Update on R&D activities

Witek Pokorski

15.02.2024

Geant4 Technical Forum

Content

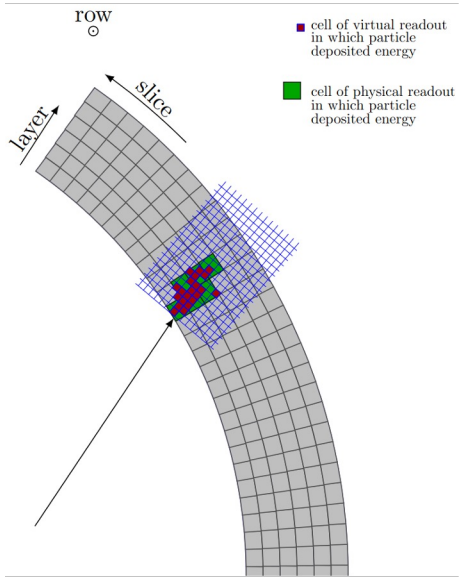
- fast simulation techniques
  - Machine-Learning based models
- exploration of new hardware (GPUs)
  - general transport code prototypes
  - domain specific application

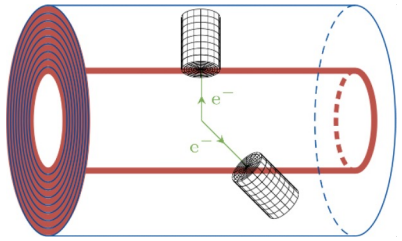
Anna Zaborowska
Piyush Raikwar

Fast Simulation

Updates in the Geant4 extended example with ML fast sim model (Par04)

- Example demonstrating use of ML models (inference within C++ framework)
- Introduced support of GPU inference with ONNX runtime
- Used to produce EM shower data published on zenodo





Exploring advantages of meta-learning: MetaHEP

Heavy training done once, quicker adaptation for new experiments

- Move burden from designing new dedicated models to implementing additional technical infrastructure for dedicated simulation with virtual readout
- Using VAE from Par04, but can be based on any architecture
- Publication in the Journal of Physics Letters B
- Technical "how-to" documentation on g4fastsim.web.cern.ch

Variational Autoencoder (VAE)

- **Fast sim model used by LHCb** (with modifications by Michal Mazurek). Our workflow can be now used in Gaussino, allowing further tests with other models for LHCb calorimetry
- Done **first tests for ATLAS** photon open dataset, but in a naive virtual meshing, needs a proper implementation to test either VAE or any other model

Anna Zaborowska
Piyush Raikwar

Fast Simulation: New models

Co-organisation of the CaloChallenge

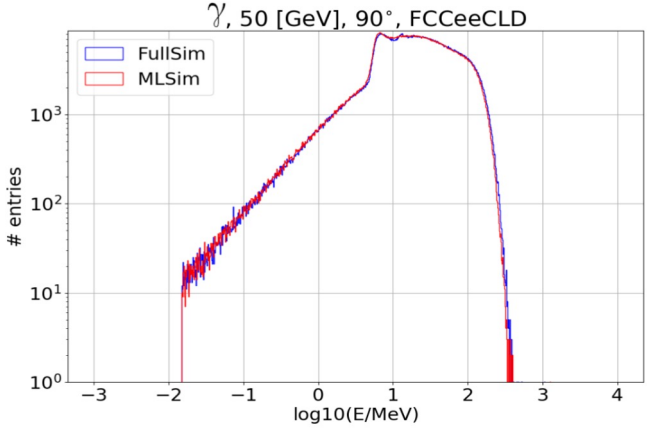
First fast calorimeter simulation ML challenge

- Aim: spur the development and benchmarking of ML fast shower models
- Around **40 submissions**, but we're aware of the datasets being used by many others
- Will directly expose experiments implementing MetaHEP(Par04)-workflow to **new models**

