GeantV Fast Simulation Mini-Workshop
CERN, 26th May 2016

LHCb Fast Simulation(s)

What exist in LHCb to produce simulated samples faster

beside making the full simulation software faster

What we are currently developing and wish list

How does it technically fit all together and constrains

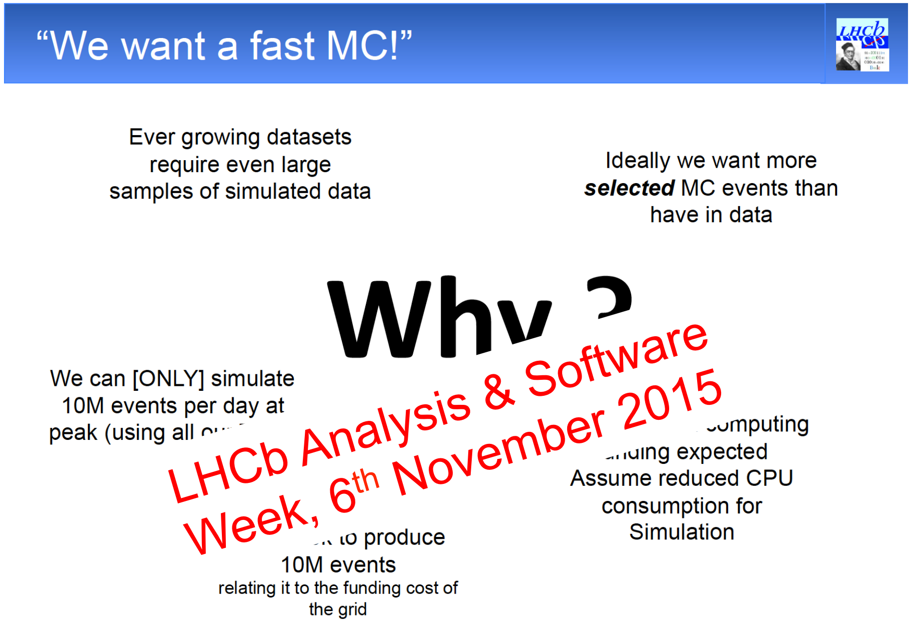
Gloria Corti, CERN

LHCb Simulation Physics Performance Working Group

Thanks in particular to: R. Cenci, S. Easo, J.-F. Marchand, D. Muller, M. Rama, A. Satta, B. Siddi

An emerging need in LHCb

- Needs for speeding up the simulation became more apparent last year
- It coalesced in new developments in view of the simulation needs for the LHCb Upgrade in Run3



> *"The richness of the LHCb physics program not only leads to the need of large MC samples, but also to widely-diversified simulation requirements"*
>
> *"PWGs already started to cope with limitations of available MC statistics"*
>
> *"It's certainly time to… find ways to reduce the hunger of CPU cycles..."*
>
> *V. Vagnoni, Paris Computing Workshop, Nov. 2015*

Simulation needs towards the upgrade era

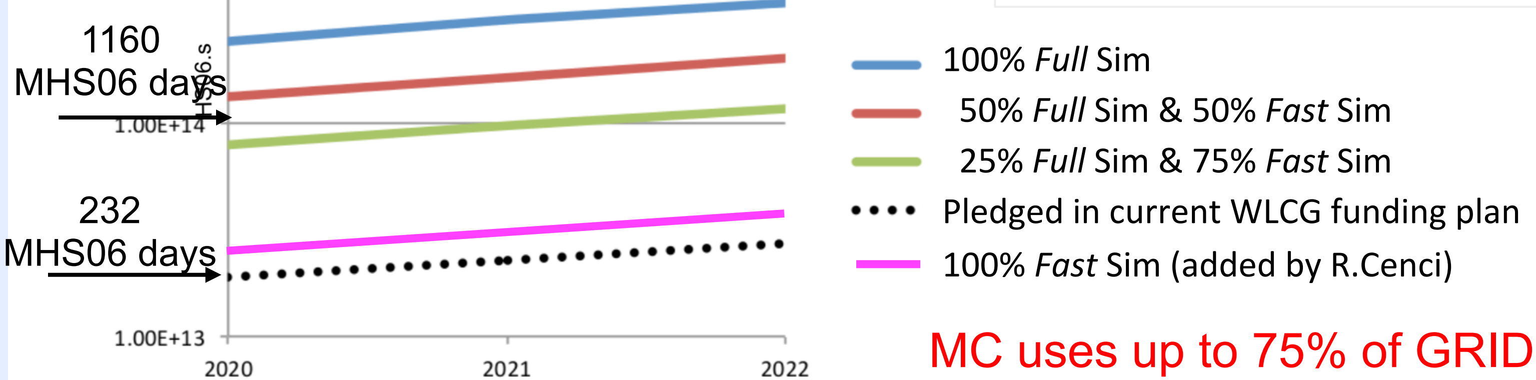
- The available resources will determine the MC statistics that we will be able to produce

- Run 1 simulated events ~ 4.5*10^9 (spring ‘15)
  - ~ 12 % of recorded
- Aim to simulate 100 % of recorded

Exercise for Computing Workshop

Assumption: Full Sim 600 HS06.s, current one is 3-5 times that

Fast Sim = 0.1 x Full Sim



MC uses up to 75% of GRID resources

MC won't fit (by far) into the pledged resources :-(KO

F. Stagni, S. Roiser, Paris Computing Workshop, Nov. 2015

‘The LHCb solution’

- No single size fits all solution but a palette of choices to pick and choose from as most appropriate
- 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
- Identified a ‘shopping list’ and decided what to focus on in order of priority and do-ability
  - Consolidated, validated and started using in production existing options with clear understanding of their applicability for physic’s studies
  - Identified what is necessary to give the analysis the full information they need and what is more appropriate for different detectors.

