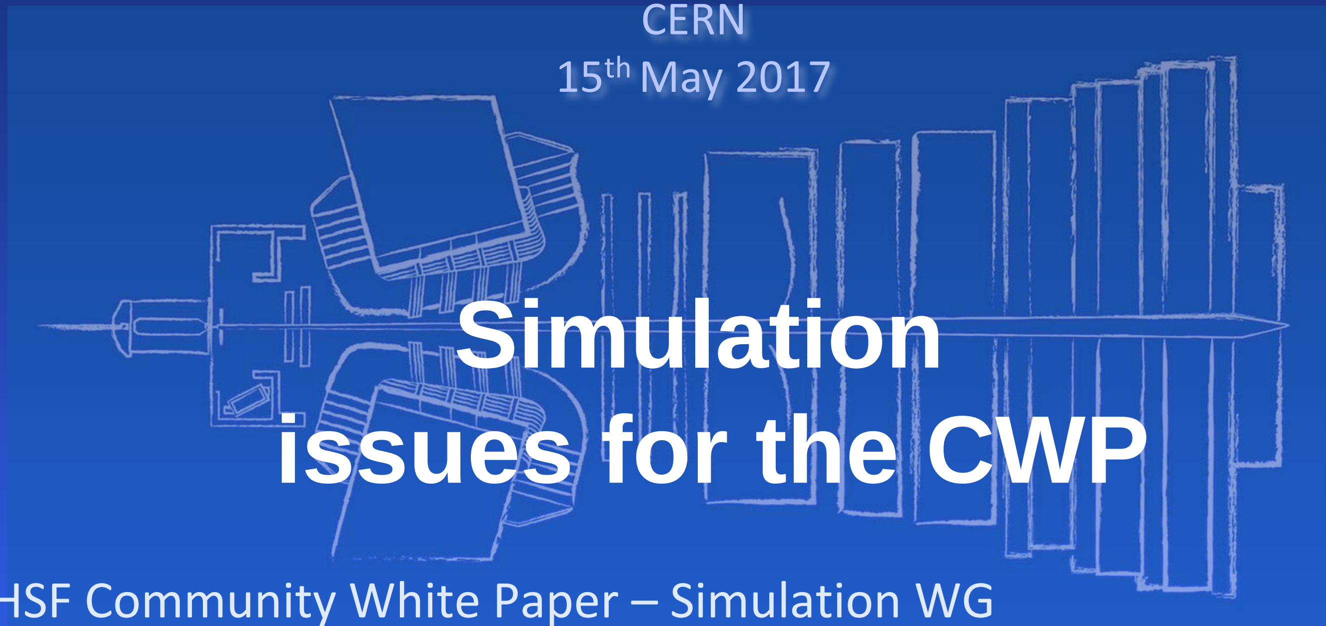
LHCb Computing Workshop

CERN
15th May 2017



Simulation issues for the CWP

HSF Community White Paper – Simulation WG

LHCb input and timeline – work for Upgrade Software Simulation

What do we think it is important to have there?

Gloria Corti, CERN

on behalf of the Simulation PPWG

HSF Community White Paper WG

Simulation in CWP

- Full and Fast Simulation of HEP experiments to achieve
  - Software efficiency, scalability and performance and make use of technological advanced (CPU, storage and network)
- To enable
  - New approaches to radically extend the physics reach of the experiments
- To ensure
  - Long term software sustainability through the lifetime of the HL-LHC

WG activities

- Topical meetings between few people of LHC experiments and detector simulation current and future toolkits, e.g. Geant4, GeantV

25-27 Oct 2017	Community Meeting on GeantV R&D
27 Feb	Computing Performance
27 Mar	Fast Simulations
24 Apr	Geometry – Digi
22 May	Physics Modeling

- Find commonality between experiments and define whenever possible common measurements matrixes
- Next summarize and formalize in document
  - https://docs.google.com/document/d/1DVyxBDTFts44-sGYrJ7tLt-bjd4jmA41XPezKOWgyil/edit# ← *LHCb copy*

LHCb upgrade needs – Input for CWP

Simulation needs for Phase 1 upgrade

"We want a fast MC!"

Ever growing datasets require even large samples of simulated data

Ideally we want more ***selected*** MC events than have in data

Why ?

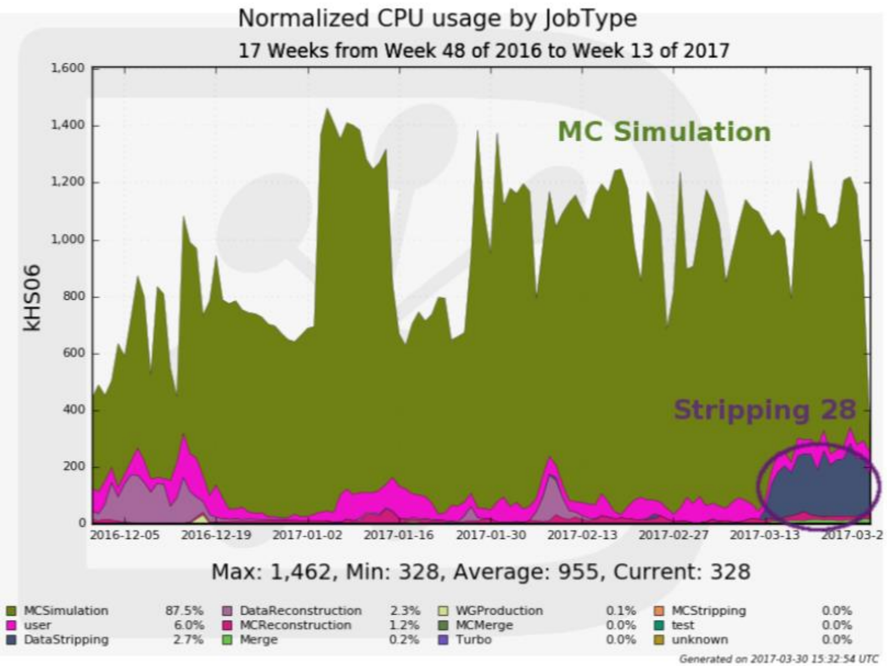
We can [ONLY] simulate 10M events per day at peak (using all our Tier 2 sites)

No changes in computing funding expected
Assume reduced CPU consumption for Simulation

It costs 6k to produce 10M events
relating it to the funding cost of the grid

The simulation is already the biggest consumer of computing resources.

