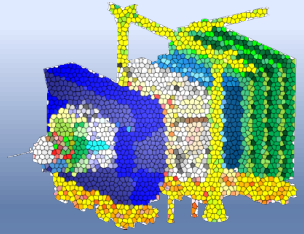
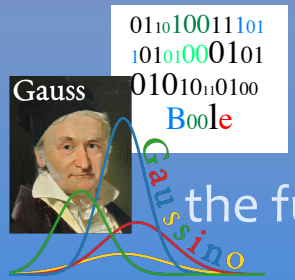


6th Workshop on LHCb Upgrade II
29 – 31 March 2023

UNIVERSITAT DE BARCELONA



R&D for simulation future

the future of simulation and simulation for the future

Gloria Corti, CERN

With special thanks to

L. Anderlini (Firenze), C. Bozzi (Ferrara), B. Couturier (CERN), A. Davis (Manchester), T. Hadavizadeh (Monash), P. Ilten (Cincinnati), M. Kreps (Warwick), M. Mzurek (CERN), W. Pokorski (CERN), F. Stagni (CERN), M. Whitehead (Glasgow)

How will we do simulation in 2030+

?

What will we need simulation for ?

- Finalize detector and data processing

- Detector commissioning

- [Prepare for] Physics analysis

- *Different needs in term of physics modeling accuracy, statistics and timescale turn-around*
- *Different requirements for s, c, b physics*

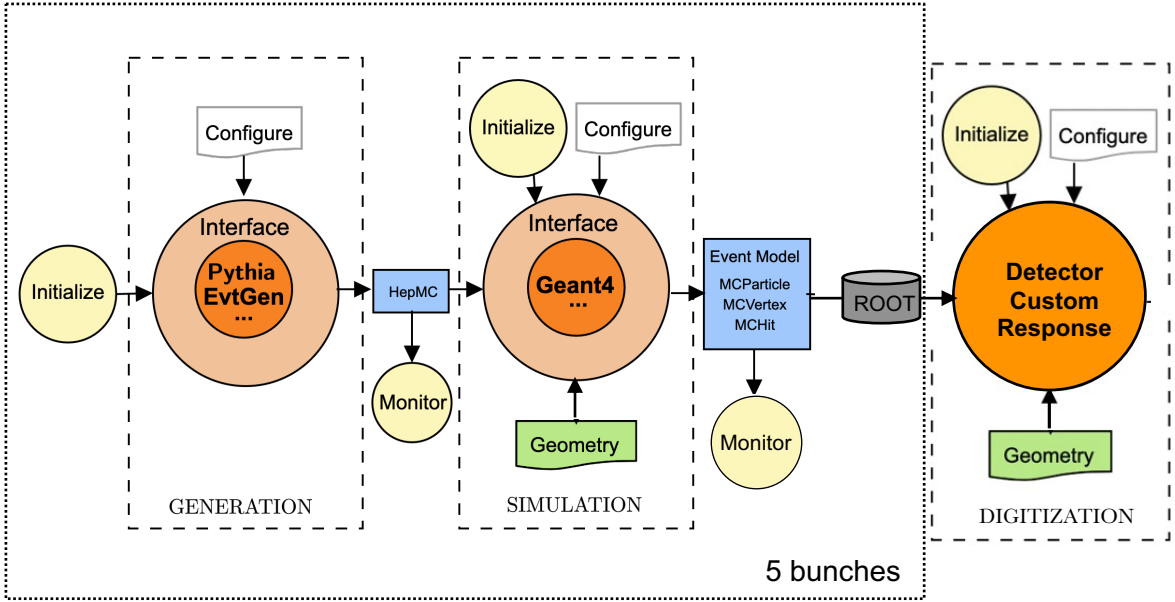
Outline

- Today's landscape
- Computing environment
- Fast simulations
- Accelerators
- Generators
- Pileup

LHCb simulation landscape today

Pythia + EvtGen are the LHCb workhorses

We use other generators for HI, double heavy baryons, CEP, EW, Higgs, …



Geant4 is the corner stone of simulation for the LHC

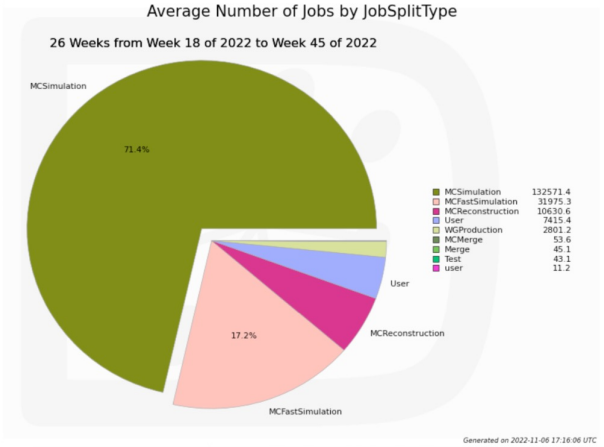
We have Fast and Ultra-fast simulations under development