The new features are essential for Run3 simulations (LHCb detector upgrade) but already useful for Run2 and will be deployed as they become available

Options (almost) available

Available for production or close to completion

Two general approaches

Simulate less detector by turning not needed components off.

Simulate less particles traversing the detector.

Or a combination of the two.

Neither gives the full picture but sufficient for some cases where either a huge MC statistic or a quick bulk-park answer is needed

Complementary to the full simulation

Simulate less detector

- Simulate partial detector keeping the whole complexity of what is simulated
  - RICH PID info not necessary for all analysis, in which case not necessary to simulate the associated physics processes
  - When [HLT and] offline selection efficiency of hadronic final states is purely tracking-based do not need to simulate Calorimeters [and Muon System]

- Changes are purely steered by configuration options of the job

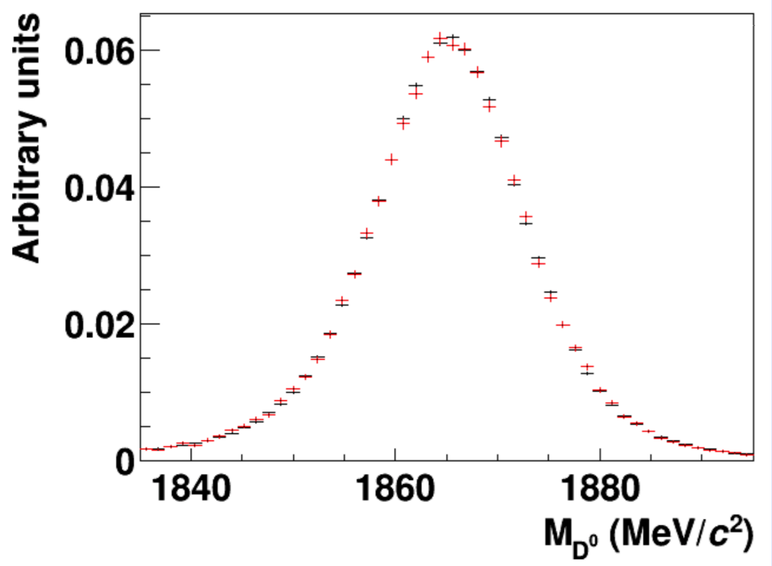
Simulating without RICH physics

- PID in simulation not needed for all analysis and in various cases corrected against the data
  - LHCb physics list defined as combination of PL at configuration. Do not use the 'RICH' processes
  - RICH material fully simulated
- Care needs to be taken when running the trigger as HLT lines including PID requirements cannot be used

✔ Saves O(30%) of simulation time
✔ Size of the data slightly reduced

✗ No PID information in MC

$B \to D^* \mu\nu X$



Validated and used in production to reduce large systematic uncertainty from MC for $\mathcal{R}(D^*)$

All relevant distribution in perfect agreement

G.Ciezarek, D. Muller

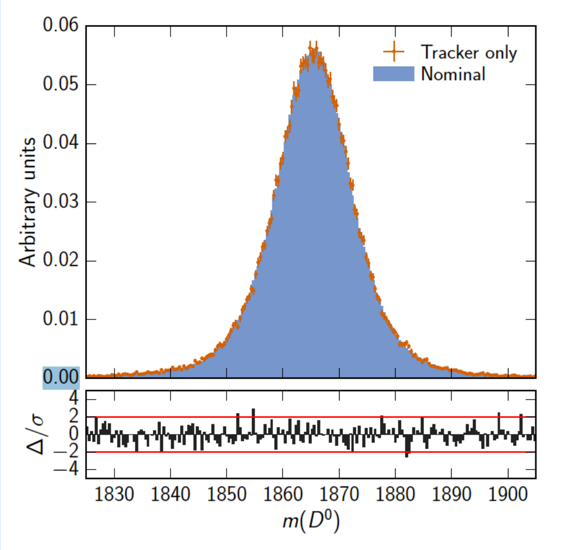
A tracker only simulation

- Basically a RICHless MC taken to the extreme
- Deactivate the geometry of the Calorimeter and Muon Systems
  - We pass a list of detectors to simulate. Give a reduced list
  - RICH material fully simulated but physics switched off
- Cannot run any L0 trigger

✔ Saves O(80%) of simulation time
✔ Size of the data reduced O(25%)

✗ No PID information in MC
✗ Cannot run the trigger

$D^{*+} \to D^0\pi^+$ with $D^0 \to K^-\pi^+$



Validated with test productions

Could [have] be[en] used in charm production

All relevant distribution in perfect agreement

D. Muller

Simulating less particles – only the signal

- Simulate only the interesting particles, those belonging to the 'signal' and ignore the rest of the event
  - Can be sufficient for some studies
- Changes are steered by configuration options where the generator (e.g. Pythia8) is replaced by a Particle Gun in-house generator and a primary vertex smearing in the reconstruction

✔ Saves O(95-99%) of simulation time (proportional to N of track)
✔ Size of the data reduced O(95%)

