Joint WLCG and HSF Workshop 2018
Napoli, Italy
26th – 29th March 2018

LHCb
Preparation for Run3

i.e. "Upgrade" of Simulation software to enable physics studies with Run3 data

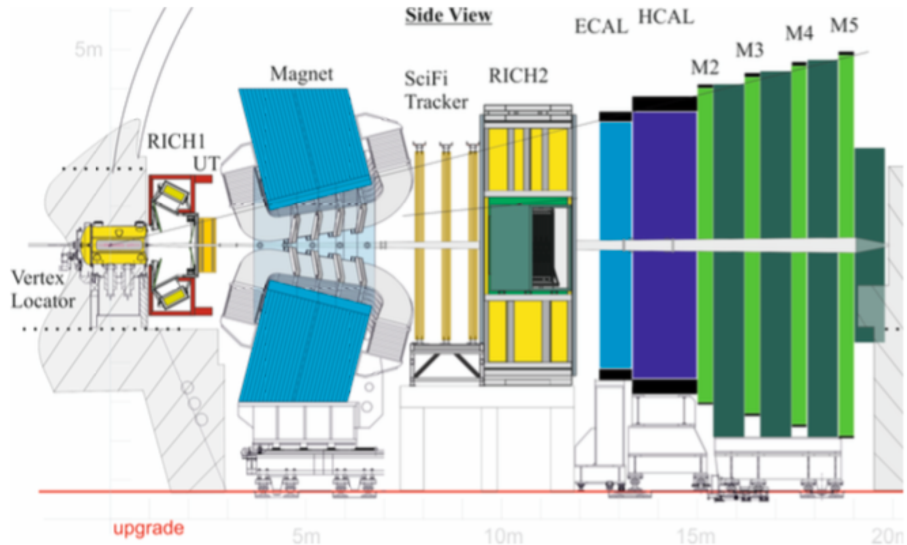
Gloria Corti, CERN

Setting the scene

LHCb Upgrade for Run3

L_{inst}: 4 10^{32} → 2 10^{33} cm^{-2} s^{-1}
μ: 1.1 → 7.6
L_{int}: 3(Run1) + 5(Run2) fb^{-1} → 50fb^{-1}

- full software trigger with high signal purity
- analysis directly on trigger output

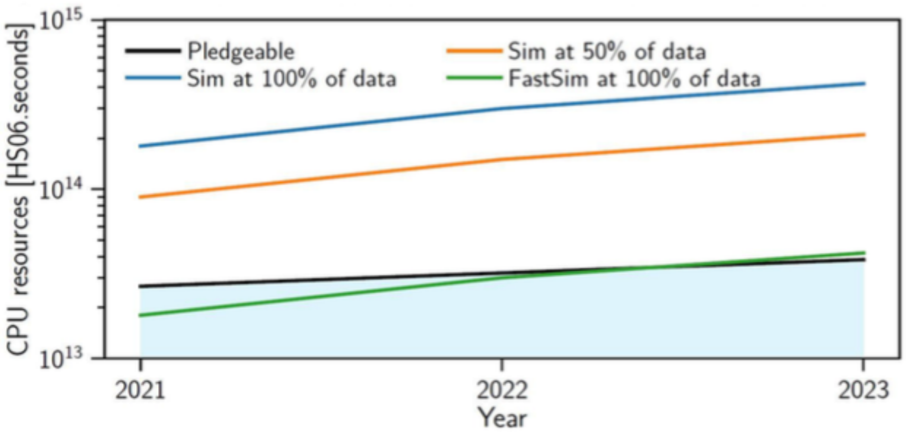


Simulation will be dominating the CPU needs, even more than today

C. Bozzi, "Challenges for LHCb trigger", 26 March @ this workshop (https://indico.cern.ch/event/658060/contributions/2939756/)
R. Matev, "Real-time analyses - the LHCb case" , 27 March @ this workshop (https://indico.cern.ch/event/658060/contributions/2889119/)

Need of faster detector simulation software in Run3

Collecting more interesting events will require more events to be simulated



Legend:
"Sim at 50% of data" = FullSim sample is 50% the datasize, FastSim sample is 50% the datasize