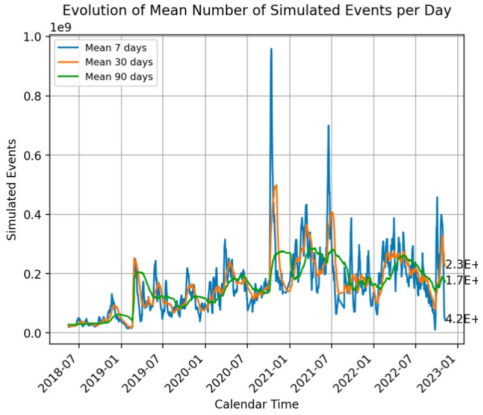
We heavily rely on fast simulation techniques

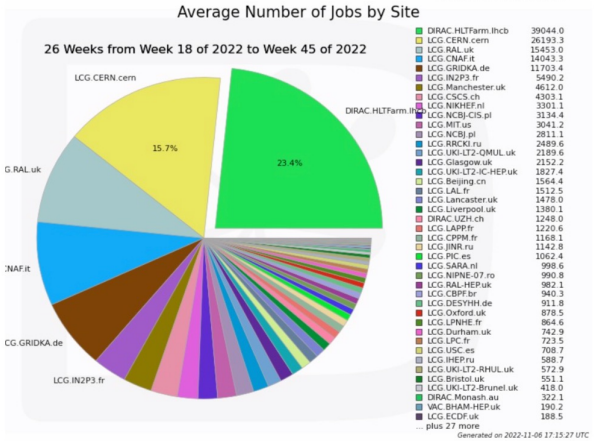
LHCb simulation and computing resources today

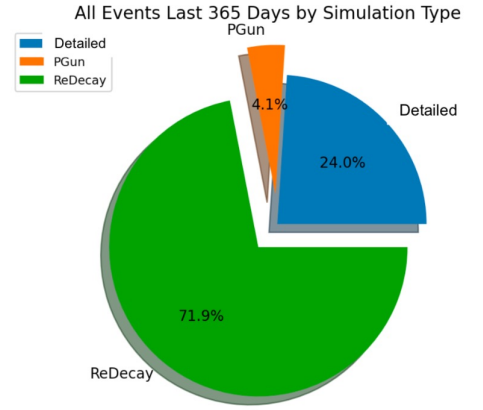
Distributed Computing Operation

- Simulation dominates (> 95%) CPU work
- Runs everywhere
- No input data required
- Runs continuously









C. Bozzi, B. Couturier – WLCG Workshop Nov 2022

Evolution of distributed computing

A complex ecosystem with **DIRAC** as LHCb standard for **workload** and **data** management

Pledged and opportunistic resources

In the next few years:

Token transition, following WLCG timeline

Largely improved **HPCs** support

- Usage still limited
- Gradually overcoming site limitations
- Proactively seeking for more resources

Barcelona Supercomputing Center (MareNostrum) in productions and used to process Monte Carlo simulation tasks (Gauss)

Full support for **non-x86** architectures

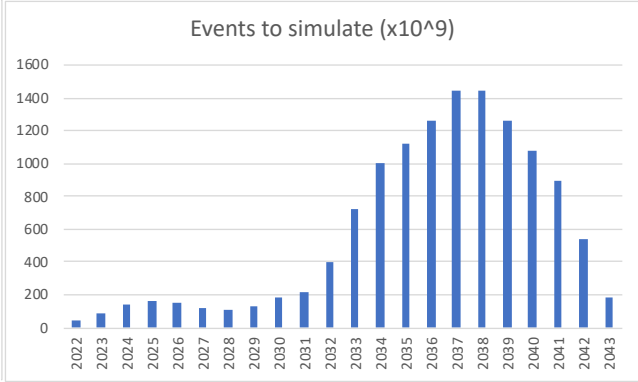
- GPU, ARM, …

Gauss ARM build in HepSCORE
Access to resources in Bologna (CNAF) and possibly in the UK this year

Simulation and computing resources for U2

More beam data requires more simulated data to support analysis

- Simulation scale with the event rate
- It does not per-se scale with pile-up (in-time + out-of-time)
- Extreme pressure on the computing budget
- How we run today will not be affordable in Run5