✗ Too good due to low occupancy

Used in upgrade studies to evaluate if detector in a new place is worthwhile before embarking in full design

Simulate less particles – most of the time

- Re-use the underlying event many times with different signal decay
  - Combine full simulation and particle gun like approach
  - Deemed a better choice then overlaying signal on data both operationally and for selecting data to use

Handled by the Gauss algorithm sequencing. For Geant4 the undlerying event and the signal are separate events

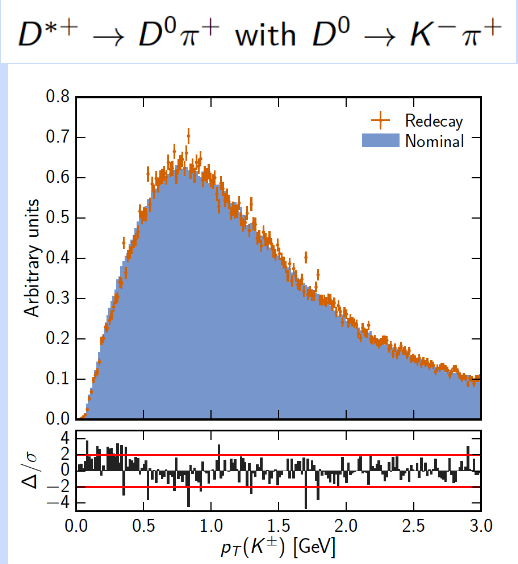
Procedure

1. Generate a full event.
2. Find and split off what should be redecayed.
3. Simulate the underlying event and make it persistent across multiple events.
4. Create particle gun like event.
5. Reattach the signal particle to the correct vertex in the event.
6. Write out this event.
7. Repeat 3. through 6. N_{Redecay} times before going back to 1.

Complementary to the other methods

Re-decay of the signal

- Compatible with different generator and detector simulations
- First prototype shown in April, work in progress to deploy it in production in the next few months
  - Working on proper integration in the Gauss configuration so that it can be controlled by a simple switch
  - Currently undergoing extensive testing



D. Muller

✔ Substantial increase in speed, depending on the number of redecays
✔ Complexity of a full event

✗ No disk space saving
✗ Statistical uncertainty more difficult

Muon low energy background fast simulation

- In the Muon System prohibitive to obtain the same hit rates as in the data with full simulation for physics studies
  - Low energy background specific to the Muon System
  - Thresholds in simulation should be as low as KeV for $\gamma/e\pm$ and eV for neutrons to get a reasonable hits rate
  - The cavern and close by LHC elements should be included
  - O(x10) slower than with standard full simulation settings
- Ad-hoc solution in place already in LHCb Geant3-based simulation and carried over

Procedure

1. Generate minimum bias samples with standard and *mu low energy background* configurations
2. Compare the two and get a parameterization of the difference
3. Add the parameterized background in the digitization

✔ Can scale the parameterization independently of luminosity
✗ No spatial correlation between low energy hits (small effect)

Options under development

Fast simulation with PID and Calo information

- The simulation of the Calorimeters and RICH are the most time consuming
  - Carrying out in depth measurements with details of processes, particles, volumes
  - Combined they account for 80-90% of the CPU time

- New project to setup a 'general purpose' fast simulation started early this year
  - Goal: up to 10x faster keeping both RICH PID and Calo information
  - Fully integrated in existing LHCb simulation framework, Gauss
  - Full/fast operating mode configurable for each system and particle type

Basic questions – overall view

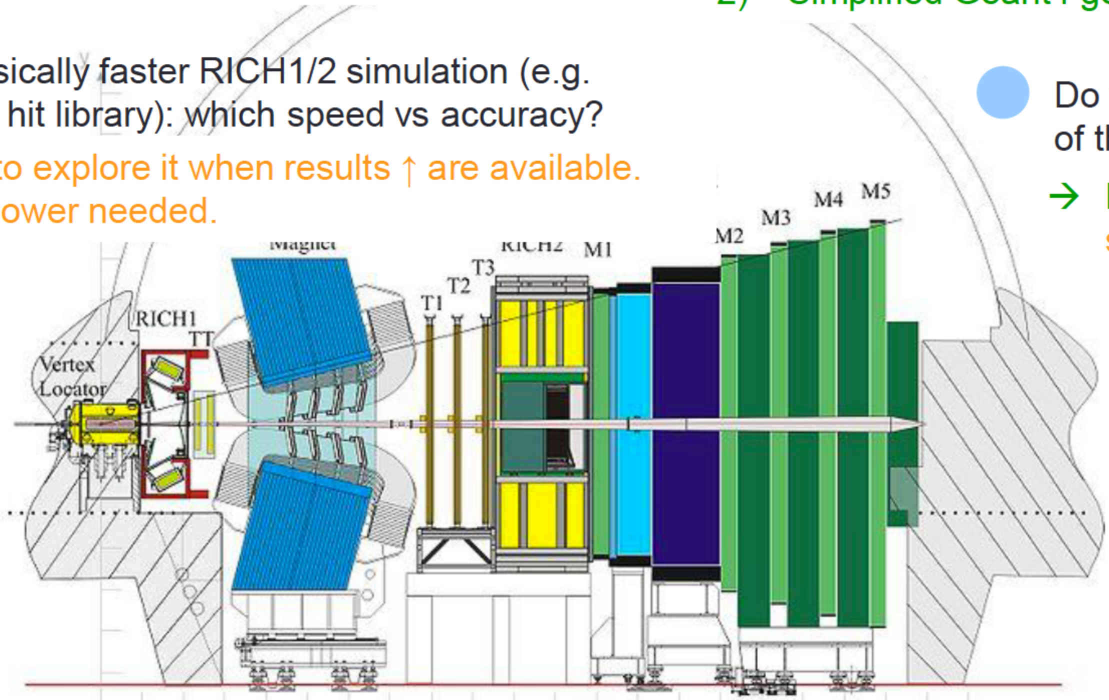
- Simulation of high-level PID variables (e.g. probNN, DLL) or low-level variables as in std simulation?
  → simulation of low-level variables

