Hardware accelerators in HEP

Computational HEP Traineeship Summer School

Charis Kleio Koraka

Wednesday July 26th 2023

Overview

- Hardware accelerators & heterogeneous computing

- Applications of hardware accelerators in HEP

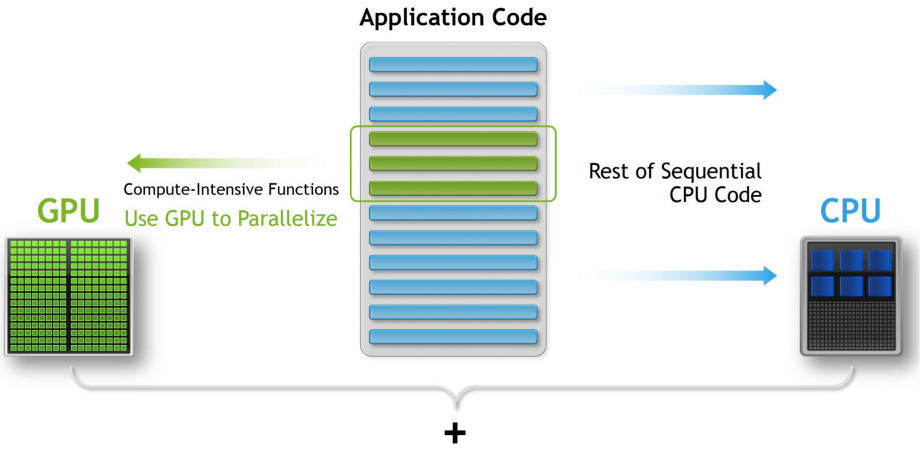
What are hardware accelerators ?

- Devices built for executing specific tasks more efficiently compared to running on the standard computing architecture of a CPU

- Come in many flavors :
  - GPUs / FPGAs / TPUs …

- Are a part of our everyday lives :
  - Encryption, video stream decoding, 3D graphics acceleration, pattern/object recognition, machine learning, AI and many more

How are hardware accelerators used?



- In accelerated computing we parallelize the compute intensive parts of the application code :
  - Typically integer or floating-point mathematical operations

- The remainder of the code (usually the vast majority) remains on the CPU
  - This is ideally serial code

Image source [i]

What about heterogeneous computing ?

- Heterogeneous computing involves using multiple different types of processors to accomplish a task

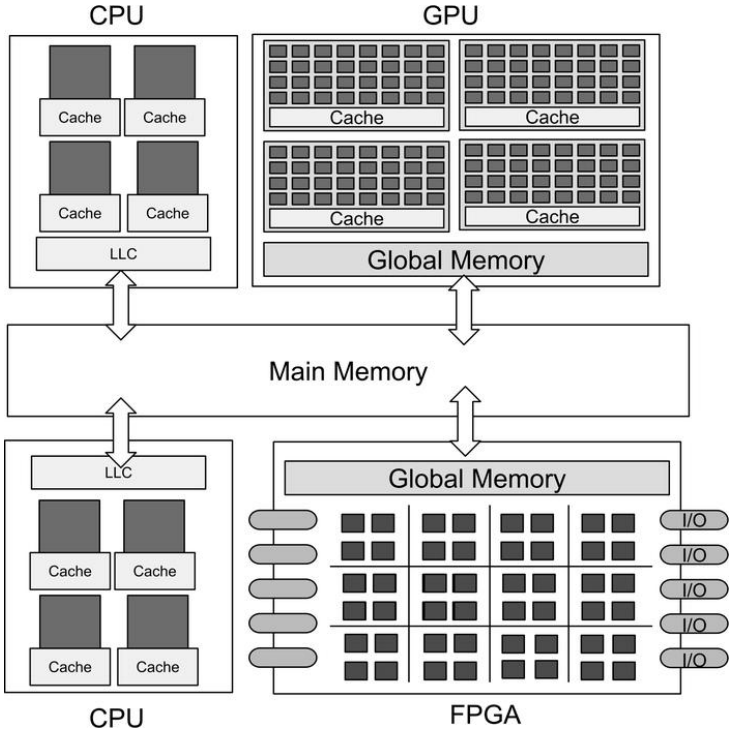


Image source [i]

What about heterogeneous computing ?

- The heterogeneous system can consist of :
  - Different types of CPUs (i.e. combine compute powerful with less compute powerful but more power efficient CPU cores)

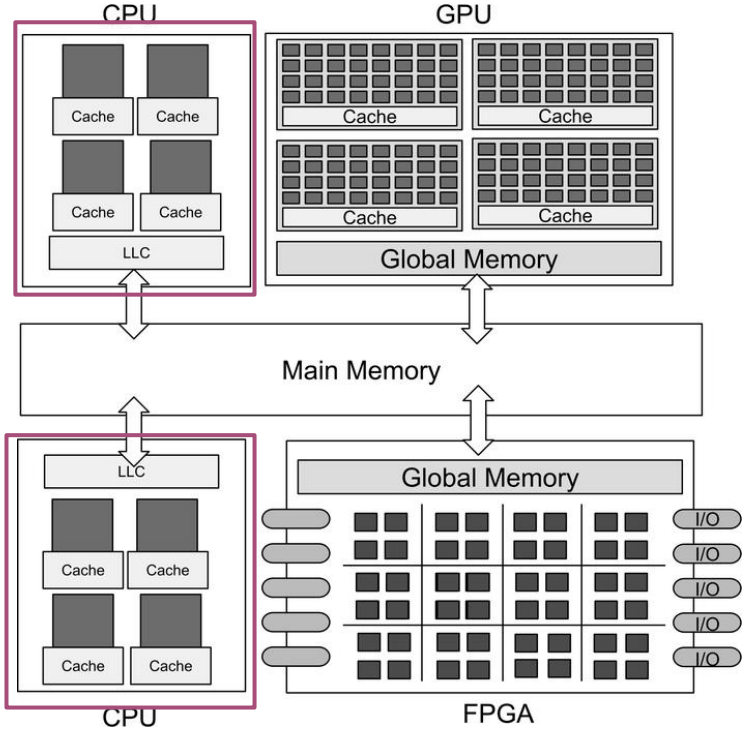


Image source [i]

What about heterogeneous computing ?

- The heterogeneous system can consist of :
  - Different types of CPUs (i.e. combine compute powerful with less compute powerful but more power efficient CPU cores)
  - Different types of hardware accelerators

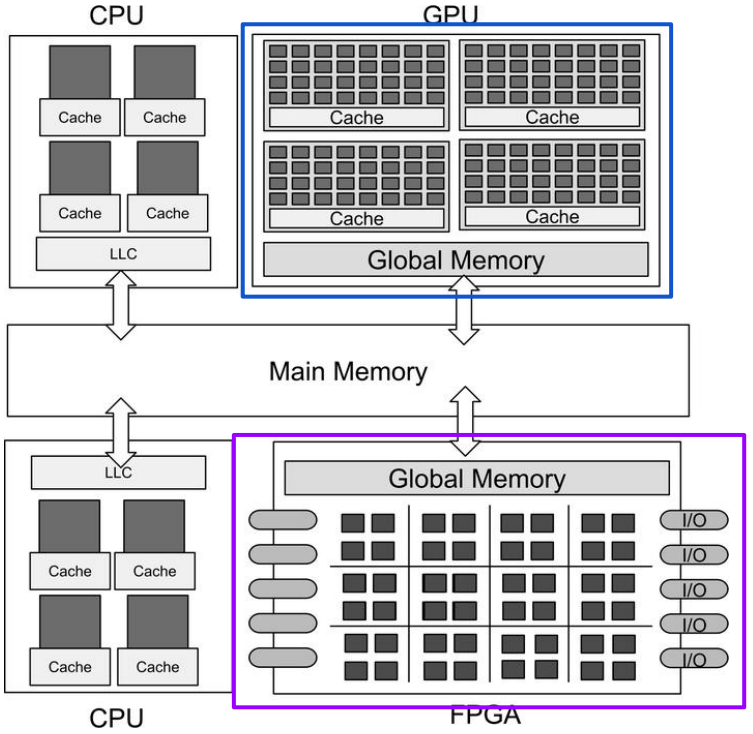


Image source [i]

What about heterogeneous computing ?