O(100) x more events to be simulated for Phase 1 analyses



Simulation needs for Phase 1 upgrade

"We want a fast MC!"

Ever growing datasets require even large samples of simulated data

Ideally we want more ***selected*** MC events than have in data

Why ?

We can [ONLY] simulate 10M events per day at peak (using all our Tier 2 sites)

No changes in computing funding expected
Assume reduced CPU consumption for Simulation

It costs 6k to produce 10M events
relating it to the funding cost of the grid

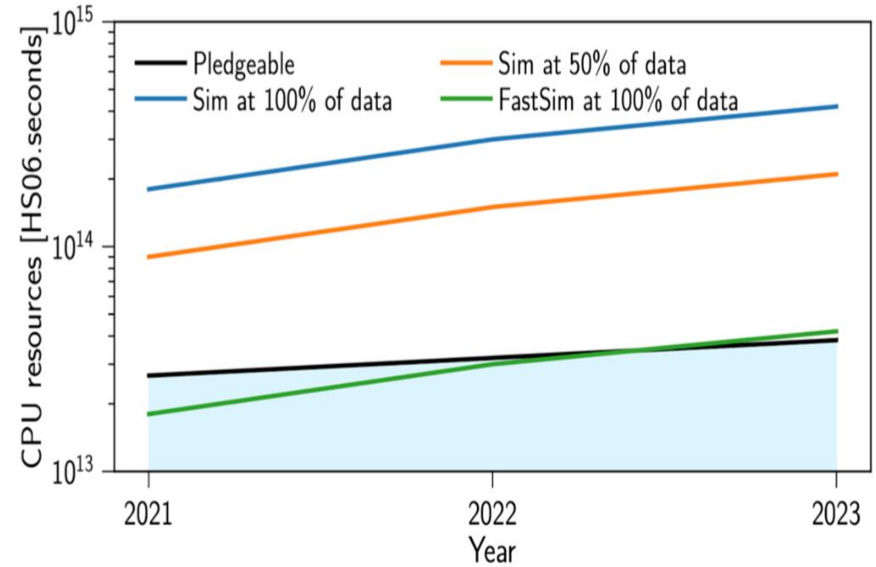
The simulation is already the biggest consumer of computing resources.

We need to make all efforts to fit within the available pledged computing resources

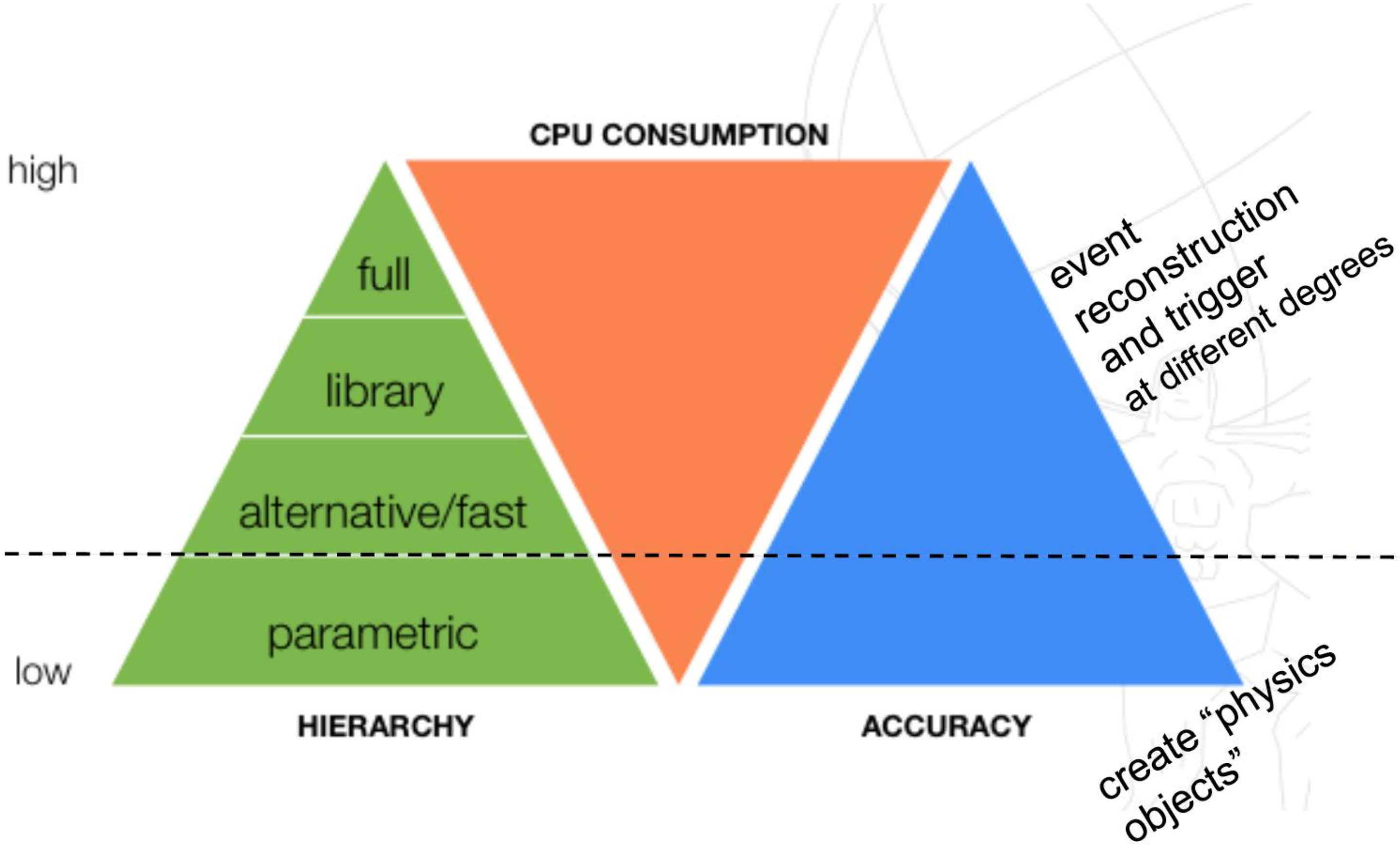
We need a working solution in place by 2020

O(100) x more events to be simulated for Phase 1 analyses

Simulation for LHCb Upgrade



… How ?