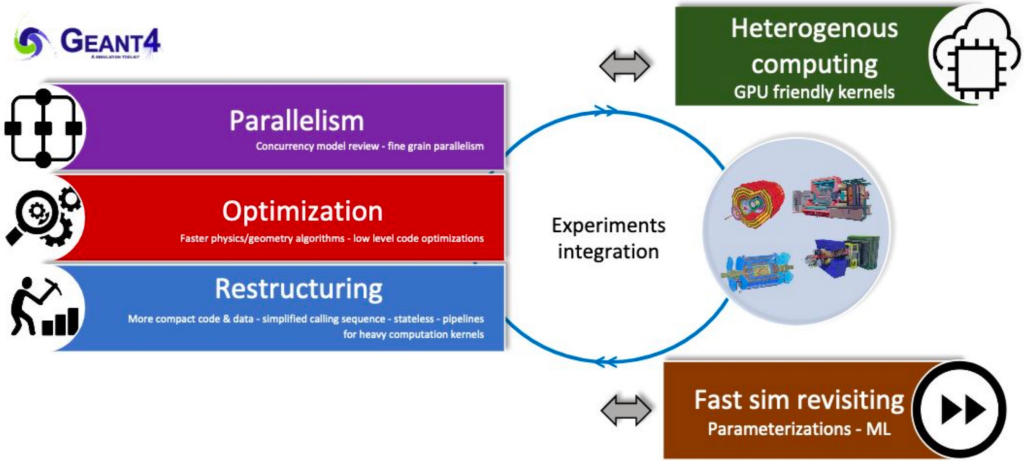


C. Bozzi
U2 Workshop May 2021

Simulation needs to be faster, without sacrificing relevant physics accuracy

Improvement on detector simulation

- Refactoring and internal improvements
  - Optimisation of current Geant4 code to run faster
  - Mostly work internal to Geant4



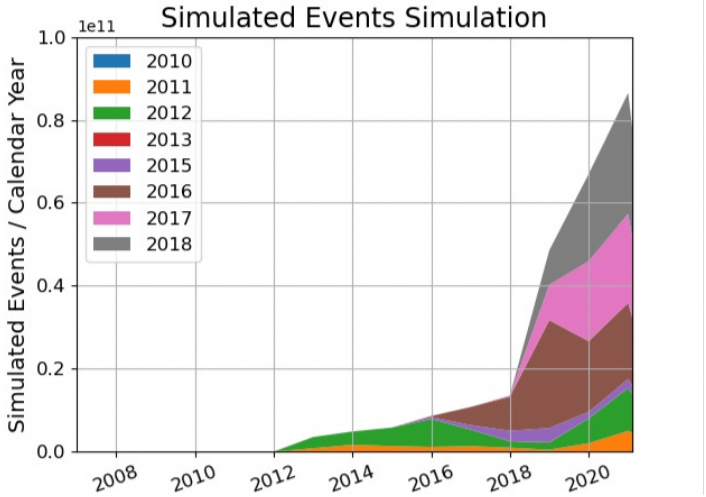
All aspects in Geant4 R&D activities
Combined with physics improvements

- Hardware (R)Evolutions
  - Increasing trend away from purely CPU based machines
  - GPUs are more and more available
  - How can we use them for detector simulation?
- Fast Simulation
  - Replace detailed particle tracking models with different methods
  - Long tradition of parametric response implementations
  - Machine Learning is the cool kid on the block

HL-LHC Computing Review: Common Tools and Community Software

Fast simulation techniques

- LHCb has already been quite successful in producing factors more events without a corresponding increase in computing resources



Year	Simulated events (10^9)	Stored events (10^9)	Ratio	CPU work kHS06.y	CPU per event kHS06.s	LFS TB
2017	10.3	4.2	40.3%	817	2.50	640
2018	12.0	3.0	25.3%	1009	2.65	550
2019	45.0	6.9	15.2%	1290	0.90	1110
2020	67.0	16.8	31.7%	1357	0.81	2010
2021	80.0	11.1	13.9%	1815	0.72	2030

ReDecay

Signal Particle Gun

Tracker Only

RICHLess

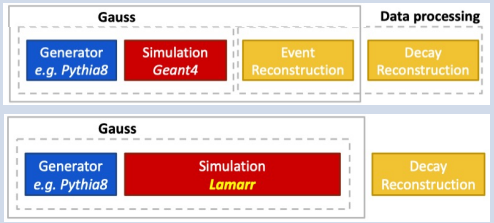
SplitSim

PoS ICHEP2018 (2019) 271

Fast simulations in LHCb

- Several R&D projects ongoing to further exploit faster simulation options

Lamarr – ultra fast simulation



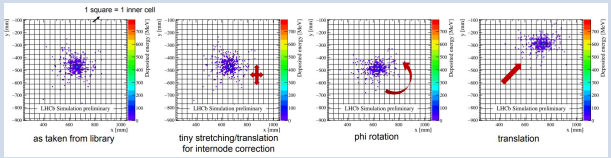
a pipeline of modular parametrizations replacing both the detector simulation and the reconstruction

more in ACAT 2022 poster by M. Barbetti; LHCb-FIGURE-2022-014

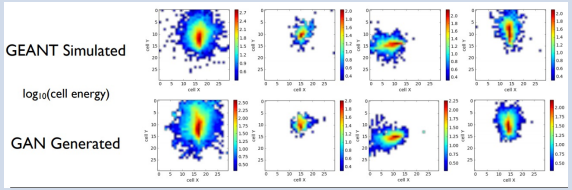
Fast simulation models

replacing Geant4 for a subdetector

- Point library for calorimeters – extract energy deposits from a collection obtained from a detailed simulation and transform them based on the property of the impinging particle [a]