- The heterogeneous system can consist of :
  - Different types of CPUs (i.e. combine compute powerful with less compute powerful but more power efficient CPU cores)
  - Different types of hardware accelerators

- **Requires development of code can run on more than one platform concurrently**

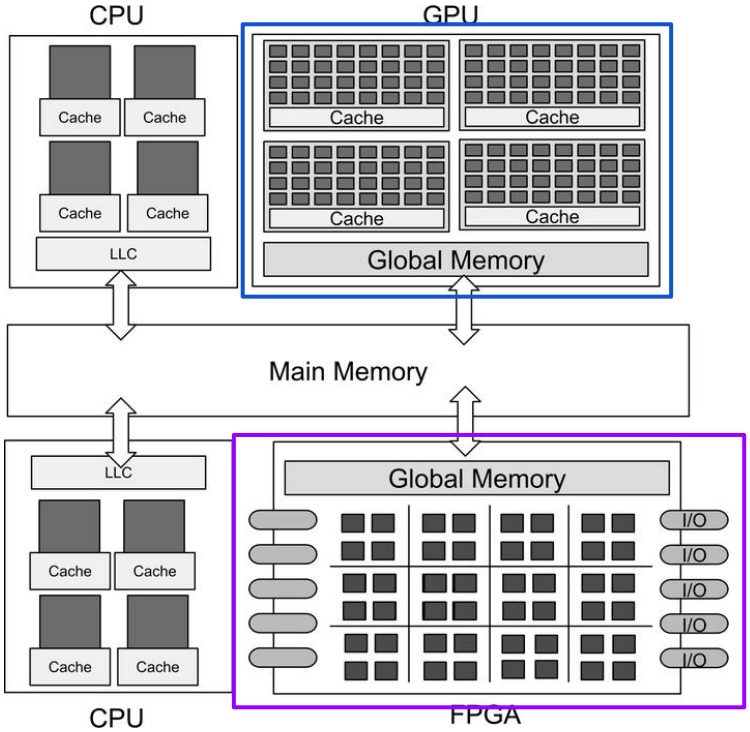


Image source [i]

Some hardware accelerators

The Graphic Processing Unit (GPU)

- Silicon based micro-processor that contain cores, registers, memory, and other components.
- Many-core processor (vs single/multi-core processor)
- Follows the Single Instruction Multiple Threads (SIMT) execution model
- GPU acceleration emphasizes on high data throughput and massive parallel computing
- Programmable using relatively easy APIs (CUDA, openCL, python etc.)

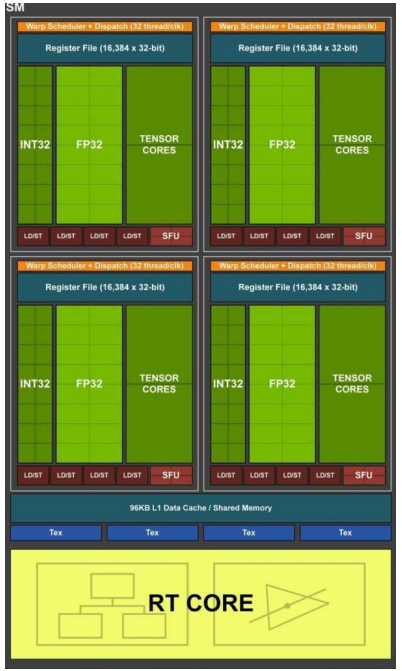


Image sources [i]

The Field Programmable Gate Array (FPGA)

- Semiconductor device that is based around a matrix of configurable logic blocks (CLBs)
  - 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)

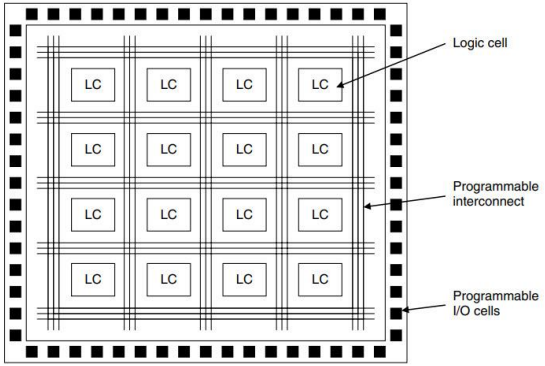


Image source [1],[2]

CPU vs FPGA - Pros & Cons

FPGA

Pros

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

GPUs

Cons

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

CPU vs FPGA - Pros & Cons

FPGA

Cons

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

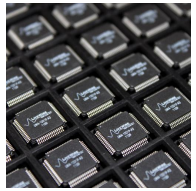
GPUs

Pros

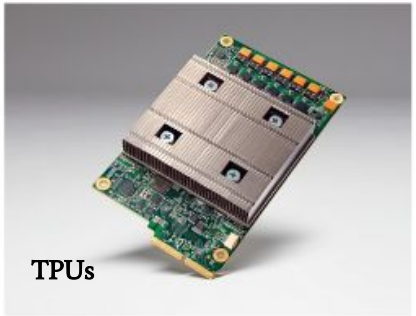
- Programmable using high level languages (CUDA, OpenCL, portability libraries etc.)
- Backward & forward compatibility
- Cheaper

Other examples of hardware accelerators

- **ASIC** (Application-Specific Integrated Circuit)
  - IC chip customized for a particular use
  - i.e. lower precision and/or optimised memory usage to maximize throughput
- **TPU** (Tensor Processing Unit)
  - Optimised to perform matrix-multiplication operations / used in e.g. NN and RF training
- **VPU** (Vision Processing Unit)
  - Used to accelerate machine vision algorithms, i.e. CNNs , AI etc.



ASIC



TPUs

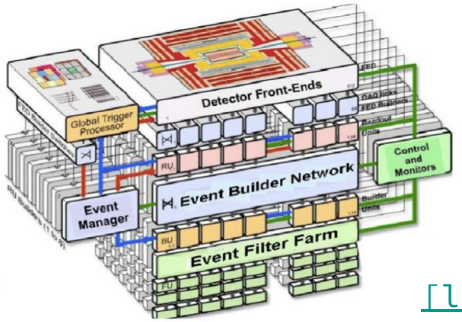
VPUs

Image sources [i] , [ii], [iii]

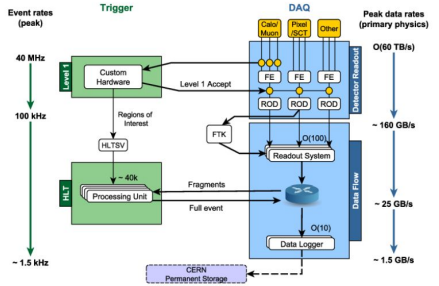
Applications of hardware accelerators in HEP

The computing needs of HEP experiments

- **Real-time data processing**

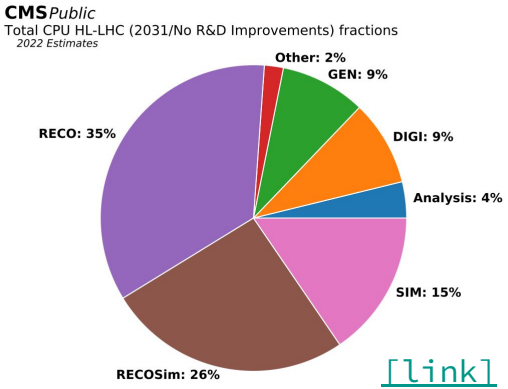


[link]

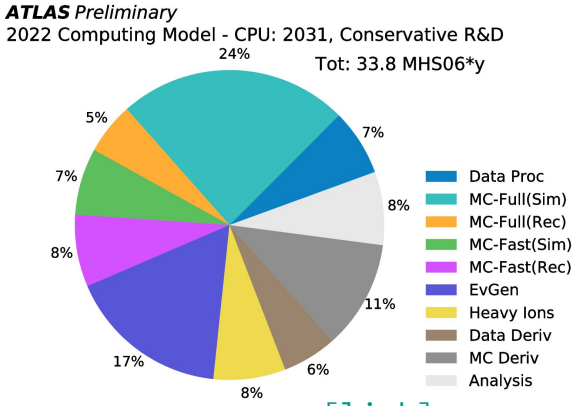


[link]

- **Offline data processing**



[link]

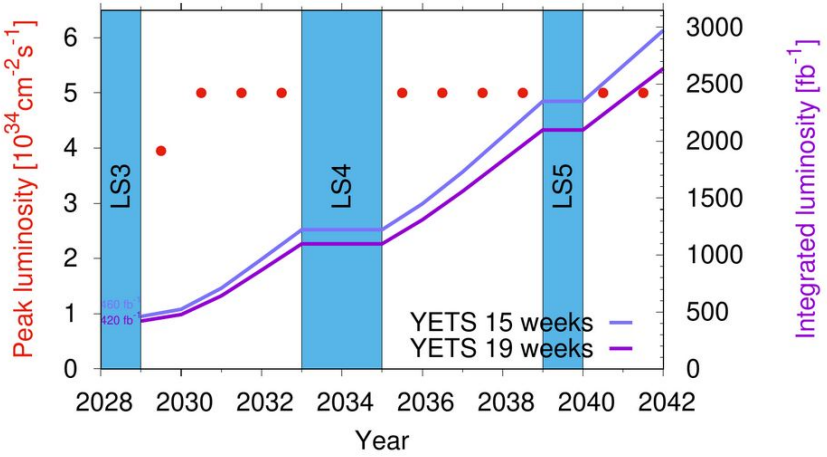


[link]

Hardware accelerators in HEP

- Hardware accelerators and heterogeneous computing already used in HEP e.g. :
  - ATLAS and CMS experiments have been using FPGAs for their Level-1 trigger
  - ALICE experiment has been using a heterogeneous system of CPUs and GPUs for its software trigger