No solution fits all!

Parallel lines of development

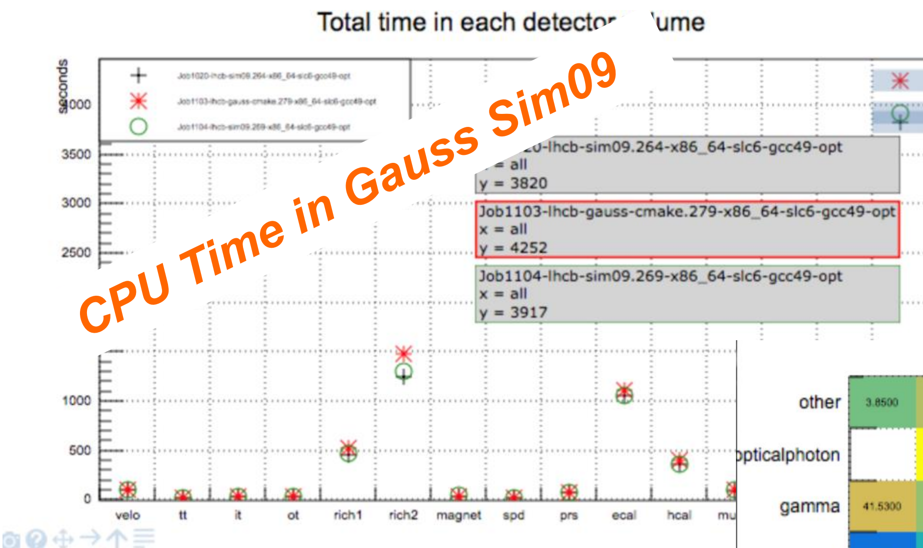
- Fast simulations – wide variety of options being explored from fully parameterized to fast detectors response to reuse of events
  - A lot of progress in developing new options
- Ways to speed up the simulation – GaudiHive, GaudiMP, Geant4 MultiTreading, Geometry
  - Co-existence of different philosophies
- Evolve Gauss to an integrated simulation framework with LHCb specific and experiment agnostic parts
  - Collaboration with FCC on Gaussino (experiment agnostic)
  - Integration of alternative ((ultra)) (fast) simulations

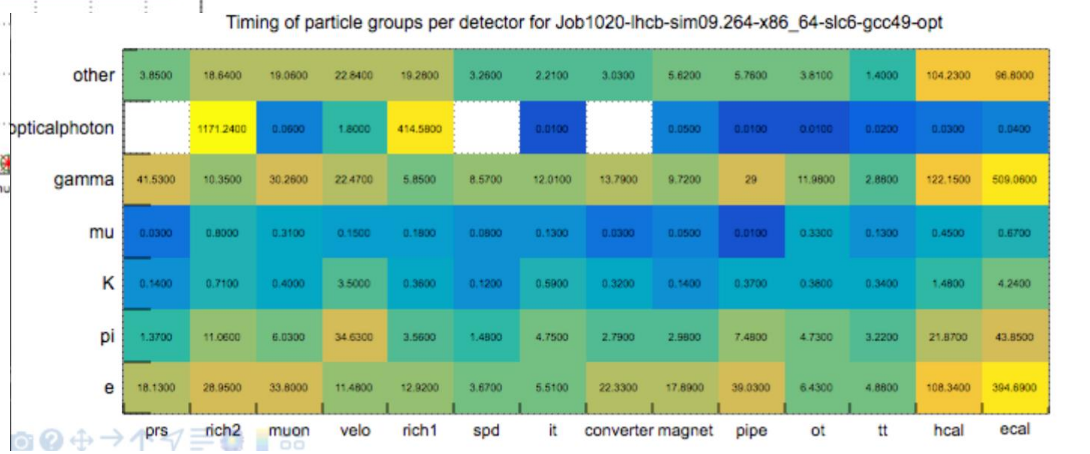
Measure where we can gain

- Concentrated in the past on disk usage
  - Identifying MC objects really needed, packing, microDST format
- Now concentrating on CPU
  - Biggest gains will come from physics changes rather then algorithmic changes
  - But we should not ignore them, for one thing we need to fit the simulation in the new future framework and exploit the online farm hardware
- Memory is not yet an issue for us
- And we should not forget purely technical issues as compilers

Measure where we spend CPU

- To identify where to focus for fast simulations
- To identify where to optimize algorithms
- To check and validate choices – physics modeling is the benchmark





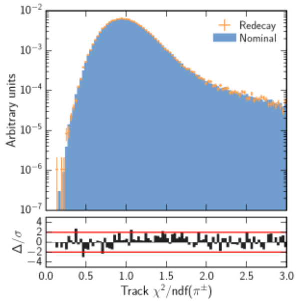
Now run in LHCb nigthtlies

Fast simulations

- Many options have or are being worked on
  - Deployed as they become available for current detector

- No single size fits all but pick and choose as most appropriate with multiple options organized under a unique framework
  - Build on and ‘upgrade’ the Gauss framework to mix simulation flavors, including for different particles in the same event
- Benchmarks and performance to choose baseline combination for most

Fast simulations options



ReDecay

Decay signal N times with same underlying event

Under code review for release

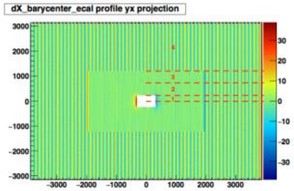
CPU and disk < O(10%)

