R&D activities

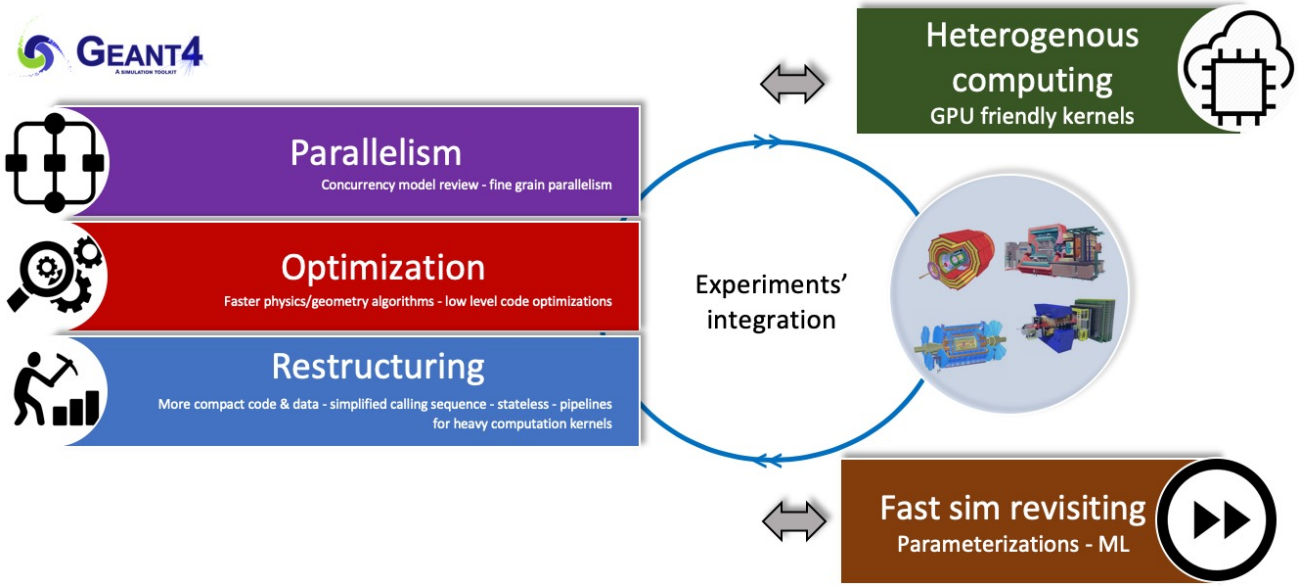
Witek Pokorski

30.09.2022

26th Geant4 Collaboration Meeting

Three Main Axes of Development

- **Improve, optimise and modernise** the existing Geant4 code to gain in performance for the detailed simulation
  - Re-structure the code to make possible major changes: task-oriented concurrency, specialisation of the physics (G4HepEm), Woodcock tracking, transportation+Msc, etc.