- GANs – use GANs trained on the data produced by a detailed simulation to generate showers in Electromagnetic Calorimeter [b]

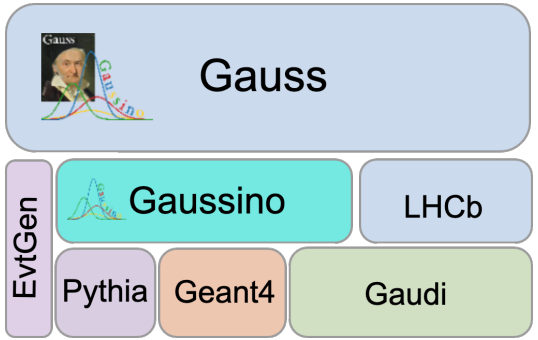


[a] LHCb-TALK-2020-108, ICHEP 2020 [b] EPJ Web Conf 245 (2020) 02026

Reminder of new simulation software structure

Gaussino – the new Core Simulation framework

- experiment-independent core elements extracted from Gauss
  - the structure and the hooks
  - components for HEP-wide software, e.g. Pythia8 and Geant4
- minimal functionality in stand-alone mode
- **ideal test-bed for new developments**



Gauss[-on-Gaussino] – the new version of the LHCb simulation framework

- built on top of Gaussino
- adds LHCb-specific parts

Optimise turn around – Fast simulations

- Exploit the Gaussino **custom physics** infrastructure
  - Can interface with custom physics, i.e. parametrisations, ad-hoc implementations and libraries and machine learning methods
  - All in a coherent and robust way
  - Details on infrastrutcutre in M. Mazurek talk from December LHCb week ECAL parallel
  - Details on machine learning software integration in M. Mazurek talk at 22 March Simulation Developmet meeting

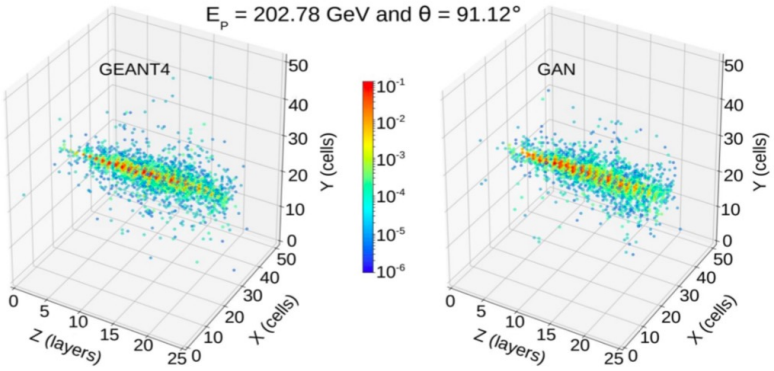
Model	Generation	Decay	Propagation	Status in G-on-G
ReDecay	✓	✓	✓	done
ParticleGun	✓	✓	✓	done
SplitSim	✓	✗	✓	done
RICHless	✗	✗	✓	under tests
TrackerOnly	✗	✗	✓	under tests
Lamarr	✗	✗	✓	(**NEW**) in progress
Point library	✗	✗	✓	(**NEW**) in progress
GANs	✗	✗	✓	(**NEW**) in progress

New challenges for machine learning in fast simulation

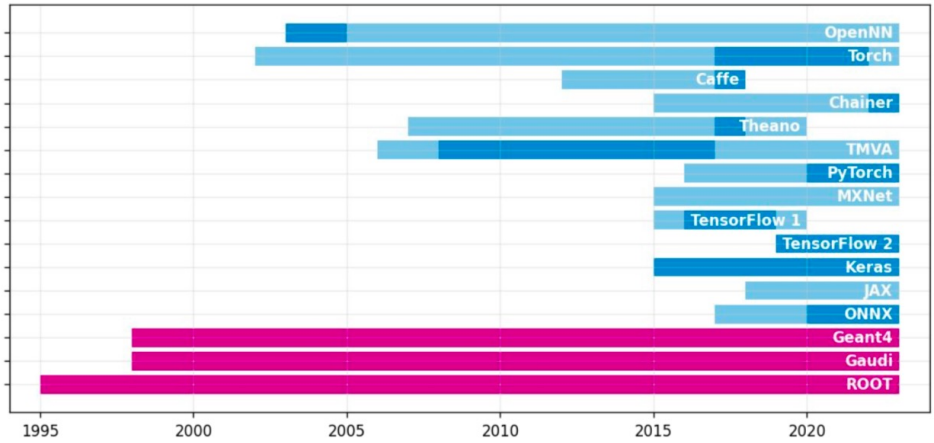
- Simulate time-evolution of ECAL showers

Showers develop in O(ns), becoming relevant for O(ps)-resolution timing detectors.