Calo Shower Library

Prototype of library set up

Studying shower characteristics

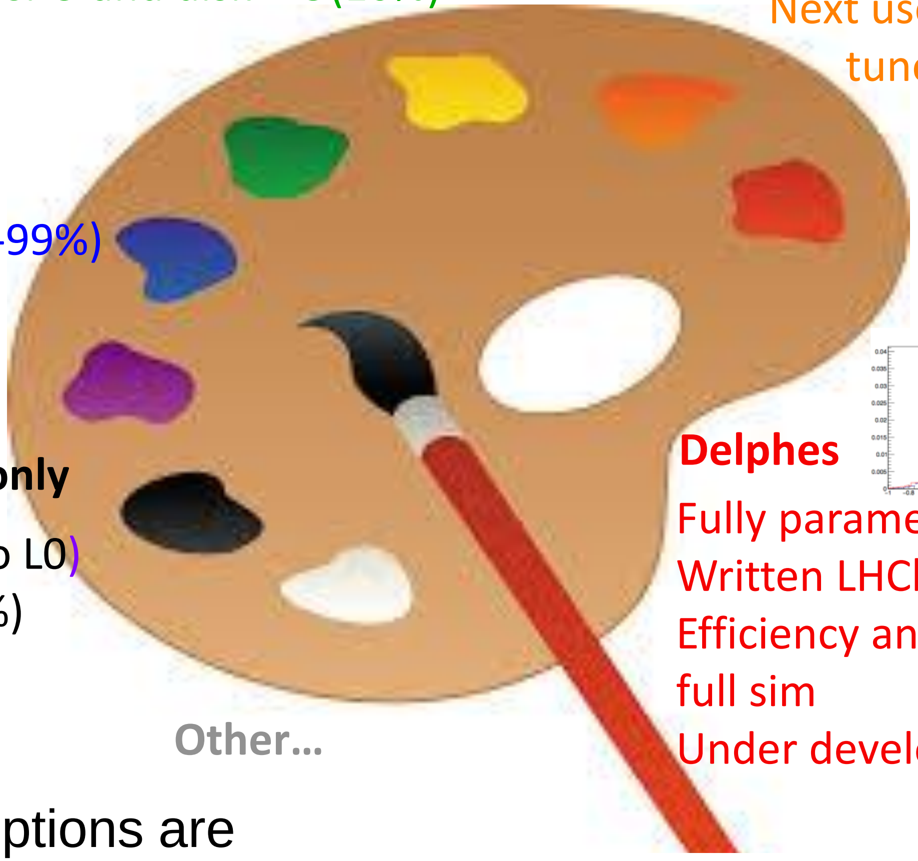
Next use and tune showers



Signal ParticleGun

In use (CEP & other)

CPU and disk < O(95-99%)

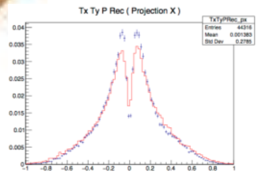


Partial detector

RICHless or Trackers only

In use (special HLT, no L0)

CPU < O(30%), O(80%)



Delphes

Fully parametric ultra-fast

Written LHCb propagator

Efficiency and resolution from full sim

Under development

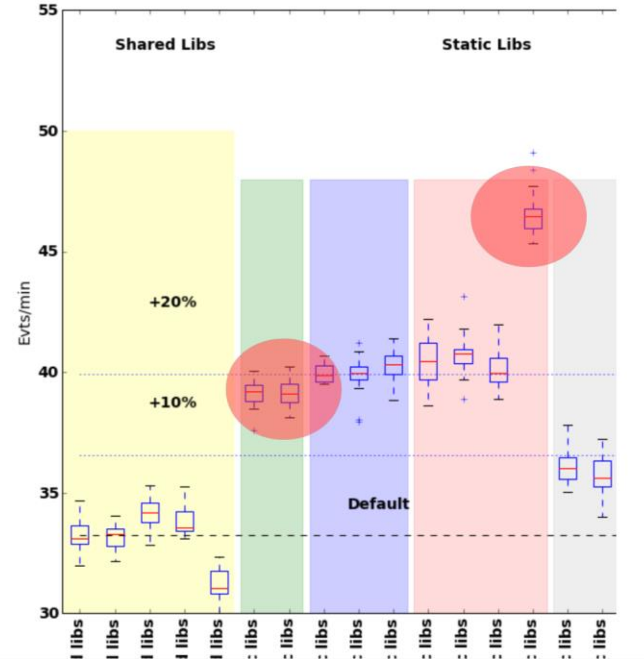
Other…

Some of these options are very specific to LHCb Physics

D. Muller *J.-F. Marchand & M.Rama* *B. Siddi*

CPU and Geant4

- Moving to Geant4 10.(2)3 in next major version of simulation for production in 2nd half of 2017 – in sequential mode
- Overall performance improvement with G4 10
  - Multi-threading with event level parallelization will likely not gain much
  - Compiler gains for free?
  - ATLAS observed ~ 15% speed up v10.1 vs 9.6 (same memory footprint)
  - ATLAS & CMS observed ~ 10-20% speed up static vs dinamic libraries – more for MT applications



