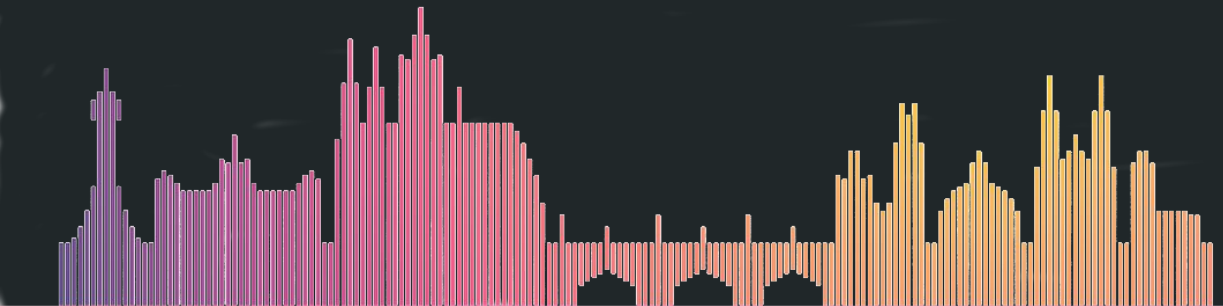
Detector Simulation Upgrades for HL-LHC

Graeme A Stewart, Guilherme Amadio, Andrei Gheata, Pere Mato, Witold Pokorski, Anna Zaborowska
CERN EP-SFT



ICHEP 2020 Virtual Prague
July 2020

High-Luminosity LHC

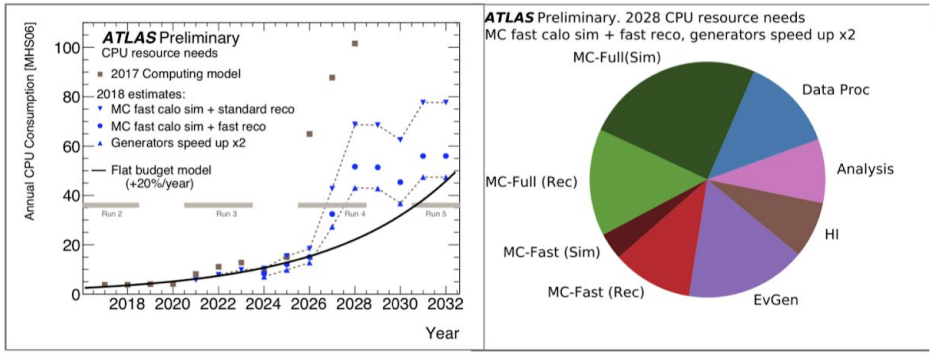
- Increase of beam luminosity to $7.5 \times 10^{34} cm^{-1} s^{-1}$
- Pile-up to reach 200 for ATLAS and CMS
- $300 fb.yr^{-1}$ and $3 ab^{-1}$ over the HL-LHC lifetime
- Huge increase in data to study Higgs physics, flavour physics, dark matter, etc.
    - About 20x more data than we have today
- *More beam data requires more simulated data to support analysis*

Anticipated Simulation Needs

Many physics and performance studies require large datasets of simulated events

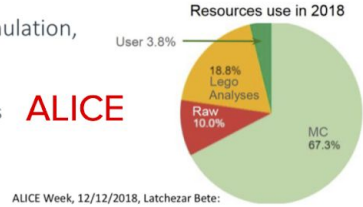
ATLAS

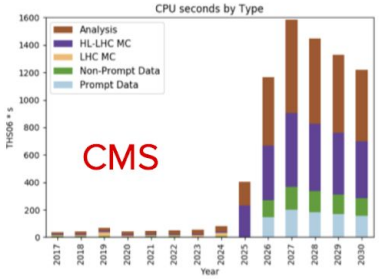
- Geant4 is highly CPU-intensive
- Already lacking statistics -- increasing luminosity poses greater challenges



- 2/3 of the computing resources are dedicated to MC simulation, all full sim
  - fast sim not used in production yet
  - fully parametrised fast simulation approach for upgrade studies
- expected 10-100 times more data in Runs 3 and 4
  - cannot cover that with current usage of full sim

