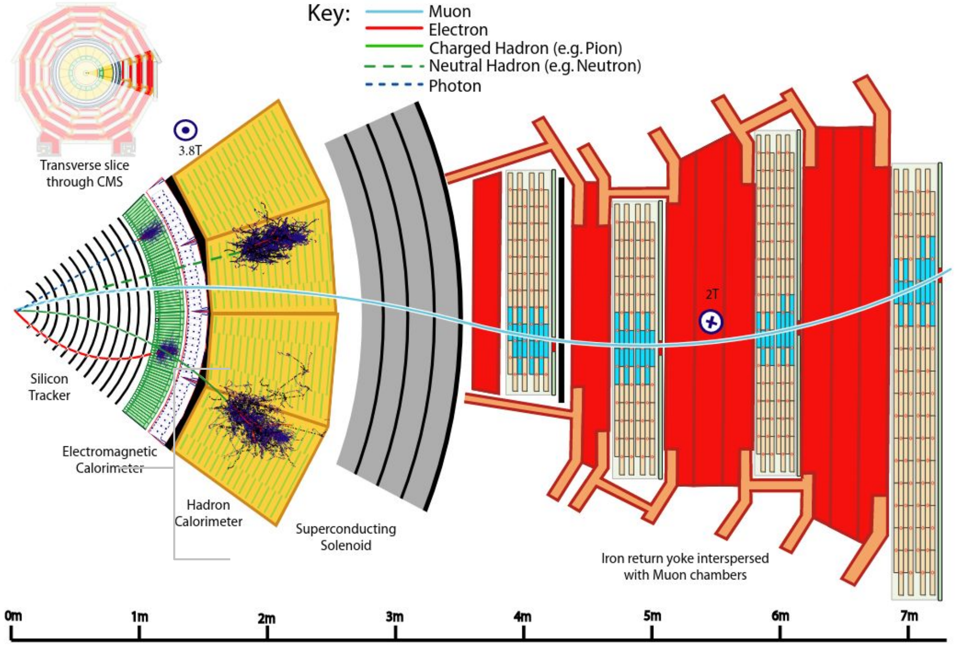

CMS Experiment at the LHC, CERN
Data recorded: 2022-Jul-05 14:49:05.562944 GMT
Run / Event / LS: 355100 / 51966930 / 54

- One of the four large experiments at the LHC

- General-purpose detector designed to measure properties of the Standard Model and observe new physics phenomena that the LHC might reveal

CMS Event reconstruction


p
MET
Jets
Taus
Event reconstruction using sophisticated algorithms
Event ready to be analyzed
p
Photons
Electrons
Muons


Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
Transverse slice through CMS
3.8T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0m
1m
2m
3m
4m
5m
6m
7m

Online reconstruction at the **High Level Trigger (HLT)** :

- Fast
- Runs on computing farm@CERN

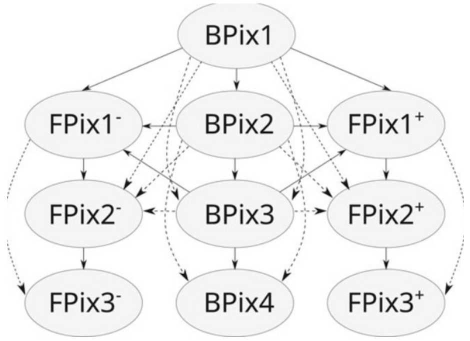
Offline reconstruction :

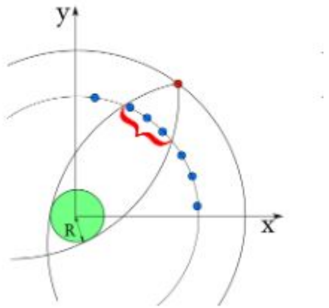
- Can be more sophisticated
- Runs on computing centers worldwide