FastSim speed assumed to be 1/10 of FullSim

MUST speed up the simulation

by implementing faster or parametrized simulations

by reducing the CPU consumption of the full Geant-based simulation

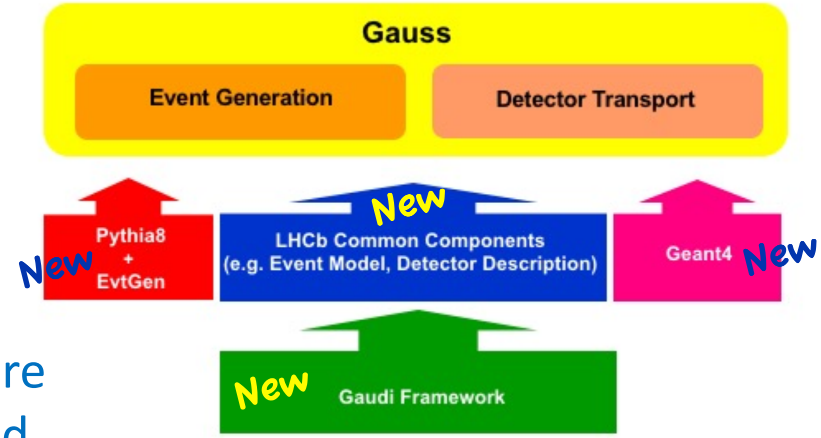
while maintaining high quality physics modeling

Setting the scene

Software modernization

Run3 is the occasion to modernize the whole LHCb software

New Multi-threaded Gaudi framework
- Better use of multi-processor CPUs
- Reduce memory usage
- Optimize cache performance
- Remove dead code
- Modern data structures
- Enable code vectorization
- Enable algorithmic optimization

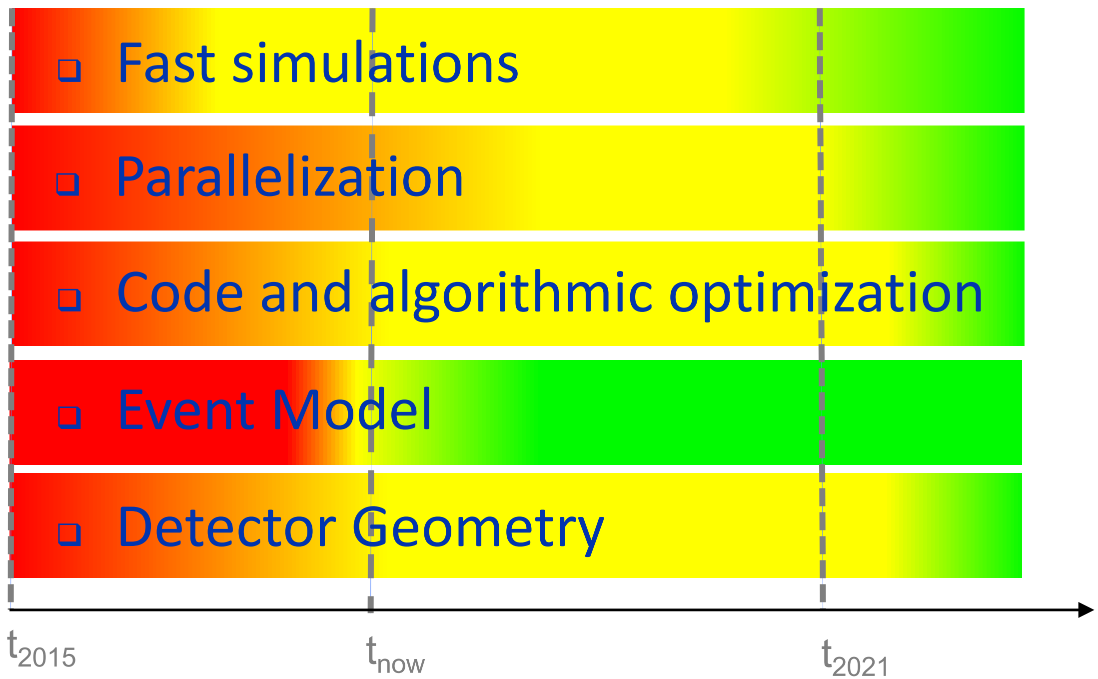


In the simulation software also exploit more modern features in HEP-wide libraries used

