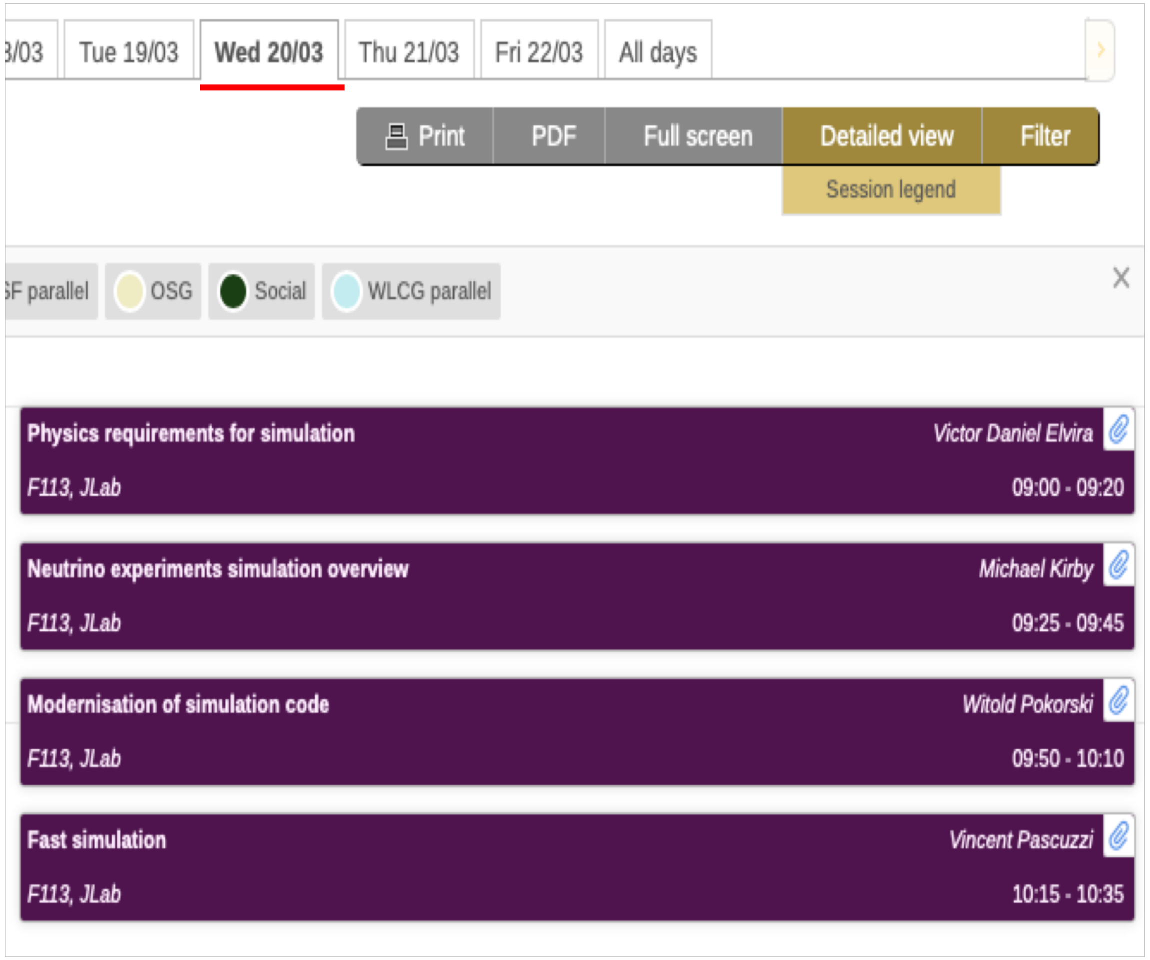
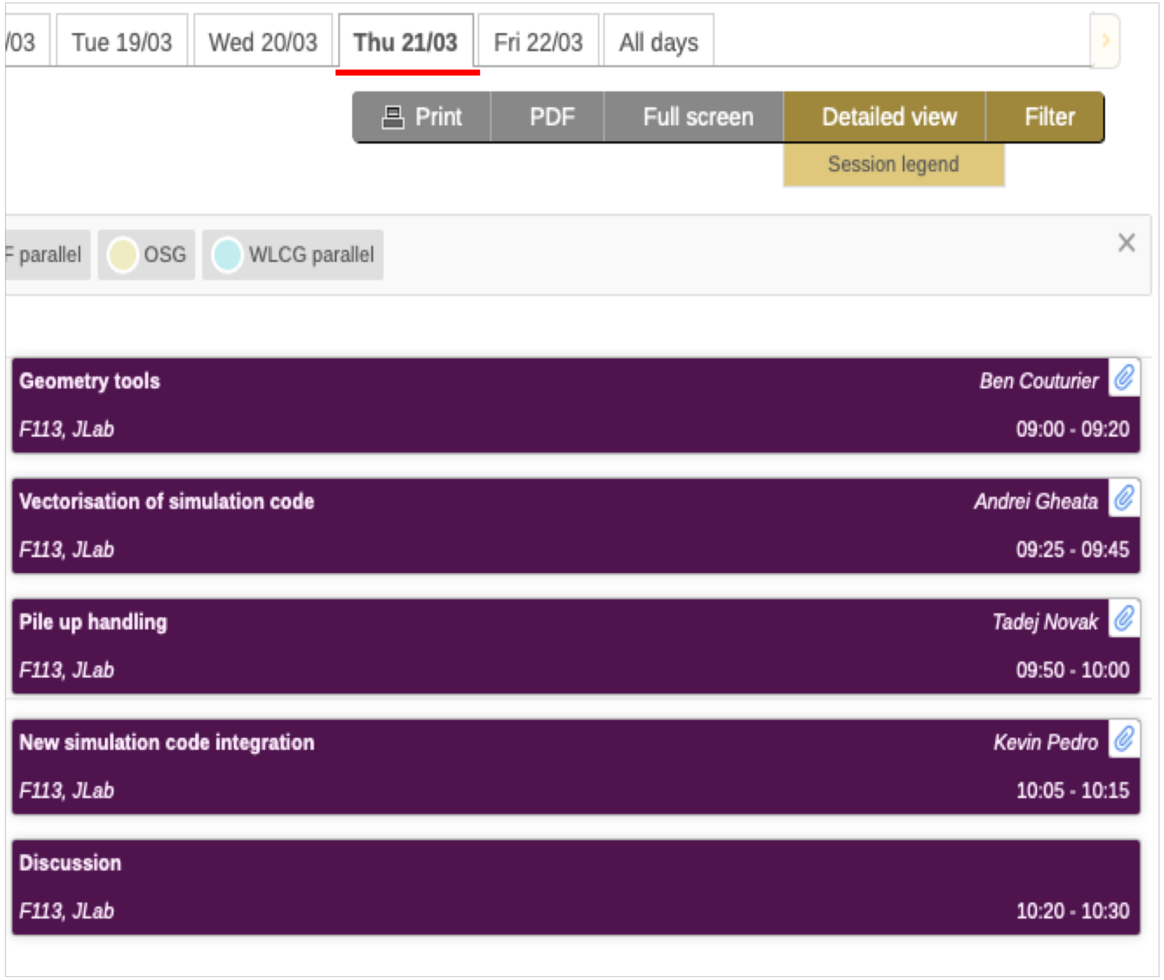
HSF Annual Workshop 2019 Detector Simulation Session