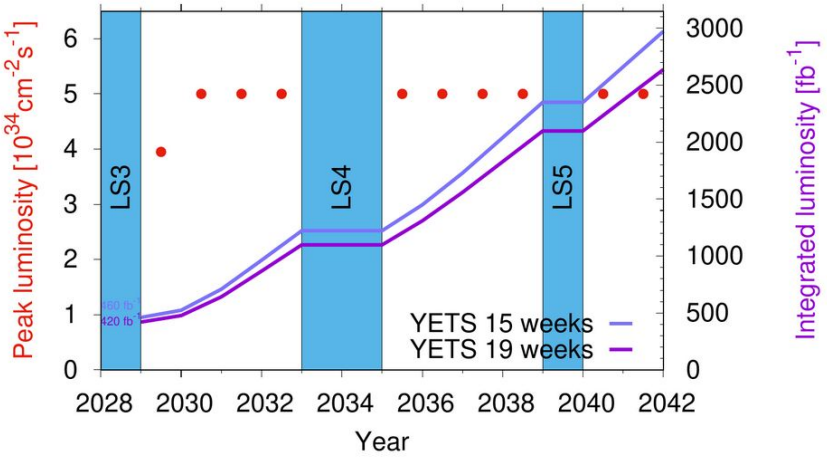
- Significant computing challenge ahead for HEP experiments :
  - Higher luminosities and PU
  - Complex detectors



Hardware accelerators in HEP

- Hardware accelerators and heterogeneous computing already used in HEP e.g. :
  - ATLAS and CMS experiments have been using FPGAs for their Level-1 trigger
  - ALICE experiment has been using a heterogeneous system of CPUs and GPUs for its software trigger

- Significant computing challenge ahead for HEP experiments :
  - Higher luminosities and PU
  - Complex detectors

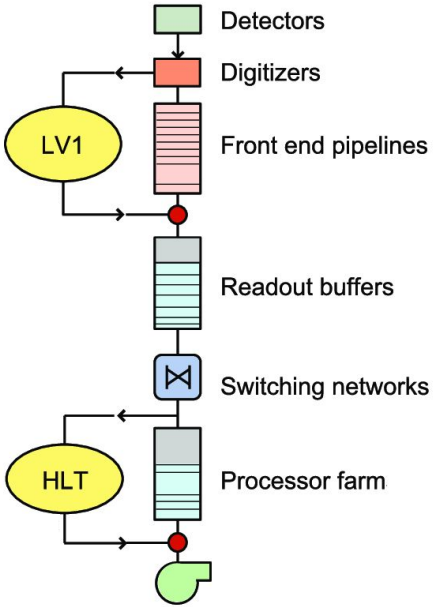


A lot of R&D to take advantage of hardware accelerators!
Only a few will be presented today!

Accelerators for real time data processing

Real time data processing can happen in two steps :

- Hardware level trigger (Level-1 Trigger)
  - Decision is based on local objects
  - Latency requirements are strict
  - Bandwidth is high

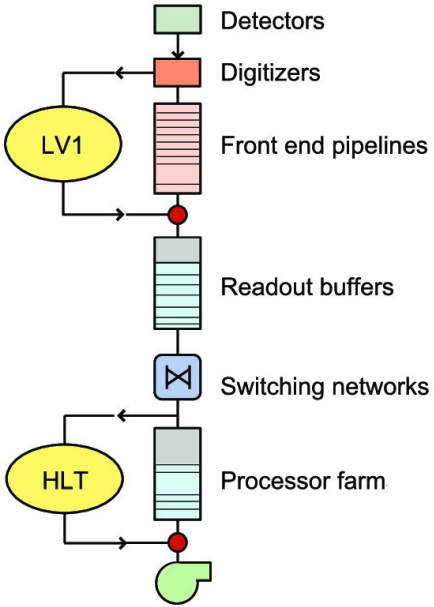


Accelerators for real time data processing

Real time data processing can happen in two steps :

- Hardware level trigger (Level-1 Trigger)
  - Decision is based on local objects
  - Latency requirements are strict
  - Bandwidth is high

- Software level trigger (High Level Trigger)
  - Necessary when information from several detectors sub-systems should be combined
  - Looser latency requirements
  - Requires the whole event stream to be read out
    - In experiments with a large data stream this can be achieved by reducing the bandwidth with a hardware level trigger



Accelerators for real time data processing

Real time data processing can happen in two steps :

- Hardware level trigger (Level-1 Trigger)
  - Decision is based on local objects
  - Latency requirements are strict
  - Bandwidth is high

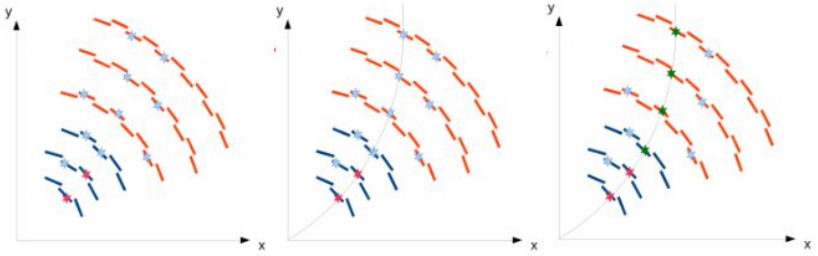
Ideal for FPGAs

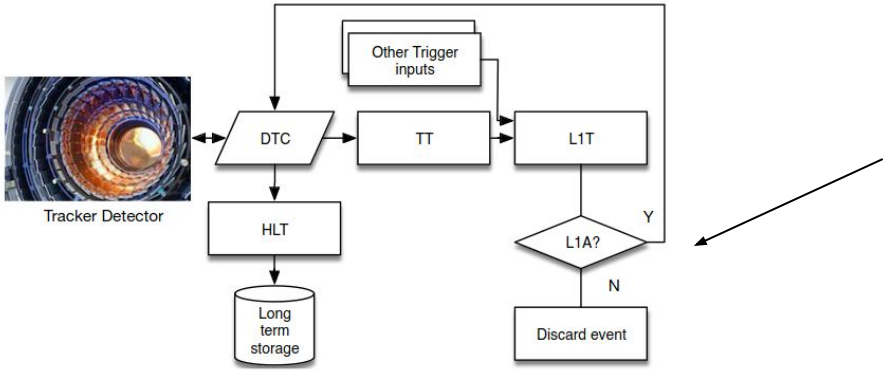
- Software level trigger (High Level Trigger)
  - Necessary when information from several detectors sub-systems should be combined
  - Looser latency requirements
  - Requires the whole event stream to be read out
    - In experiments with a large data stream this can be achieved by reducing the bandwidth with a hardware level trigger

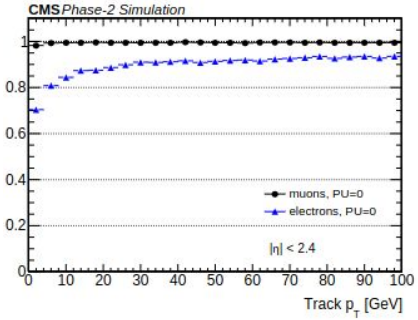
Ideal for heterogeneous systems

FPGA-based tracking for the CMS Level-1 trigger

- Pattern recognition and charged particle trajectory reconstruction using an all-FPGA solution.
  - pairs of stubs combined to seeds
  - a helical trajectory formed
  - linearized χ2 fit to calculate trajectory parameters



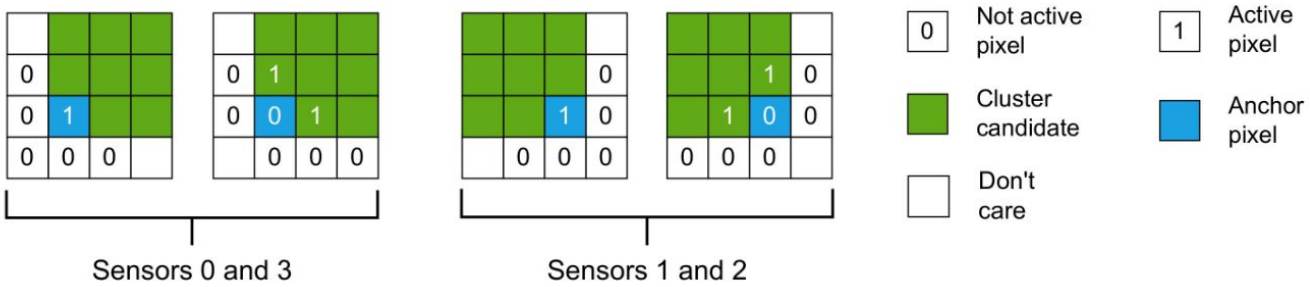




https://arxiv.org/pdf/1910.09970.pdf

FPGA-based cluster finding for the LHCb silicon pixel detector

- Currently in use at Run-3
- In a fully parallel way, each pixel checks if it belongs to one of the following patterns, if so a cluster candidate is identified
- Each cluster candidate is resolved using a LUT