'Newish' Multi-threaded Geant4
- Same as above

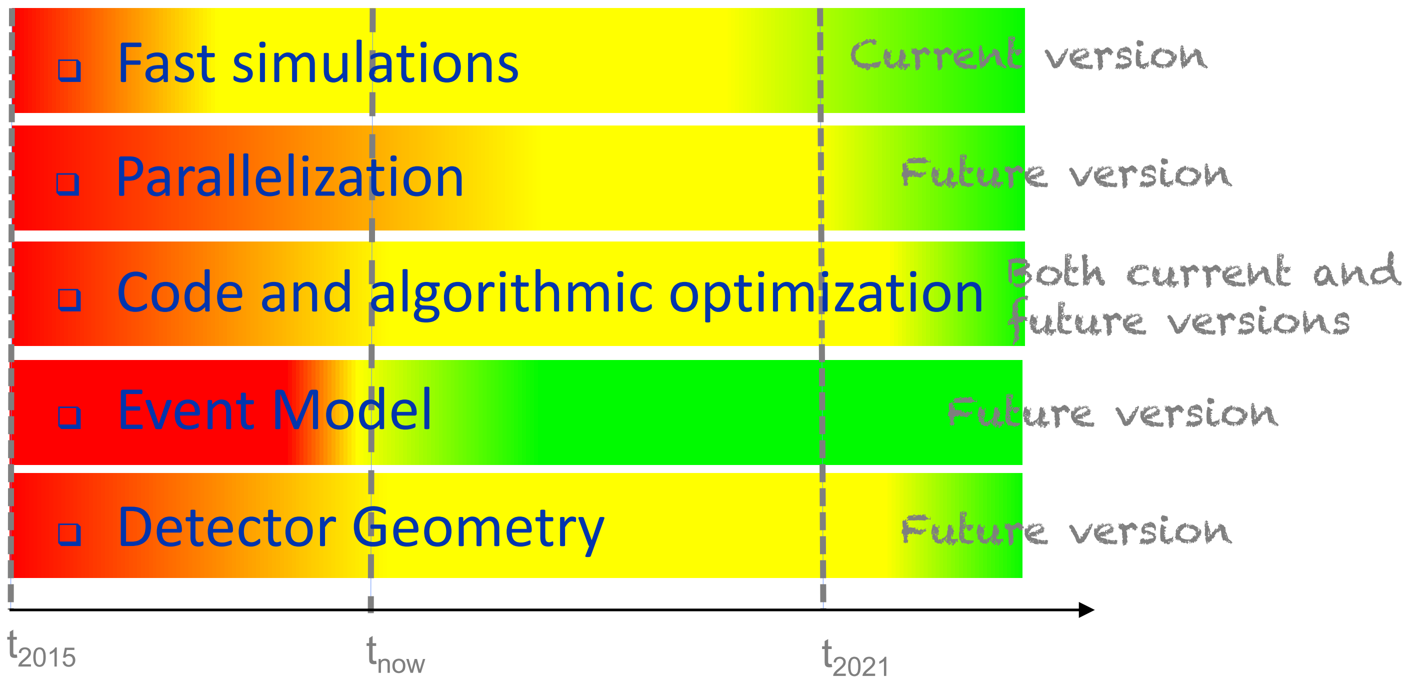
Thread-safe generators, e.g. HepMC 3

Strategy & main area of development in LHCb simulation



Develop & deploy in current simulation framework as much as possible, while implementing and testing new simulation framework