Witek Pokorski

01.04.2019

Agenda (Heather Gray (ATLAS), Gloria Corti (LHCb), W.P. (SFT))





https://indico.cern.ch/event/759388/timetable/#20190320

Physics Models in Detector Simulation

V. Daniel Elvira
HSF Workshop – how 2019
March 20th, 2019

Scope and goals, Outline

Overview of the detector simulation physics models needs of HEP experiments

- Restricted to LHC experiments (ALICE, ATLAS, CMS, LHCb) and Belle II
  - Other experiments are discussed in other talks
- Focused on the Geant4 toolkit

Discuss differences and commonalities among experiments

Identify opportunities for collaboration

- Introduction
  - Physics in Geant4, challenges identified during the HSF roadmap and Simulation CWP process
- Reports from the experiments – recent developments, tests, needs
- Summary and outlook

Summary and outlook

HL-LHC and B factory experiments in the 2020's require higher physics simulation accuracy and lower execution times, simultaneously

- ALICE, ATLAS, CMS, and LHCb report some disagreements of Geant4 predictions for shower energy fluctuations and lateral shapes, particularly in the low energy range
  - Best models for detailed shower description come at a high time performance cost
- LHCb reported a low kaon cross section asymmetry in G4 prompted a fix in v10.3.p03
- Belle II is developing a dedicated physics list to increase the contribution of hadronic models (Bertini) and achieve broader showers at low energies
- LHCb needs reliable modeling of Cerenkov processes
- ALICE needs correct light nucleon interactions ported to the FTFP_BERT physics list

The experiments and G4 collaborate effectively in the area of physics validation

Introduction: physics in Geant4

Very few of our HEP colleagues know how physics is handled within Geant4

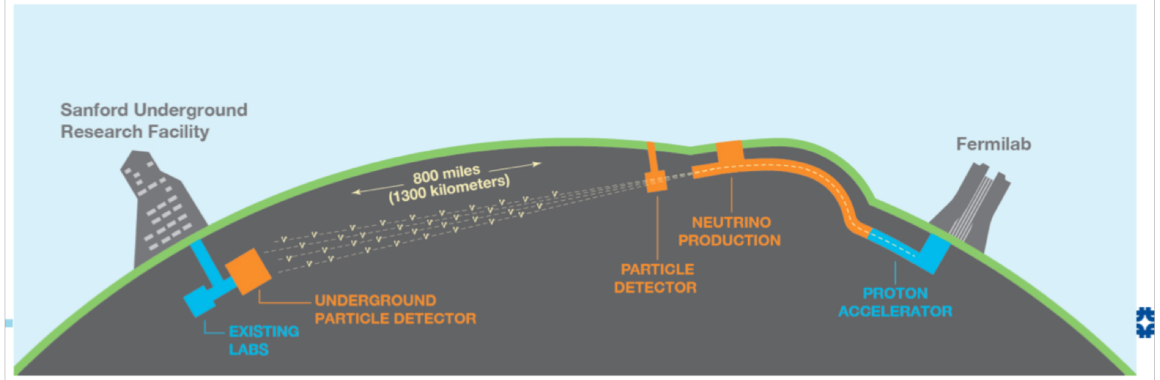
- Tens of models to describe different EM, hadronic, decay processes (sub-eV to TeV)
  - E.g. of EM: Compton, Photoelectric, ionization, bremsstrahlung, multiple scattering, ionization
  - E.g. of HAD: stopping, decay, elastic and inelastic models, capture models, fission
- Theory-based and parametric models
  - Theory-based preferred for prediction power in regions with little or no data

Neutrino Experiment Simulation Overview

Michael Kirby, Fermilab/Scientific Computing Division
Mar 20, 2019
Thomas Jefferson National Accelerator Facility