Recently succeeded training 3D GANs for full spatial correlations [Khattak (2021)]. 4D was never attempted.



L. Anderini



- The deployment of these models in our framework can hardly be achieved without a **solid integration of third-party machine learning software**.

Which one? It's a bet. **Torch in Gaussino**

Particle-to-particle correlations in ultra-fast simulation

L. Anderini

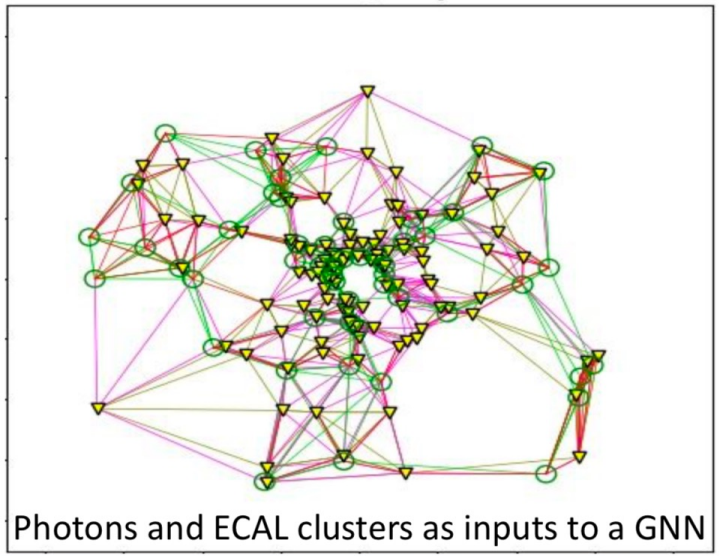
Ultra-fast (or *parametric*) simulation, today relies on the assumption that **each particle can be processed separately**, accounting *statistically* for the rest of the event.

Treating particle-to-particle correlations:

- will become critical with higher multiplicities
- will extend the application of ultra-fast simulation to more applications

More details in the slides of the Lamarr Workshop

Modern architectures such as **GNNs** and **Transformers** provide tools for describing complex relations between ***spatially-correlated objects***, and applications in HEP are just dawning.



Photons and ECAL clusters as inputs to a GNN

Learning from Run 3 for Run 4 and Run 5

Several challenges are emerging with Run2 → Run3 migration, already, involving many different aspects.

Reconstruction & Analysis

- Support for **multiple Event Models** for reconstructed quantities;
- Interactions between **Python and C++ frameworks** used for training and deployment, respectively;

CoreSoft

ML Theory

- Assess **uncertainties** arising from the adoption of generative models;
- Access to **important, distributed resources** for training the models, extremely difficult to provide and account as part of WLCG pledges;

Distributed Computing

Management & Social Science

- Overall, a novelty for the HEP field, creating new opportunities of **inter-experiment collaborations**.

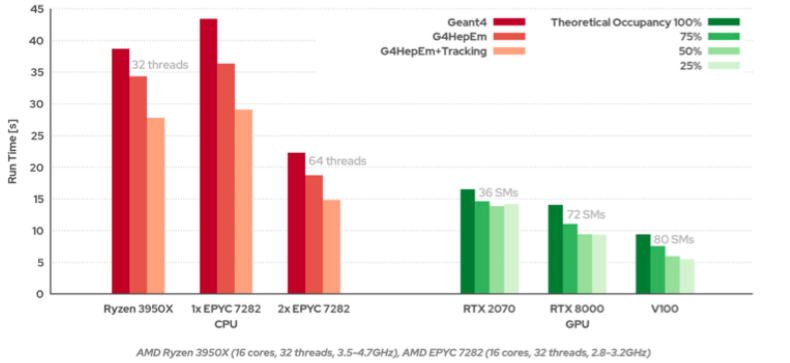
Compute Accelerators

- General simulation not a natural candidate for GPUs
  - Complex physics (many models), branches and special cases
  - Workload not known in advance due to stochastic nature
  - But, we need to try it out: more common and better performance then x86 CPUs
- Successful projects in medical physics, optical photons (Juno), neutron transport
- **Three R&D prototypes** in the last year for testing HEP use case
  - Can particle transport be **efficient** enough on GPUs?
  - How much **effort** for a production quality tool?
  - Tackle a limited scope problem computationally intense
  - Concentrate on one particular physics area
  - Deal with evolving populations of particles
  - Stay on the GPU for all key operations: e.g. geometry, magnetic field