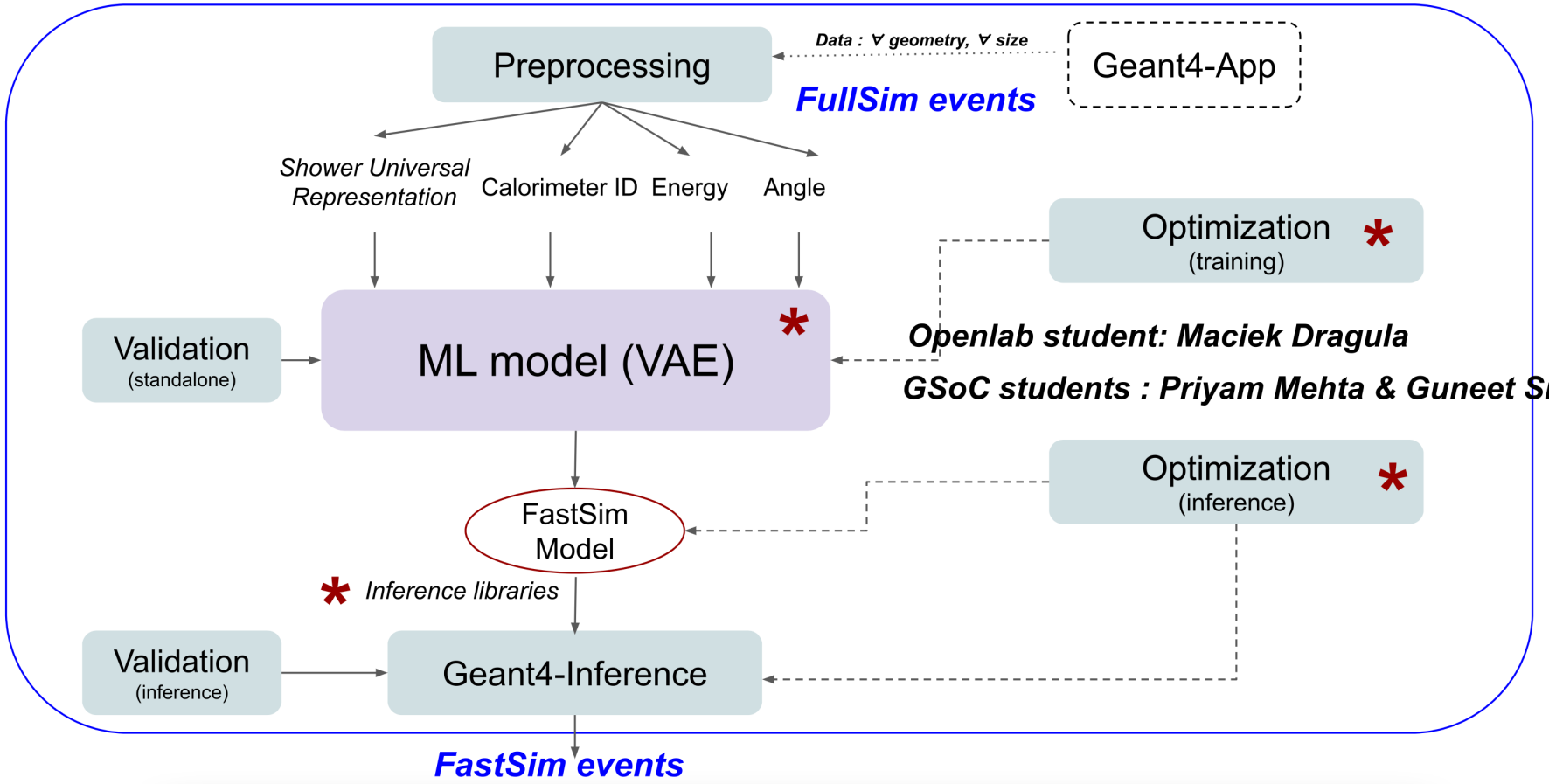
- Trade precision for performance using **fast simulation techniques** both with parameterisations and with ML methods, and integrate them seamlessly in Geant4
  - Use detailed simulation to 'train networks' or to 'fit parameters' that later can deliver approximative detector responses well integrated within Geant4
- Investigate the **use of accelerators** such as GPUs
  - With novel approaches for organising the computational work

Fast simulation

Dalila Salamani

+ **ML pipelines**
+ **Code base**
+ **Documentation**

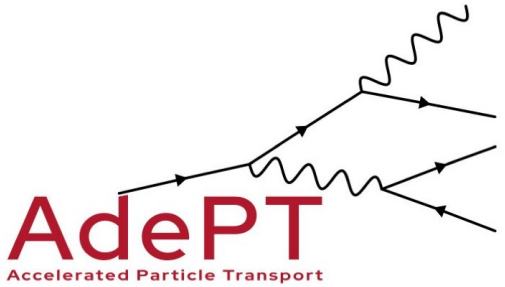
New fast simulation developments



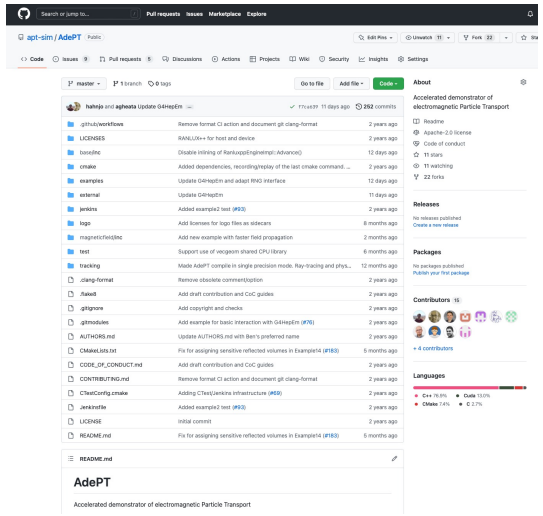
Use of compute accelerators

- General HEP transport prototypes
  - AdePT
  - Celeritas
- Specialized applications
  - optical photon transport - Opticks

AdePT project targets



- Understand usability of GPUs for general particle transport simulation
  - Prototype e+, e− and γ EM shower simulation on GPU, evolve to realistic use-cases
- Provide GPU-friendly simulation components
  - Physics, geometry, field, but also data model and workflow
- Ensure correctness and reproducibility
  - Validate the prototype against Geant4 equivalent, ensure reproducible results in all modes
- Integrate in a hybrid CPU-GPU Geant4 workflow
  - Understand possible limitation in such an environment
- Understand bottlenecks and blockers limiting performance
  - Estimate feasibility and effort for efficient GPU simulation
- Git repository
  - Initial commit in Sep 2020, $\mathcal{O}(10)$ contributors