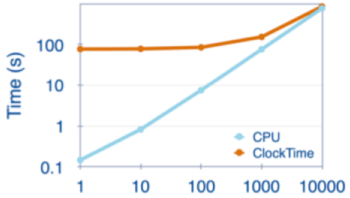
Outline

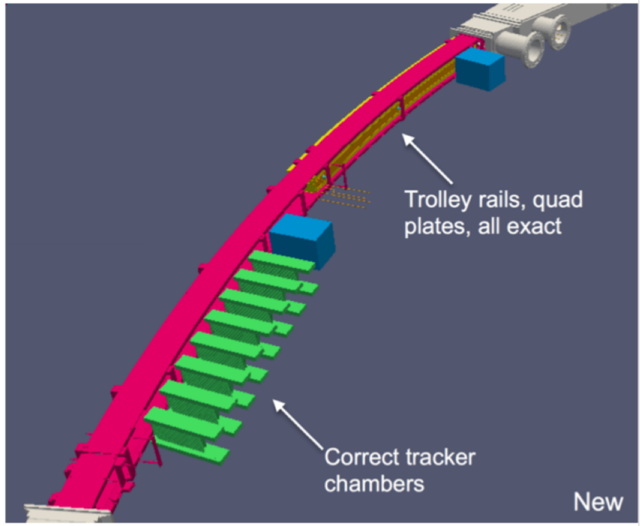
- outlook on precision measurements in neutrino oscillations
- where simulations come into the picture
- simulation of neutrino beam fluxes and systematics
- neutrino interaction event generators and cross sections
- detector simulation with GEANT4
- slight diversion about other IF experiments at Fermilab



CADMesh utilized by the Muon g-2 Experiment

- Translates CAD files into GDML for simulation in GEANT
- allows for precise shape and location of detector components without recreation in GDML by hand
- does require greater precision than engineers are sometimes focused on
- gaps in volumes and overlapping volumes can be serious problems in GEANT

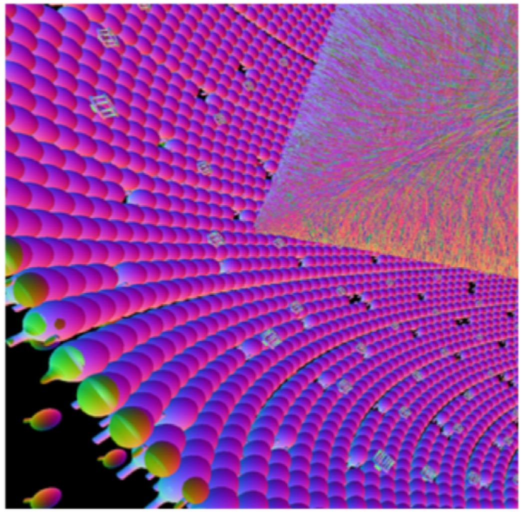




Leah Welty-Reiger, Renee Fatemi

Opticks: GPU-based full simulation

- ***State-of-the-art GPU ray tracing (NVIDIA OptiX) applied to optical photon simulation***
  - replace Geant4 optical simulation with GPU equivalent
  - translate G4 geometry to GPU without approximation, (CSG implemented on GPU)
  - port G4 optical physics to CUDA
- ***Optical photons generated + propagated entirely on GPU, highly parallel***
  - only photons hitting PMTs require CPU memory
  - expected speedup : *Opticks* > 1000x *Geant4*
  - eliminates memory + time bottlenecks
- ***Status : validation iteration ongoing***
  - validation by direct comparison of random sequence aligned GPU and CPU simulations



http://bitbucket.org/simoncblyth/opticks

Simulation Code Modernization

Witek Pokorski
EP-SFT, CERN
20.03.2019 HOW

- Motivation
- Experiments status and plans
- Simulation toolkits status and plans
- Conclusion

Motivation

- future accelerators (HL-LHC, FCC) experiments need a large speed-up in detector simulation (one of dominant CPU-time consumers)
  - requirement of at least an order of magnitude speed-up in simulation (more to simulate pile-up)
    - HSF Community White Paper
      - https://arxiv.org/pdf/1712.06982.pdf
      - https://arxiv.org/pdf/1803.04165.pdf
- we need
  - better algorithms
  - better code
  - efficient use of current (and future) computing architectures
- **we need to modernize our code!**

Summary of experiments input

- Full simulation using multi-threading as current (near future) 'production mode'
  - ALICE (and Belle II) using multi-process framework with 'late forking' and messaging system
- new geometry library (VecGeom) demonstrates how modern code (and internal vectorization) can help
  - although in case of geometries dominated by simple solids (like LHCb) the gain would come from navigation and not just solids
- Fast simulation is (very) seriously taken into account by all the experiments (can't survive without it)
  - see Vince's talk
- GPUs fit naturally in conjunction with Machine Learning techniques being explored in the context of fast simulation
  - not yet possible to use in full simulation for HEP use-case, but efforts ongoing