GPU Prototype – AdePT

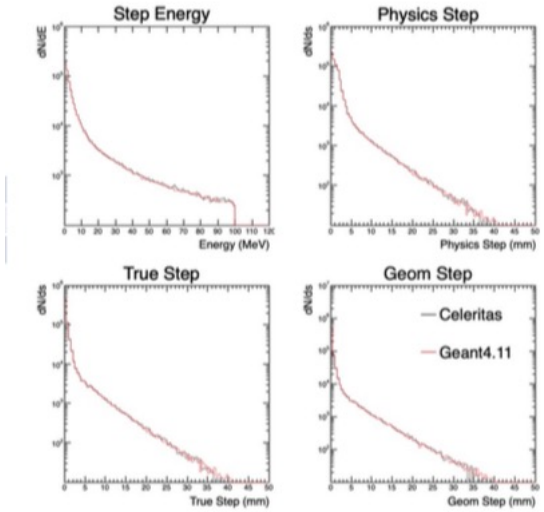
- First prototype for e-, e+ and gammas shower on GPU presented and discussed at HSF Detector Simulation on GPU Community Meeting
  - Full set of interactions of e-, e+, gammas (implemented by G4HepEm)
  - Navigation in complex geometry models using VecGeom (read from GDML, slabs and CMS geometry)
  - Helix propagation of charged particles in a constant magnetic field
  - Simple hit generation code (transferred from GPU to host)
  - HepMC interface for input events
  - Implemented both standalone and G4-integrated workflows (using fast simulation interface)

A. Gheata, W. Pokorski



GPU Prototype – Celeritas

- Prototype for electromagnetic showers on GPU presented and discussed at HSF Detector Simulation on GPU Community Meeting
  - GPU-focused implementation of HEP detector simulation
  - Roughly 10–25 × speedup for tested problems using Celeritas on GPU vs CPU on a full Summit node
  - Good agreement with Geant4 for preliminary test problems (energy deposition and step length distributions)
  - Support for standard EM physics, GDML geometry, magnetic fields
  - “MC truth” output and other diagnostics
  - interface for integrating directly into Geant4 to offload EM tracks
  - Version 0.2.0 released January 2023



EXASCALE COMPUTING PROJECT

Multi-institution collaboration (4–5 FTEs)
Funded through US DOE

P. Canal, S. Johnson

GPU Prototype – Opticks/CaTS

- **Opticks**: open-source project that accelerates optical photon simulation by integrating NVIDIA GPU ray tracing, accessed via NVIDIA OptiX
  - Re-implementation of Opticks for OptiX 7 required huge changes due to the new and very different OptiX API (>7.)
  - Moved code that doesn't require OptiX or Cuda out of GPU context.
- **CaTS**: interfaces Geant4 user code with Opticks using the G4CXOpticks interface provided by Opticks.
  - It defines a hybrid workflow where generation and tracing of optical photons is offloaded to Opticks (GPU), while Geant4 (CPU) handles all other particles.
  - Included since Geant4 11.0 as an advanced example
  - Very preliminary benchmarking results with the new workflow: ~ 200 fold speed up compared to single-thread Geant4 (11.0.p3). Results vary depending on geometry, photon yield, computing hardware ...

Developed by Simon Blyth (CAS, Bejing)

https://doi.org/10.1051/epjconf/201921402027

Gaussino and GPUs

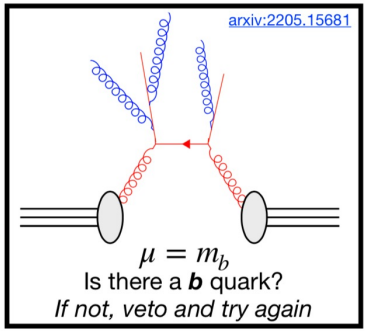
- What level of changes will be required for experiment's production frameworks?
- AdePT in Gauss-on-Gaussino for ECAL starting this year
  - Fits well with AdePT 2023 plans for integration in experiments' frameworks
  - Exploit custom simulation interface in Gaussino
  - Proof-of-concepts for passing part of the physics to GPUs
  - In parallel AdePT will try out LHCb geometry via GMDL
- Opticks/CaTS for RICH optical photons
  - Being explored by LHCb UK institutes

Generators

- For some LHCb EventType already the reason of productions `time-out` due to signal filtering
- Pythia8, and EvtGen combined, developments including on speed optimisation
  - Monash-Warwick Particle Physics Alliance
- GPUs
  - Generators are 10%-20% of CPU budget in ATLAS
  - ME generators (NLO) more time consuming. MG5aMC on GPUs work by A. Valassi et al. in context of HSF Generators
  - Ideas to tackle some aspects of Pythia8 on GPUs + GPU by A. Valassi https://arxiv.org/abs/2109.14938
- Machine Learning
  - Better suited than detector simulation
  - ML for hadronization by P.Ilten et al in Cincinnati
  - ML to evaluate hadronization uncertainties in Pythia8 https://arxiv.org/abs/2203.04983