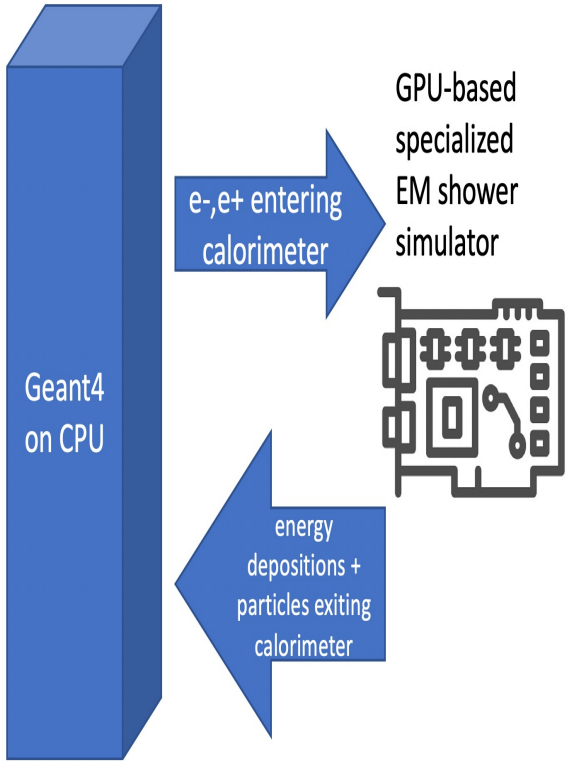


AdePT status

- Near-complete prototype for e-, e+ and gammas shower on GPU
  - Full set of interactions of e-, e+, gammas (implemented by G4HepEm)
  - Navigation in complex geometry models using VecGeom (from slabs to CMS)
  - Propagation of charged particles in a magnetic field using helix for constant B-field as first version
  - Simple hit generation code, which is then transferred from GPU to host
    - validation of simpler setup successful, ongoing for others
- Workflow
  - Implemented both standalone and G4-integrated workflows
- First performance assessment
  - Understood main performance bottlenecks (current geometry model)
- Initial results encouraging and motivating
  - Full desired functionality can be provided on GPUs
  - Path to GPU efficiency still long, passing through important restructuring of critical components.

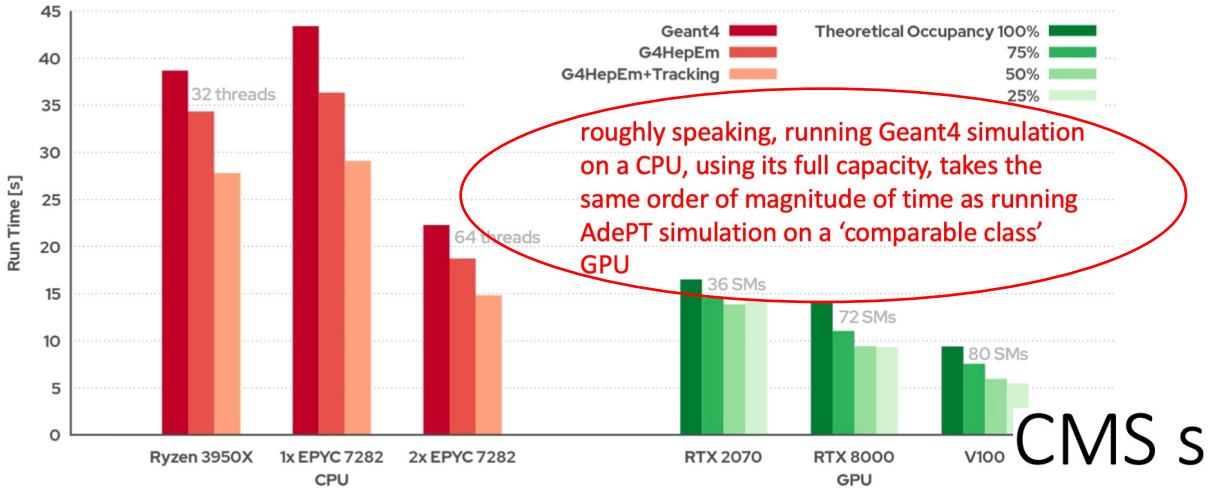
AdePT prototype integration strategy

- region-based approach for delegating simulation to an external transport
  - particle killed on the Geant4 side and passed to the other transport engine
  - energy depositions and ‘outgoing’ (from that region) particles returned
- this follows ‘fast-simulation’ approach in Geant4
  - allows the use of (most of the) existing fast-simulation hooks
  - easy integration with the physics list
  - ability to switch between full Geant4 and Geant4 + AdePT at runtime (from macro file)
- one difference:
  - we buffer particles to process them together when some threshold is reached (or when there are no more Geant4 particles on the stack)



CPU vs GPU Performance

(Sampling Calorimeter example)