Conclusion

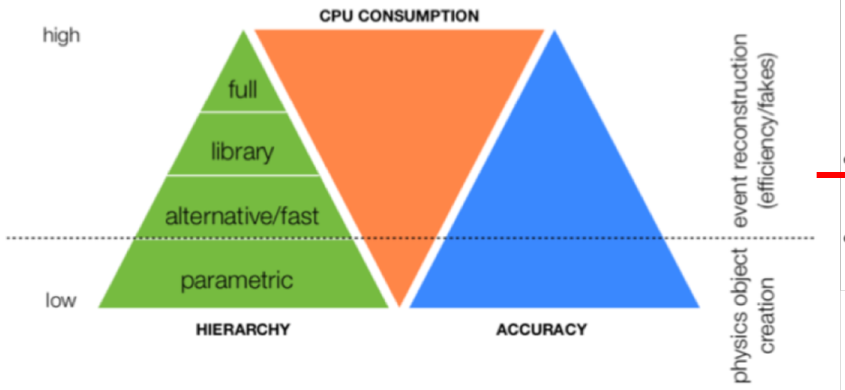
- experiments are moving (sooner or later…) to new solutions offered by the toolkit developers
  - they have no choice… they need it…
- several different R&Ds for simulation toolkits ongoing, but maybe too disjoined?
- new technologies available on the market (HPCs, GPUs, Machine Learning) but we seem not to be using them so far, why?
  - not adapted to our use case?
  - or are we too slow with modernizing our code…?
- seems that we (toolkits developers) are dragging a bit…?
- essential to have common, dedicated effort on simulation R&Ds
  - agile development with quick prototyping and testing of ideas
  - implementation of the successful ones in the production
- personally, I see it extremely positive to have the new R&D Geant4 Task Force
  - really looking forward to a lot of new, interesting developments

Fast simulation

HSF Parallel Session
2019 Joint HSF/OSG/WLCG Workshop
Jefferson Laboratory, Newport News, VA

Vincent R. Pascuzzi
University of Toronto

J. Phys.: Conf. Ser. 664 072024



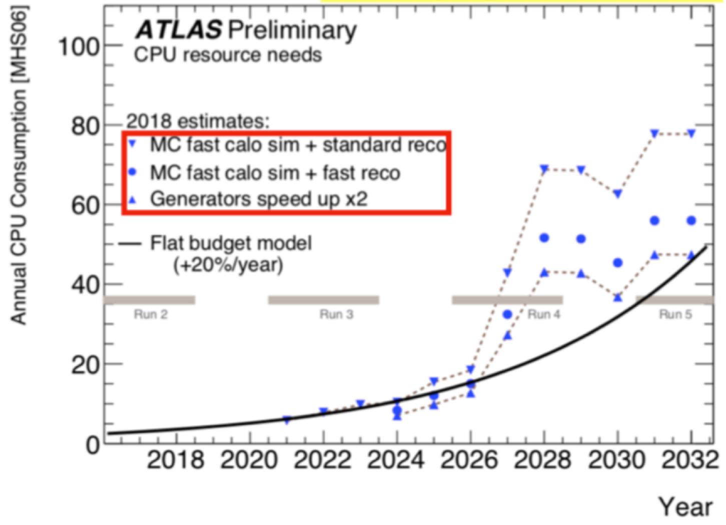
- ReDecay used for O(100M) events already, up to 50x speed-ups; additional 30-50x from frozen showers
- Other ideas: disable subdetectors, fully parameterized detector with Delphes

The problem

Large MC statistics to model recorded data

- Increasing luminosity poses greater challenges

ATLAS:ComputingandSoftwarePublicPlots



Summary

With an increasing (HL-)LHC luminosity, we need to look towards fast simulations

- Some techniques employed for nearly a decade, worked well
  - Parameterizations, reduced geometries, other simplifications
- We already know these won't suffice in the future
  - Some cases accuracy is also a problem