- Is a faster simulation of RICH1/2 without loss of accuracy possible? Which speed gain?
  → work in progress, first results in a few weeks

- Which fast simulation solution for the calo system?
  → 2 solutions being investigated in parallel.
  1) Shower library (shower param. might come later)
  2) Simplified Geant4 geometry

- Intrinsically faster RICH1/2 simulation (e.g. using hit library): which speed vs accuracy?
  Plan to explore it when results ↑ are available. Manpower needed.

- Do we need a fast simulation of the muon detector?
  → No, but need to estimate/ simulate the punch-through



S. Easo, J.-F. Marchand, M. Rama

Fast simulation of the LHCb calorimeters

- Use of shower libraries to simulate the calo hits
  - Proved to be fast and accurate in ATLAS after proper tuning
  - Issue of punch-through in muon system to be considered
- Parameterized simulation of ECAL/HCAL showers
  - Simpler then shower approach
  - Probably difficult to achieve a good description of the shower
- Simplified description of the calorimeter geometry [as given to Geant4]
  - It would preserve the punch-through
  - Can we gain 10x in speed and with which accuracy?

The LHCb calorimeters are sampling calorimeters in forward region. The use of shower parameterization may be difficult and the shower library be preferable

Learning from what ATLAS has done

J.-F. Marchand, M. Chefdeville, M. Rama

Calo: what has been and need to be done

- Beside investigating the possibility of implementation and what is done in detail in the full simulation…
- First idea, being implemeted:
  - Each (e, γ, π) entering the calorimeter is killed in Geant4

    Via G4 actions for now. Intend to investigate using Geant4 regions and parallel geometry

  - Definition of the format of the shower libraries, which will be used to fill the LHCb MCCaloHits to be processed by the digitization
  - While building the shower library will check if a parameterization is possible
- To be done:
  - Building the shower libraries
  - Validation
  - Tuning to data

  The time consuming part looking at the ATLAS experience

J.-F. Marchand, M. Rama

Fast RICH [and PID] simulation

- RICH ParticleID is sensitive to several parameters and it is not clear if all correlations can be taken into account in fast simulations

Charged tracks → Cherenkov photons traveling through the RICH optical elements

Photons → Photelectrons → Hits in HPDs

- Nevertheless even for ParticleID variables the overall preference by the Physics community is to have low-level PIDs that then the high level one as DLL, ProbNN
  - High level variables can be determined with data driven methods (already done for PIDCalib) but no lower-level PID available. Muon, Calo and Velo also used.
  - Difficult to know a priori speed vs accuracy in RICH fast simulation at hit level

S. Easo, C. Jones, M. Rama

Faster RICH simulation

- Proceed in steps
  1. Implemented a number of changes in current full Geant4-based RICH simulation with are expected to significantly improve the speed without loss of accuracy.

     Track less photons based on max Quantum efficiency of the HPDs, relying at the start on max detection efficiency (depends on data taking period), …

     Changes implemented reduced simulation time O(20%) with no impact on results. To be validated in production

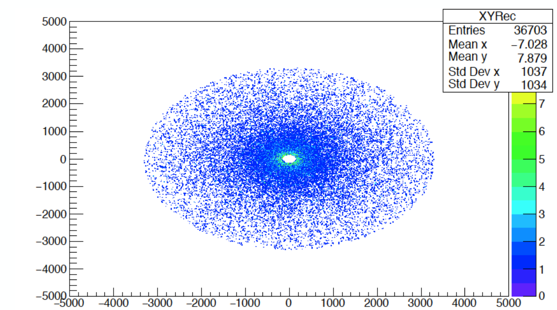
  2. Deep investigation of where the remaining CPU time is spent to see if further improvements can be made

  3. In parallel explore the feasibility of an intrinsically faster RICH simulation, e.g. using simplified hit library

Fully parameterized

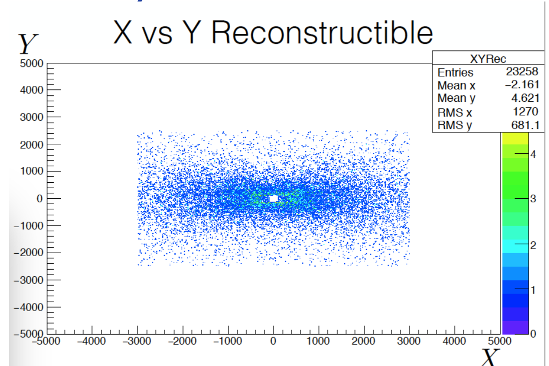
A SuperFast simulation

- We are also working on a fully functional ***SuperFast*** option based on full parameterization using DELPHES(*)
  - Proof of principle of DELPHES adaptor in Gauss beginning of 2015
  - Fits in the Gauss framework reusing the generators and replacing Geant4 for the whole detector
  - Identified and implementing extension necessary in DELPHES to be used for physics studies in LHCb (shape, parameterization)