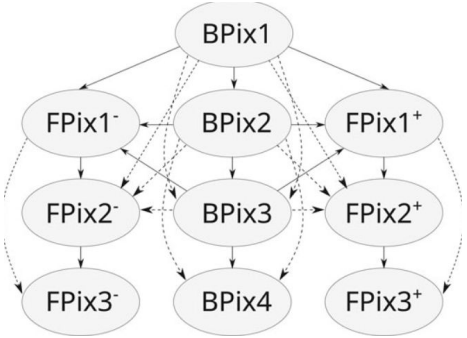


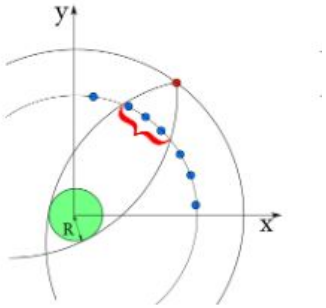
- FPGA-based cluster finding lead to :
  - 11% increase in HLT1 event rate
  - 14% reduction of the required bandwidth
  - O(50x) reduction in power consumption

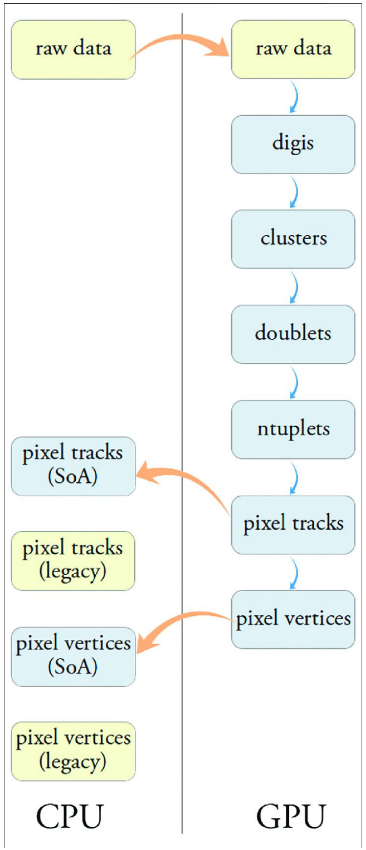
https://indico.jlab.org/event/459/contributions/11812/attachments/9206/14093/CHEP2023-clustering-final.pdf

Patatrack track reconstruction in CMS

- Used in Run-3 for tracking @HLT
- Pixel RAW data for each event is initially transferred to the GPU (~250kB/event)
- Sequential steps run on the GPU to minimize data transfers
  - 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)



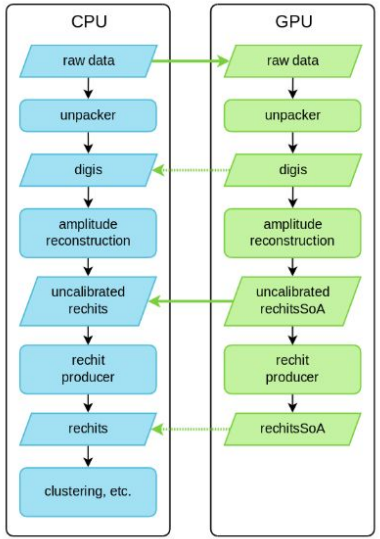




Sources [1], [2]

ECAL and HCAL local reconstruction at CMS

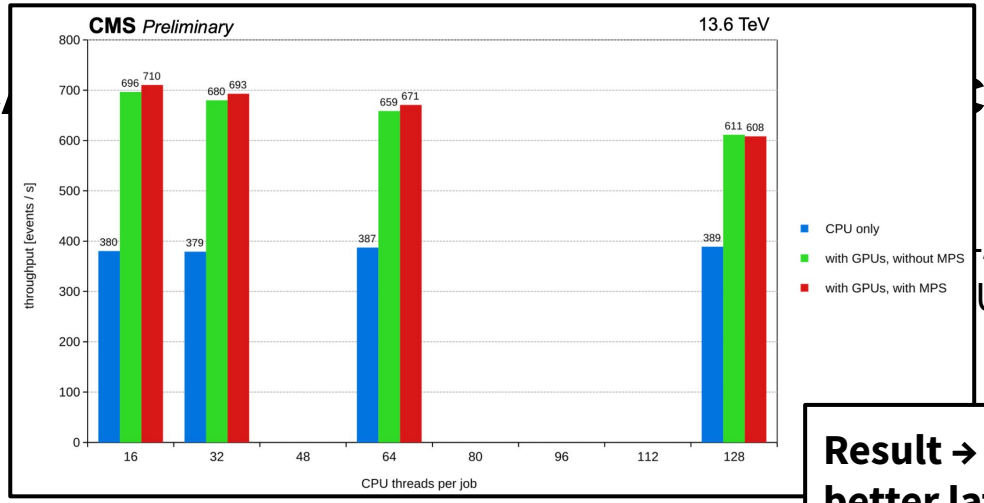
- Electromagnetic and Hadronic Calorimeter (ECAL/HCAL) local reconstruction at the HLT stage is performed on the GPU
- This includes :
  - Unpacking of the raw data from the detector into consecutive samples read out from a single ECAL channel.
  - Amplitude reconstruction.
  - Calculation of energies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

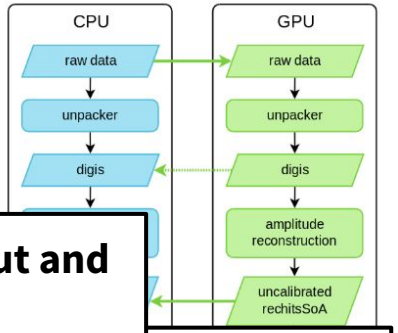


https://cds.cern.ch/record/2802591/files/CR2022_029.pdf

EC… …tion at CMS

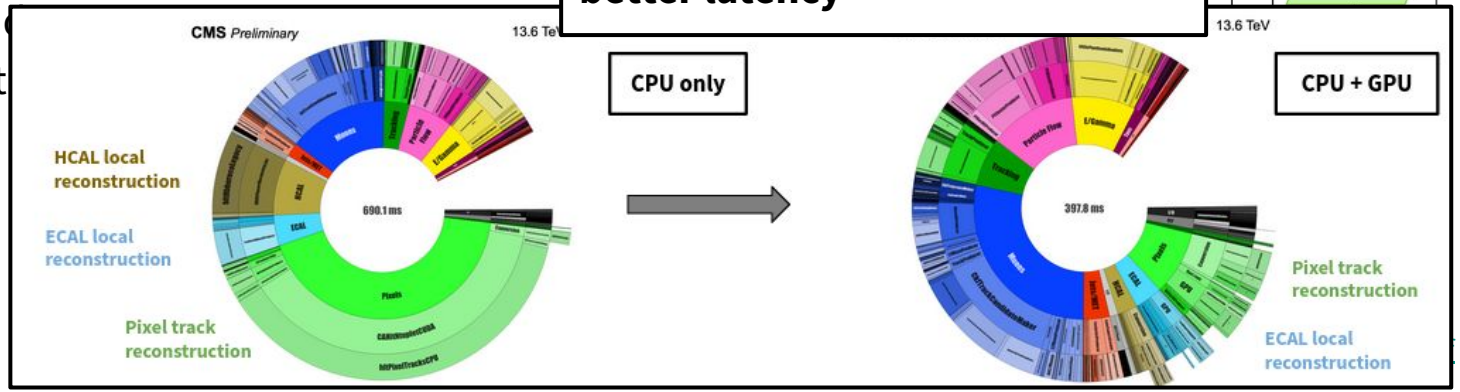
- …/HCAL) local …U
- …





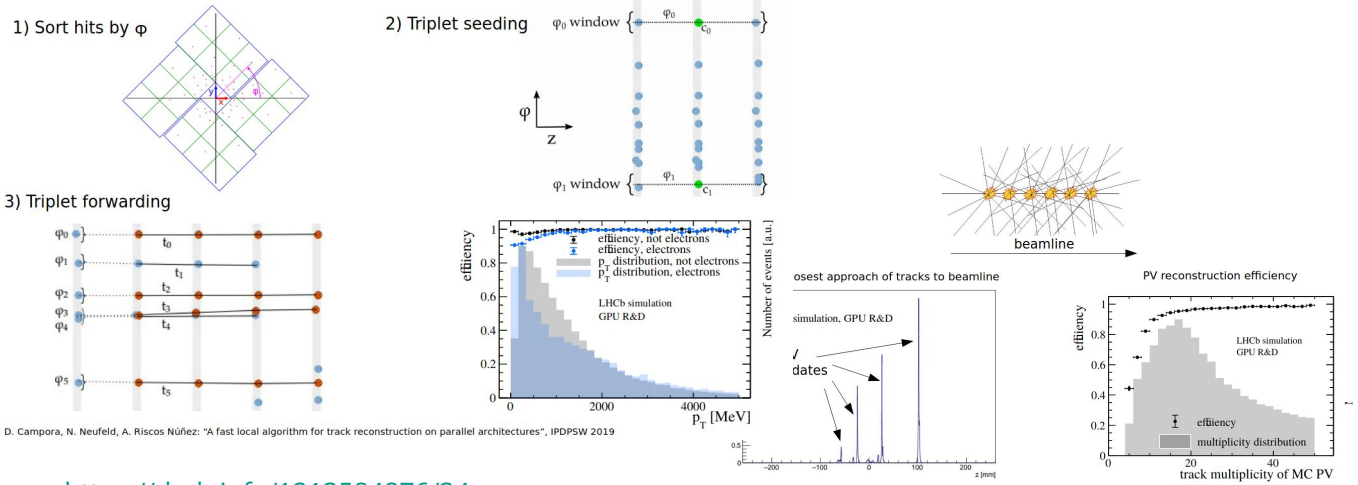
Result → Larger throughput and better latency