Next generation of fast simulation requires new strategies

- Integration of full+fast techniques into harmonized framework
- Much effort ongoing to exploit advanced statistical techniques and fast hardware
  - Machine learning
  - Accelerators and HPC

Deep learning shows promise

- LHC experiments devoting human-power into generative algorithms
  - Detector-agnostic implementation, use of GPUs, FPGAs, HPC
- Still some R&D before hitting mainstream/production
  - Scientific requirements and considerations are crucial

Geometry Tools for Simulation

HOW2019, Jefferson Lab
B.Couturier CERN

- Experiments have different ways to describe their geometries, in the persistent form on disk or in memory (mostly Constructive Solid Geometries)
    - ROOT TGeo (C++) representation: ALICE/FAIR
        - Loaded from GDML for LArSoft
    - Geant4 solids: Belle2
    - DD4hep (XML + C++ constructors, TGeo in memory)
        - FCC, STCF, ILD, Sid and CLICdp
    - Experiment designed frameworks:
        - CMS: DDCMS (XML, C++ constructors)
        - ATLAS: GeoModel (XML, C++ constructors)
        - LHCb: DetDesc (XML representation on disk, custom memory representation)

- Several descriptions are therefore needed depending on the cases
    - E.g. CMS have tracking *(reconstruction)* and simulation geometries *(built from the same description)*
    - Fast Simulation may require simplified detector
- Example solutions
    - Manual tuning of geometry for various purposes
        - LHCb: Simplified geometry, Delphes model etc
    - ATLAS Fatras https://iopscience.iop.org/article/10.1088/1742-6596/898/4/042016/meta
    - CMS Fastsim https://arxiv.org/abs/1701.03850

Representation model vs Implementation model

How to ensure consistency between all representations ?

Can we automatically simplify the geometry ?

Conclusions

- Many commonalities between experiments
    - But also many differences so the toolkit approach is appropriate
- Community efforts for new tools
  that are being adopted by experiments
    - LHCb moving to DD4hep
    - CMS investigating using it as a DD Mediator
      (i.e loading the description, XML and C++ algorithms remaining the same)
      Expected to complete by the end of 2019
- Designing, simplifying, misaligning geometries is not easy
    - (how) can the tools be improved to help ?
- HSF maybe can also help with good practices, advice etc...

Vectorization of simulation code

Andrei Gheata for GeantV R&D team

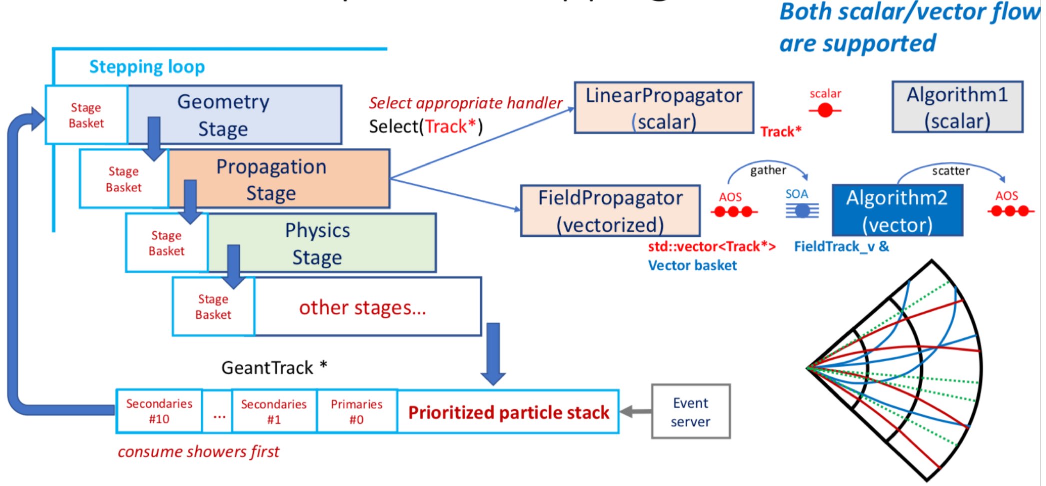
Vector Simulation R&D

- GeantV: performance study for a vector simulation workflow
  - An attempt to improve computation performance of Geant4