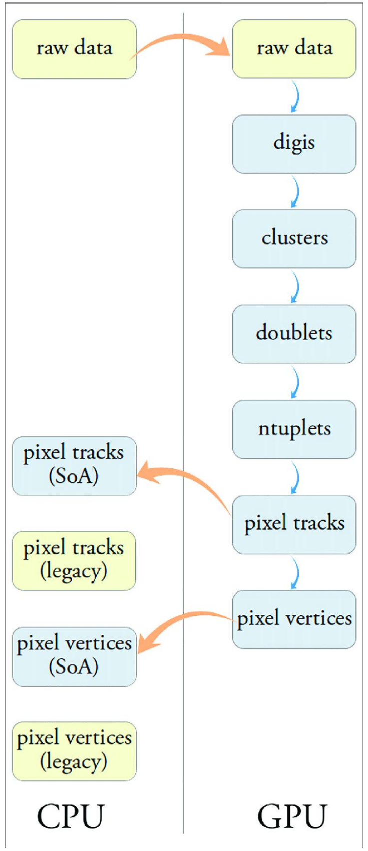
CMS - Patatrack track reconstruction

What is done on the GPU :

- Raw data decoding
- Clustering of nearby active pixels (Hits)
- Linking of doublets of hits on different layers
- Pattern recognition linking doublets segments (Cellular Automaton)
- Fitting of the found n-tuplets (Pixel Tracks)
- Clustering of Pixel Tracks (Vertices)
- Pixel RAW data for each event is transferred to the GPU initially (~250kB/event)

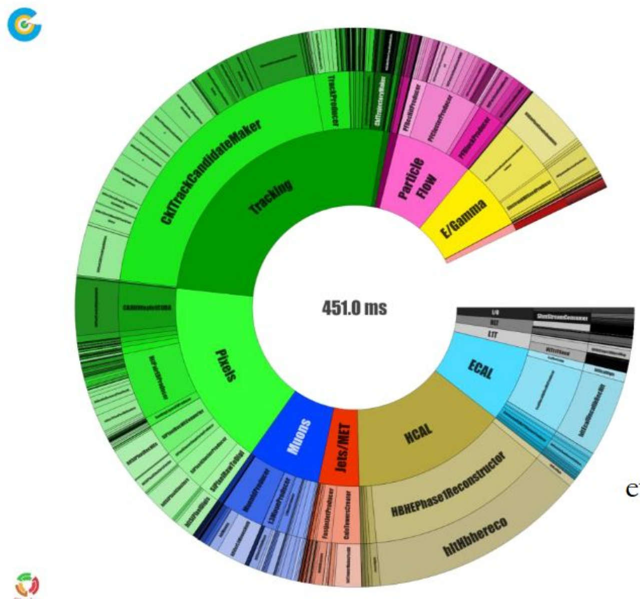






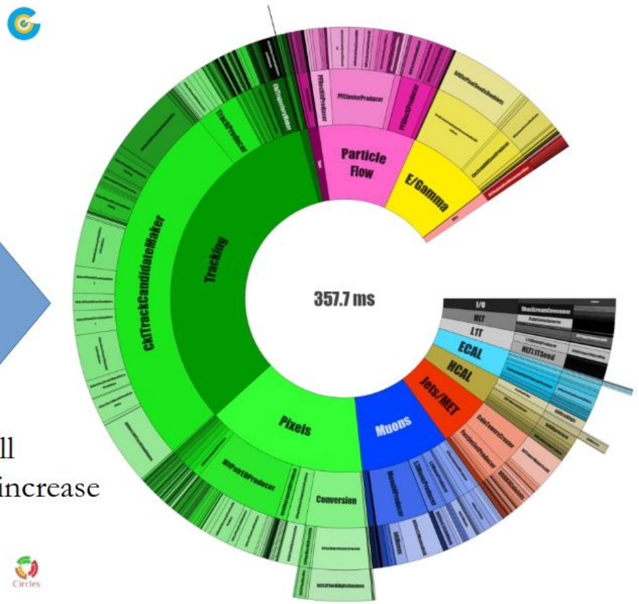
Sources [1], [2]