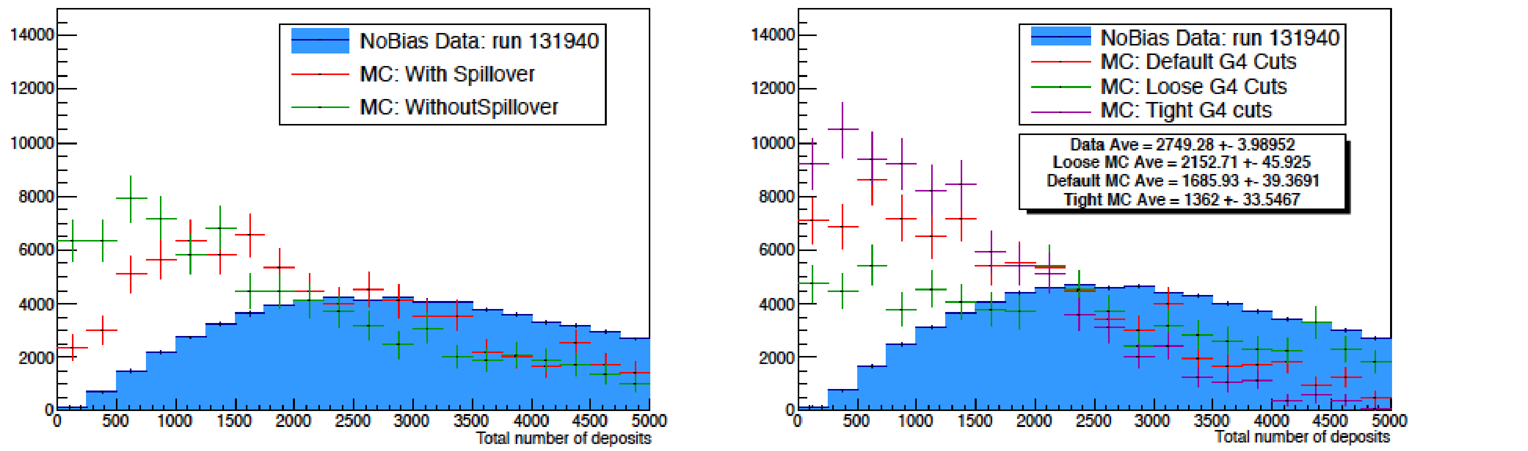
@ ATLAS

Is it the same for us? Need to measure it!

We should be able to do it with Gauss

CPU and Geant4

- And we should not forget we need to review our choices of simulation parameters and effect on CPU



We need a more fine grained setting of parameters this will impact the physics most

- One more issue will be the higher pileup and spillover
  - Although they may not have the same solution

Parallelization (G4 and Gaudi)

- Investigated how to use G4 10 MT in Gauss as of now
  - We could use it for spill-over or pileup
  - We need to re-implement quite a few GiGa classes
- We also need to understand how to make Gaudi parallel and G4 MT play nice in a parallel world
  - Different concurrency models and parallelism
  - They manage their 'components' in different ways
- We are not the only nor the first
  - Area of collaboration with FCC and potentially with ATLAS (that has[is] integrating G4 MT in Gaudi parallel)

"We should move Gauss to G4 MT and Gaudi parallel in a single go and develop a minimal fully functional Gauss' to try this out"

c.f. D. Popov, G. Corti

Parallelization (G4 and Gaudi)

- Investigated how to use G4 10 MT in Gauss as of now
  - We could use it for spill-over or pileup
  - We need to re-implement quite a few GiGa classes
- We also need to understand how to make Gaudi parallel and G4 MT play nice in a parallel world
  - Different concurrency models and parallelism
  - They manage their ‘components’ in different ways
- We are not the only nor the first
  - Area of collaboration with FCC and potentially with ATLAS (that has[is] integrating G4 MT in Gaudi parallel)

“We should move Gauss to G4 MT and Gaudi parallel in a single go and develop a minimal fully functional Gauss’ to try this out”

c.f. D. Popov, G. Corti

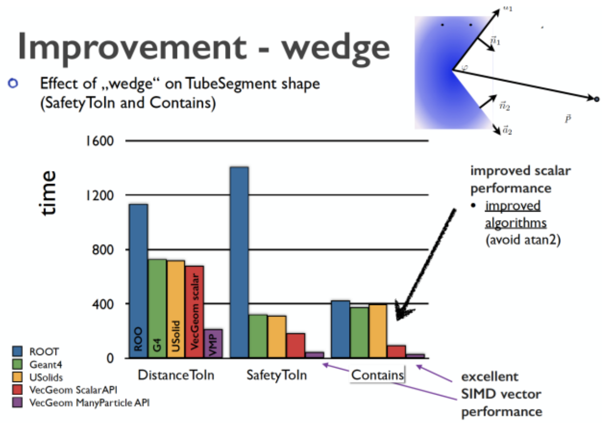
Geometry

- We need to test new geometry mechanisms and how they impact the simulation
  - A lot of time in the simulation is spent transporting particles through volumes and in the magnetic field

- We can check different simulation engine descriptions – developments in Geant4/GeantV

Different options with Gauss based on G4 10
(a Geant4 compilation choice)
- USolid shape library in G4 10.2
- VecGeom library in G4 10.3

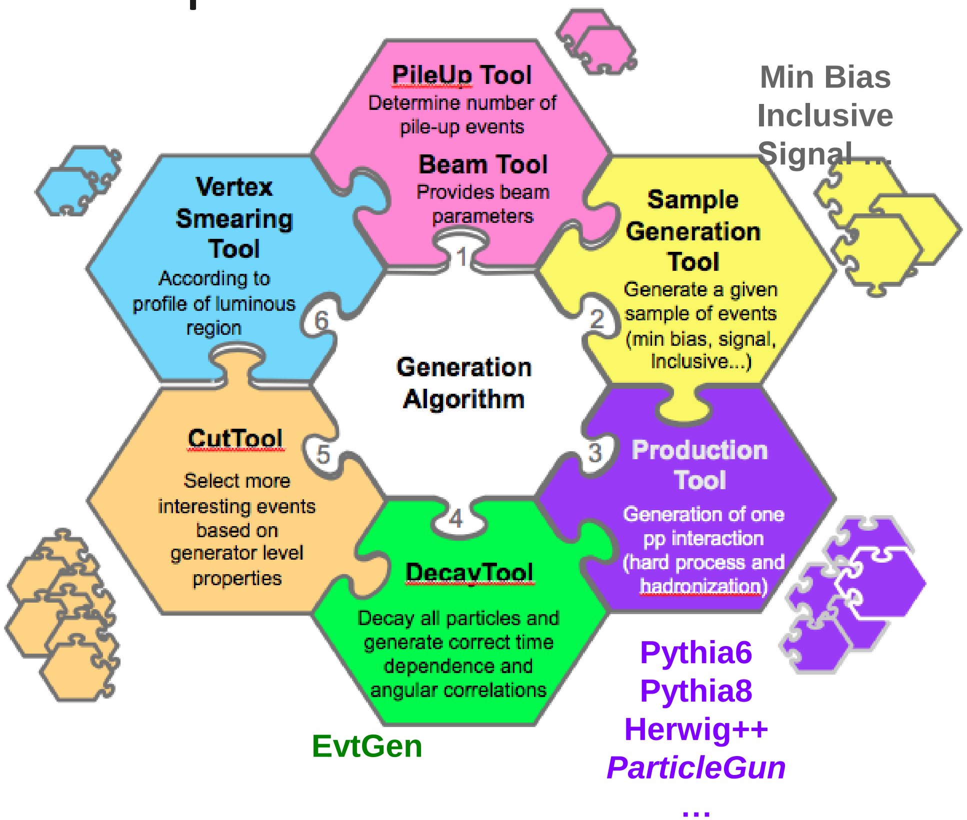
Preliminary tests do not show differences in CPU with Gauss+USolid



