Detector simulation requirements from HEP Experiments

HSF/WLCG/Geant4 workshop
23rd November 2020

Marilena Bandieramonte (University of Pittsburgh)

With contributions from …

- **LHCb:** Adam Davis[1], Gloria Corti[2], Michal Kreps[3]
- **ALICE**: Sandro Wenzel[2], Andreas Morsch[2]
- **CMS**: Danilo Piparo[2], James Letts[4], Samuel Bein[5], Vladimir Ivanchenko[6]
- **ATLAS**: John Derek Chapman[7] , Michael Duehrssen-Debling[2]

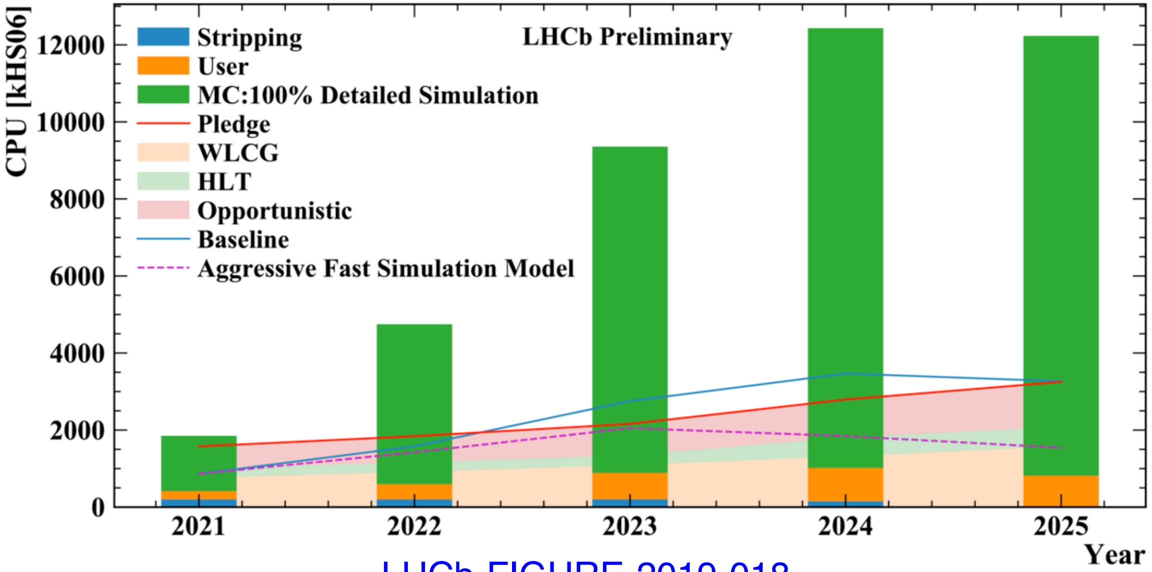
… many thanks!

[1] University of Manchester
[2] CERN
[3] University of Warwick
[4] University of California San Diego
[5] University of Hamburg
[6] Tomsk State University
[7] University of Cambridge

CPU requirements for detector simulation

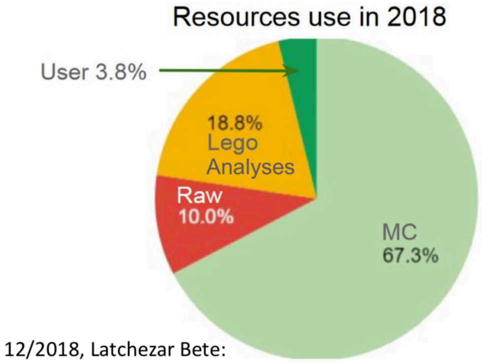
- **Detailed detector simulation**, is one of the largest consumers of WLCG computing resources for LHCb (85%), ATLAS (40%) and CMS (25%) [1]
- Already in Run 3 **LHCb** will process more than 40 times the number of collisions that it does today, and **ALICE** will read out Pb-Pb collisions continuously at 50 kHz.
  - As a consequence of the increased luminosity and interaction rate, a significantly larger amount of data will have to be processed and selected.
  - ALICE estimates that the simulation requirements will increase **by a factor of 20** [2]
- **Simulation is the leading CPU consumer** for LHCb and ALICE and will continue to be in the near future
- Only with adoption of aggressive alternatives to detailed Geant4 based simulation will we be able to stay within pledged resources

LHCb



LHCb-FIGURE-2019-018

ALICE

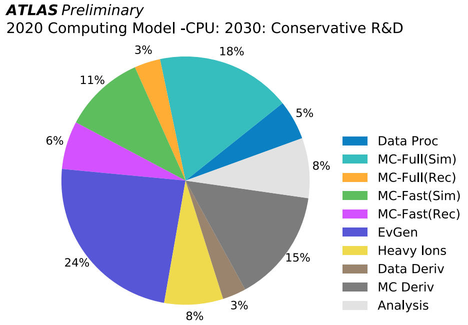


2/3 of the Computing Resources are dedicated to MC simulation, all full simulation

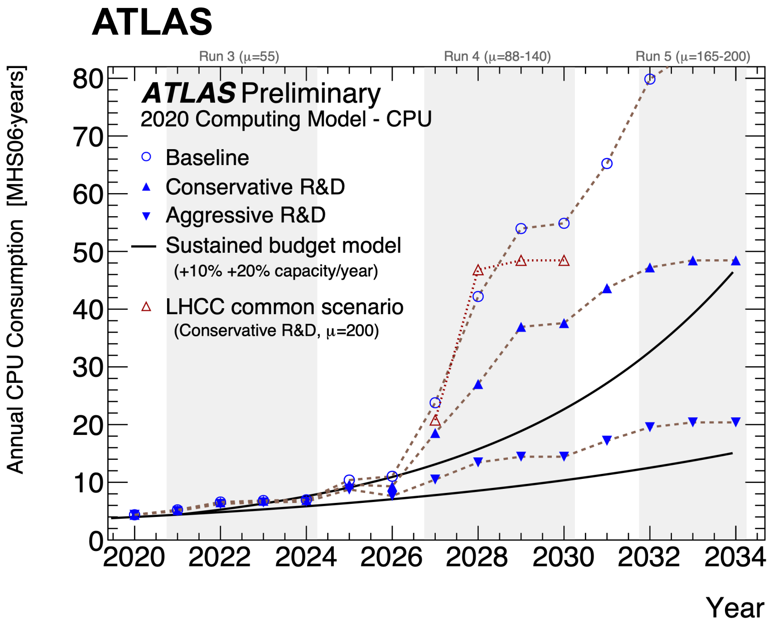
ALICE resource use in 2018 [2]

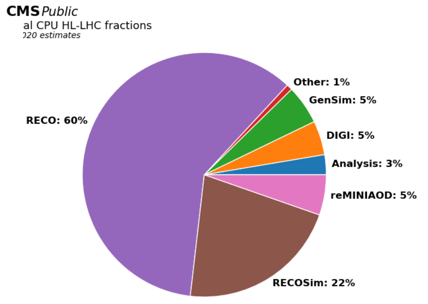
CPU requirements for detector simulation

- The upgrade to the **HL-LHC for Run 4** produces a step change for **ATLAS** and **CMS**.
- The beam intensity will rise substantially, giving bunch crossings where the number of discrete proton-proton interactions (pileup) will rise **to about 200**, from about 35 today (2018 and foreseen for 2022)
- Accurate simulations and larger Monte Carlo samples will be needed to achieve the desired precision in physics measurements, while avoiding that simulation dominates the systematic uncertainties
  - ATLAS plans different R&D lines to reduce the need for *detailed full simulation*
  - Last CMS projections point out *reconstruction* as the biggest computing power consumer. Big effort ongoing to improve it: simulation and other components will become more and more important [1]