CMS - Patatrack track reconstruction performance



CPU Only Reconstruction

+26% overall event throughput increase

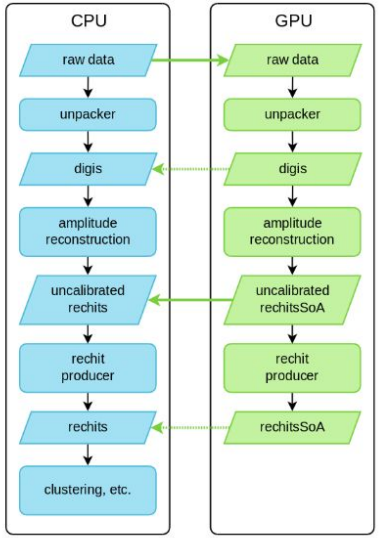


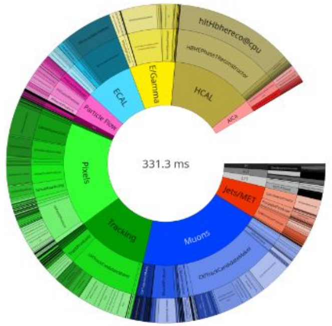
Heterogeneous Reconstruction

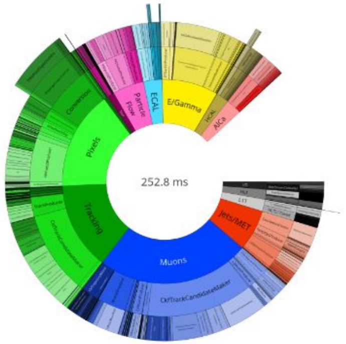
Sources [1], [2]

CMS - ECAL & HCAL local reconstruction

- Electromagnetic and Hadronic Calorimeter (ECAL/HCAL) local reconstruction at the HLT stage is performed on the GPU
- This includes :
  - Unpacking of the raw data from the detector into consecutive samples read out from a single ECAL channel.
  - Amplitude reconstruction.
  - Calculation of energies from the uncalibrated reconstructed hits

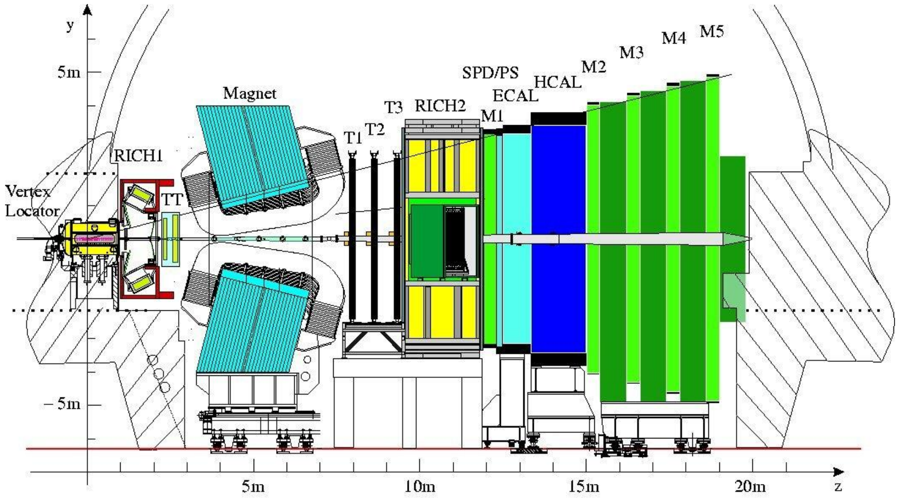






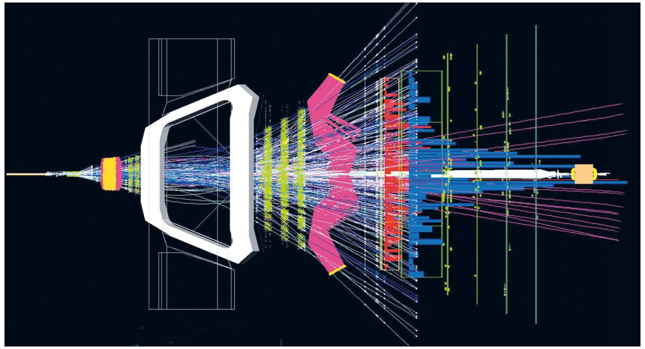
https://cds.cern.ch/record/2802591/files/CR2022_029.pdf

LHCb (**L**arge **H**adron **C**ollider **b**eauty)