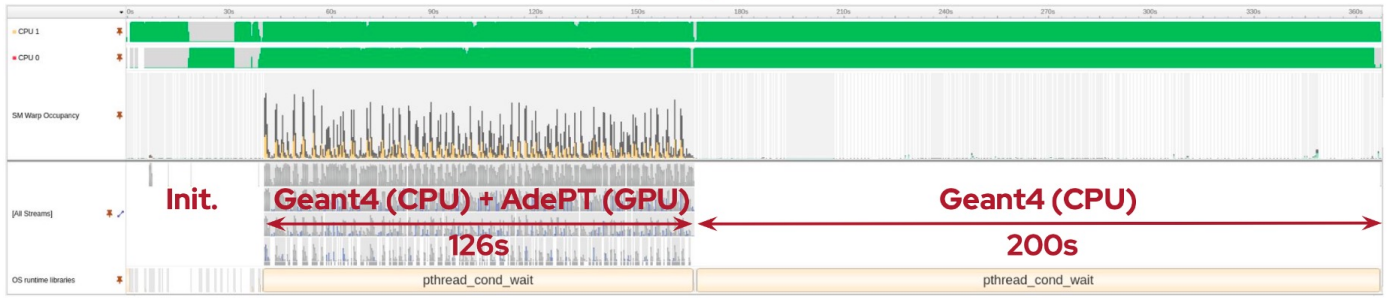


AMD Ryzen 3950X (16 cores, 32 threads, 3.5–4.7GHz), AMD EPYC 7282 (16 cores, 32 threads, 2.8–3.2GHz)

roughly speaking, running Geant4 simulation on a CPU, using its full capacity, takes the same order of magnitude of time as running AdePT simulation on a 'comparable class' GPU

AdePT
Accelerated Particle Transport

CMS simulations: integrated and standalone

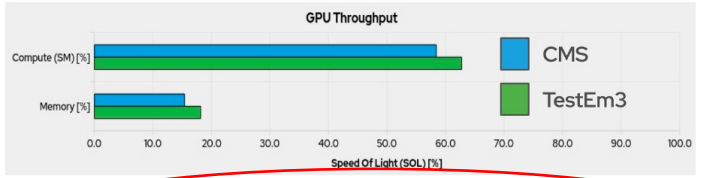


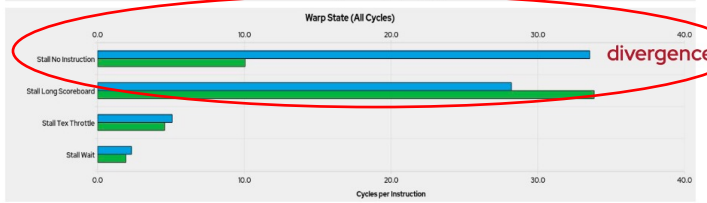
Comparison to Geant4

Above is the timeline of CMS simulation comparing AdePT integrated into Geant4 to Geant4 (Ryzen 3950X, RTX2070), with a speedup of 37% when using 2 CPU threads + 1 GPU vs only 2 CPU threads.

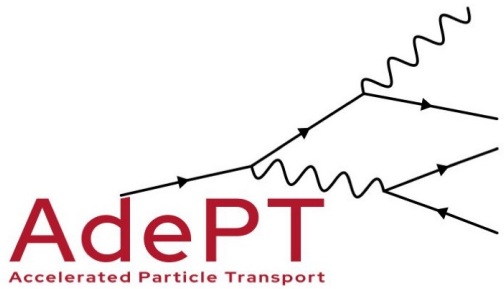
Impact of detector geometry

On the right, 10^6 electrons at 10GeV on Nvidia Tesla V100 with TestEm3 geometry vs the CMS geometry. The total simulation run time for the simplified calorimeter (TestEm3) setup is 549s vs 1455s for the CMS geometry





AdePT summary

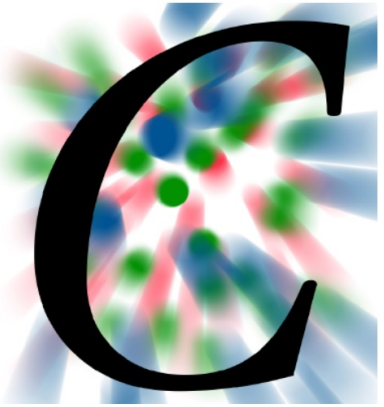


- challenging project, simulation clearly not an easy problem for GPUs
  - need to recast the diverse physics interactions and geometry elements for the regularity of the GPU
- AdePT GPU prototype provides full EM physics and geometry support to run simulations of CMS calorimeter complexity in standalone and Geant4 integrated modes
  - encouraging and motivating first result!
- current geometry model is a big bottleneck, we are addressing it with development of a new surface-based model
- further work on integration with experiments software frameworks will allow to better understand other potential stumbling blocks
- we invite collaborators to join this R&D !