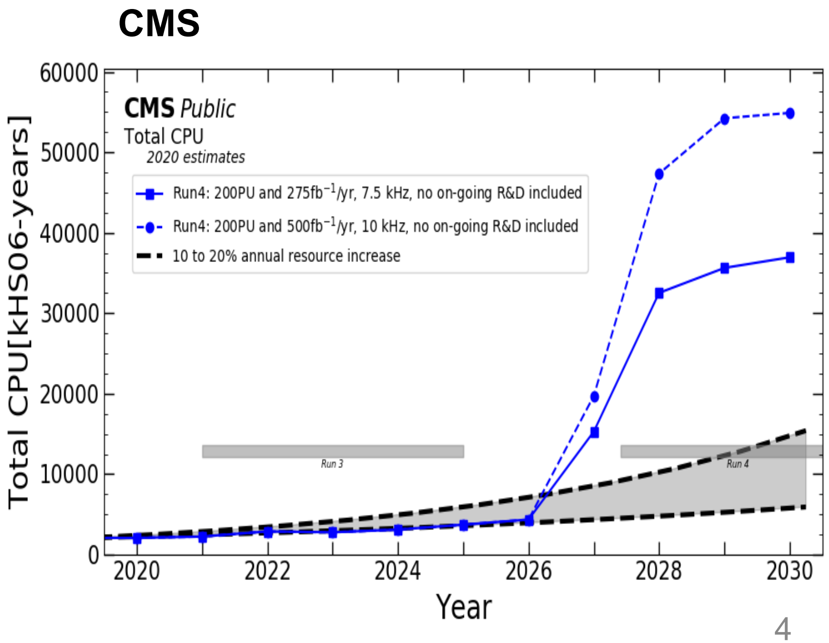


ATLAS Public plots





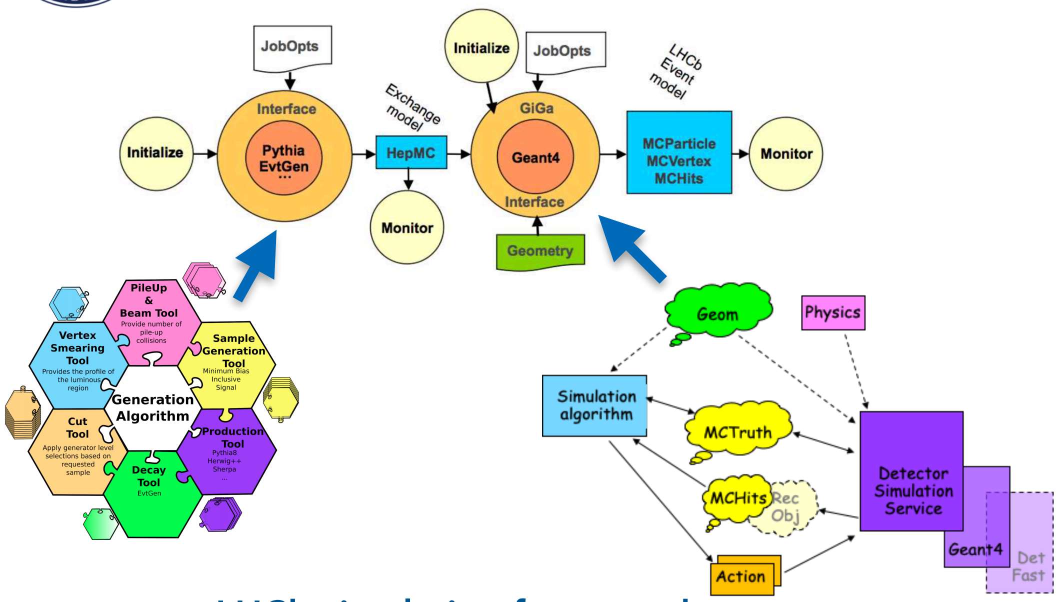
CMS Public plots



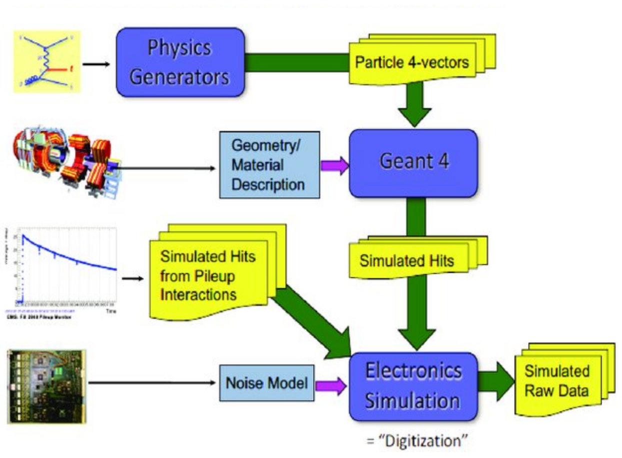
HEP simulation frameworks

- **Geant4** is at the core of almost all HEP experiment simulation infrastructures

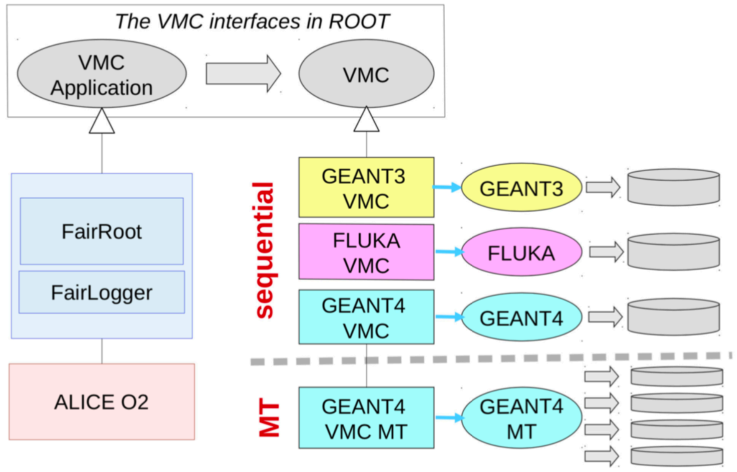
- Typically, **detailed Simulation** (a.k.a. Full Simulation) performed using Geant4, with **fast simulation options** available
  - Integrated within Geant4 using fast sim hooks
  - Available with a different framework (i.e. Integrated Simulation Framework in ATLAS)