Strongly depend on overall LHCb choices and their implementations



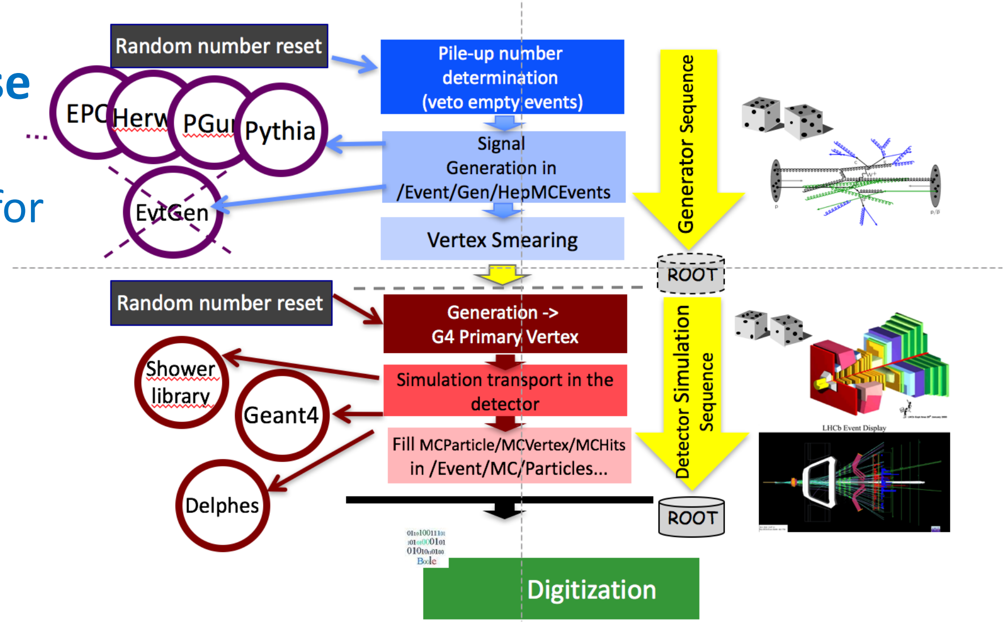
Develop & deploy in current simulation framework as much as possible, while implementing and testing new simulation framework

Integrated simulation framework

No single solution for all needs but different simulation options organized under the Gauss unique framework

Building an integrated, easy to use and **safe** simulation framework

- Working fast simulation solutions for the whole event
- Prototypes allowing a mix of simulation flavors for different detectors and particles
- Output compatible with what expected by later processing



Benchmark and physics performance measurements to choose baseline production setting

Fast simulation options

Broad investigation deploying solutions when mature for physics

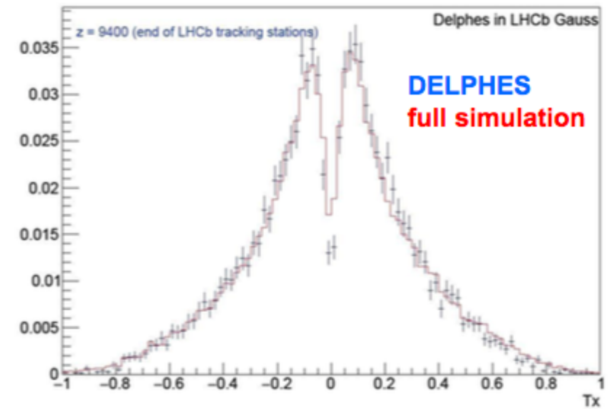
- Simplified detector simulation
  - Reduced detector: RICH-less or tracker-only. In production
  - Calorimeter showers fast simulation. Under development
  - Muon lower energy background, used with full muon detector simulation. In production
- Simulation of partial event
  - Simulate only particles from signal decay. In production
  - ReDecay, e.g. use N-times the non-signal decay part of the event. In production
- Fully parametric simulation
  - Parametrized tracking, calorimeter and particleID objects with a DELPHES-based infrastructure. Under development

M.Rama, "Fast Simulations in LHCb" @ 2017 IEEE NSS and MIC
D. Muller, "ReDecay, a method to re-use the underlying events to speed up the simulation in LHCb", @ 2017 IEEE NSS and MIC

Fully parametric fast simulation

- Work in progress on a fully parametric ultra-fast simulation based on the DELPHES package
  - Parametrizes not only the detector response but also the reconstruction
- Crucial to cope with large amount of simulated statistics needed for Run3 and future Upgrade II. **Goal: 100-1000x faster then full simulation.**
- Functional prototype integrated in the current Gauss framework
  - **Tracking** efficiency and resolution
  - **Primary vertices** reconstruction
  - **Photon** calorimetric objects
  - Output **LHCb reconstructed high level objects**, compatible with the experiment analysis tools

px/pz distribution



J. De Favereau et al., JHEP 02 (2014) 057
D. Muller, B. Siddi, "Fast simulation options in LHCb from ReDecay to fully parametrised", LHCC 2017 Poster Session

Fully parametric fast simulation

- Deployment in steps with a first beta release for user to try out with some physics analysis coming soon

- In parallel extensive development to obtain all relevant quantities and replace all applications between Gauss and physics analysis input
  - Include error covariance matrices in **charged particle tracking**
  - Parametric **calorimetric response** for all electromagnetic particles
  - **Particle identification hypothesis** to be integrated adapting particle ID calibration mechanism for data