Original Delphes output shows a cylindrical symmetry not suitble for LHCb

New particle propagator for LHCb trapezoid shape and dipole P_T kick



Output from Delphes in LHCb acceptance, TES location

  - Defined what output to provide for analysis since it replaces the full reconstruction in a way that LHCb analysis framework can be used as-is, i.e. the LHCb end of reconstruction high level object

(*) by P. Demin, M. Selvaggi

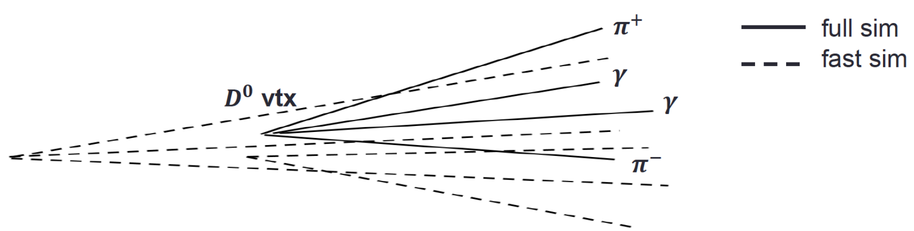
Putting it all together

Making it all seamlessly available

- Towards a flexible framework to mix fast and full simulated particles in the same event – similar to ATLAS Integrated Simulation Framework
  - Muons always fully simulated
  - Possibility to select full/fast mode according to particle type
  - But could have more complex criteria
  - Treat differently out-of-time events
  - and in-time pileup
  - … we can even consider mixing all
- User configuration has to be as simple as possible and coherent for all fast options

Example

Simulation of decay $D^0 \rightarrow \pi^+\pi^-\pi^0[\rightarrow \gamma\gamma]$:
full simulation for $\pi^+, \pi^-, \gamma, \gamma$ from signal
fast simulation of calo and/or RICH for the other particles of the event



- Speed close to that of particle gun
- Accuracy close to that of fully simulated event. In particular:
  - reconstruction of PVs
  - track reco degradation
  - possible to implement track isolation

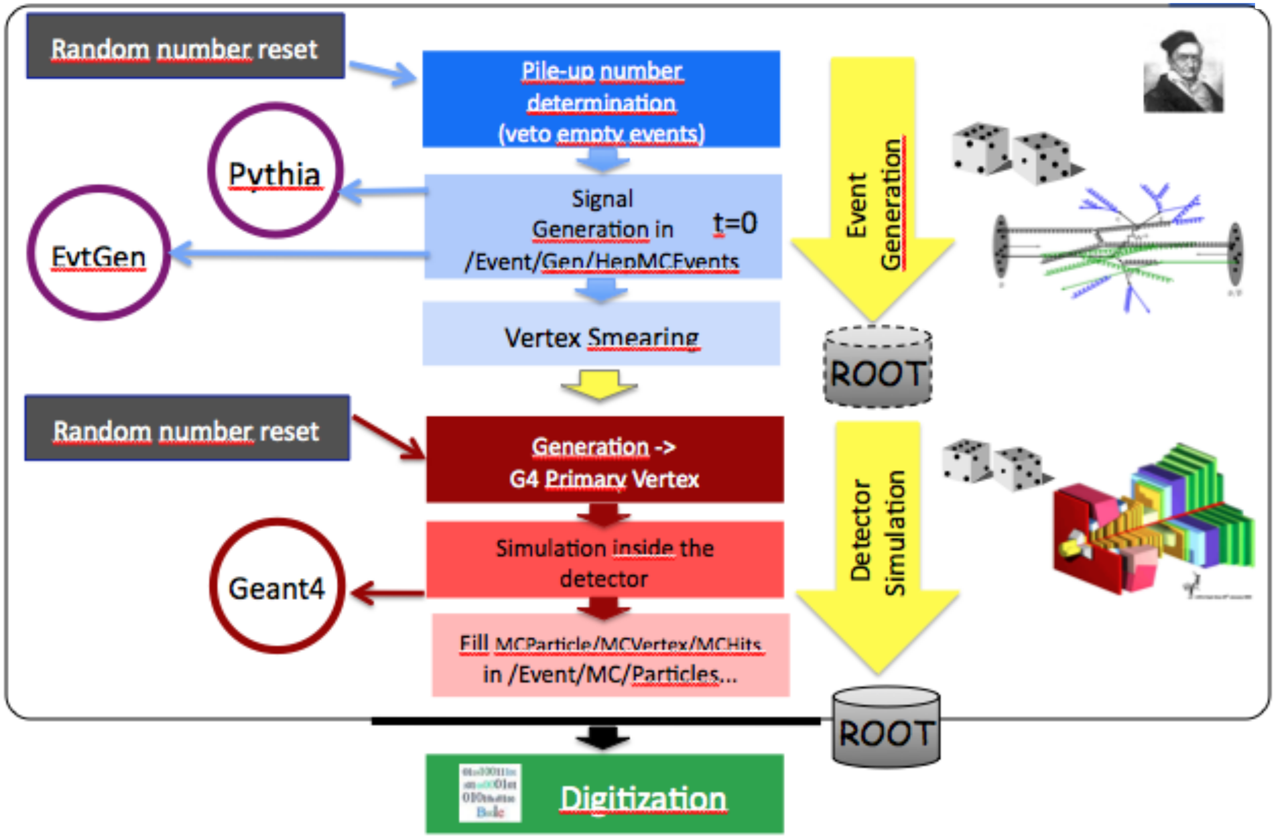
Under investigation

NB: the fast sim mode might even be “RICH-less and calo-less”