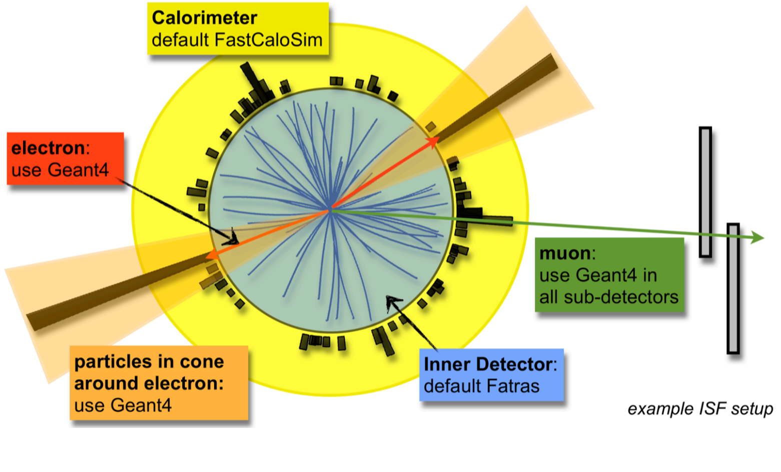
LHCb simulation framework



CMS simulation framework



ALICE simulation framework



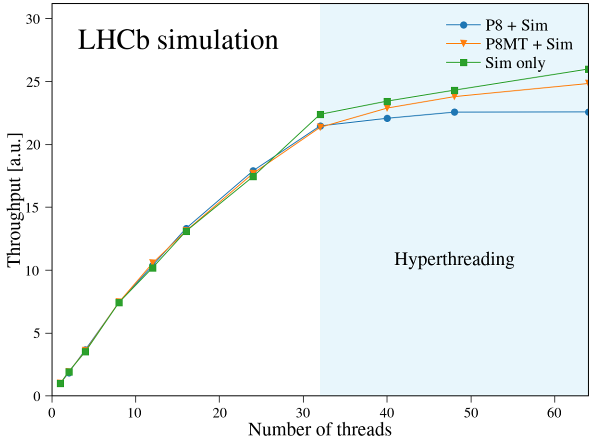
ATLAS Integrated Simulation Framework

Three major simulation efforts and R&D's

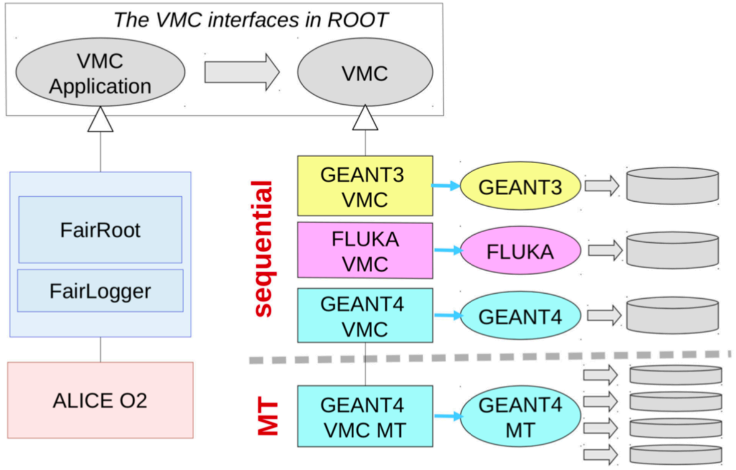
1) Efficient use of parallel computing architectures
2) GPU-based HPC exploitation
3) Fast simulation

Simulation efforts and R&Ds: MT

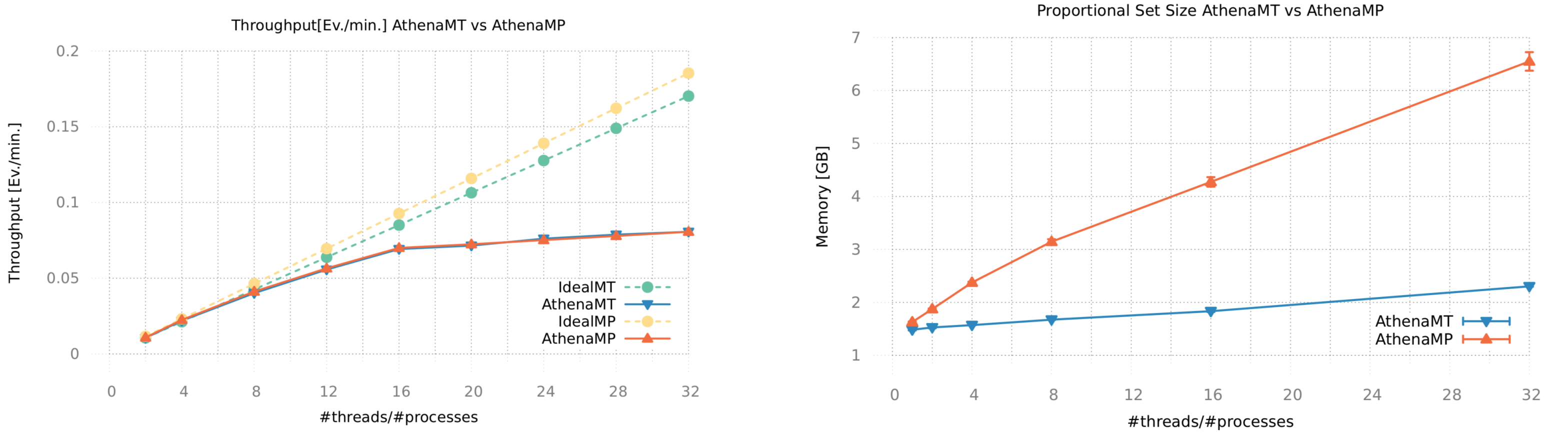
- Most of the simulation requirements increase will be met by using **fast or parameterised** Monte-Carlo simulations.
- For the remaining full simulations, it is essential to use a **computationally efficient** particle transport code able to make **efficient use** of computing resources available on the GRID/Cloud:
  - **Geant4** offers a multi-threading capability since version 10.0
- **Heavy effort to Migrate to MT paradigm and optimise them**:
  - Gauss-on-Gaussino (LHCb), AthenaMT (ATLAS), O2+VMC (ALICE), CMSSW (CMS)
  - All CMS workflows are MT since Run-2 start. Heavy effort to make them more and more efficient since then



Performance of the multithreaded prototype for the future LHCb simulation framework (Gauss-on-Gaussino), uses GaudiMT + Geant4MT[3]



The ALICE Virtual Monte Carlo context in O2 framework [1]



ATLAS Wall-Time speedup (a) and Memory Occupancy - PSS (b) of AthenaMT (blue) vs. AthenaMP (orange), as a function of the number of threads or processes, respectively. [2]

Simulation efforts and R&Ds: GPU-based HPC

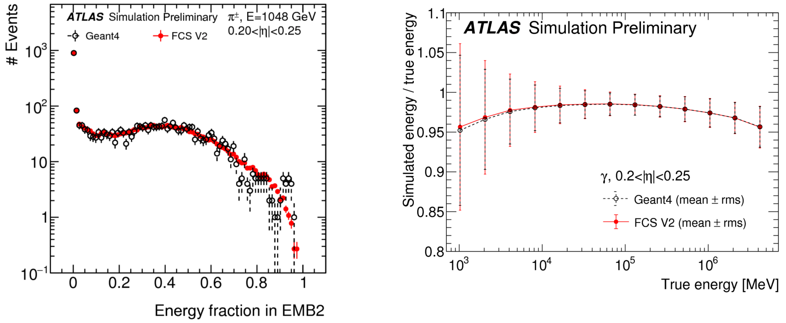
- **GPUs** are currently one of the **main challenges** for the exploitation of HPC centres by the HEP experiments [1]
  - the trend for new HPC systems is to provide an increasingly larger fraction of their computing power through GPUs

- All **HEP experiments have been actively working** in the last few years on the reengineering of their software workflows and also on their **porting to heterogeneous hardware environments** including GPUs (WLCG sites or HPC centres)
  - **Detailed MC simulation:**
    - **Geant4** cannot currently **exploit GPUs or other accelerators:**
      - heavy R&D ongoing, see *adePT*, *VecGeom* on GPUs, *G4HepEM*
    - New interesting ongoing community efforts for **faster detector simulation on GPUs**
      - see: *Celeritas*, *Optiks*, *ExcaliburHEP* [see talks in this session]
  - **Fast detector simulation:**
    - **R&D efforts using parametrised calorimeter responses, ML techniques**
      - Calorimeter shower parameterisation on GPUs (i.e. FastCaloSim-GPU [2])
      - Network trainings or hyperparameter optimisations
      - ML training requires inputs and I/O from/to HPCs, not too easy