Celeritas project overview

Seth Johnson

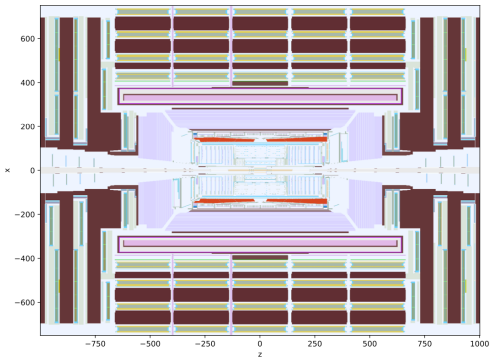
- Motivated by HL-LHC computational challenges *and* by recent success in GPU MC (*ExaSMR*)
- **GPU**-focused implementation of **HEP** detector simulation
  - Physics derived from Geant4 methods and implementation
  - Cross sections, materials, etc. loaded directly from Geant4
- Multi-institution collaboration with external contributors
- Funded through US DOE ASCR/HEP



Near-term goal: integrate with Geant4 and experimental frameworks to offload EM tracks to GPU

Physics and geometry

Particle	Process	Model
γ	photon conversion	Bethe–Heitler
	Compton scattering	Klein–Nishina
	photoelectric effect	Livermore
	Rayleigh scattering	Livermore
$e^{\pm}$	ionization	Møller–Bhabha
	bremsstrahlung	Seltzer–Berger
		relativistic
	Coulomb scattering	Urban MSC
e^{+}	annihilation	→(γ, γ)
μ	bremsstrahlung	μ brems

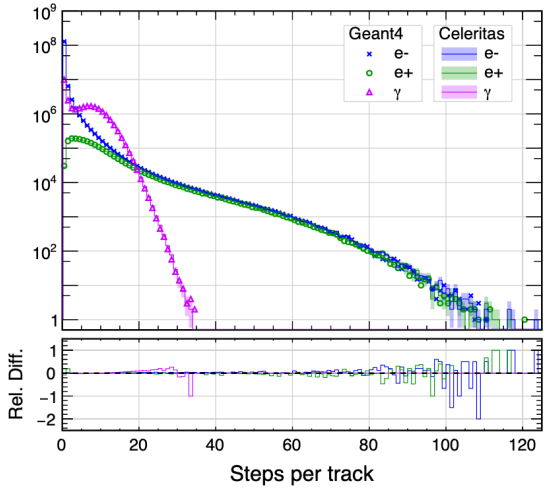


- Geant4 Standard EM physics *(verification in progress)*
- VecGeom *(GDML)* + ORANGE *(experimental, AMD/HIP compatible)*

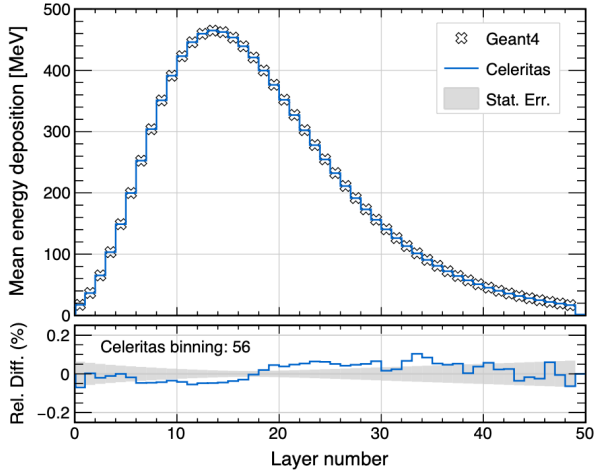
OAK RIDGE National Laboratory

Initial physics code comparison results

- No fields
- No multiple scattering



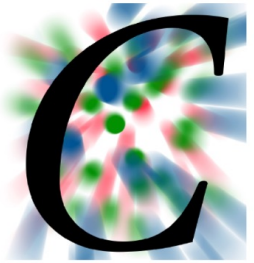
"Simple CMS" step counts
Isotropic 1 GeV γ



TestEM3 (Pb+LAr) energy deposition
Monodirectional 10 GeV e⁻

OAK RIDGE National Laboratory

Early performance testing



- TestEm3 — simplified calorimeter
  - 50 alternating layers of Pb and lAr
  - 10000 10 GeV electron primaries split between 7 events
- Equivalent configurations of Celeritas/Geant4
  - No magnetic field
  - Disabled multiple scattering, energy loss fluctuations, Rayleigh scattering
  - Excludes initialization time
- No spline interpolation in Celeritas (for now)
  - ~3% performance penalty for Geant4 with spline
  - Compensate by using 8× cross section grid points: <2% slower

Early performance testing (results)

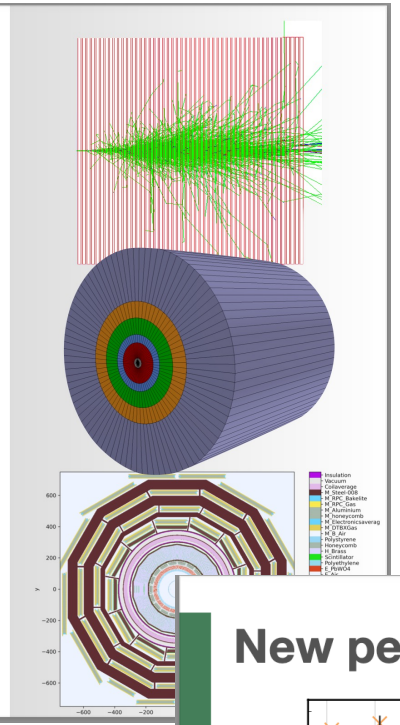
- 1–2 batches of 6 simultaneous runs on Summit (OLCF)
  - CPU (Power9): multithreaded with 7 cores
  - GPU (V100): one CPU plus one GPU
- 30–45× faster with GPUs
  - Apples-to-apples: Celeritas CPU vs GPU
  - Similar order-of-magnitude improvement irrespective of code
  - 210–315 CPU core to GPU equivalence