ALICE

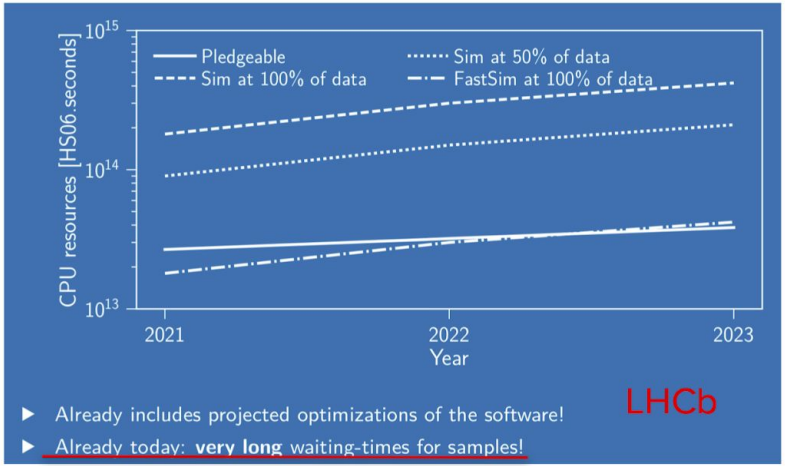




CMS

- Simulate more events to keep up with HL-LHC data volumes: 10×(Phase1)
- May also need to improve accuracy of physics lists to simulate HGCal
- Reconstruction will take longer due to high pileup and granular detectors

➢ Need more events, more accuracy, in more complicated geometry… w/ relatively smaller fraction of total CPU usage



LHCb

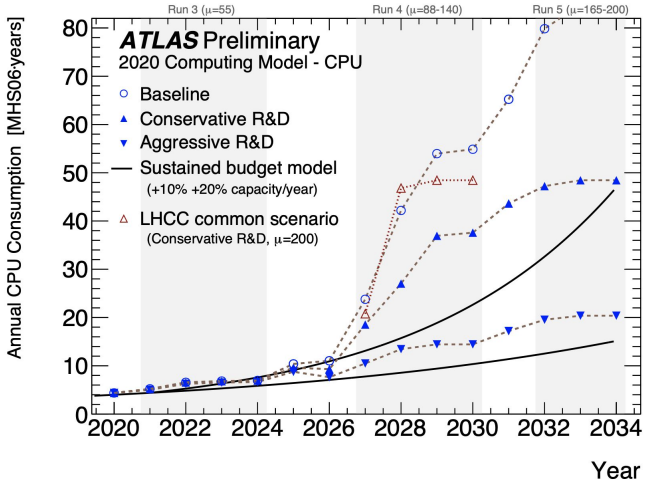
- Already includes projected optimizations of the software!
- Already today: **very long** waiting-times for samples!

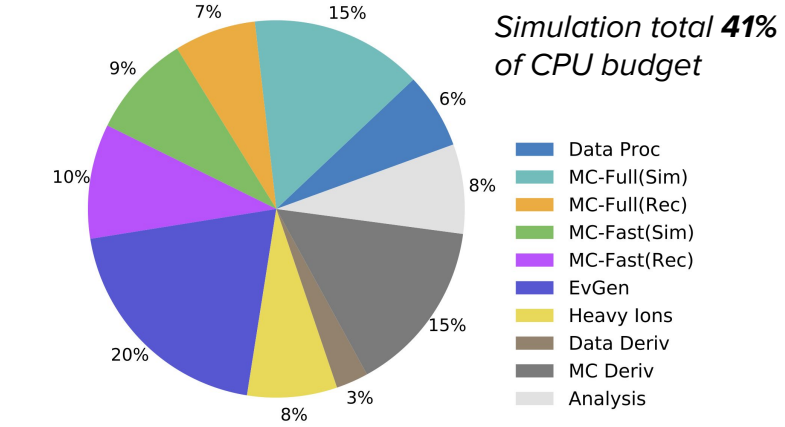
Simulation Software

- The cornerstone of simulation for the LHC is Geant4
  - Toolkit born in the 1990s, providing a highly flexible simulation framework in C++
  - Allowed progress in physics by comparison of competing models in the same energy ranges and complementary models to be combined to cover large energy ranges.
  - This delivered a physics precision that allowed the experiments to perform analysis of the collected data and to produce new results pushing the boundaries of our understanding in HEP
    - arXiv:1706.04293
- Geant4 mission
  - Provide production-quality simulation toolkit and support to experiments
  - Provide good long-term maintenance while making the toolkit more sustainable
  - Improve the physics models with better precision and energy range extensions, in particular the hadronic ones
  - Improve the overall computational performance of simulation