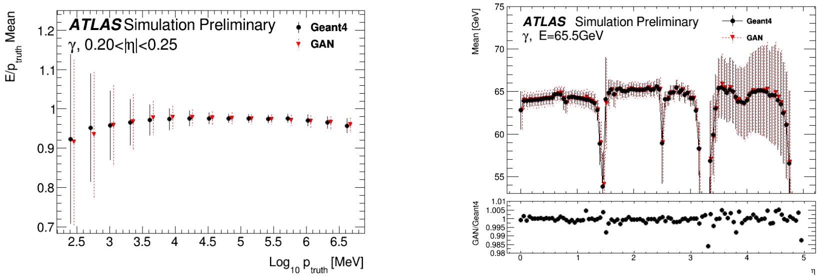
Simulation efforts and R&Ds: Fast simulation*

- **ATLAS**
  - In production: Frozen showers and Fast Calorimeter Simulation V1 for ~40% of all MC
  - New Fast Calorimeter Simulation : Hybrid: FastCaloSim V2 (parameterization-based Fast Calo Simulation) [1] for e/gamma, GAN-based [2] approach for high energy hadrons, Geant4 for muons and exotic particles
  - FastChain: fast simulation, fast digitization, and fast track reconstruction [3]

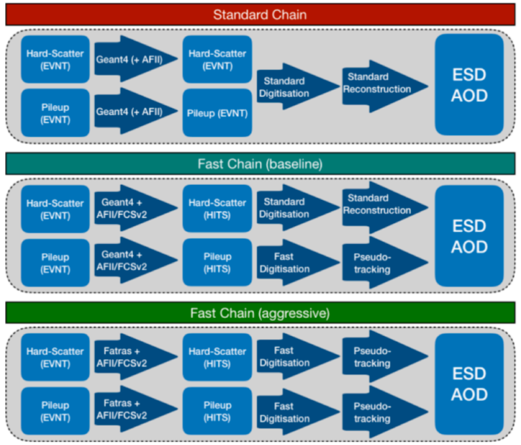
ALICE is trying to simulate TPC clusters using GANs [4]



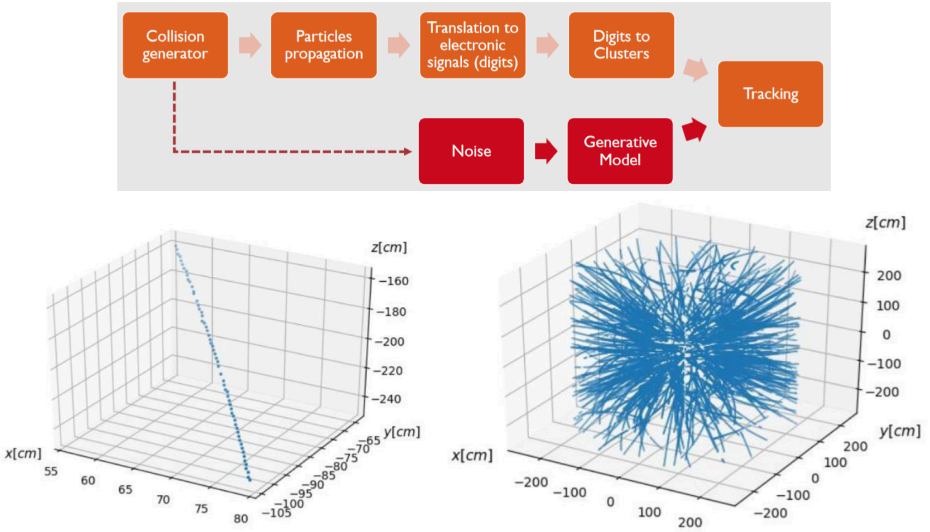
ATLAS FastCaloSimv2 [1]



ATLAS FastCaloGAN [2]



ATLAS FastChain [3]



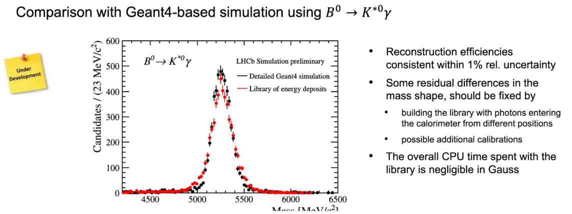
ALICE TPC clusters simulation using GANs [4]

*For details on Machine Learning techniques in HEP experiments please see the next talk from K Pedro

Simulation efforts and R&Ds: Fast simulation*

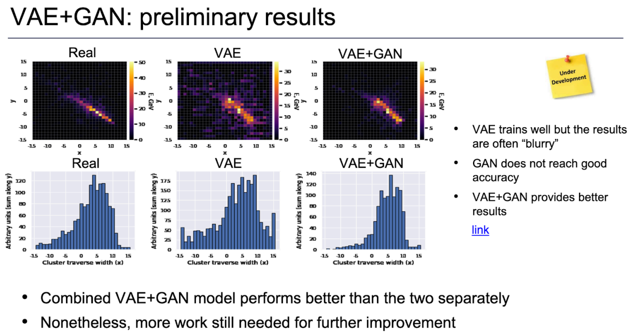
- **LHCb recent effort on fast simulation:**
  - **Calorimeter Shower Libraries:** Replace Geant4 calorimeter simulation with point libraries
  - **GANs for PID**
    - Replace Geant4 calorimeter simulation with ML techniques (GANs so far but VAE studies ongoing)
  - **Lamarr (Combination of parametric and ML based techniques)**
    - After generation phase, replace all steps to final analysis with parametric simulation
- **CMS: Monte Carlo Fast Processing Chain: Sim+Rec**
  - **Since Run 2, CMS Monte Carlo events produced via**
    - Standard approach: G4 based simulation + detailed reconstruction: baseline
    - Fast-Chain (sim+digi+reco): about 10% of the samples
      - simplified geom, infinitesimal layers, simplified tracking, analytical interaction mo
  - **About 6x faster than the "high-fidelity" chain (all steps from simulation to analysis dat**
  - **Fast-chain had a scarce utilisation in Run 2: G4 sim + reco fast enough**
  - **Re-evaluating its use for Run-3-4-5-…**
    - Improvements/validation underway to establish wider applicability, e.g.,
      - fake/displaced tracks, impact on b-tagging
      - ML techniques to improve accuracy, e.g., of jet shape observables
    - Outreach to various analysis groups to see where Fast-Chain could or couldn't be adopted

Simulation of $B^0 \to K^{*0}\gamma$ decays M. Rama, ICHEP2020



Comparison with Geant4-based simulation using $B^0 \to K^{*0}\gamma$

- Reconstruction efficiencies consistent within 1% rel. uncertainty
- Some residual differences in the mass shape, should be fixed by
  - building the library with photons entering the calorimeter from different positions
  - possible additional calibrations
- The overall CPU time spent with the library is negligible in Gauss

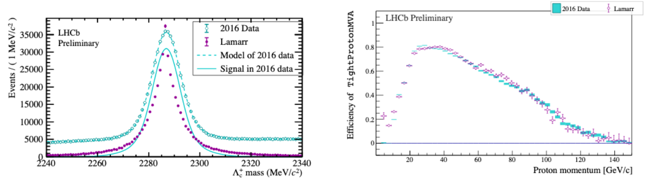


VAE+GAN: preliminary results

- VAE trains well but the results are often "blurry"
- GAN does not reach good accuracy
- VAE+GAN provides better results

link

- Combined VAE+GAN model performs better than the two separately
- Nonetheless, more work still needed for further improvement



[LHCb-FIGURE-2019-017]