M. Rama

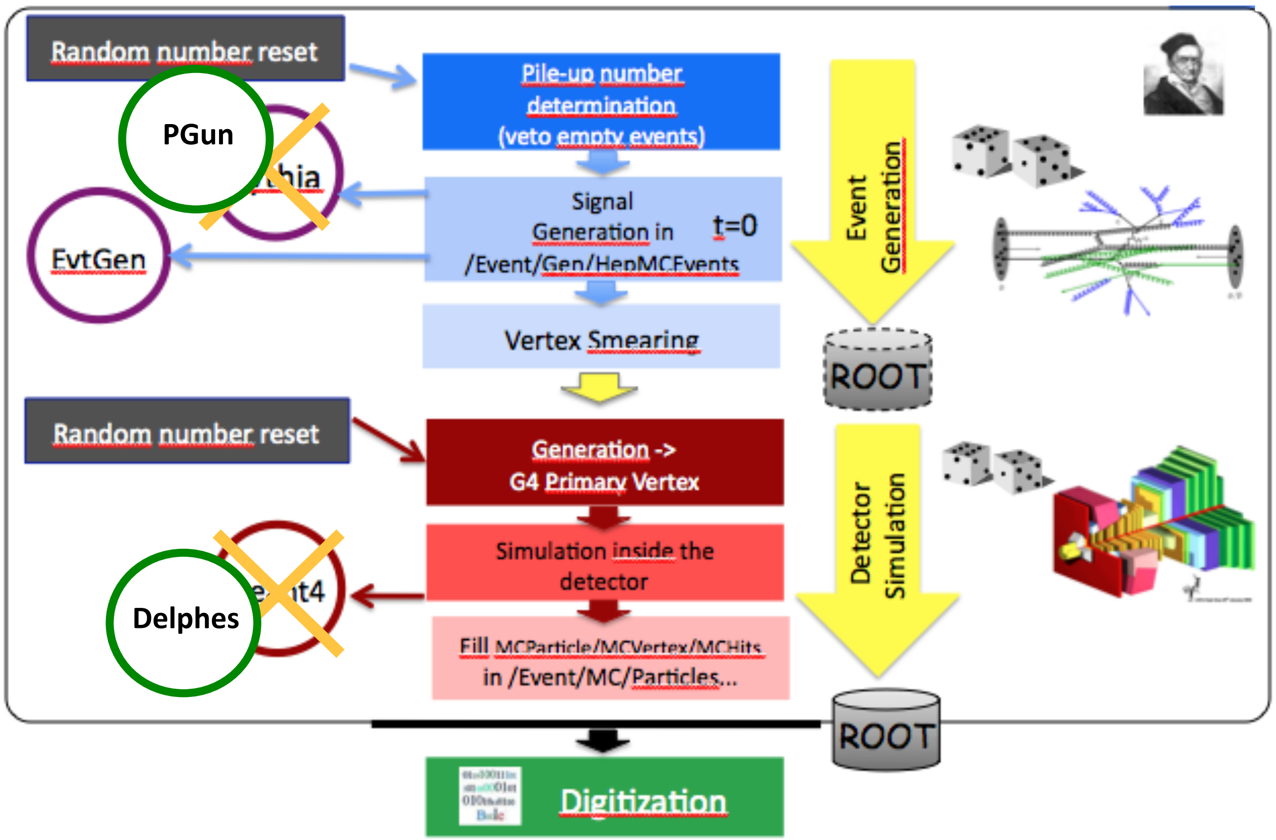
Under development

- Gauss is a framework
  - It fully exploits the Gaudi architecture and 'plug-and-play' philosophy