Bridging the Gap

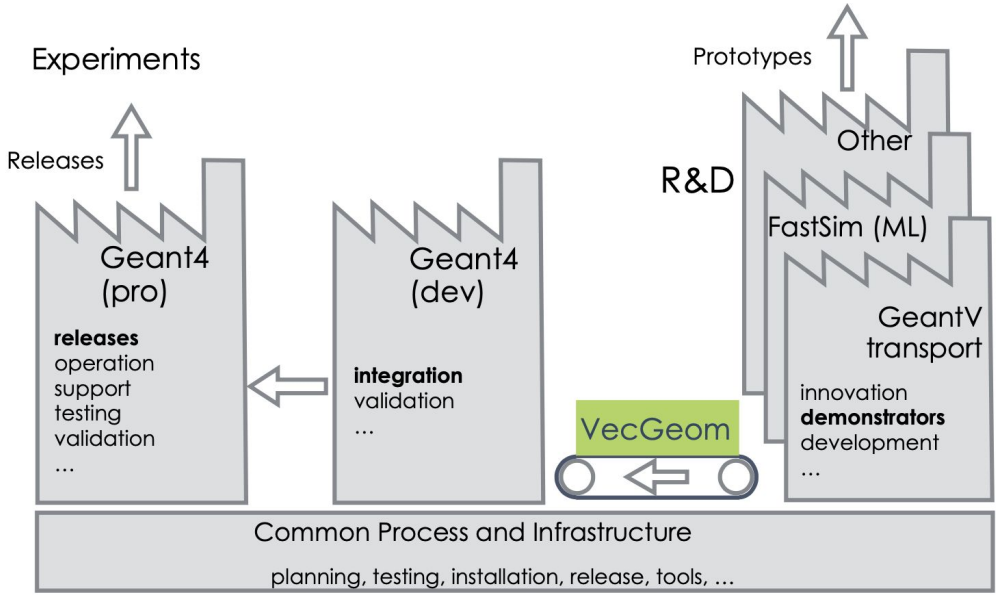
- Simulation does not per-se scale with pile-up (watch out for digitisation!)
- It does scale with event rate
  - Which will rise by about x10
- This puts extreme pressure on the computing budget
- Running today's full Geant4 simulation for the needs of HL-LHC would simply not be affordable
- *Simulation needs to be faster, without sacrificing relevant physics accuracy*





Strategic Development Model

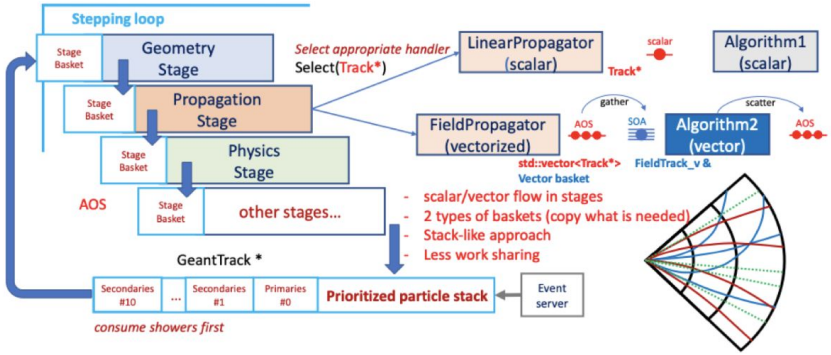
- Recognise that Geant4 is the critical production simulation software for the experiments
- Changes must be able to be introduced adiabatically



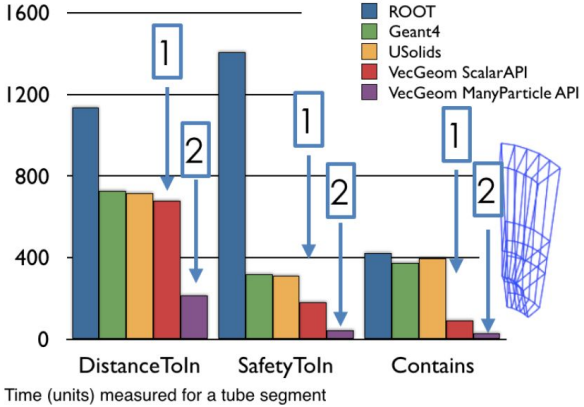
Three Tracks for Improvement

- Refactoring and internal improvements
  - Optimisation of current Geant4 code to run faster
  - Mostly work that is internal to Geant4, little direct impact for user code
- Fast Simulation
  - Replace detailed particle tracking models with different methods
  - Long tradition of parametric response implementations
  - Machine Learning is the hot topic here
- Hardware (R)Evolutions
  - Increasing trend away from purely CPU based machines
  - In particular GPUs become more and more common
  - So we have to start looking at how we could use these machines for detector simulation