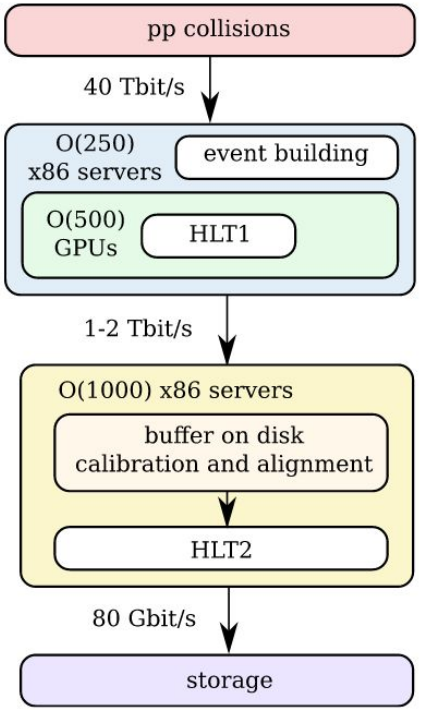
- Amplitu…
- Calculat… hits



Allen : HLT on GPUs for LHCb

- In use for Run-3
  - Primary vertex reconstruction
  - Reconstruct charged particle trajectories
- Allows for several thousand events to be processed in parallel

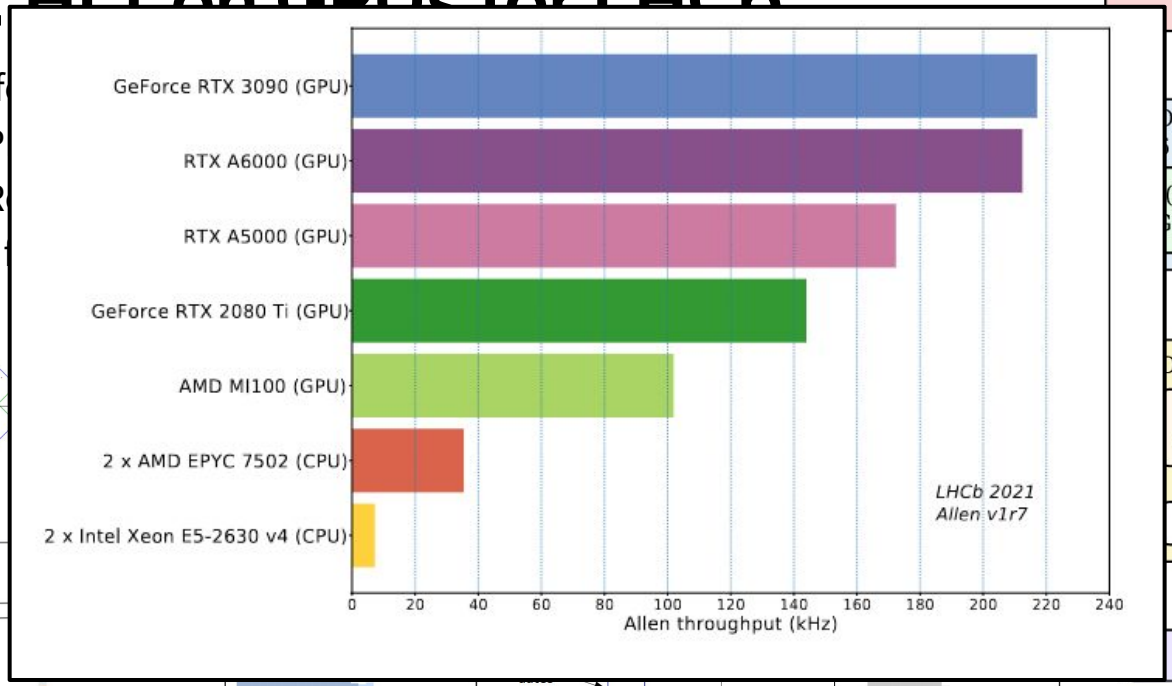


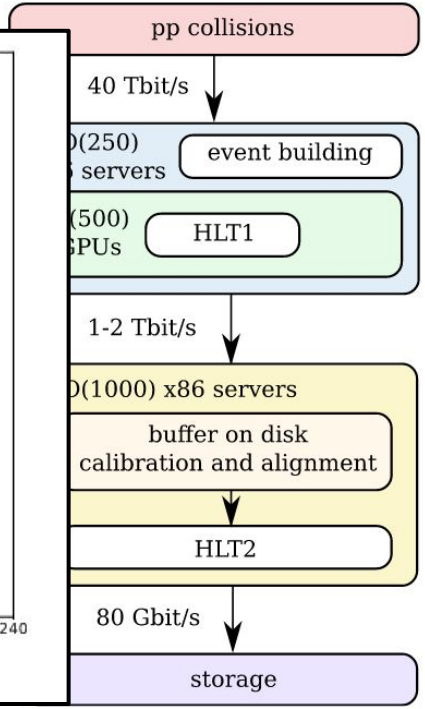


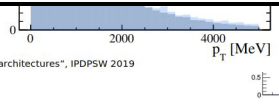
https://d-nb.info/1213584876/34
https://cds.cern.ch/record/2699553/files/vom_Bruch_Allen_chep2019%2004.11.pdf

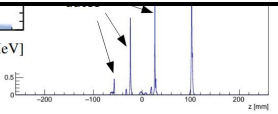
Allen : HLT on GPUs for LHCb

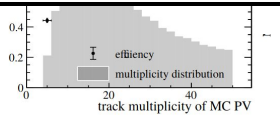
- In use f
  - P
  - R
- Allows







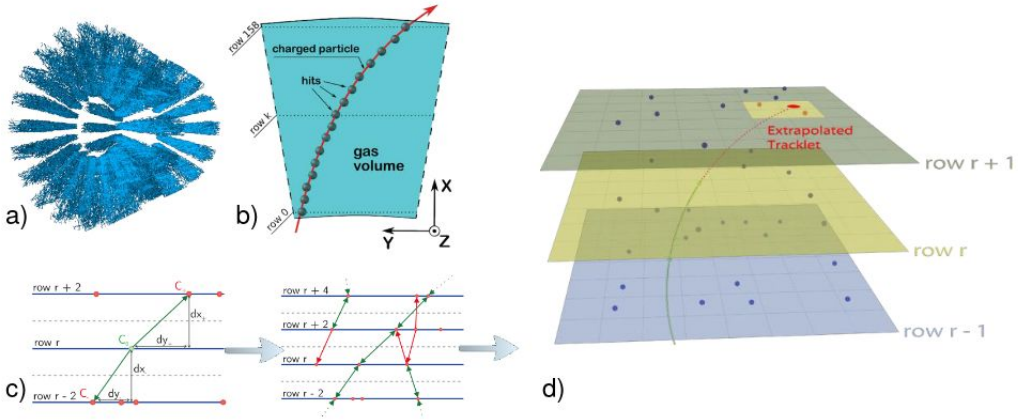




https://iopscience.iop.org/article/10.1088/1742-6596/2438/1/012017/pdf

TPC track reconstruction with GPUs at ALICE

- 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

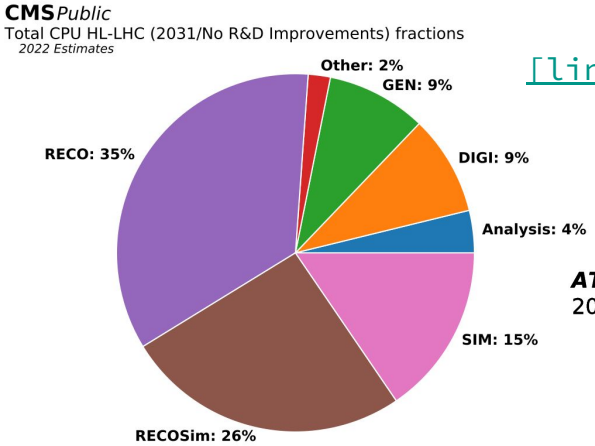


- **Several events and different sectors of the TPC are able to run in parallel**