- Parametrization fully experiment dependent and specific to a given data taking period
  - To be properly calibrated for precision measurements it requires both extremely reliable full simulation and highly precise data-driven measurements of the detector performance

Fast simulation of the Calorimeter system

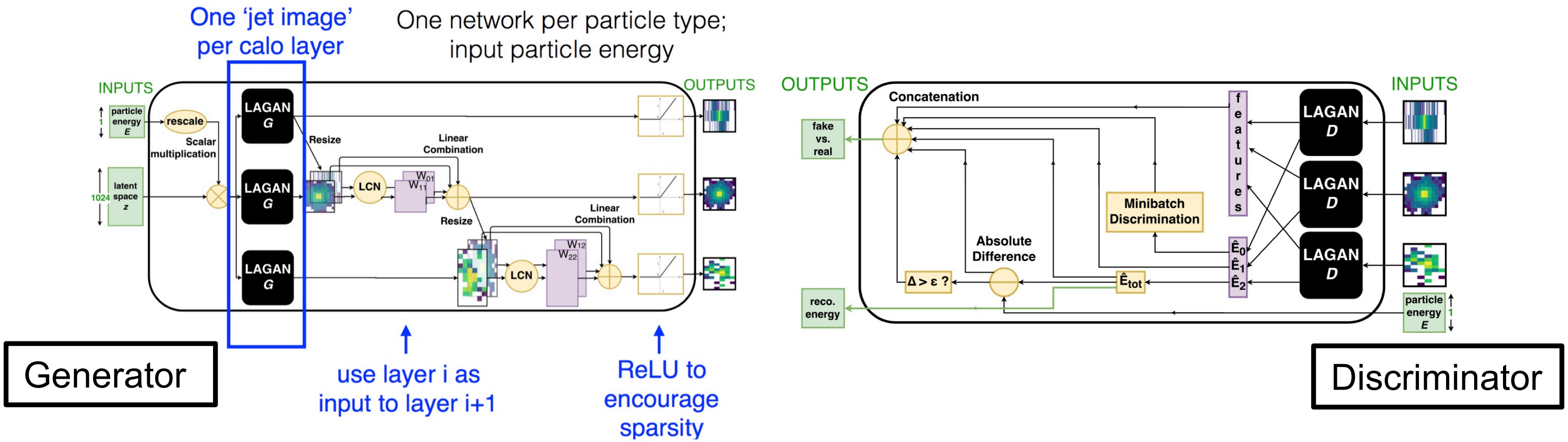
- The modeling of the calorimeter system is the most time consuming in the LHCb Geant4-based simulation
- A generic infrastructure has been developed in the Geant4 interface of the current Gauss framework to allow to interrupt the G4 transport for a given particle and inject a given fast simulation counterpart
  - Makes use of Geant4 utilities (e.g. G4Region)
  - Create the same calorimeter cell hits with identical format as those produced from Geant4
  - Works with any fast implementation – also for detectors other then calorimeter
  - Expose to physics community in a few months – early feed back
- Potential issue with punch-through to downstream muon detector could be solved with muon low energy background dedicated parametrization

Fast simulation of the Calorimeter system

- Two fast parametrization solutions currently under development
  - *Classic* Frozen Shower Libraries
  - Hits generation based on Generative Adversial Networks (GAN)
  - … not necessarily mutually exclusive ☺. Could solve the Shower Library problem of a fast search in multi-dimensional phase space by reducing the dimensions with Machine Learning techniques, e.g. autoencoders

- Aim to speed up by factor 3 to 10 the simulation of the calorimeters
  - Timing study with dummy filling of calorimeter cells shows overall speed of full LHCb detector reduced by a factor of 2

GAN for LHCb Calorimeters



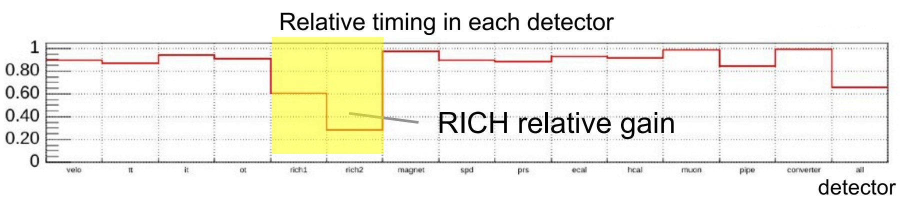
- Starting from latest configuration of **CaloGAN,** a new Machine-Learning method based on a **generator**, trained to maximize goodness of produced sample, and a **discriminator** to classify **images** (in HEP applicable to jets, clusters)
- Very fast response, but generally long training
- First look with simple mock-up of LHCb ECAL and signal particle gun reproduce the shape reasonably well , need to now tackle variativity. Huge range of energies may be difficult to cover by single generator