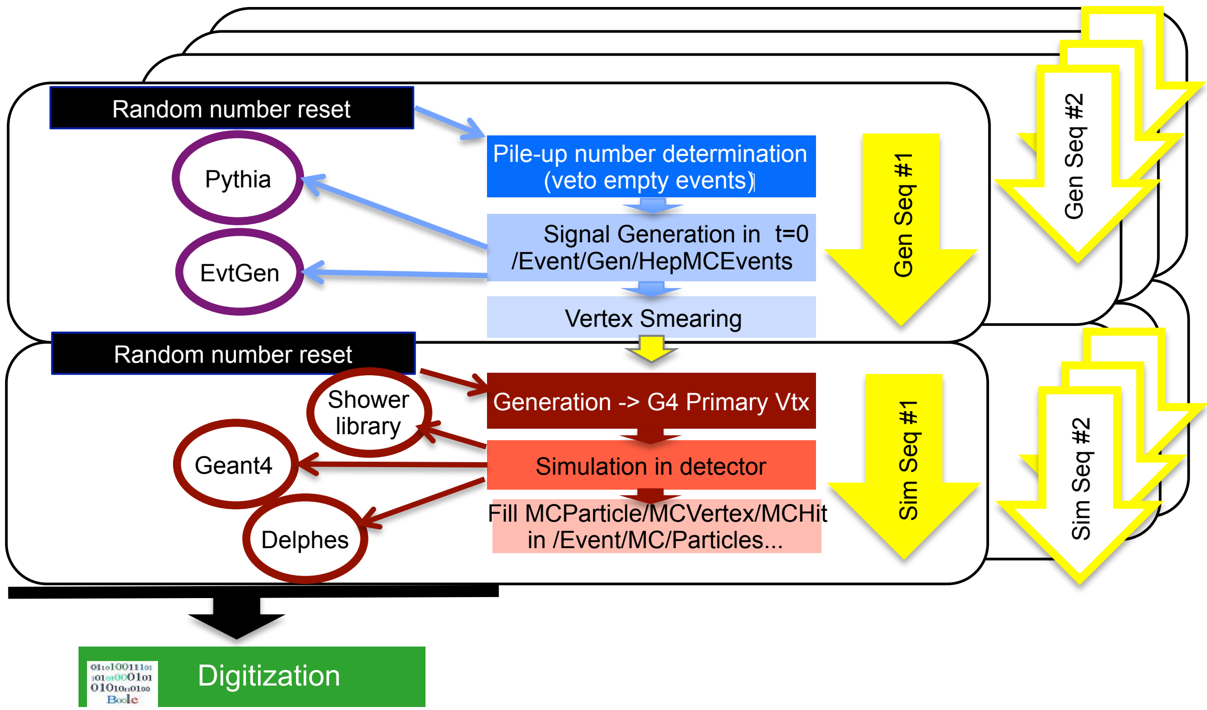
Gauss sequence

- Things can be easily replaced at the whole event level via the Gauss() configurable sequencing (once they are available ☺)



Gauss algorithms and tools

- A little bit more complicated but foreseen how to change things at the event level
  - algorithms need to be extended/specialized
  - multiple generator and simulation sequence can be specialized



What about the end user?

- As much as it can be a complicated gymnastic in the Gauss() configurable it has to be as simple and as safe as possible to use

```
Gauss().GenType= 'PGun', 'Phys'
```

already exists

```
Gauss().Reuse =       True
```

```
Gauss().SimType =     'Full', 'Fast', 'Delphes', 'None',
                      'Skip', 'Custom',…
```

some exists, but by setting Phases, PhysicsList, DetectorGeo/Sim

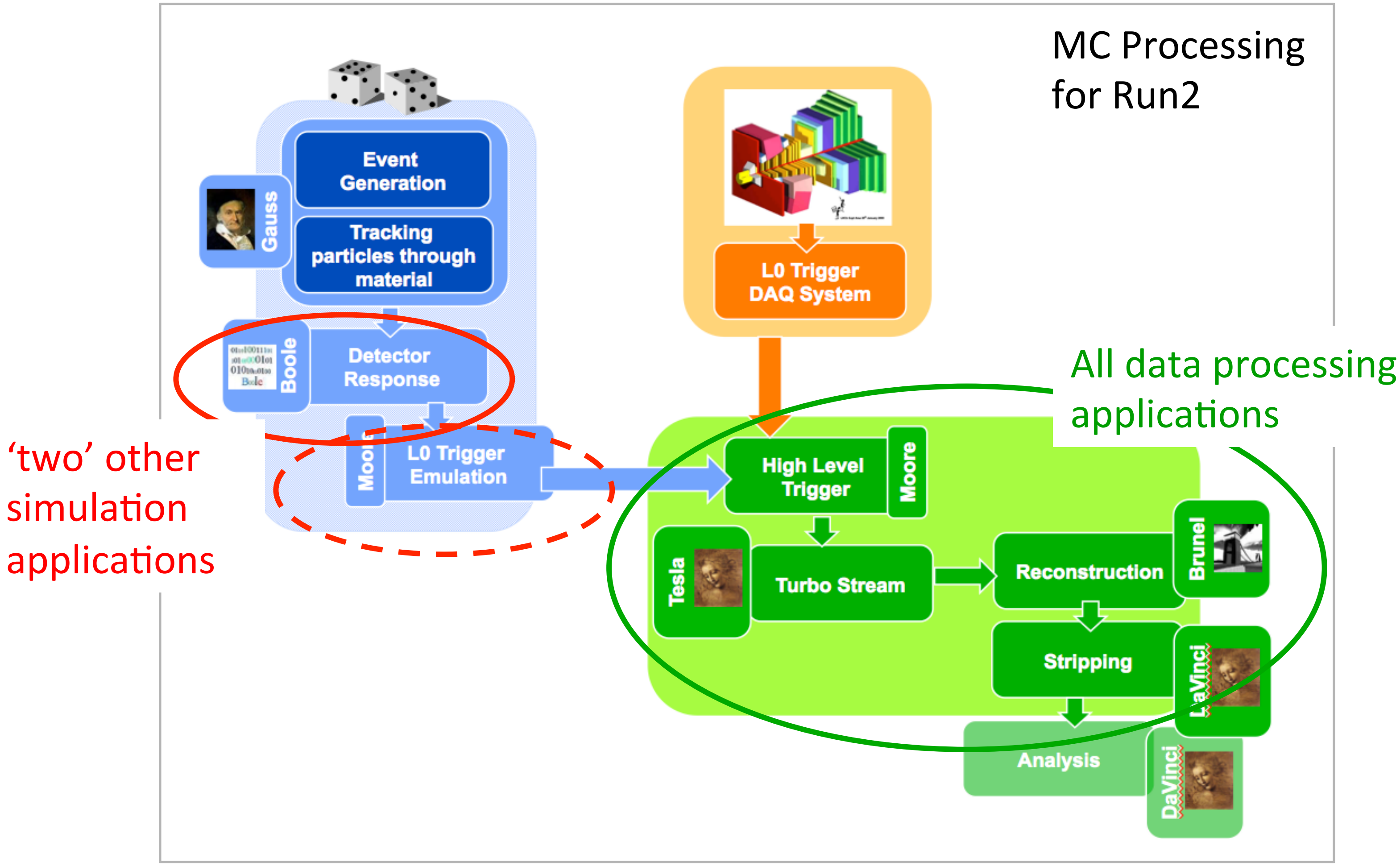
and granular to select fast/full for each detector

```
Gauss().DetectorSim = { "Detectors": ['Velo: Full',
                                      'Rich: Skip',
                                      'Calo: Fast'] }
```