Pythia User hooks

- The Pythia `Userhook` classes allow us to inspect and veto events during the generation
  - This can speeds up inclusive generation of events with b-quarks by **~10x**
  - Reduce time spent **evolving+hadronising** we know can't produce b quarks



- Developments aim make the generation of rarer hadrons or events with multiple heavy quarks feasibly with Pythia
  - e.g. heavy baryons, B_c^+ mesons

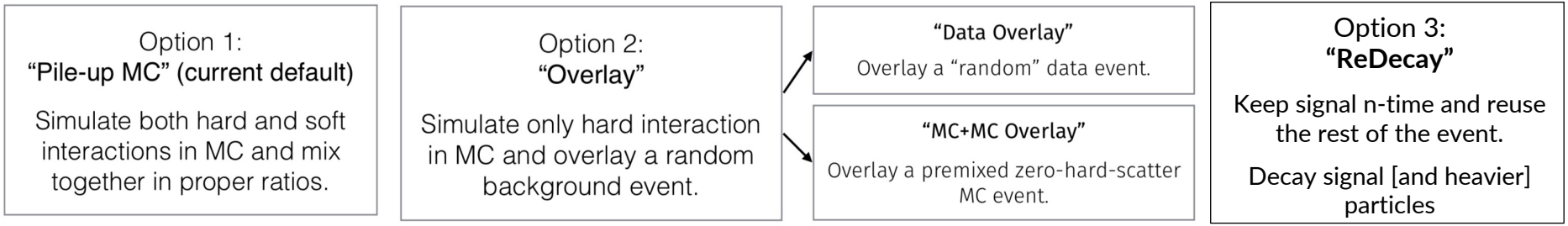
LHCb SimDev Meeting
Alice Week

Collaborative effort between LHCb and Pythia colleagues in Monash

Validation underway in LHCb to determine its impact on generator-level distributions and timing

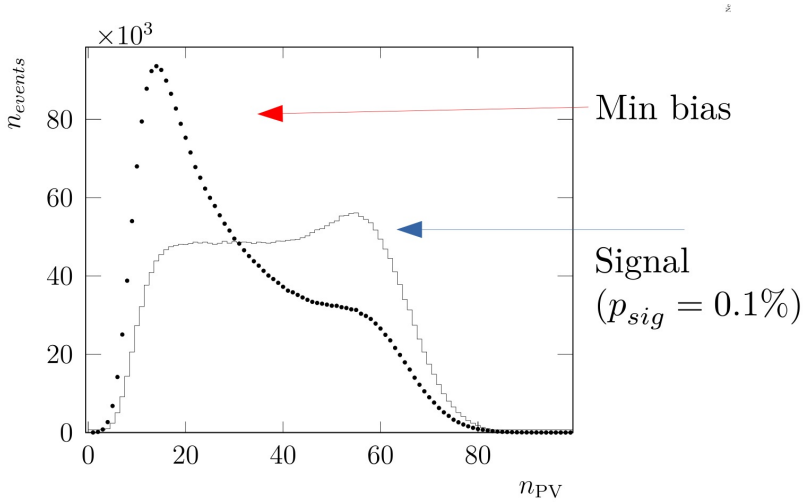
Tackling Pileup

- Other collisions in current and additional bunch crossing
  - Varies during the fill
- Rather urgent
  - Cannot keep the same implementation as we have (**no factor 10** in CPU pledges nor speed)



[On-time] Pileup

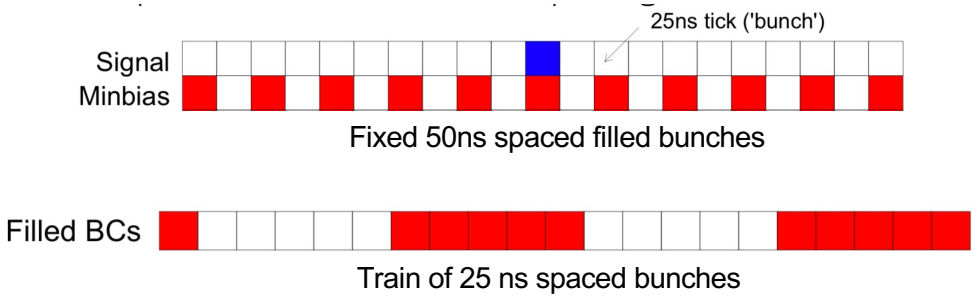
- Additional pp collisions, **10-80**, per bunch crossing
- Related to number of [reconstructed] PVs
- Need accurate vertex position [and time]!