Outlook and conclusions

- GeantV prototype demonstrates that vectorizing a large-scale complex HEP application is possible
  - Most of the available DP-ops vectorized, about 50% visible
  - Still some vectorization potential left, more difficult to harvest
- Efficient vectorization is not a piece of cake (for simulation)
  - The limits of the "basket" model now visible, ongoing performance study to outline them
  - Having more computation hotspots would have helped…
- Contributions from basket workflow and vectorization do not explain the full performance gain, the major part (60-70%) is coming from other sources
  - improved instruction cache use, more compact code, less virtual calls, …
  - Currently trying to disentangle these effects
- Finalizing this performance study will outline the directions to go
  - Technical document (facts, numbers and lessons learned) to be prepared
  - What are the directions for adopting some of these benefits in Geant4

GeantV multi-particle stepping

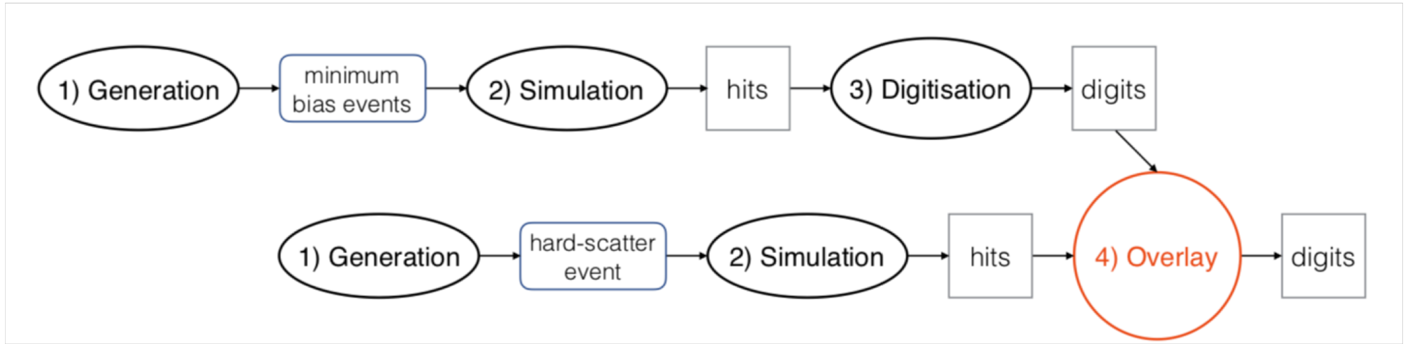


Pile-up Simulation at the LHC

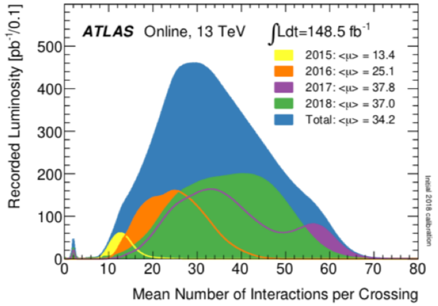
HOW Workshop 2019, HSF parallel: Detector Simulation
March 21, 2019