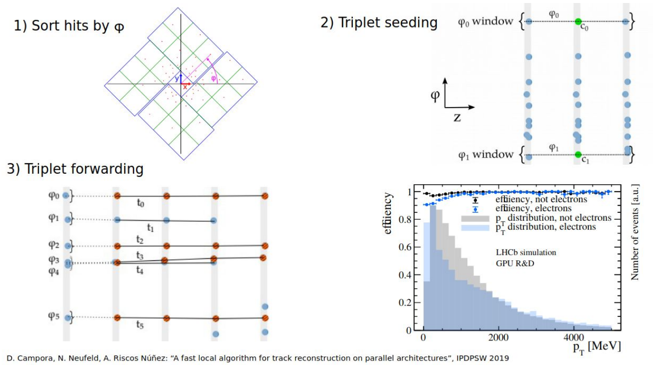
http://cds.cern.ch/record/5701

- One of the four large experiments at the LHC
- Series of sub-detectors dedicated to detect mainly forward particles
- Aim to measure small differences between matter and antimatter by studying properties of the b-quark

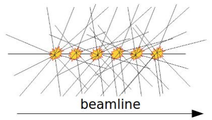


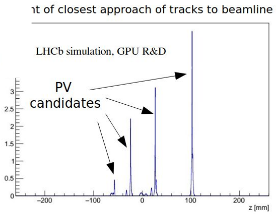
Allen : An HLT on GPUs for LHCb

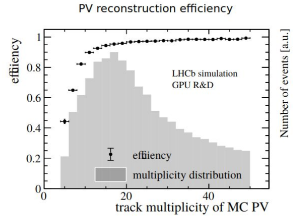
- Primary vertex reconstruction
- Reconstruct charged particle trajectories
- Several thousand events can be processed in parallel on the GPU

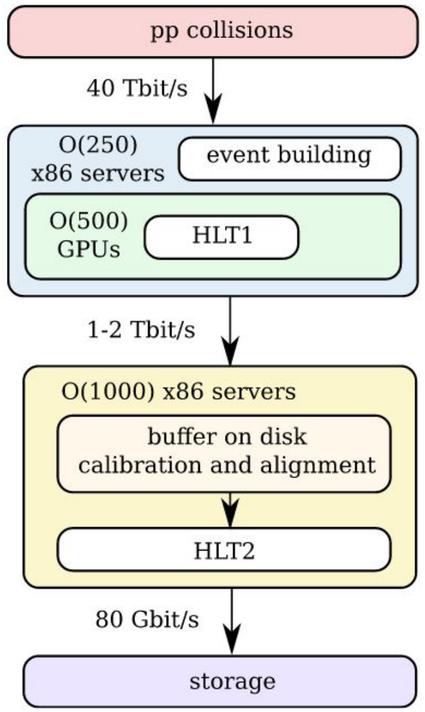


D. Campora, N. Neufeld, A. Riscos Núñez: "A fast local algorithm for track reconstruction on parallel architectures", IPDPSW 2019