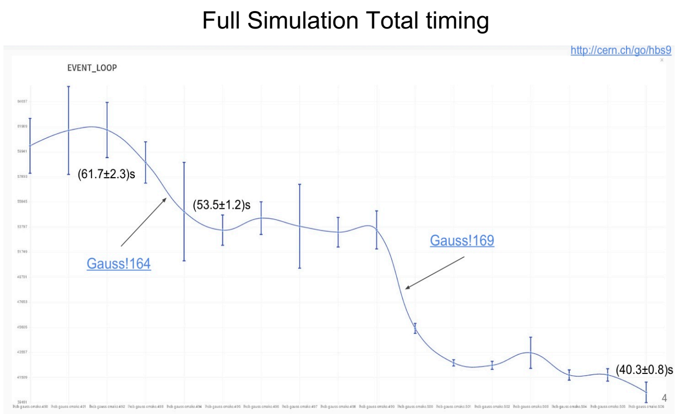
B. Nachman, M. Paganini, L. de Oliveira, http://arxiv.org/abs/1712.10321

Algorithmic optimization and timing

Complementing fast simulations with algorithmic optimization in full simulation

Dedicated studies for speed up of RICHes





- improve treatment of fraction of background scintillating photons
- 'stop' Cherenkov photons as early as possible in the chain rather then at the end

Benchmarking is essential to know how we are doing and in which direction we should proceed, keeping also the physics results under control

- Timing test running in the LHCBPR2 framework on a dedicated machine together with checks of basic and higher level physics quantities

Geant4 and timing

- Most of the time to produce the majority of the simulated samples in full simulation is spent in Geant4
- LHCb is in the process of validation new productions with Geant4 10.4 and exploring faster ways to use it
  - Clear indication of a significant speed-up moving from Geant4 9.6 to 10.4 combined with the move from gcc 4.9 and 6.2.
  - No indication of speed-up using new VecGeom geometry package. Trying to understand why that is the case: mostly simple shapes?

Geant4 v10.3.3 and VecGeom v0.3.rc.
Tested with:
- Scalar backend + SSE4.2
- Scalar backend + AVX2
- Vc backend + AVX2

Shapes used in LHCb for current detector

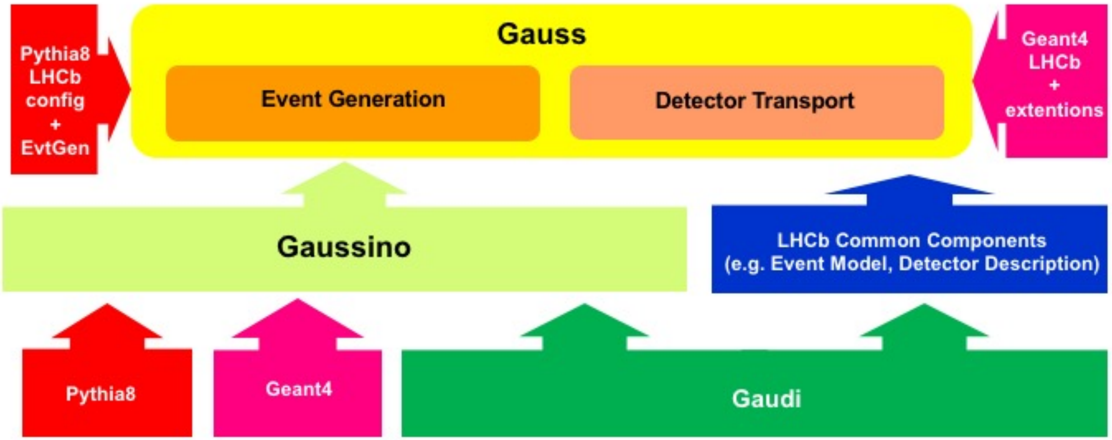
Shapes	Quantities
G4Box	2219
G4Cons	122
G4Polycone	20
G4Sphere	133
G4Trap	399
G4Trd	18
G4Tubs	1490