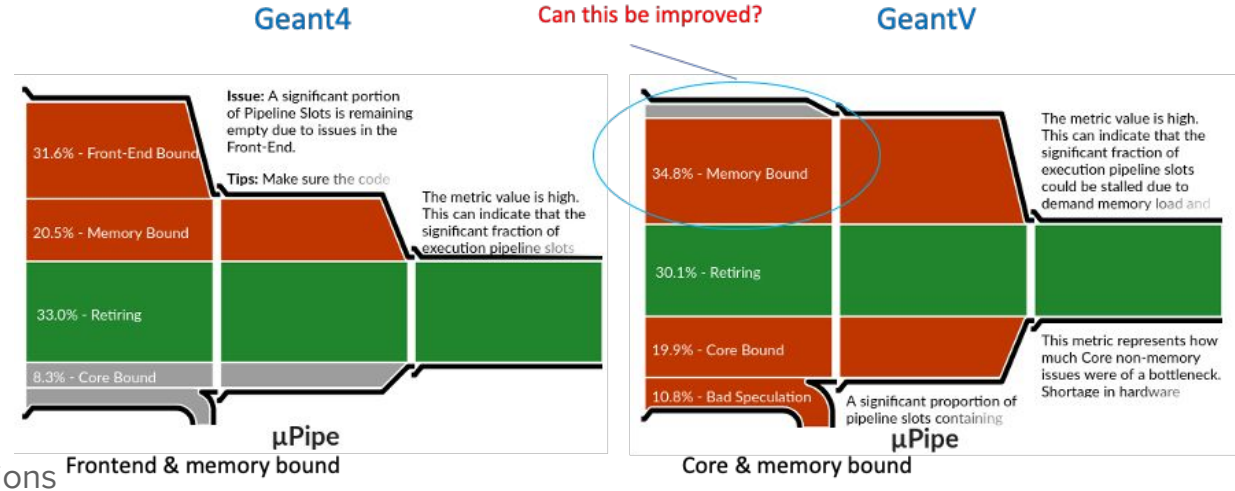
GeantV Prototype



- R&D in exploring a next generation simulation toolkit that would exploit vectorisation for particle transport
- Prototype with EM physics developed using *baskets* to gather work of the same type into *CPU vector register lanes*
  - Results in [arXiv:2005.00949] with careful comparison with Geant4
  - X2 speed up observed, but *highly architecture dependent*
- Developed optimised libraries that are used now in Geant4 and in ROOT as well as other projects
  - VecCore, VecGeom, VecMath