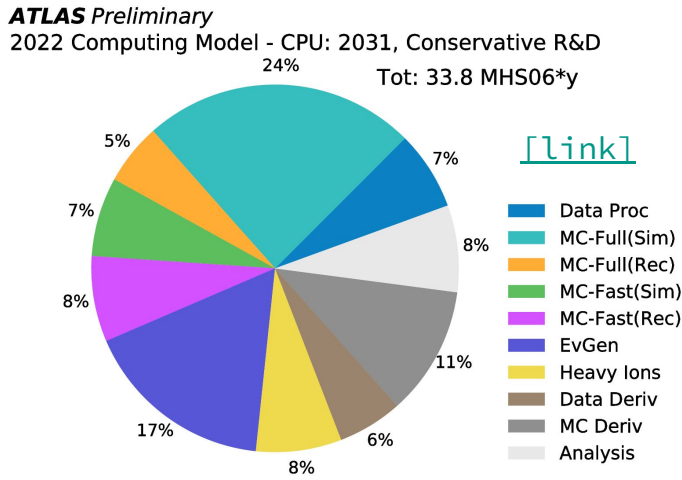
https://arxiv.org/pdf/1712.09430.pdf

Computing needs for offline data processing

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



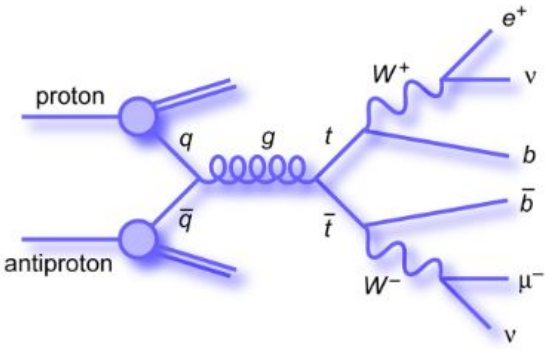
[link]

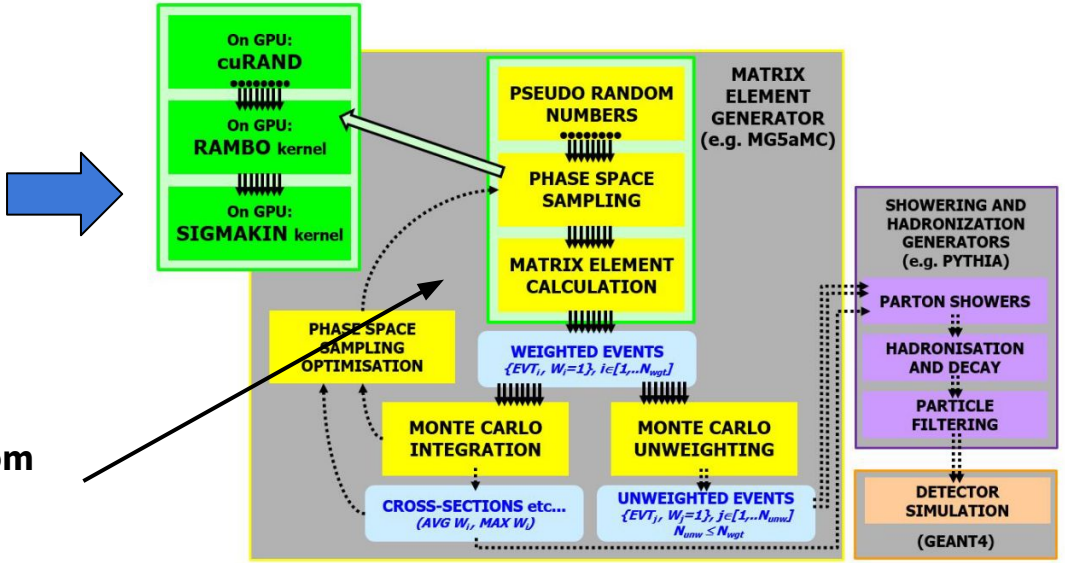


[link]

GPU enabled madgraph generator

Most commonly used software for generating simulated particle collision events / calculating cross sections of SM and BSM processes etc.

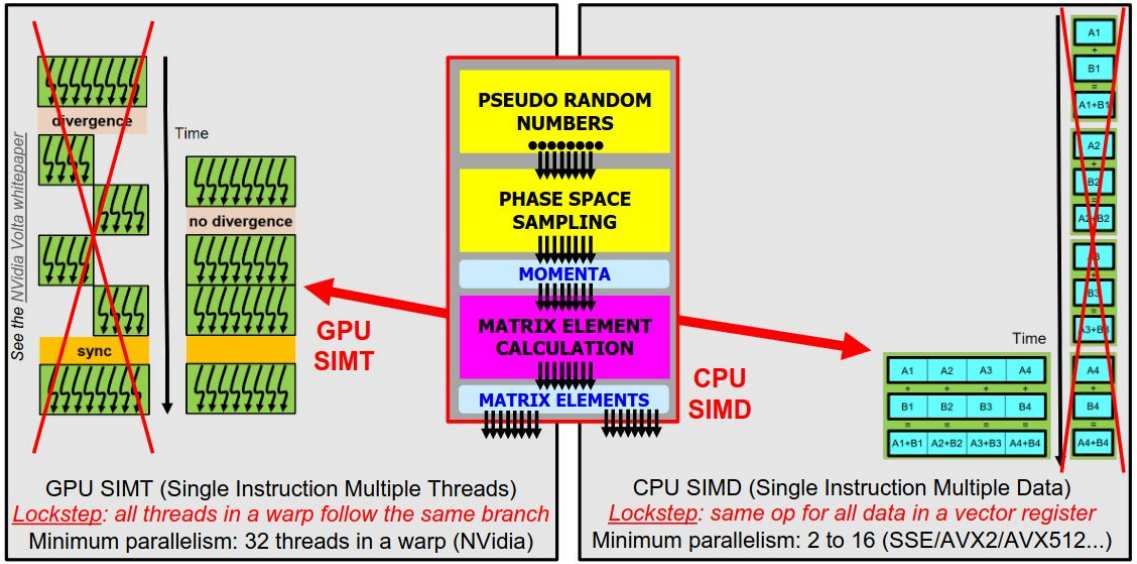




Use of cuRAND to speed up the random number generation

https://cds.cern.ch/record/2774080/files/2106.12631.pdf
For more info also check out this CHEP2023 talk

GPU enabled madgraph generator



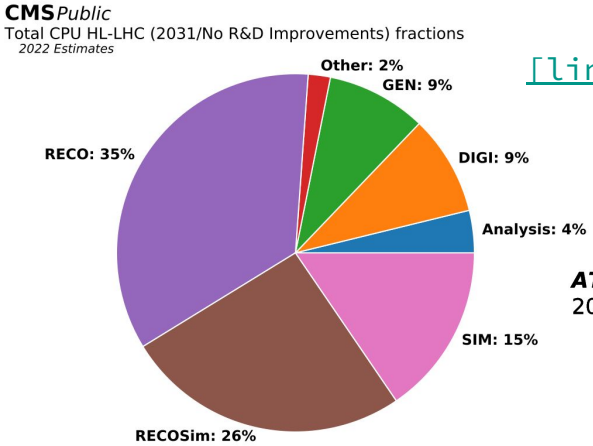
Matrix element calculation:

- Good fit for lockstep processing on GPUs (SIMT) and vector CPUs (SIMD)
- Data parallelism strategy in is event-level parallelism

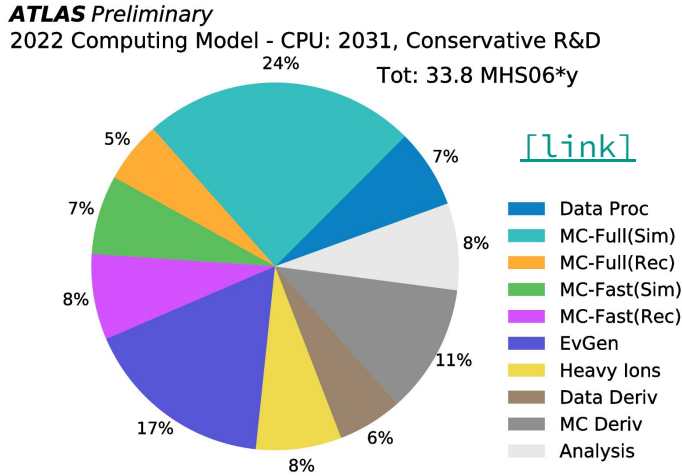
ME calculation essentially calculation of the same function on different data points

Computing needs for offline data processing

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



[link]

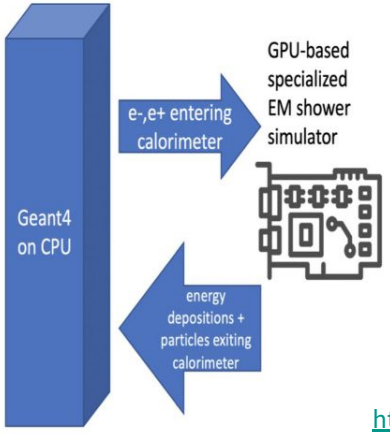


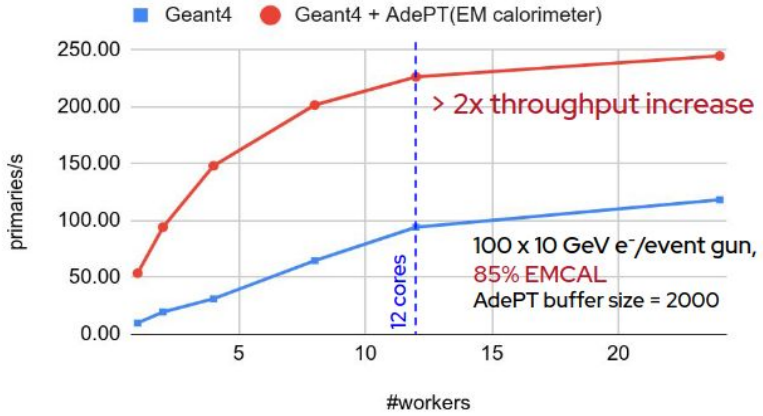
[link]