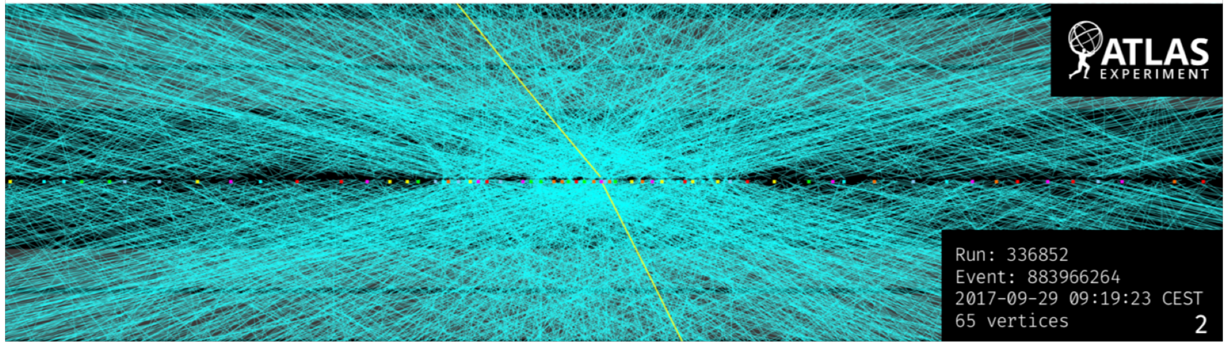
Jožef Stefan Institute
Ljubljana, Slovenia

Tadej Novak, on behalf of the
ATLAS, CMS and LHCb experiments



- Pile-up: soft collisions in current and surrounding bunch crossings.
- Number of interactions per crossing (μ) in Run 2 at ATLAS and CMS up to 80.
- LHCb observed on average 1.6 collisions per crossing in Run 2, secondary vertexing very important.





Conclusions & Outlook

- Pile-up simulation and handling very complex and CPU intensive.
- A problem for all experiments, especially in the HL-LHC.
- ATLAS and CMS will use pre-mixing with overlay for Run 3.
- LHCb considering to generalise parametrised pile-up hits for out-of-time pile-up for Run 3 to only need one heavy computation.

Integration of New Simulation Technologies in the Experiments

Kevin Pedro (FNAL)
on behalf of ATLAS, CMS, LHCb
Joint HSF/OSG/WLCG Workshop
March 21, 2019

VecGeom in the Experiments

CMS:
- VecGeom used in scalar mode with Geant4
- 7–13% speedup with similar memory usage
  - → Just from code improvements, no vectorization!
- ➢ Included in production for >1 year

ATLAS:
- VecGeom tested in scalar mode with Geant4: just Cones and PolyCones used
- 1–3% speedup observed
- Testing with all shapes from VecGeom in progress
  - Also testing w/ different Geant4 versions (10.4.2, 10.5)

LHCb:
- VecGeom tested in scalar mode with Geant4
- No speedup observed: likely due to simple shapes in detector geometry

DD4hep in the Experiments

- Not a vectorized package, but a common solution for detector description
- Uses ROOT TGeo to handle geometry construction

CMS:
- Infrastructure for migration in place, tested w/ muon system (drift tubes)
- Provided significant feedback to DD4hep developers
- Contacts established to migrate other subdetectors

ATLAS:
- Testing DD4hep as a description language
- Need to use GeoModel for backend rather than TGeo (not supported)

LHCb:
- Testing w/ Gaussino, new lightweight simulation framework (CHEP2018)
- Provided feedback for TGeo (optical surfaces)
- Plan to use DDG4 simulation toolkit to convert geometry for Geant4

Conclusions

- New packages relevant to simulation are available:
  - Common solutions (DD4hep)
  - Vectorized components (VecGeom, GeantV, etc.)
- Experiments are making progress testing and integrating these packages
  - Providing frequent feedback to developers
  - Continued communication is essential to the success of these projects
  - Observed speedups vary; many factors at play, and still early
- In particular, CMS integration testing of GeantV is maturing
  - Next step: performance testing w/ beta release
    - ➢ Check if speedup translates to experimental software framework
    - ➢ Check if existing CMS speedups are compatible w/ GeantV
    - ➢ Understand full cost of migration to new interfaces
  - Provide a path for other experiments to follow

Conclusion

- nice set of overview talks
  - Agenda a bit too packed
  - not enough room for discussion
  - longer session next time?
- several useful messages for the near future
  - important to focus effort on common solutions and reusable tools
- looking forward to new R&Ds taking shape
  - important to involve the community as much as possible
- HSF Detector Simulation WG topical meetings prove very interesting
  - allow to identify possibilities of collaboration