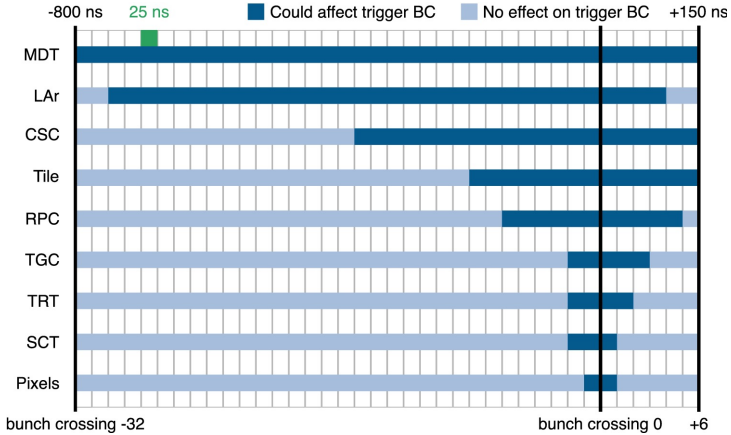


© T. Evans

Spill-over a.k.a. [Out-of-time] Pile-up

- Sensitivity varies between detectors
- Depends on structure of filled and empty bunches





- Can use different implementations
  - Parametrisation
  - Partial simulations
  - Correlation between subdetectors introduce complexity

- Do we need to worry about it with ps detectors?

How to put things together?

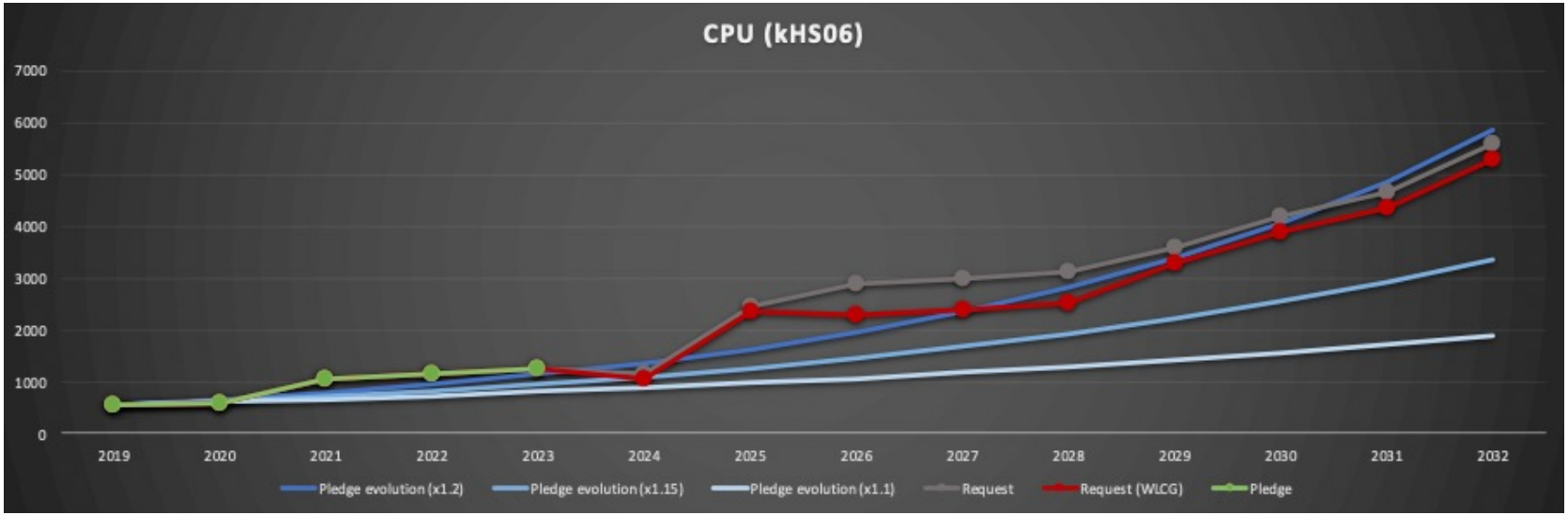
- The simulation framework is the key
  - Use Gaussino as a test bed
- Ideas of
  - Mix of fast / ultra-fast / detailed simulation for different collisions in same ‘event’
  - Simulation of partial detector
  - Simulate only events interesting, à la SplitSim…. More complicated if interst is based on reconstructed quantities
- Docker container to exploit clouds and HPC…

Outlook

- We need to invest in R&D and be open to new strategies
  - Exploring different “dimensions” of simulation
  - Exploit heterogeneous architectures
- Flexibility is key
  -
- Try out new idea, learn from Run3 and already profit in Run4

BACKUP

Computing resources forecast for U2



Aim of simulation

Provide the **underlying simulation framework** to **explore different solutions** and promote their seamless **integration**, while continuing to support the **immediate needs of the experiment**