GPU based Geant4 application AdePT

Geant4 →Toolkit for the simulation of the passage of particles through matter

- AdePT project tries to use GPUs to accelerate the simulation
  - Currently only provides EM physics for e^+, e^- and γ
- Main difference with CPU based Geant4 :
  - All active tracks available are stepped at once exposing the parallelism to the GPU

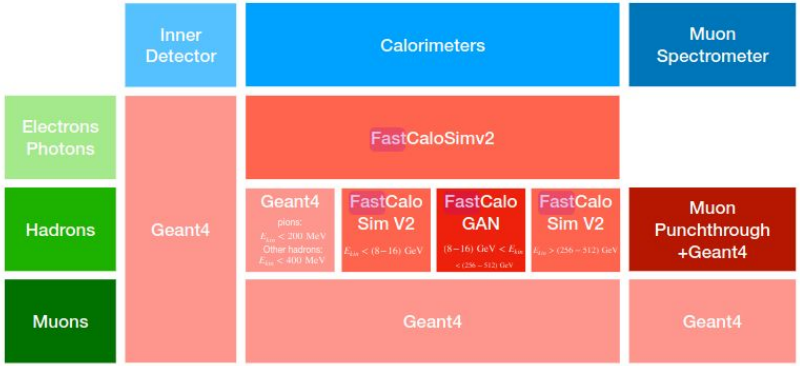




https://indico.jlab.org/event/459/contributions/11427/attachments/9538/13835/CHEP2023-AdePT.pdf

AI/ML enabled Fast Simulation AtlFast3

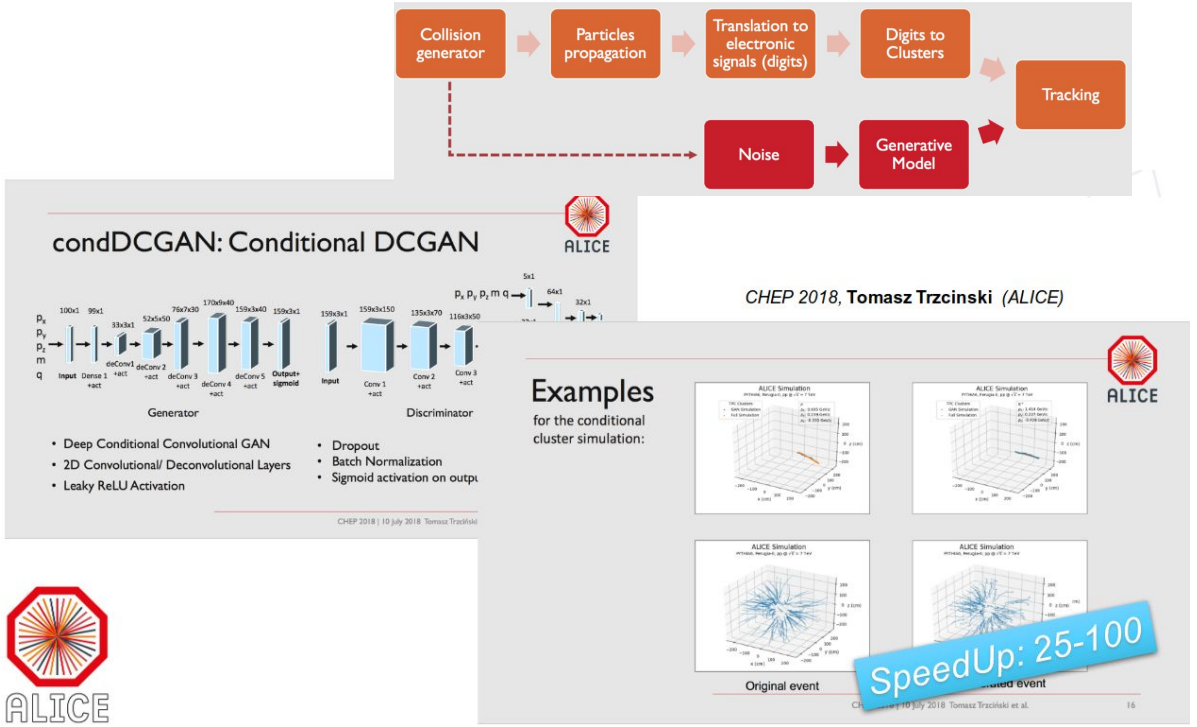
- Many experiments have been looking into taking advantage of ML for fast detector simulation
- In ATLAS simulation of shower development particularly CPU intensive when using the Geant4
- Combination of 2 fast sim approaches :
  - FastCaloSimV2 parameterizes the longitudinal and lateral development of showers in the calorimeters
  - FastCaloGAN is a fast calorimeter simulation tool that parameterizes the interactions of particles in the calorimeter system using GANs different particle types and η slices



https://arxiv.org/pdf/2109.02551.pdf

AI/ML enabled Fast Simulation DC GAN

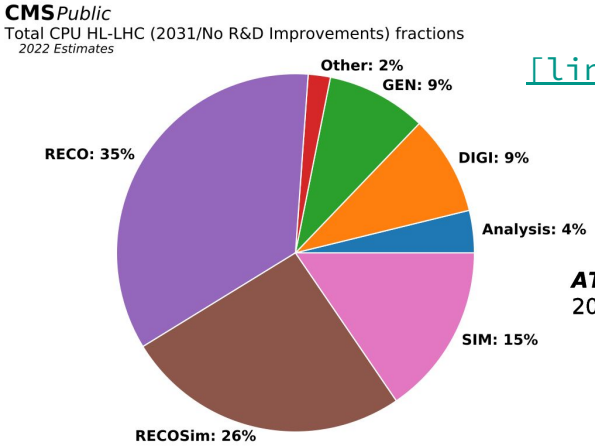
- Using generative models for fast simulations in the TPC (Time Projection Chamber) detector
- Substitute part of the simulation pipeline, namely particle propagation and translations to digits and clusters, with a generative model



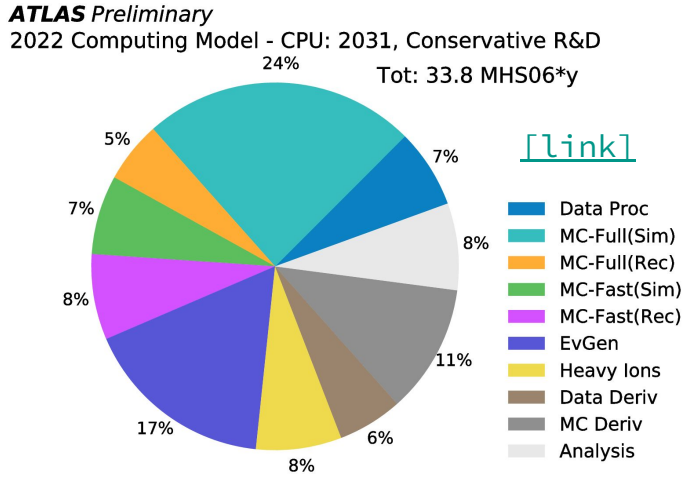
https://indico.cern.ch/event/587955/contributions/2937515/attachments/1683183/2707645/CHEP18.pdf

Computing needs for offline data processing

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



[link]



[link]

Computing needs for offline data processing

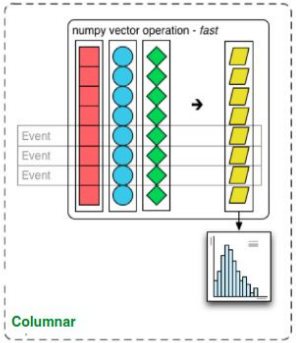
- Event generation
- Simulation
- **Event reconstruction**
- Event post-processing
- Data analysis

- Most experiments use common frameworks for performing real-time reconstruction (trigger) and offline reconstruction (data re-reconstruction and reconstruction of simulated events.
- Can take advantage of developments in real-time data processing and use the same algorithms offline

Computing needs for offline data processing

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

- Training and inference of ML models using GPUs
- Perform HEP analysis using columnar analysis paradigm tools (e.g. coffea [i])
- Check out also talks by Nick and Lukas from yesterday



Summary

- In front of a big computing challenge in HEP
  - Cannot rely solely on using CPU processors and homogeneous architectures
- Hardware landscape is changing
  - Shift towards hardware accelerators and heterogeneous computing
- Active R&D by many HEP experiments in most fronts :
  - Event generation
  - Simulation
  - Reconstruction
  - Analysis tools

BACK-UP

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

Performance comparison of CPUs and GPUs (1)

FLOPS : Floating-Point Operations per Second

- Measure of computing performance useful in fields that require floating-point calculations (such as HEP)
- GPUs can deliver more FLOPS compared to CPUs