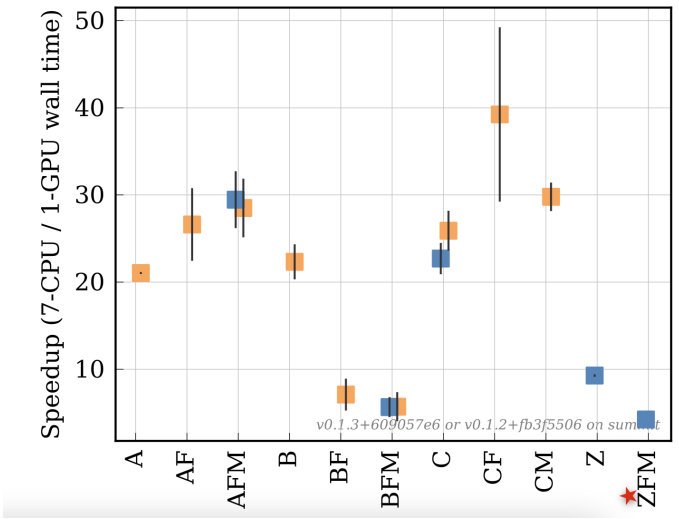
Work rate (events/s)

	geo	arch	mean	σ
Geant4 10.7.1	Geant4	CPU	0.24	0.010
Celeritas 8d83ebab (29 Apr 2022)	ORANGE	CPU	0.33	0.003
		GPU	15.09	0.375
	VecGeom	CPU	0.36	0.006
		GPU	11.17	0.075

New regression/timing suite

- 1300 10 GeV electrons per event, 7 events per run (1 per CPU, 7 per GPU)
- Very preliminary set of problem definitions *(working with AdePT team to develop)*
- Not currently optimized *(more on this later)*
- Run on single node of Summit (6 separate runs simultaneously, different seed for each)
- Initial results are apples-to-apples (Celeritas only)



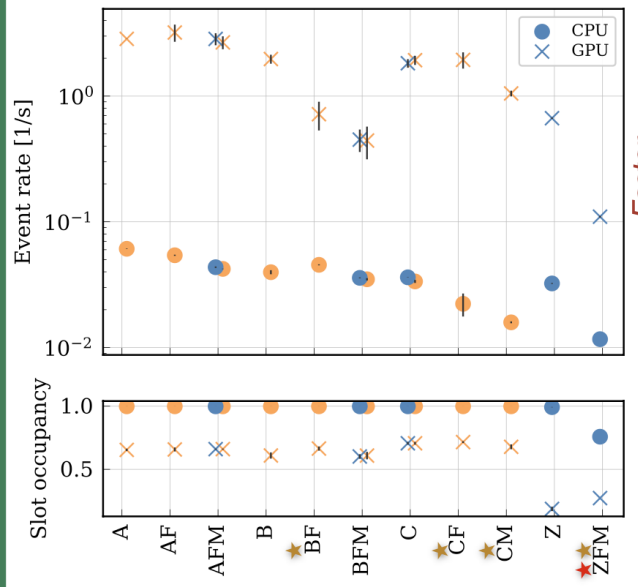


New performance results

ORANGE
VecGeom

	Problem definition
A	testem15
B	simple-cms
C	testem3
Z	cms2018

Modifier: F +field, M +msc



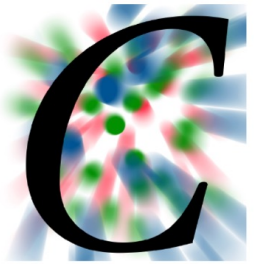
Faster

- Higher "slot occupancy" *(fraction of GPU track slots in use)* → better performance
- MSC slows GPU tracking by 2×
- Occasional tracking failures in field
- ORANGE and VecGeom show approximate performance parity

*unconverged
*some instances fail

Key areas of continuing work

- Physics ***validation*** (physics models, progression problems, experiment-specific)
- Experiment ***integration*** (Acceleritas + direct)
- Performance ***experimentation*** (there's a long list)
- International ***collaboration*** (AdePT, VecGeom, CMS, ORANGE)



Conclusions

- Celeritas is a new *specialized* detector simulation code
- Current test problems show **~10–30× performance boost** using GPUs on Summit (70–210× GPU/CPU core equivalence)
- Laundry list of fixes, features, validation, optimization to do
- Version 0.1.3 now available (install from source or Spack)

https://github.com/celeritas-project/celeritas

Fermilab | U.S. DEPARTMENT OF ENERGY | Office of Science

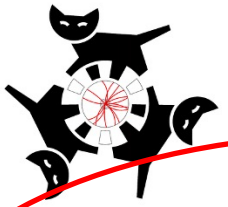
Integration of Opticks[1] and Geant4 (an advanced example: CaTS)

Hans Wenzel
Soon Yung Jun
Krzysztof Genser
Alexei Strelchenko

Fermilab

GEANT4 A SIMULATION TOOLKIT

[1]developed by Simon Blyth
(Institute of High Energy Physics, Chinese Academy of Sciences)



Opticks/G4Opticks/CaTS

Opticks is an open-source project that accelerates optical photon simulation by integrating NVIDIA GPU ray tracing, accessed via NVIDIA OptiX.