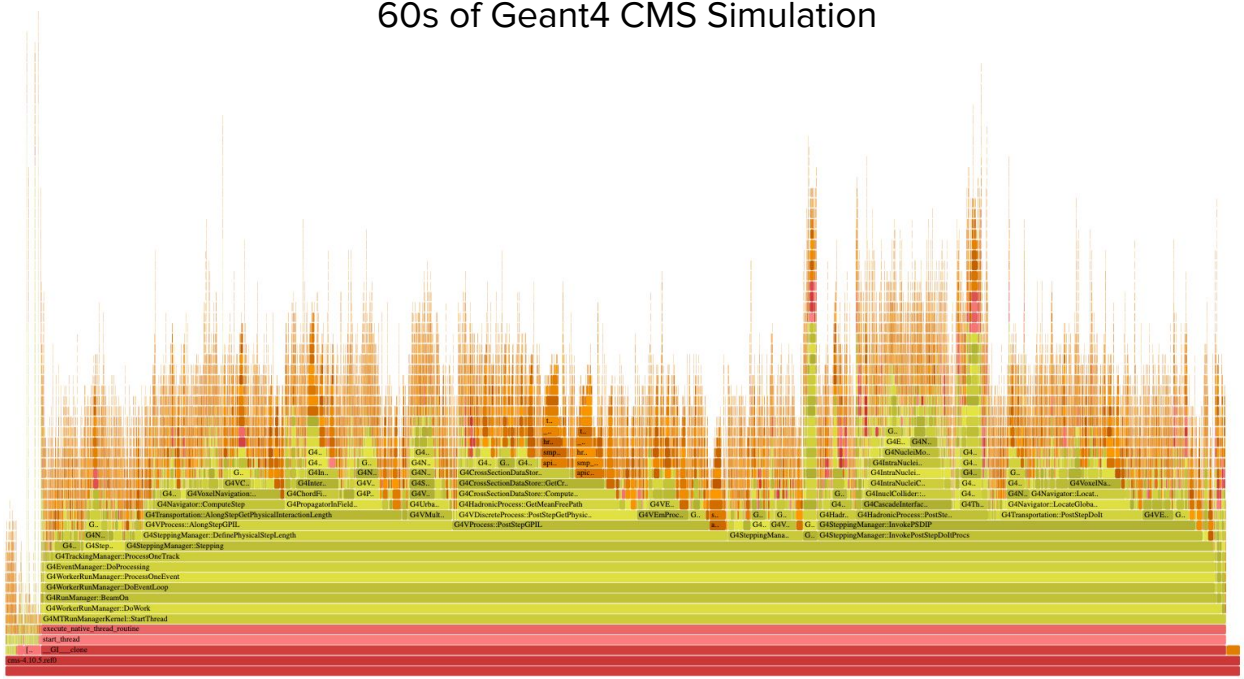
GeantV Lessons



- Main speed-up factors
  - Better data cache usage
  - Better instruction cache use (simpler code with fewer classes, less indirections and banching)
- Vectorisation impact was much smaller than was hoped for
  - Small fraction of code could be vectorised or run efficiently in SIMD
  - Overheads of data reshuffling (basketisation) canceled gains from vectorisation
    - Pay for extra memory copy in track collection or destroy coherency (array of pointers)
  - Basketisation did bring benefits for floating point hot-spots (B field, multiple scattering)
- Conclude that rewriting and modernising parts of Geant4 could bring tens of % speed-up, depending on CPU and caches
  - Compact code, use better data layouts, reduce virtual function calls

Geant4 Profiling

60s of Geant4 CMS Simulation

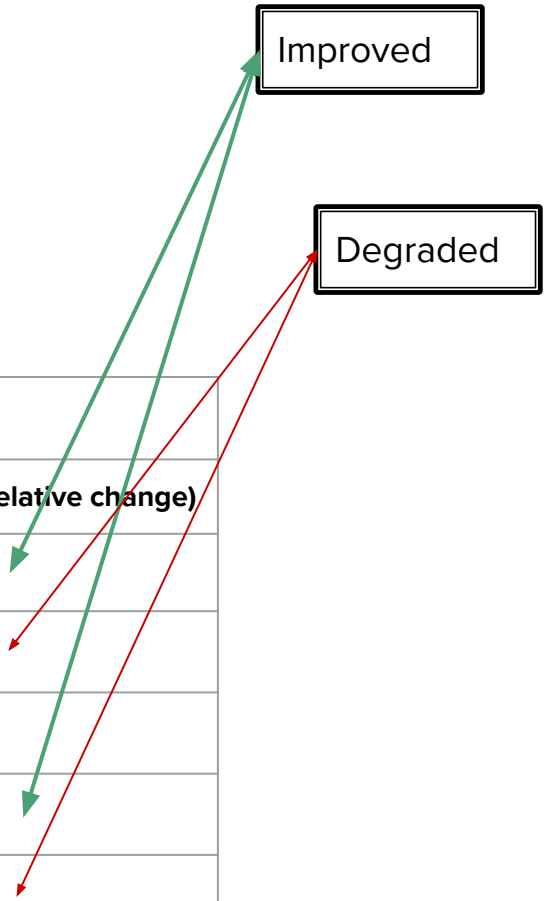


- Flame graph of a CMS simulation using Geant4
- Call graph appears as the “tower” at each point in time
- Characteristic short narrow spikes mean lots of deep stacks of calls that descend then unwind
- Puts pressure on the instruction cache, compounded by many threads all acting independently of each other
  - Threads calling geometry, physics (many processes!), magnetic field, etc.
  - R&D on sub-event parallelism may help alleviate this (J Masden, LBNL)