- Investigation of DD4Hep vs. current detector description not a priority at the moment (and no resources) – synergy with general work on geometry
  - DD4HEP adaptor available from FCC
  - LHCb detector description adaptor to be implemented

Generators

- Not much of a gain in parallelizing the generator code but should make it thread safe – and in fact not in the scope of Simulation CWP



- Thread safety requires migration to HepMC 3
  - But only a beta release
- Many special generators still in FORTRAN

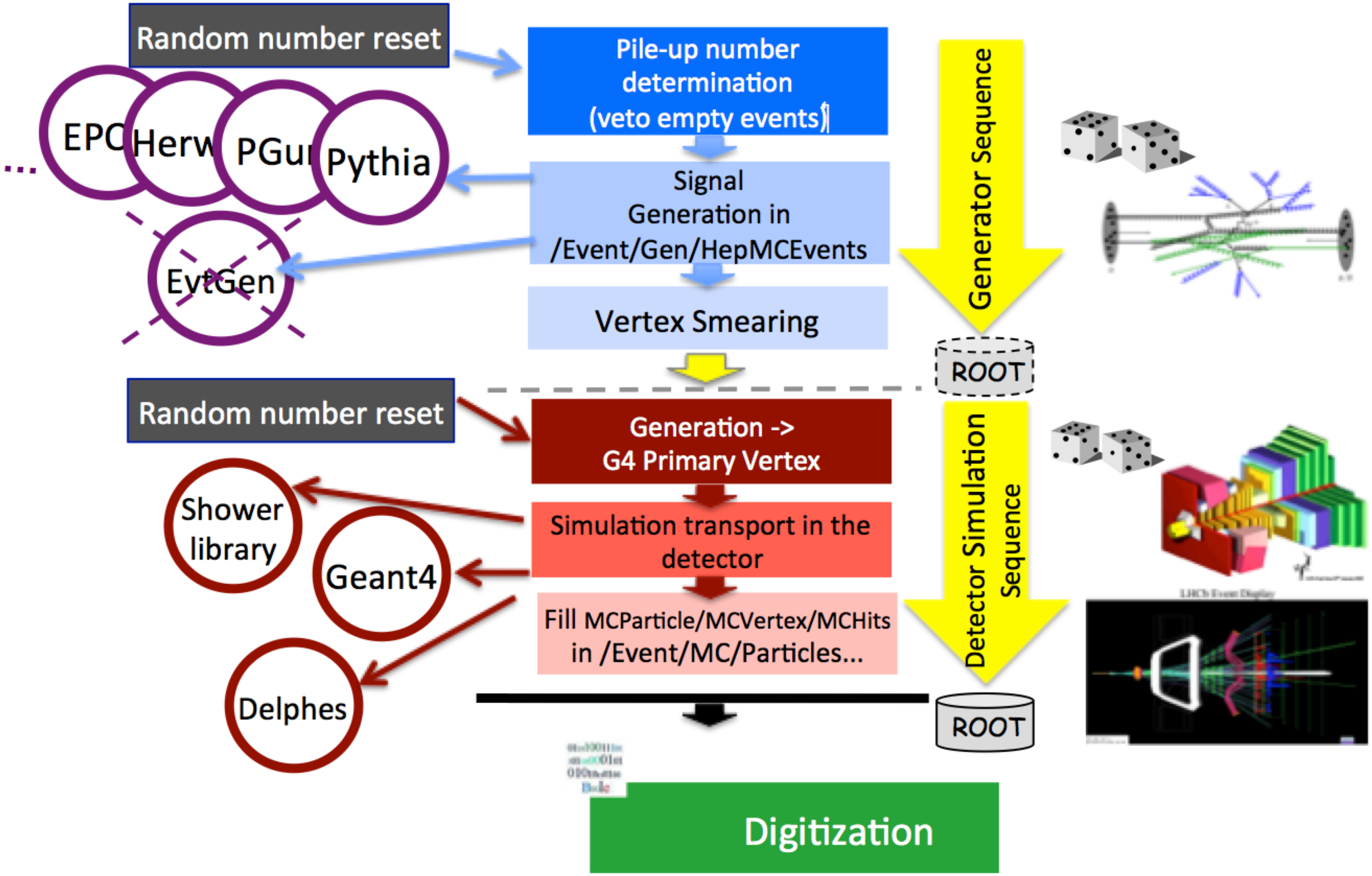
Not the most urgent topic but look into running generators as FIO to parallel simulations
Rather how they fit with the rest – for LHCb a single job v

And selection of interesting events at this stage pays off

The MC truth & MC history

- Generator to Geant4 and MC truth
  - Pass only particles from generator to Geant4 which will interact with detector, that is to say particles with non-zero travel lenght.
  - All other particles are saved directly in MCParticle container, and the decay chains are restored at the end of the processing by Geant4.
- MCHistory (i.e. what happened during the tracking of particles) is essential to understand efficiencies and physics effects
  - Geant4 does not have a tree structure to keep history
  - Introduced use of HepMC internally to Geant4 to provide such a tree structure
- Here it is a potential bottle neck in particular for vectorised approaches!