Developed by Simon Blyth:

https://bitbucket.org/simoncblyth/opticks/

CaTS: interfaces Geant4 user code with Opticks using the G4Opticks 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. CaTS was included in Geant4 11.0 as an advanced example:

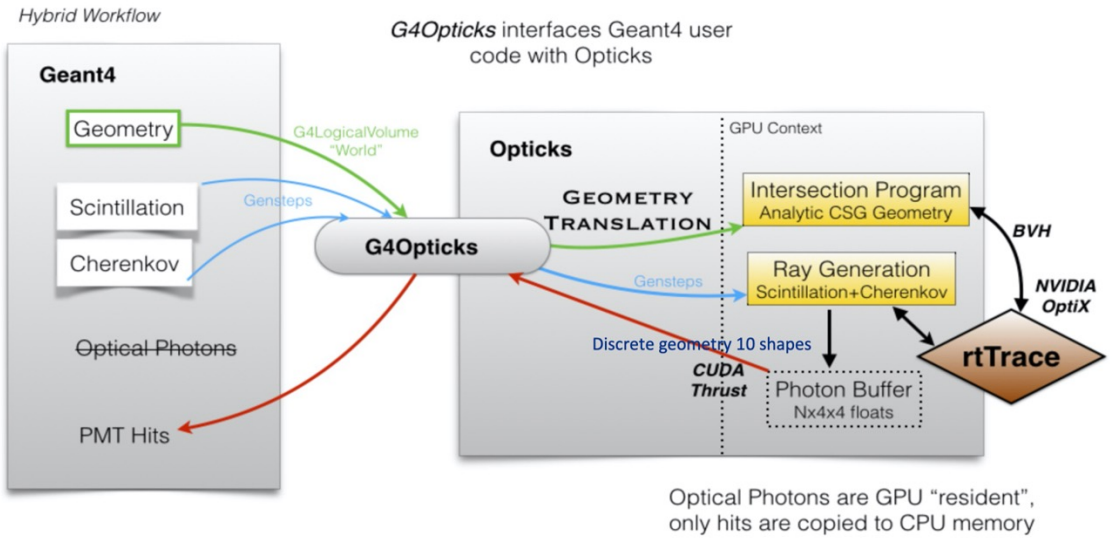
https://geant4.kek.jp/lxr/source/examples/advanced/CaTS/

Integrate NVIDIA OptiX with Geant4

Geometry Translation + "Gensteps"
=> entirely offload photon simulation to GPU

- upload translated geometry at initialization
- only hits need to consume CPU memory