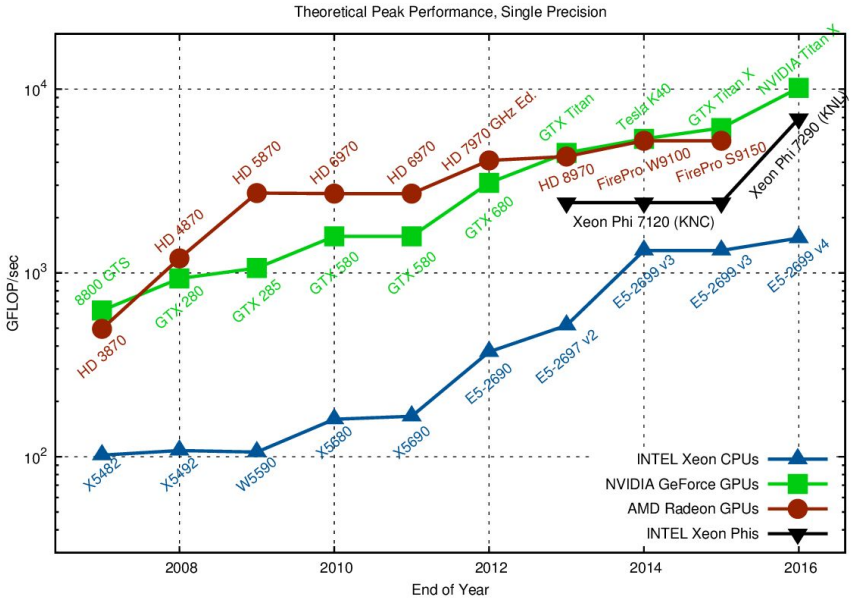


Image source [i]

Performance comparison of CPUs and GPUs (2)

FLOPS per Watt :

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

Important since power consumption is limiting factor in hardware manufacturing/usage:

- Peak performance constrained by the amount of power it can draw and the amount of heat it can dissipate

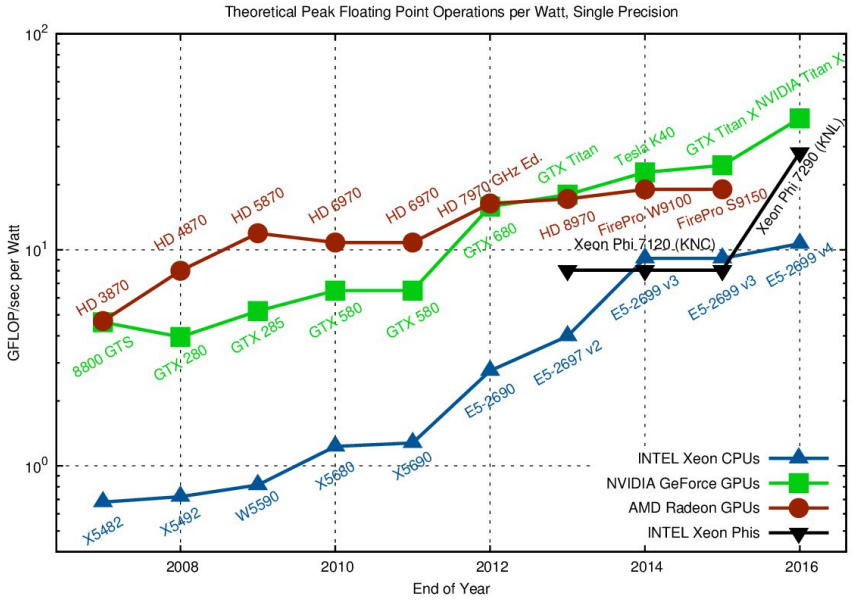
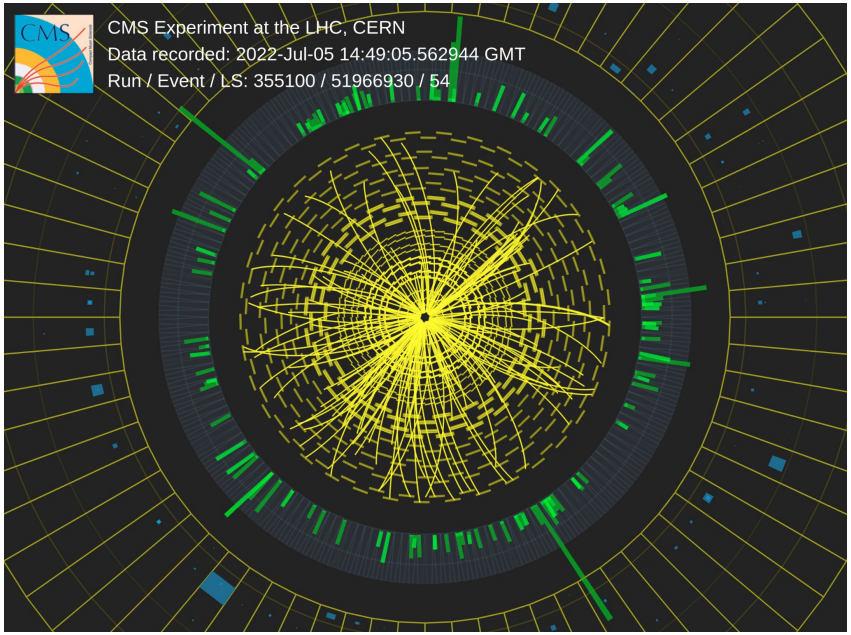


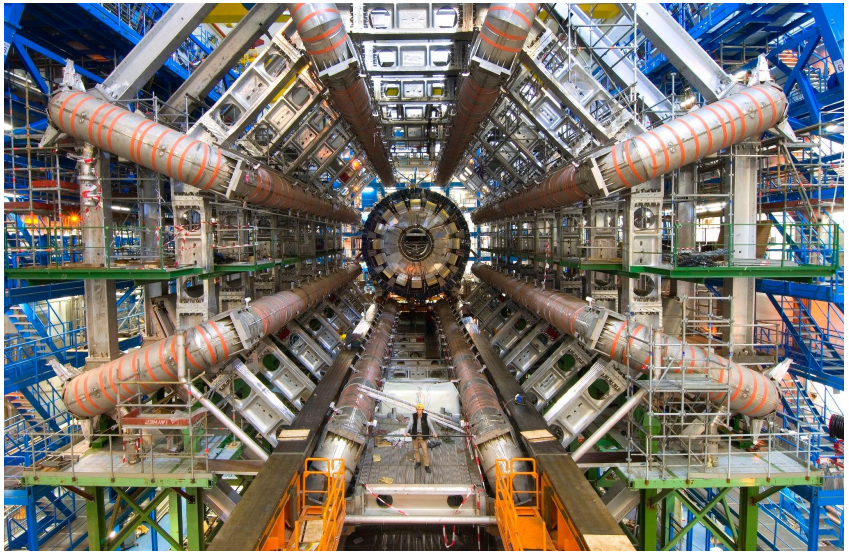
Image source [i]

CMS (Compact **M**uon **S**olenoid**)**



- 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

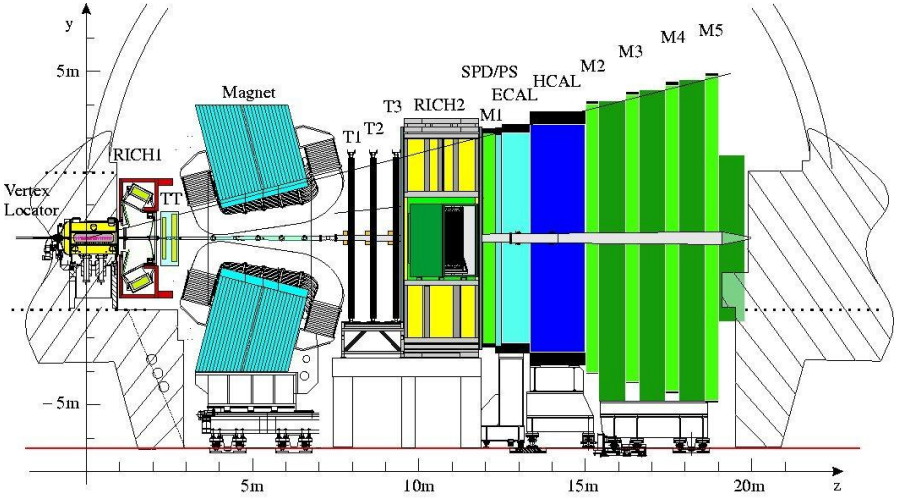
ATLAS (A Toroidal **L**HC **A**pparatus)



- One of the four large experiments at the LHC

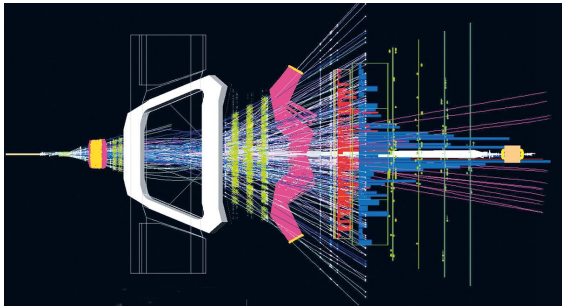
- General-purpose detector designed to measure properties of the Standard Model and observe new physics phenomena that the LHC might reveal

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



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)**