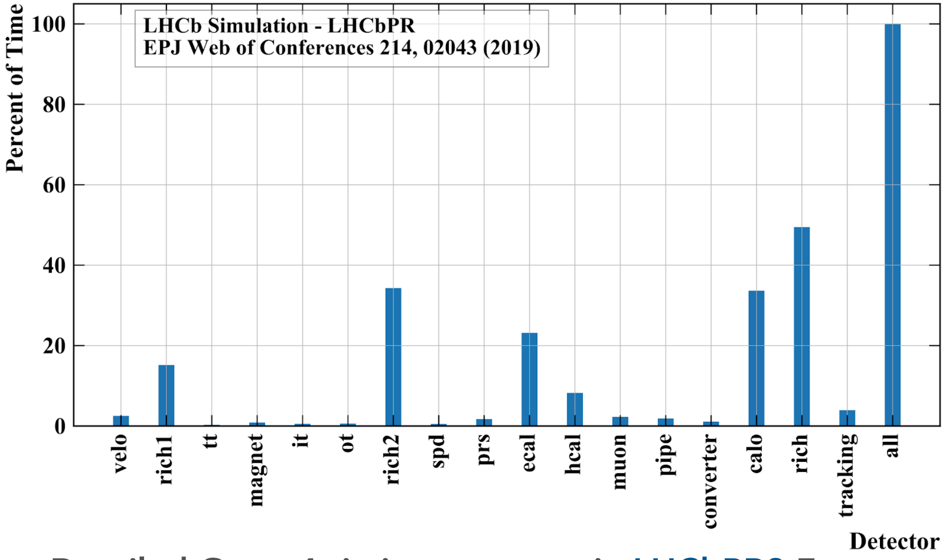
A.Davis, CHEP2019, A.Morris ICHEP2020

*For details on Machine Learning techniques in HEP experiments please see the next talk from K Pedro

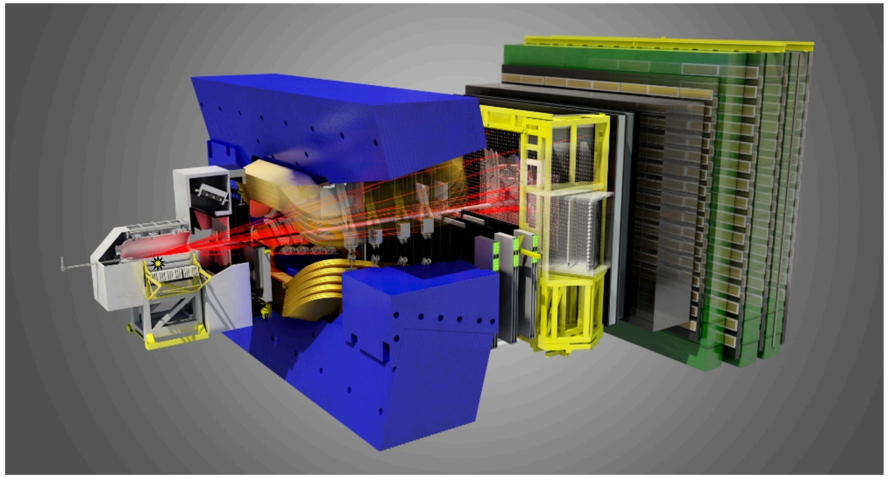
Experiment specific simulation requirements

Detector simulation requirements: LHCb

- Detailed Geant4 timing tests run in LHCbPR2 Framework
  - Major users are RICH detectors (optical photons) and Calorimeters (especially ECAL)
  - 20-30% improvement in all volumes moving from Geant4 9.6.p04 -> 10.3.p03 (LHCbEm PL)
    - Still true in current configuration, using Geant4 10.6.p02 (EmOpt2)
- Any improvement in **speeding up the code base** will be highly appreciated – it is understood that tuning is the responsibility of the experiment
- Appreciate continuing support on **more complex fast simulation** integration
- LHCb will have access to new and varied computing resources, e.g. **GPUs at the HLT farm**. It will be more and more important to be able to access heterogenous resources for the simulation



Detailed Geant4 timing tests run in LHCbPR2 Framework



The LHCb detector - Image credit

LHCb ongoing and future effort on Opportunistic Resources

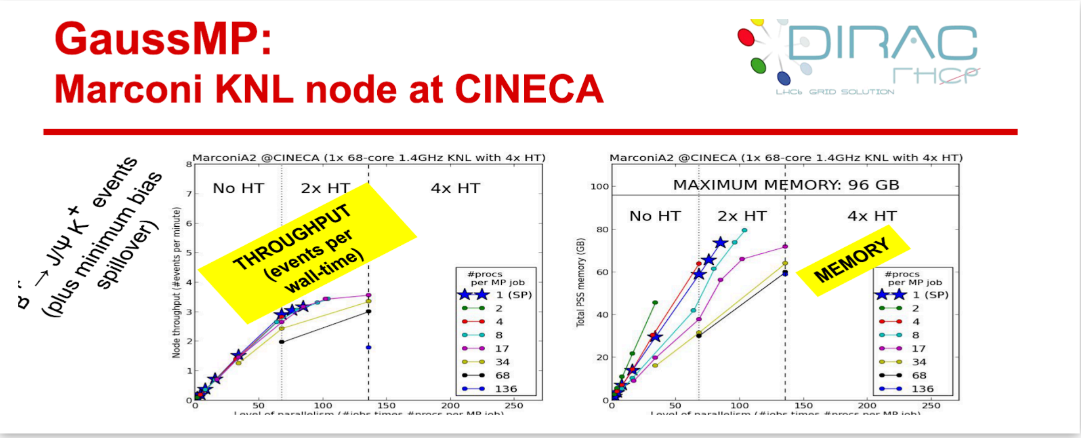
The usage of **opportunistic resources** is a necessity

Ongoing studies:

- MultiProcessor version of Gauss tested at **CINECA** [2006.13603v1]
- Eventually will use Gauss-on-Gaussino Multi-Threaded on **HPCs**

- More and more important to exploit heterogeneous resources for simulation:
    - e.g. **LHCb Run3 HLT GPU farm**
- Preliminary look at Opticks/Optix for ray tracing of optical photons on GPUs for use in RICH simulation
- Happy to be exposed early on to R&D and try it out in Gauss-on-Gaussino

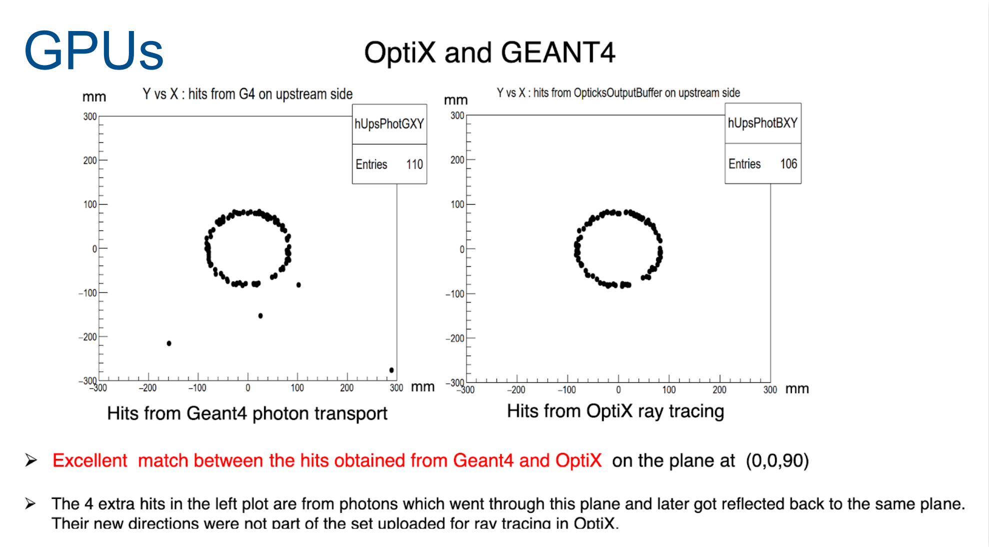
HPCs



GaussMP:

Marconi KNL node at CINECA

F. Stagni @ CHEP2019



GPUs

OptiX and GEANT4

Hits from Geant4 photon transport

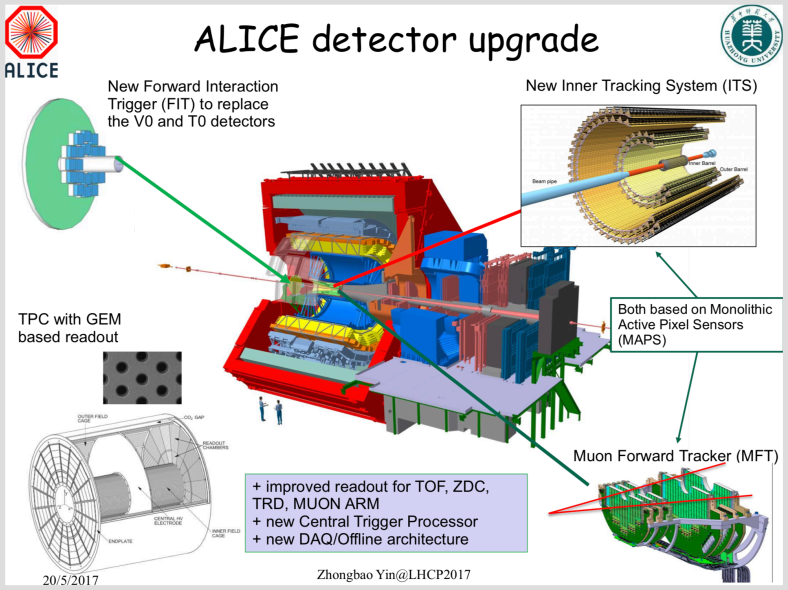
Hits from OptiX ray tracing

- ➢ Excellent match between the hits obtained from Geant4 and OptiX on the plane at (0,0,90)
- ➢ The 4 extra hits in the left plot are from photons which went through this plane and later got reflected back to the same plane. Their new directions were not part of the set uploaded for rav tracina in OptiX.

S.Easo @ HSF Sim Forum

Detector simulation requirements: ALICE

- With an increase of the data volume by **two orders of magnitude**, the Runs 3 and 4 simulation requirements will notably increase ~ **by a factor of 20** [1]:
- To address the challenges of the major upgrade of the experiment, the ALICE simulations must be able to make efficient use of **computing and opportunistic supercomputing resources** available on the GRID.
- In this scenario having **faster simulation toolkits**, that:
  - allow to increase the event rate substantially
  - are ready to make good use of novel hardware (GPU when available, but also on traditional servers)
  - integrate fast simulation methods within the framework
  - continue to improve the speed on ordinary hardware (integration of VecGeom, novel steppers, …)
- Having **more flexible "data" handling:**
  - for example: less centric on "an event"
  - provide easier sub-event parallelisation (for big events)
  - or allow to treat groups of events in one chunk (for really small events)



ALICE detector upgrade

New Forward Interaction Trigger (FIT) to replace the V0 and T0 detectors

New Inner Tracking System (ITS)

TPC with GEM based readout

Both based on Monolithic Active Pixel Sensors (MAPS)

Muon Forward Tracker (MFT)

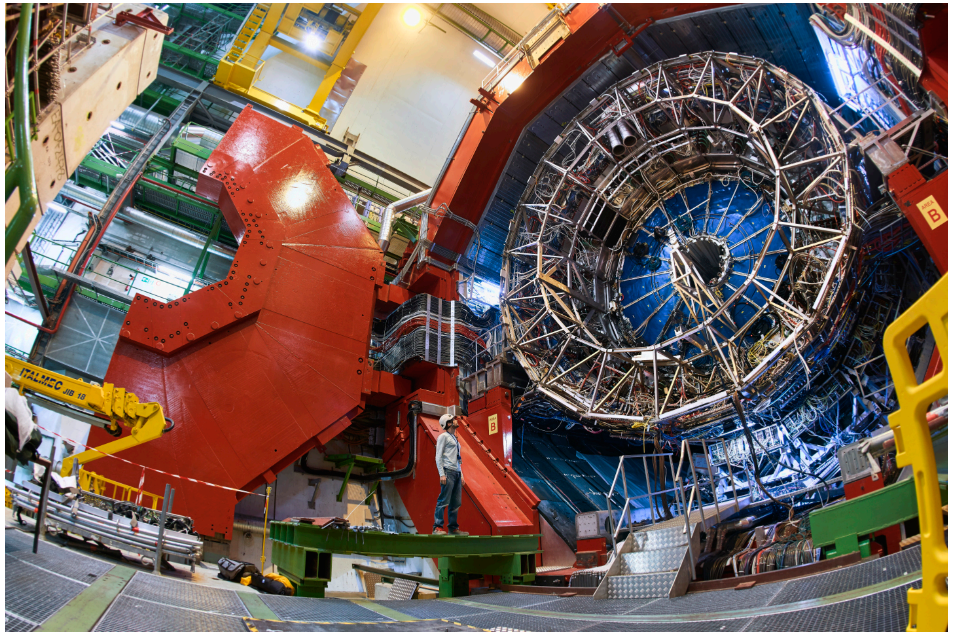
+ improved readout for TOF, ZDC, TRD, MUON ARM
+ new Central Trigger Processor
+ new DAQ/Offline architecture

20/5/2017

Zhongbao Yin@LHCP2017

Detector simulation requirements: ALICE

- Offering more **tools around simulation**, for instance to allow for convenient/fast experiment specific tuning:
  - automatic **geometry optimization** tools or "expert" advisors during design
  - automatic **physics cut-optimization** frameworks:
    - tune the cuts (under a certain physics constraint) until we have minimal steps/CPU time etc…
  - automatic **geometry cut optimization** tools:
    - tune material budget / space (under a certain physics constraint) until we have minimal steps/CPU time etc…
  - automatic **tuning of parameters** used by Geant4:
    - self selection of best stepper parameters etc.

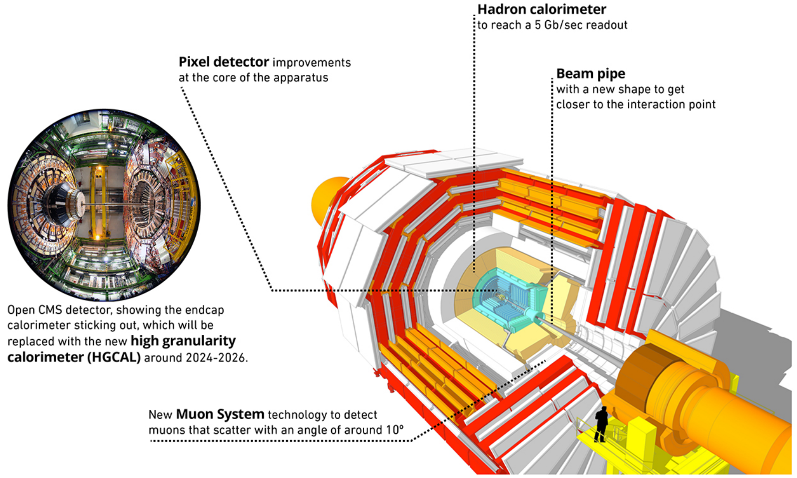


The ALICE detector - Image credit

Detector simulation requirements: CMS

- CMS needs to use heterogeneous hardware efficiently: important to use accelerators in simulation
  - **Run 3 HLT farm will feature GPUs and CPUs**: necessary to exploit this resource as well as present and future HPC allocation
  - More infrastructure for **plugging classical and ML fast sim** techniques would be useful also in this context
- **Full simulation needed** in copious quantities for Run 3 and Phase-2
  - Phase-2 fullsim 1.5x-3x slower than Run 3 (already accounted in the HL-LHC resources projections)
  - **CPU optimisation**s are needed as well as a plan for their implementation in the following years.
    - Quantitative estimation of the foreseen G4 timing improvement needed.