From current Gauss to future Gauss

- Take advantage of modern technologies by building the future version of Gauss for Run3 on the new Gaudi and LHCb software
- Take the occasion to restructure the code introducing an experiment independent layer to also use as test bed for new idea and benchmark alternative options

Gauss future: built on top of Gaussino, new evt model, has the LHCb-specific parts

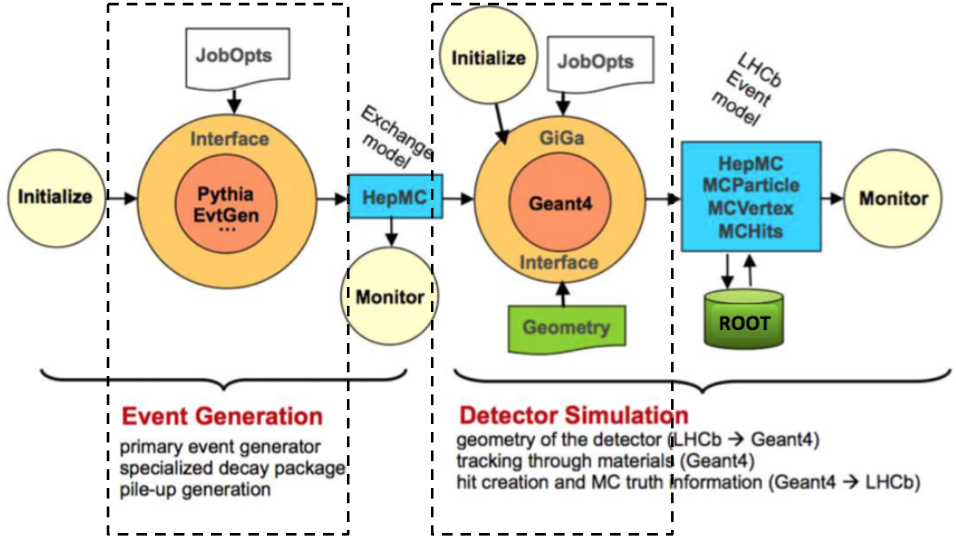
Gaussino: generic core, born as LHCb-independent Gauss, includes GaudiFunctional, thread-safe, Geant4 MT, const TES



Gaussino

- A complete simulation framework following the basic Gauss architecture

in collaboration with SFT (FCC)



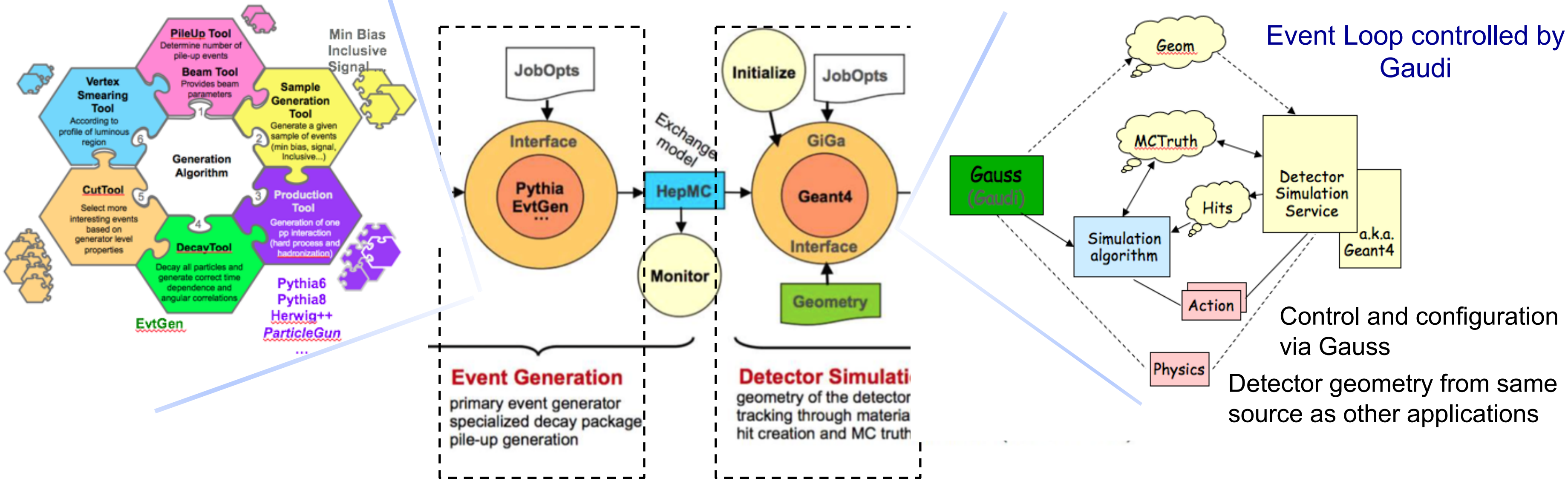
Generator phase
can be used as-is
even though it should be made thread-safe