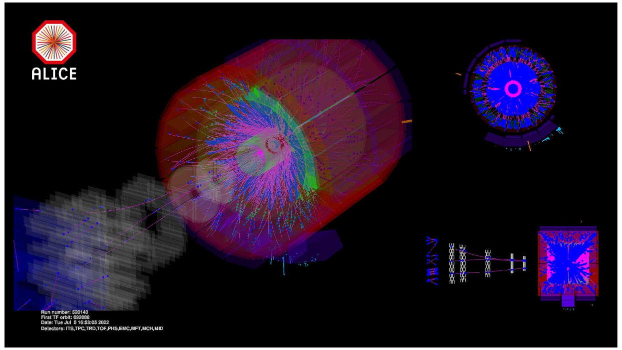


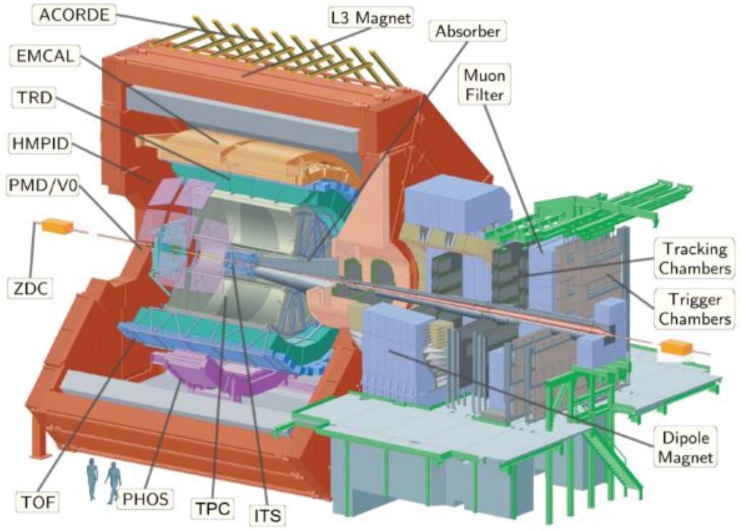


Source:
https://d-nb.info/1213584876/34
https://cds.cern.ch/record/2699553/files/vom_Bruch_Allen_chep2019%2004.11.pdf

ALICE (**A L**arge **I**on **C**ollider **E**xperiment)

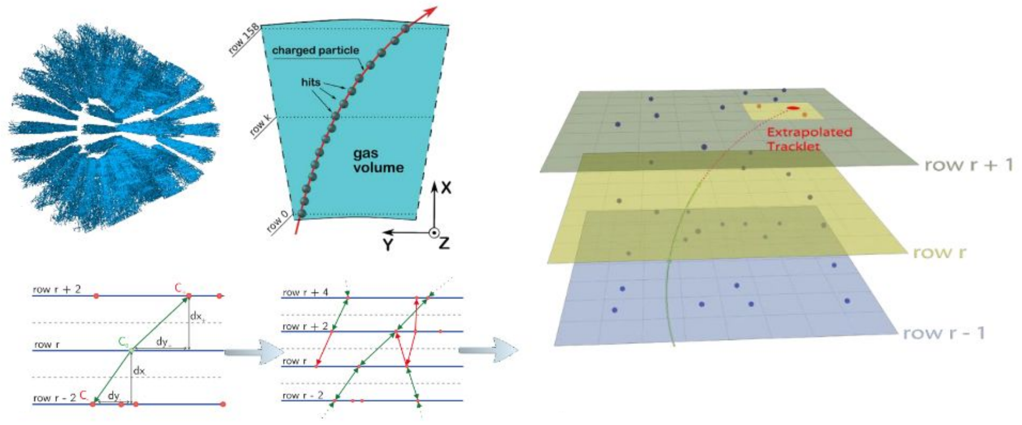
- One of the four large experiments at the LHC
- Detector dedicated to heavy-ion physics
- Designed to study the physics of quark-gluon plasma
- Trying to recreate conditions similar to those right after the Big Bang
- Main detector: large **time projection chamber (TPC)**


ALICE
Run number: 520143
First TF orbit: 692888
Date: Tue Jul 5 16:53:05 2022
Detectors: ITS,TPC,TRD,TOF,PHS,EMC,MFT,MCH,MID



ALICE - TPC track reconstruction with GPUs

- Most compute-intensive part is the reconstruction of particle trajectories in the TPC.
- The HLT uses a GPU-accelerated algorithm for TPC tracking that is based on the Cellular Automaton principle and on the Kalman filter.
- In operation since Run-1



- Several events and different sectors of the TPC are able to run in parallel

https://arxiv.org/pdf/1712.09430.pdf

Wrapping-up

Overview of today's lecture

- GPUs can provide more FLOPS/watt that CPUs :
    - They do not replace CPUs but are used in combination to maximize performance when the task at hand is parallelizable
- Heterogeneous computing involves using multiple different types of processors to accomplish a task
- The next decades will pose a significant computing challenge for HEP experiments, especially for the LHC experiments in light of HL-LHC
- Many HEP experiments are already exploring the use of accelerators and heterogeneous computing

Next week

- We will brush up our knowledge of C++
  - Core syntax and types
  - Operators
  - Arrays, Pointers & References
  - Control instructions (if/switch,for/while loops etc.)
  - Compound data types (structs, enums etc.)
  - Functions
  - Scopes / namespaces
  - Object Orientation (objects and classes/ constructors & destructors/ inheritance)

BACK-UP