- **Thread friendliness** to be improved, i.e. removal of all serialisation points as well as dependency of throughput on progress of the event loop due to initialisation
- **Startup time** should be reduced or eliminated where possible (single threaded), e.g. not performing identical calculations for each job
- Allow to **remove** (compile time or configuration) **G4 functionalities** not needed for CMS, e.g. gravity



Hadron calorimeter to reach a 5 Gb/sec readout

Pixel detector improvements at the core of the apparatus

Beam pipe with a new shape to get closer to the interaction point

Open CMS detector, showing the endcap calorimeter sticking out, which will be replaced with the new **high granularity calorimeter (HGCAL)** around 2024-2026.

New **Muon System** technology to detect muons that scatter with an angle of around 10°

Detector simulation requirements: CMS

- CMS traditionally integrates early **new versions of Geant4** in its IBs
  - CMSSW + G4 10.7 + VecGeom last week

- Successful experience with the **GeantV prototype**
  - Very useful exercise even if it could not be used in production
  - Technical path for integration: template wrappers, ExternalWork to manage GeantV tasks
  - Speedup in CMSSW framework similar to standalone GeantV results

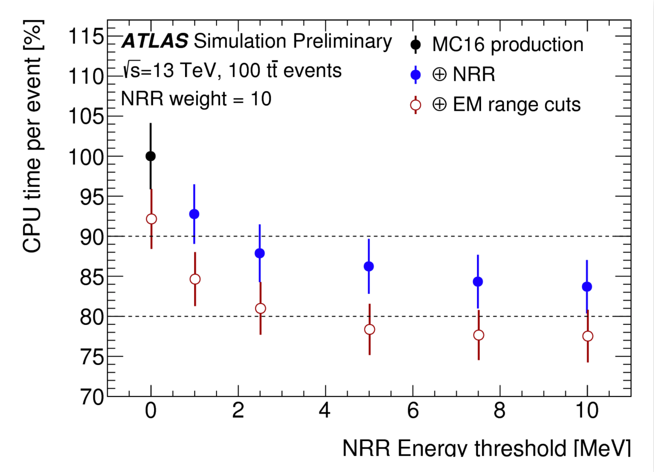
- **CMS will continue to**:
  - **explore** experiment specific optimisations
  - **adopting** new optimisations from the Geant4 team,

    e.g. interest in the forthcoming VecGeom navigator

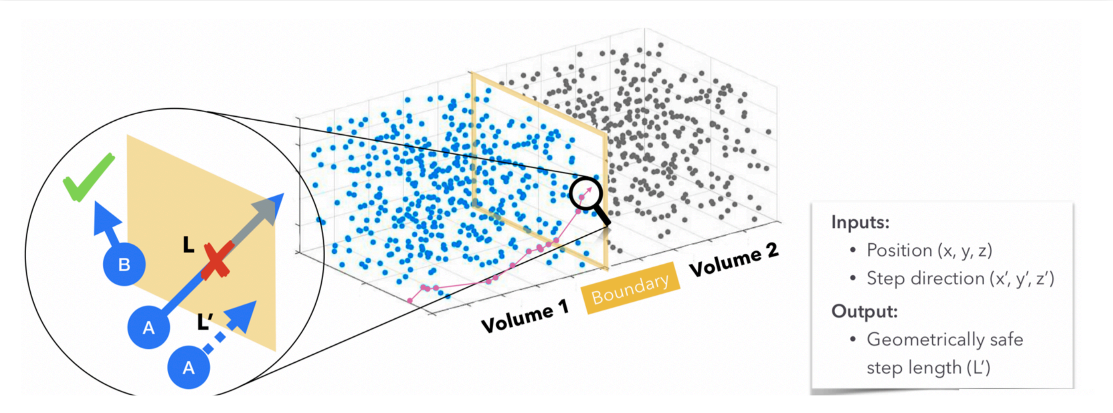
The CMS detector - Image credit

Detector simulation requirements: ATLAS

- Very active **Geant4 Full Simulation Optimization** work ongoing
  - tackling RUN3 but also RUN4
- Taking advantage of **intrinsic performance optimizations** coming with newer Geant4 versions:
  - confirmed ~5% speedup coming from new GammaGeneralProcess + ~7% speedup btw Geant4.10.5 and Geant4.10.6 due to logarithmic calls reduction in EM physics
- Optimization with **tuning of G4 parameters** (physics models, physics lists per regions)
- **Neutron** and **Photon Russian Roulette** + **EM range cuts** (ongoing physics validation)
- **Geometry** optimisations (new EMEC variants + R&D on ML guided steppers in geometry)
- **Magnetic field** tailored switch-off
- **Geant4 linking** as static library (a.k.a. Big library)
- Explore **machine learning options** especially for **simulations optimization:**
  - Machine learning solutions to optimise the detector simulation and optimally tune/re-weight parameters (i.e. physics models, physics lists per regions, range cuts, magnetic field)



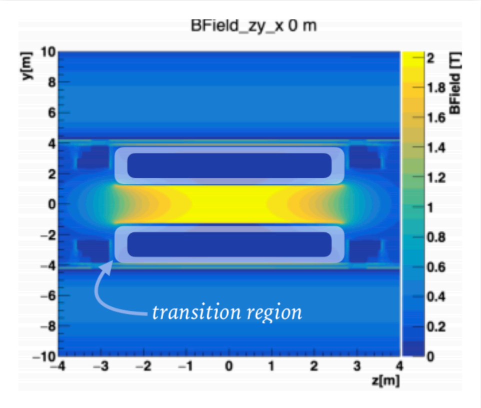
CPU time with NRR + EM range cuts [1]



Inputs:
- Position (x, y, z)
- Step direction (x', y', z')

Output:
- Geometrically safe step length (L')

Machine Learning approach to Geant4 steppers in complex geometries [2]

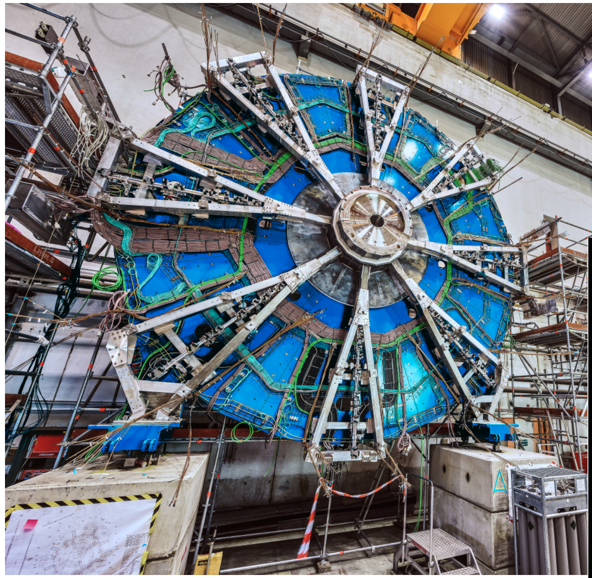


Magnetic Field tailored switch-off [3]