Perf analysis of Geant4

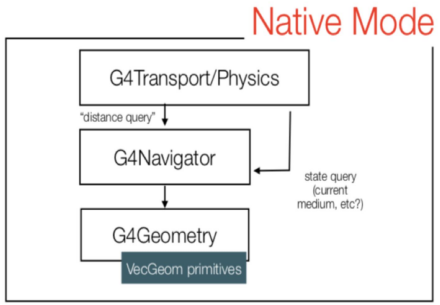
- Run full Geant4 jobs with Linux perf monitoring
  - Gives detailed access to the performance data from the CPU
  - Wealth of information (too much!)
- Look for changes across different releases

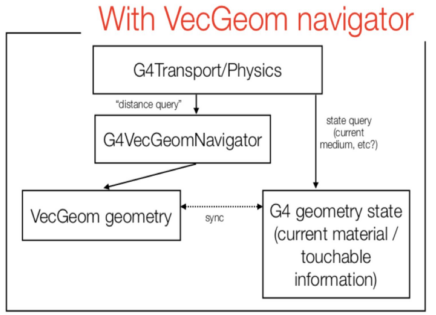
Metric	**Geant4 Version**	
branch-misses	**Master (absolute value)**	**10.5.1 (relative change)**
G4PhysicsVector::Value	4.56%	+6.18%
G4VEmProcess::PostStepGetPhysicalInteractionLength	4.68%	-2.05%
L1-dcache-load-misses		
G4PhysicsVector::Value	1.36%	+13.39%
G4VEmProcess::PostStepGetPhysicalInteractionLength	12.86%	-7.21%

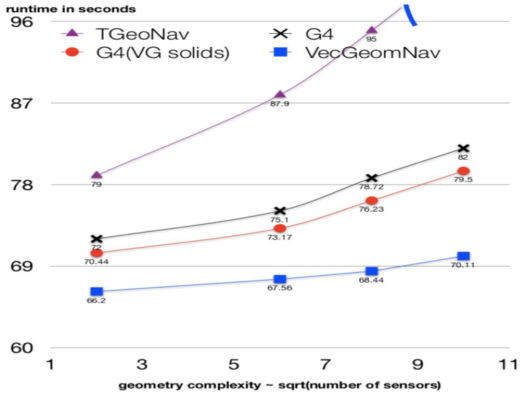


Improved Navigation

- VecGeom already integrated into Geant4
  - Reimplementation of geometry primitives
  - SIMD exploitation (from GeantV)
  - Modernised and improved algorithms
- Can also be used for improved navigation
  - Performance benchmark transporting 100k primary electrons through complex geometries yields encouraging numbers
  - Gain of 8 to 17% over current Geant4