Gauss as an integrated simulation framework



Proof of principle exist and we use it

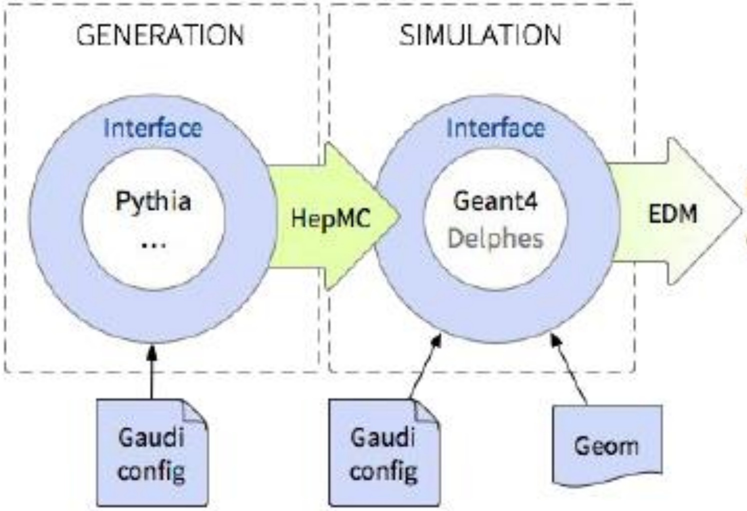
BUT we need to make it easy and fail-safe at the same time

Use Gaussino as a test bed

Gaussino — An experiment independent Gauss

Work so far carried out by FCC.
Input from LHCb on design of new/modified components – geometry ‘ready’
others to be defined

Integrating:
generators, Geant 4
& Delphes(?)



Hand-off to the experiment SW

Event data exchanged with HepMC

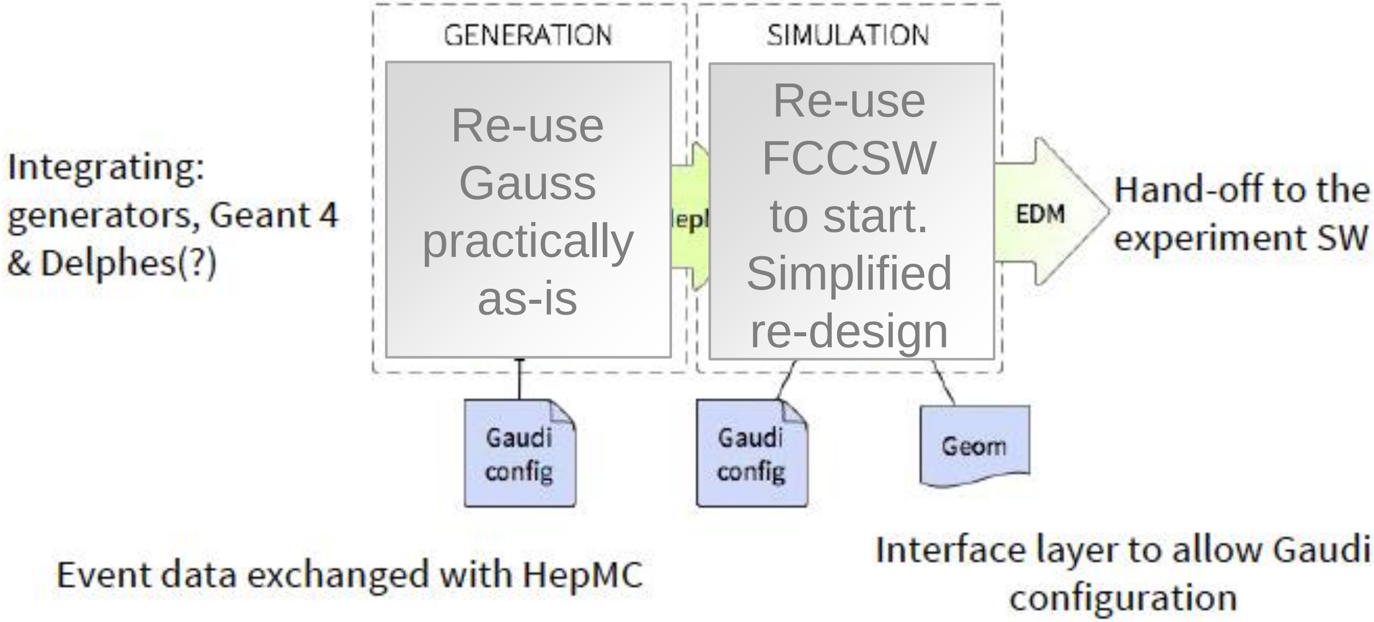
Interface layer to allow Gaudi configuration

Gaussino is the ideal place to try out Gaudi parallel and G4 MT, the new event model, …