and for different particles

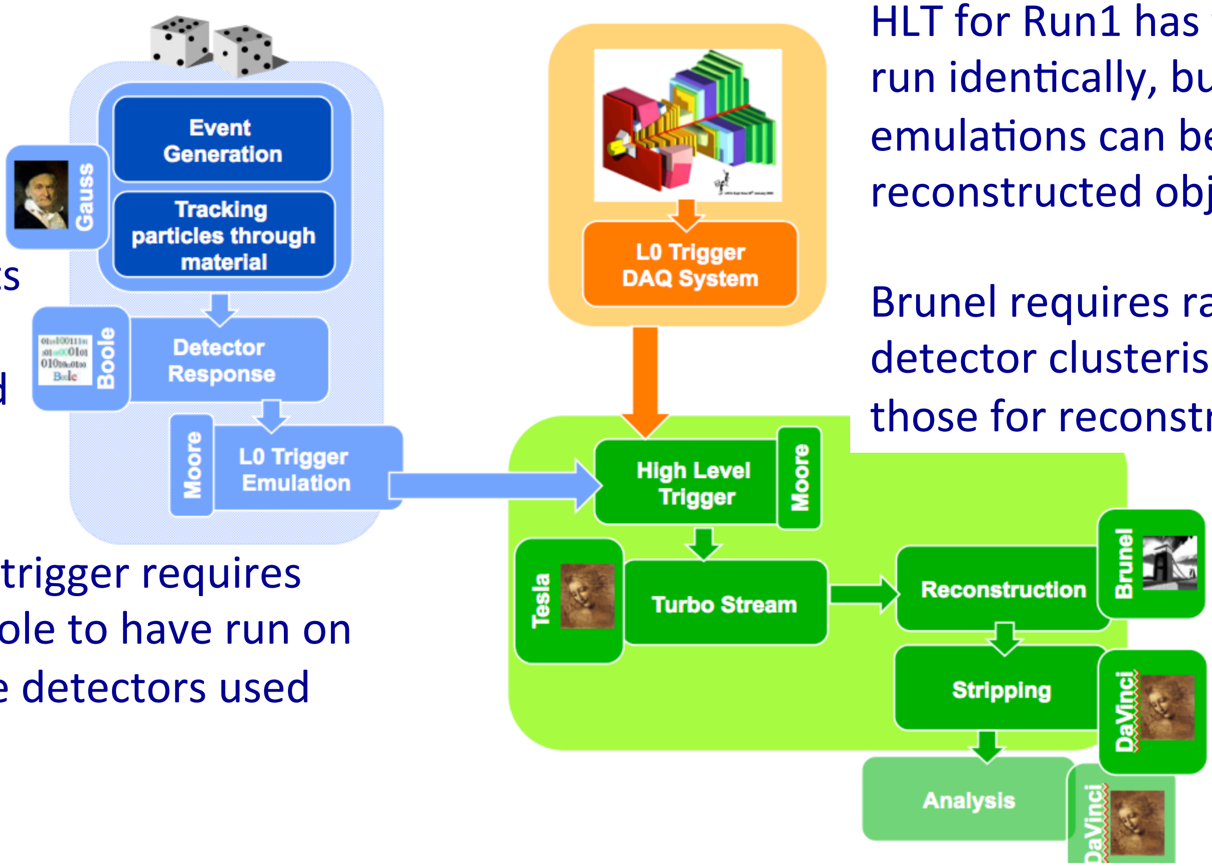
All done? Not quite…

- Gauss is a necessary but not sufficient step.



Processing remarks

- All applications will work as-is as long as the input is there as expected



HLT for Run1 has to be re-run identically, but for Run2 emulations can be setup with reconstructed objects

Brunel requires raw data for detector clusterisation and those for reconstruction

Boole is fine if MCHits are there regardless of how they are filled and for what part of the event

L0 trigger requires Boole to have run on the detectors used

DaVinci requires at least the final objects of the reconstruction, i.e. ProtoP and PV
The one that MUST work independently of the type of simulation we run

Final remarks and long term wishes

- Gauss is starting to fulfill its potential as **Integrated Simulation Framework**
  - we want to mix simulation flavors for different particles
  - we are working to provide an easy to use palette of choices
- Fast simulations are not only a simulation issue but a wider simulation processing
  - Tracking, particleID, trigger of MC samples …
- To understand what happened we rely on connection to MC truth and MC truth history
  - We need to provide that in the fast simulation, too
- Eventually we may be CPU limited by the tracker where we need highly reliable simulations
  - Reliable transport through a simplified geometry? Tools to help ‘averaging’ where appropriate would be nice but may be an unfulfillable dream

BACKUP

LHCb applications and data flow