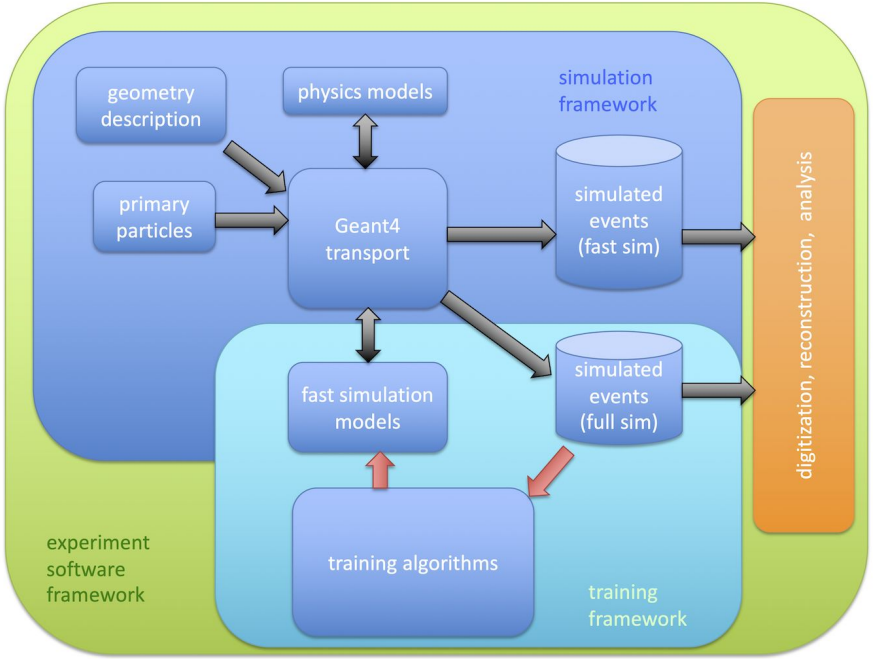


J Apostolakis, G Cosmo, S Wenzel

Alternative Transport

- Current simulation framework used in Geant4 (process interface, track, stepping algorithm, etc.) is completely general
  - provides a high level of flexibility
  - brings many potential sources of inefficiency
    - well defined simulation problems such as HEP detector simulations require a small fraction of these functionalities
- New R&D on simplified physics simulation "framework", tailored specifically for HEP detector simulations
  - targeting performance critical particles and their interactions with the highest level of specialization for HEP usage
  - highly reduced complexity and size of the corresponding code, more cache efficient data storage designed by considering the run time access pattern, etc
  - by creating a separate, self-contained part of the simulation provides a natural starting point for GPU-related R&D where the corresponding work is performed on the device

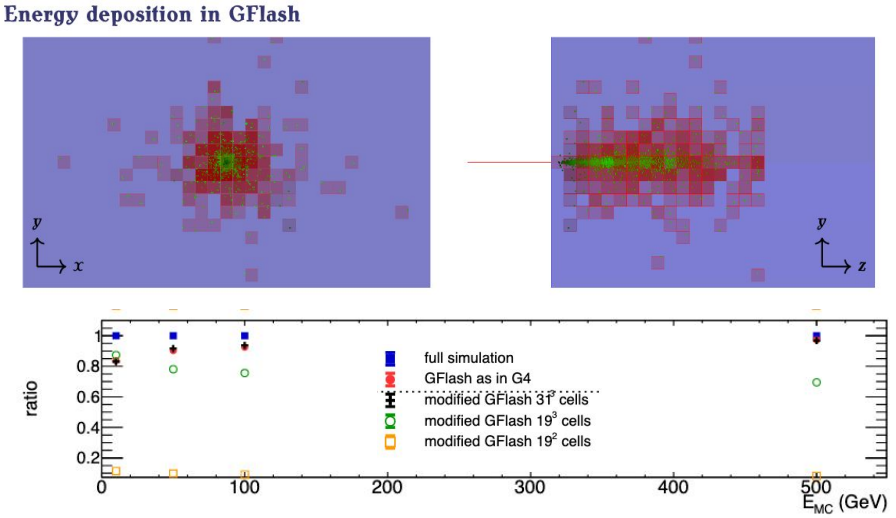
Fast Simulation



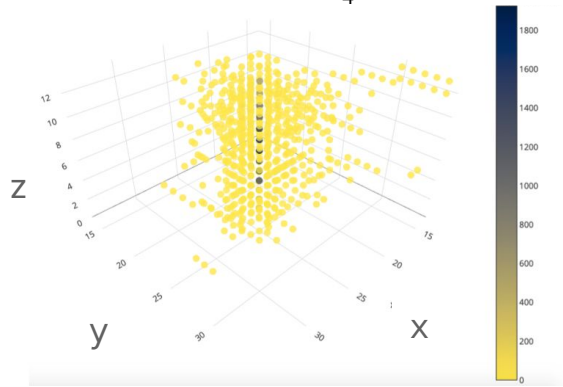
- Fast simulation is usually experiment dependent
  - Custom procedures to extract parameterisations
- Ongoing work to streamline this procedure
  - Users establish their setup
  - Full simulation producing standard information (usually hits)
  - Simplified, automatic easy way to extract parameters
  - Plug trained network back into simulation via Geant4 ML models
    - Should be able to import any model, howsoever trained
- More time spent on precision, less on integration

Fast Simulation Parametric and Machine Learning

- GFlash - historical parameterisation of calorimeter energy deposition
  - Dates back to CDF
  - Working to improving both the speed and accuracy of the implementation
- Auto-regressive neural networks training on different calorimeter data
  - Scalable and stable technique
  - Captures cell interdependencies well
  - $PbWO_4$, Pb/LAr, Pb/Sci, W/Scint
  - Validation against full simulations