Use Gaussino as a test bed

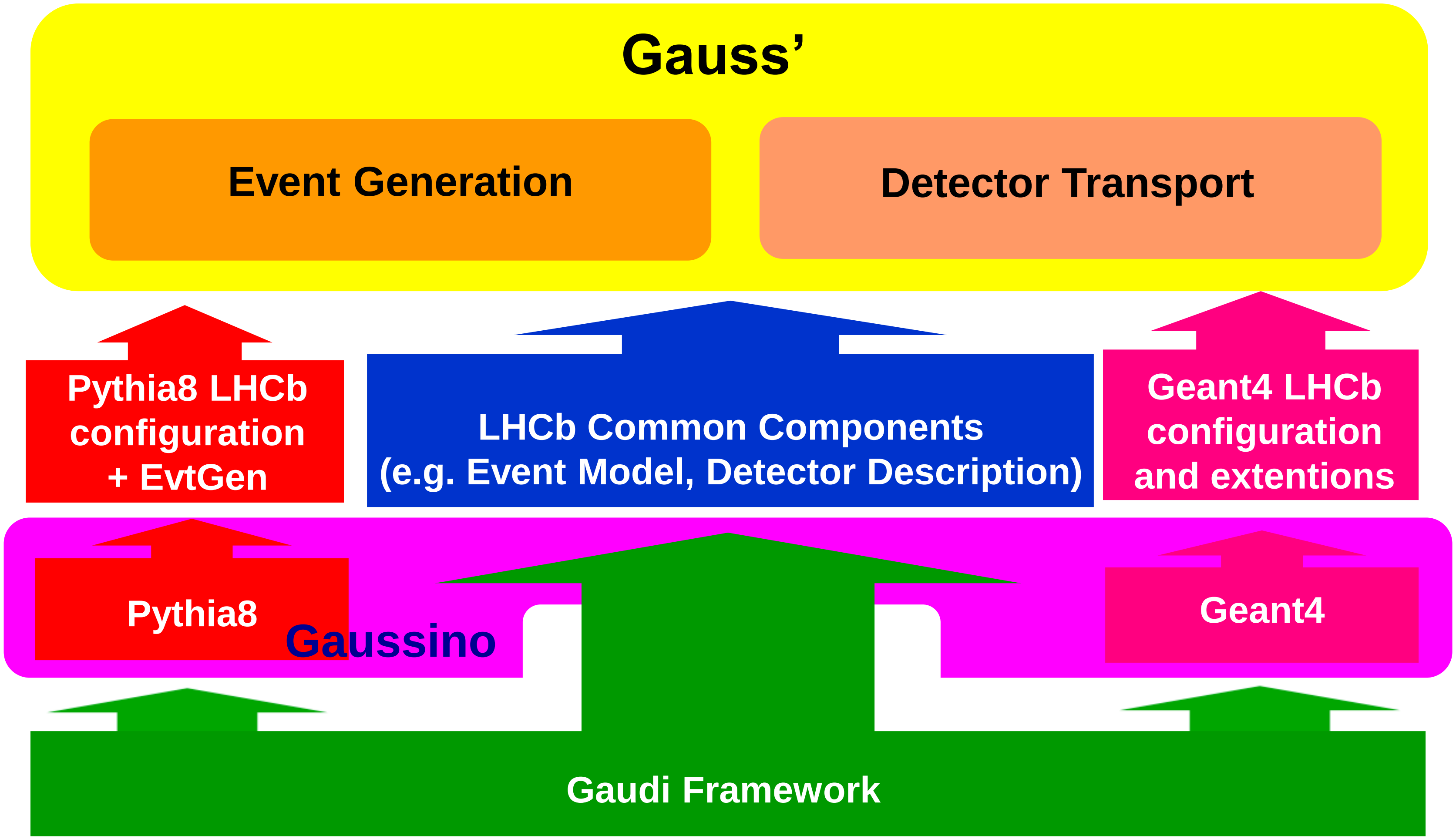
Gaussino — An experiment independent Gauss

Work so far carried out by FCC.
Input from LHCb on design of new/modified components – geometry ‘ready’
others to be defined



Gaussino is the ideal place to try out Gaudi parallel and G4 MT, the new event model, … and we can share the work with FCC ...
BUT it is not enough and we[(LHCb)] need to be more involved in its development.

A version of Gauss built on Gaussino



GeantV – long term future

Talk by Witek Pokorski

- R&D for vectorised simulation
  - **Developing** bits and pieces needed by **concurrency, vectorization** of geometry/physics, data structure and management, new model development
  - **novel transport** engine based on 'basketization' approach
  - **high performance vector-aware geometry** package interfaced in multi-particle mode with GeantV
- Proof-of-principle demonstrator
  - Speed ratio GEANTV/GEANT4 ~2.5-3.5 *in similar conditions*

With LHCb geometry via GDML and tabulated physics

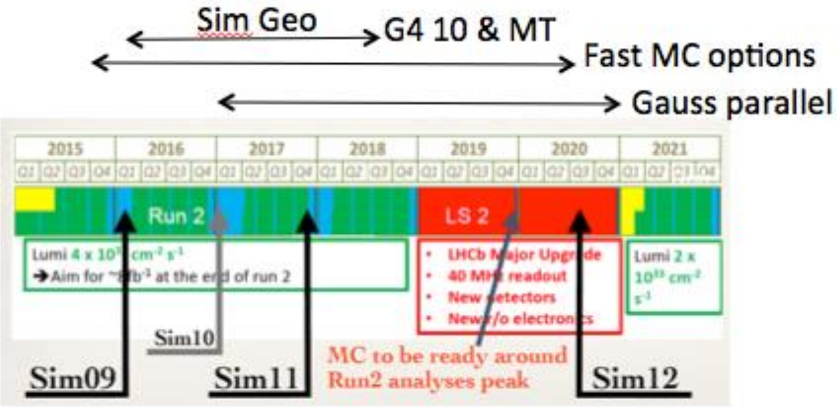
- Fully functional prototype end of 2018.
  - Interface to experiment framework will require extensive re-write
  - EM physics and at least one hadronic physics choice
  - Recommended to expose interfaces as early as possible

Digitization

- Detector response handled by Boole
  - Tiny CPU time with respect to Gauss
- Special cases of Muon low energy background handled at the digitization level from dedicated time consuming simulations

Roadmap for LHCb Upgrade

- To do for Computing Upgrade TDR
  - Fast Simulation options part 1
  - Test alternative Geometry descriptions within Geant4
  - Demonstrator for LHCb of multithreaded/multitask Gauss → Str



- To do before Run3 starts:
  - Fast Simulation options part 2
  - Fully implement new geometry and new event model
  - Gauss on Gaudi future
- Future:
  - GeantV - development from LS2 to use it sometimes in Run3

Give input from our needs and experience for LHCb Upgrade to CWP