Simulation phase
requires some re-design
also to take advantage of Gaudi and Geant4 MT

Gaussino

- A complete simulation framework following the basic Gauss architecture

in collaboration with SFT (FCC)



Generator phase with Pythia8 interface implemented

Code migrated to HepMC 3

Simulation phase under redesign with identification of elements.

Aim to keep it ‘simulation engine’ independent

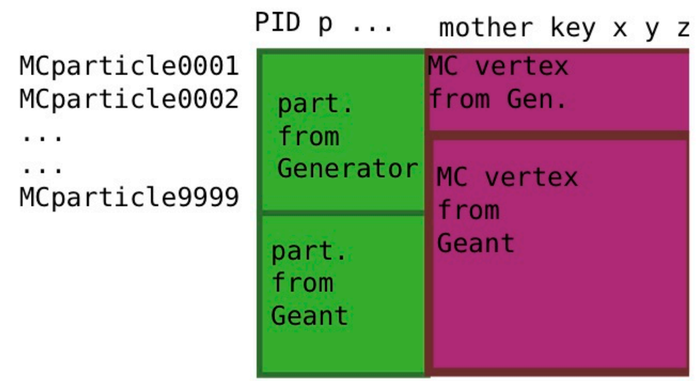
Multi-Threading in Gauss

- Many open issues that require prototyping to choose
- Detailed design of MT interplay for Gaudi to control Geant4
  - MT at event level: event = Gaudi event?
    event = Out of time bunch crossing?
    event = proton-proton collision?
  - Generation in separate thread preferred option
  - Reproducibility: how to distribute random seeds to threads
  - Drop use of double-inheritance classes (similarly to ATLAS)
- Fast simulations and multi-thread or thread-safe?
  - Inherently faster but what happens when mixing them

Event Model for LHCb Simulation

- Simulation of **sequential processes involving particles and vertices** provide results at different time, not trivial to manage with **const TES**

- **Modular/fragmented structure**
  - add columns (container) if needed
  - To be linked together for reading later, **must be transparent for the user**



- **Relations** tree of MC truth has to be accessed in both directions. Decision: relations created once and stored

- Trying out few options for MC particles and Vertices with Gaussino

- Replaced obsolete HepMC2 with **HepMC3** (thread-safe, C++11, etc) in Gaussino. Drop it from LHCb Event model and transfer all information needed for physics studies in MC truth LHCb objects

What about the simulation for Run1 and Run2?

- The simulation software has to keep providing **samples for physics analysis with Run1 and Run2** data until 2022 with latest improvements in physics models
- We have limited amount of human resources

The future version of Gauss should simulate events also for Run1&Run2

Boole, the digitization application, instead simulate the response of a different detector, hence the version for Run1&Run2 will be frozen

Two additional constrains for Gauss to support all Runs

- Write out the Event Model classes needed by the subsequent applications
- Use the same geometry, calibration and alignment that the trigger and reconstruction use for a given data taking

Summary

- The challenge for the LHCb simulation for Run3 is to cope with the amount of simulated samples needed both in term of statistics and accuracy
- Extensive development is being carried out on parallel lines to meet this challenge
  - A wide variety of fast simulation options are actively pursued ranging from fully parameterized to fast detectors response to reuse of events
  - Ways to speed up the simulation are being explored from code modernization to algorithmic optimization
- A major rewrite of the LHCb simulation framework, Gauss, is in progress to adopt the new Gaudi and Geant technologies and at the same time encapsulate experiment agnostic and LHCb specific parts
  - Gaussino as underlying generic simulation framework and prototyping test bed
  - Gauss as a fully Integrated Simulation Framework allowing the most suitable mix of simulation options for a given Run3 LHCb physics analysis