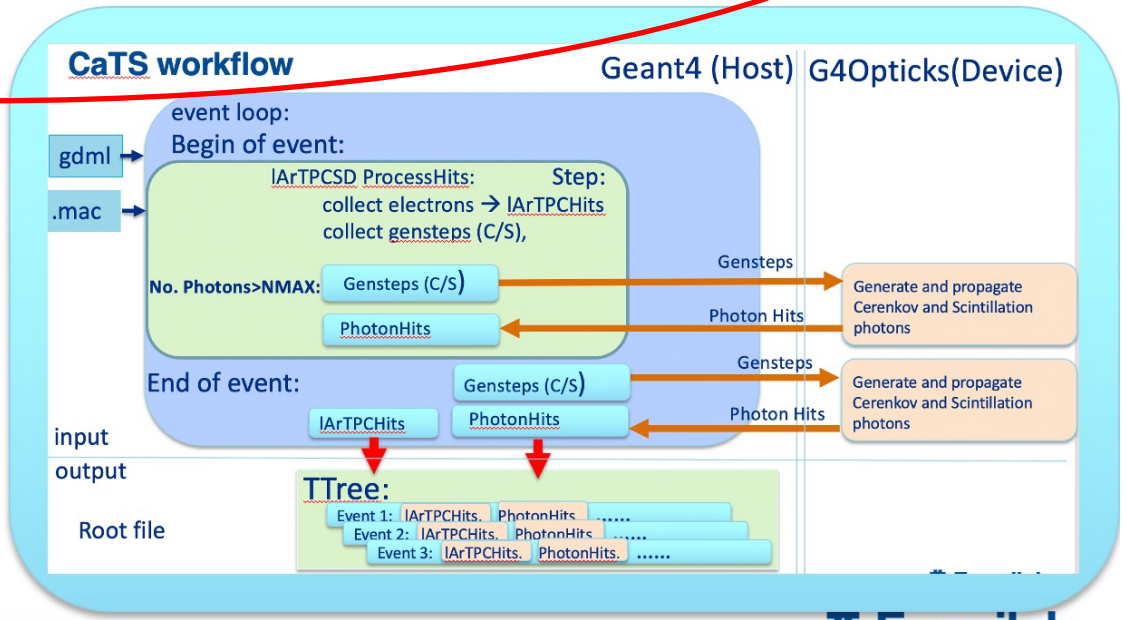
Figure from Simon's presentation



CaTS: advanced Geant4 example

- Uses Geant4 to collect Scintillation and Cerenkov Gensteps. A Genstep collects all the data necessary to generate Cerenkov/Scintillation photons on the GPU. The harvesting is done in Sensitive Detectors(SD) (RadiatorSD/lArTPCSD). The number of photons to be generated is calculated by Geant4 and constrained to be identical whether one uses the Geant4 optical physics or G4Opticks.
- Use of G4Opticks is both a build and runt-time option.
- The PhotonHits collected by the PhotonSD sensitive detector have the same content whether Geant4 or G4Opticks is used.
- Uses GDML with extensions for flexible Detector construction and to provide optical properties at runtime. The gdml extensions include:
  - Assigning Sensitive Detectors to logical Volumes.
    Available:
    - **RadiatorSD**, **lArTPCSD**, **PhotonSD.**
    - TrackerSD, CalorimeterSD, DRCalorimeterSD, **...**.
  - Assigning step-limits to logical Volumes.
  - Assigning production Cuts by regions.
  - Assigning visualization attributes.
  - Note there are Opticks specific keywords!
- Uses G4PhysListFactoryAlt to define and configure physics.
- Uses Root IO to provide persistency for Hits.

Achieved speed up in the order of a few times 10^2, depends strongly on detector geometry, hardware and settings.



Latest developments:

- For the Geant4 advanced example:
  - Lots of code and cmake cleanup, make use of C++17 features, follow Geant4 code conventions added visual code configuration.
- More examples: (e.g. scintillation crystals, WLS examples, …).
- Help users getting started with CaTS.
- Ensure that gdml examples are compatible with Geant4 10/11 and Opticks.
- Various optimizations.
- Changes to RootIO.
- Allow for Geant4 event-level multi-threading for offloading Opticks– autolock for G4Opticks call (generate and propagate Photons). But poor scaling due to mutex (as expected as kernels calls are sequential).

Status of re-implementing Opticks for Optix 7[1].

Huge change unavoidable from new OptiX API → So profit from rethink of simulation code → **2nd implementation advantage**. Goals of re-implementation : flexible, modular GPU simulation, easily testable, less code:

- COMPLETED: Full Simulation re-implementation for OptiX 7 API, but new workflow not ready for testing yet.
- Many packages were removed or are planned to be removed.
- Move code that doesn't require Optix or Cuda out of GPU context (SYSRap, not QUDARap).
- Rather monolithic .cu was replaced by many small GPU+CPU headers many GPU+CPU headers.

1) Extracted from status report by Simon Blyth for more details see:
https://simoncblyth.bitbucket.io/env/presentation/opticks_20220718_towards_production_use_juno_collab_meeting.html

Plans

G4Opticks/Opticks:

- Add opticks to the framework used by the liquid Argon TPC community. That requires Cuda, Optix, Opticks … being packaged in the way required by these frameworks (ups/upd, spack).
- Try out the new Opticks as soon as Simon gives the go-ahead.
- Use the same implementation of the scintillation process on CPU and GPU, use the same optical properties/keywords.
- Implement Wavelength shifting process (WLS).
- Have a look at simplifying cmake, e.g., get rid of obsolete modules.

CaTS:

- Achieve true concurrency by using G4Tasking. Allow to fully utilize GPU resources like vram, cpu cores, multiple GPU's.
- Change to use Root TBufferMerger for RootIO instead of using separates file and merging them when in multithreaded mode.
- Provide benchmarking and physics validation results with realistic (including WLS) liquid Argon TPC geometry.

Geant4:

- Move the harvesting of Gensteps to the Geant4 optical producer processes (G4Cerenkov, G4Scintillation)→ general interface to external ray tracing programs like Opticks. Trigger the G4Opticks from UserSteppingAction whenever sufficient photons/gensteps are collected for efficient processing on GPU.

Summary

- nice progress along all R&D directions
  - improvements to G4 code, fast simulation, usage of GPUs
- ML4FastSim is now a complete tool to manage full ML cycle
  - meta-learning provides huge gain in adaptation for new geometries
- AdePT and Celeritas delivered first prototypes to run EM shower on GPUs
  - encouraging first results, with clear bottlenecks (geometry) identified
- CaTS/Opticks provide a production quality tool for fast optical photon simulation
  - growing number of users