Energy deposits from 19GeV Photon Generated Event in $PbWO_4$ Calorimeter

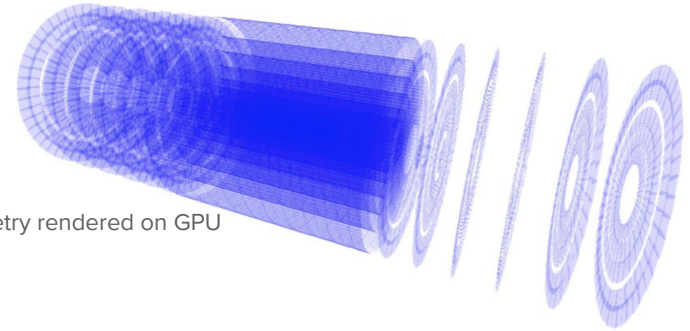


I Ifrim

Compute Accelerators

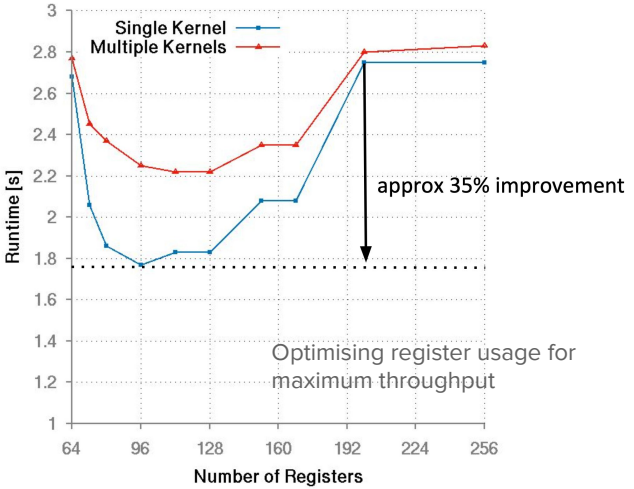
- General simulation not a natural candidate for GPUs
  - Complex physics (many models), branches and special cases
  - Workload not known in advance due to stochastic nature
  - But, this is a resource that becomes more common and where year-on-year performance improvements continue to outstrip x86 CPUs
- Successful projects in medical physics, optical photons (Juno), neutron transport
- Key questions for HEP a use case
  - Look for a problem that limits scope, but is 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 and field

R&D Prototypes



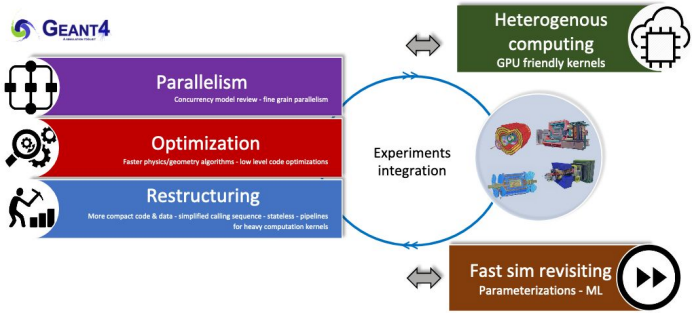
Track ML Geometry rendered on GPU

- Raytracing on GPUs
  - Upload of GDML geometry into GPU
    - Using VecGeom library
  - Ray tracing (no physics), using GPU threads for each ray
  - Tiled mode that uses streams on the GPU to batch workloads
  - Recent work has improved the code in a number of areas
    - Reducing the size of the geometry state to fit more tracks in device memory
    - Simple GPU-aware global navigation
- EM Physics on GPU
  - First prototype working to introduce evolution of particle populations
    - Pair production, bremsstrahlung
    - Allows testing how these populations can be managed on a GPU
      - Avoid too much sorting (overheads will kill performance)
      - Limit divergence and cut event tails
  - Later evolution to more realistic physics envisaged
- Aiming to then recombine different R&Ds into computationally realistic example



In collaboration with B Morgan, D Costanzo, M Hodgkinson (UK ExCALIBUR)

Conclusions



- High-Luminosity LHC brings huge physics opportunities to the field, but lays down a challenge for detector simulation
- Production simulation for the experiments needs to become faster to cope with greatly increased event rates
- Three pronged approach to this problem
  - Improve current Geant4 toolkit to become incrementally faster
  - Easier integration of fast simulation methods and techniques
  - R&D programme for the use of non-CPU devices
- All of these are required if simulation is to reach the required speed and accuracy
- We look forward to seeing these activities come to fruition over the next several years