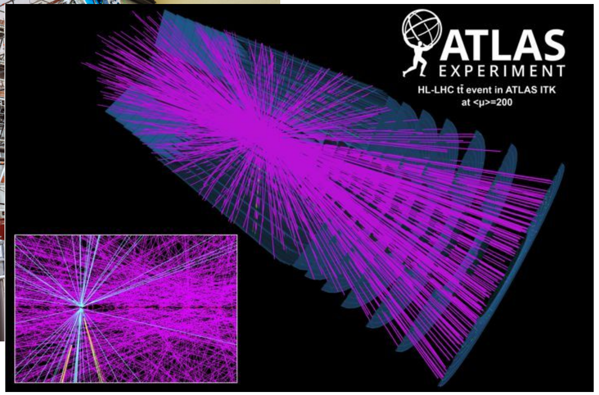
Detector simulation requirements: ATLAS

- ATLAS is undergoing an intensive R&D effort to meet the CPU requirements for HL-LHC
  - fast simulation will cover most of the MC production, but full simulation will be nevertheless needed
- It would be useful to know the projected **simulation speed-ups** for Run3 and Run4
  - Interest in the new Geant4 optimizations, VecGeom navigator plugin [1], new steppers
- Better support of pre-defined decays in G4 EDM (for QS particle sim)
- Making effective use of **GPUs/accelerators** within simulation
  - Interest in trying out R&D simulation prototypes/libraries running on GPUs (i.e. *G4HepEM*)

- Make each **component** of the detector simulation **as independent as possible**, such that it becomes easier to integrate and mix fast and detailed simulation options
- Explore **machine learning options** especially for **fast simulations:**
  - ATLAS plans to use a GAN for some kinematic range of hadrons in the fast calo simulation



ATLAS New Small Wheel Upgrade project [2]



A simulated $t\bar{t}$ event at average pile-up of 200 collisions per bunch crossing [3]

Conclusions and common highlights

- **Heterogeneous hardware interest is high** in all experiment collaborations
  - Intensive R&Ds ongoing in every experiment
    - exploitation of large HPC facilities: major re-engineering/re-implementation of the code
  - Redesigning and developing **detector simulation toolkits to be more efficient** when executed on current vector **CPUs and emerging new architectures**, including **GPUs and Accelerators** is of vital importance to the HEP experiments

- **Great enthusiasm** about new R&D activities that can address the **shortage in CPU power**
  - Big interest in CPU/memory optimization improvements in Geant4

- Request for extended support for **different Fast Simulation options**, where the full detector simulation is replaced, in whole or in part, by computationally efficient techniques:
  - Common frameworks for **fast tuning, validation and optimisation.**

- Developing, improving and optimising **machine learning tools** that can be shared among experiments to make the **modelling** of complex detectors computationally **more efficient, automatic and optimal**.

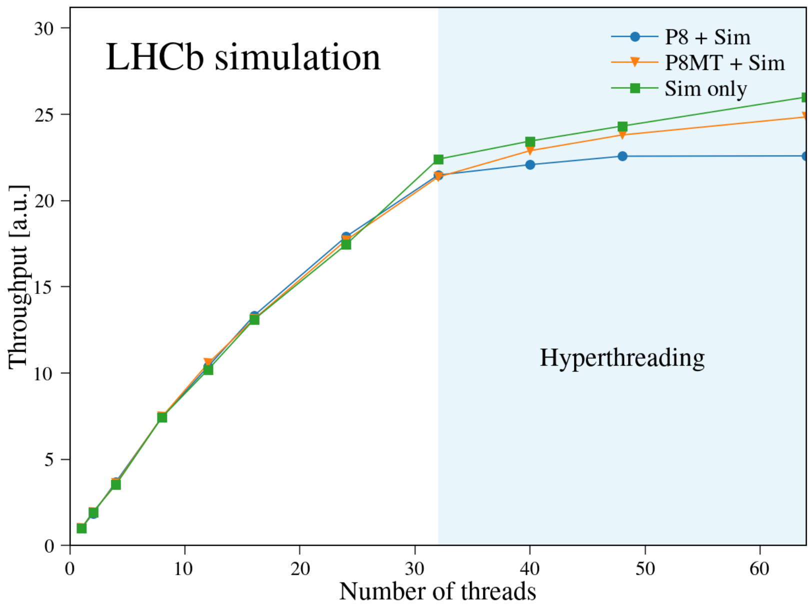
Thanks for your attention!

Marilena Bandieramonte
marilena.bandieramonte@cern.ch

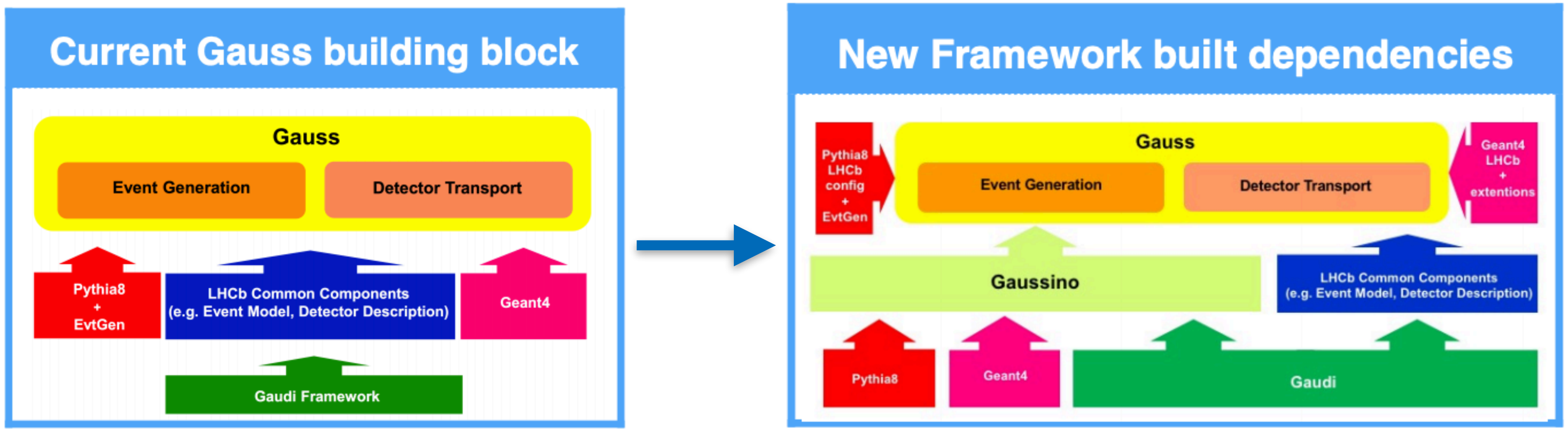
Backup slides

Detector simulation requirements: LHCb

- Future LHCb Simulation will use Gauss-on-Gaussino
- Use what's good in current Gauss
  - Modular
  - Integrated generation and simulation phase
- Use multi-threaded Gaudi
  - Easy Python configuration
- Use multi-threaded Geant4



LHCb-FIGURE-2019-012



Current Gauss building block
Gauss
Event Generation
Detector Transport
Pythia8 + EvtGen
LHCb Common Components (e.g. Event Model, Detector Description)
Geant4
Gaudi Framework
New Framework built dependencies
Pythia8 LHCb config + EvtGen
Gauss
Event Generation
Detector Transport
Geant4 LHCb + extentions
Gaussino
LHCb Common Components (e.g. Event Model, Detector Description)
Pythia8
Geant4
Gaudi