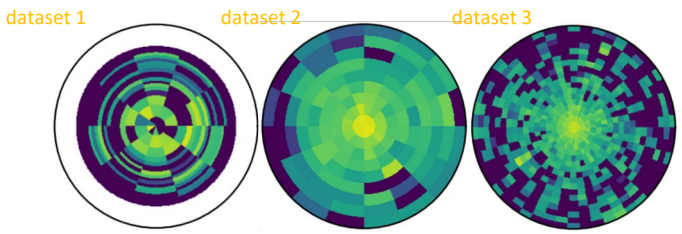
Exploration of transformer-based models

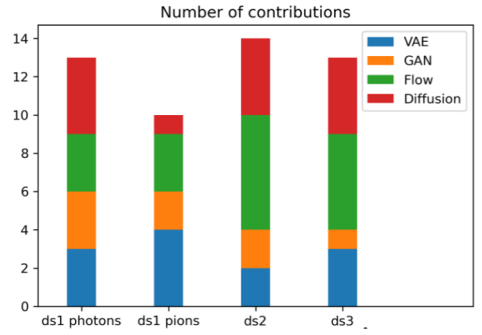
Use of transformers to build a model that better describes correlations between cells and generates better cell energy distribution

- First tests with VAE-type models presented at CHEP
- More promising results obtained with a diffusion architecture
- Work in Collaboration with IT and IBM Research
- Use of energy scored in the virtual mesh like MetaHEP/Par04, directly usable for any detector (currently implemented in ddsim as well as in Gaussino by LHCb).
- Tested for FCC-ee.



cell energy distribution

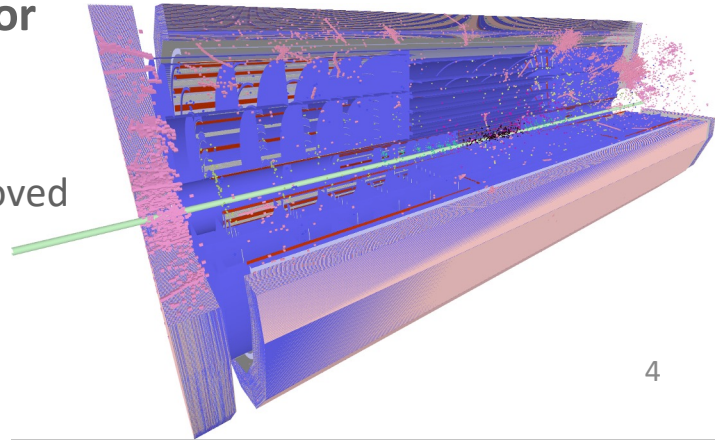




Work on the Open Data Detector

Benchmark detector for algorithmic studies and a source of the new benchmark for ML studies with improved complexity (**CaloChallenge2** ?)

- added hadronic calorimeter

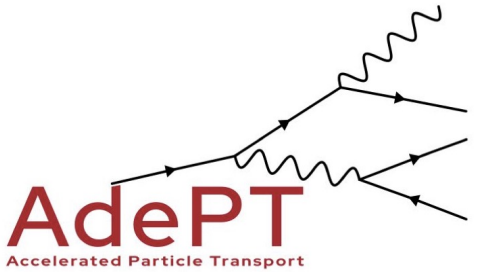


GPU-based transport codes

- Geant4 assessment of the R&D prototypes organized 13-14 December 2023
- AdePT and Celertitas projects presented in details and discussed
  - current status
  - main challenges
  - possible synergy (common interface)

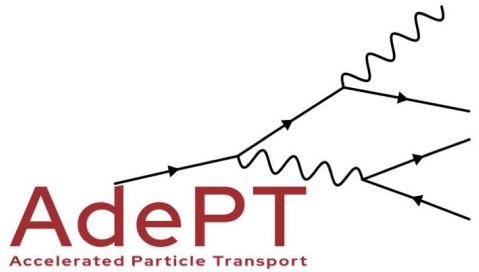
- Report from the assessment panel available the coming days

AdePT 2023 status



- Achieved the initial goals of the R&D
  - Understand usability of GPUs for general particle transport simulation, seeking for potential speed up and/or usage of available GPU resource for the HEP simulation
    - Prototype e+, e− and γ EM shower simulation on GPU, evolve to realistic use-cases
      - Geometry: VecGeom library, Physics: G4HepEm library
- Transport for EM particles working on GPUs for LHC-complexity geometries
  - Excellent physics agreement within statistical fluctuation
  - Reproducibility of the simulation achieved
- Full integration with Geant4 applications
  - Possible to plug AdePT into existing Geant4 applications with minimal extra code
    - reusing existing sensitive detector implementations
- Main bottleneck: geometry – being addressed now (see following slides)

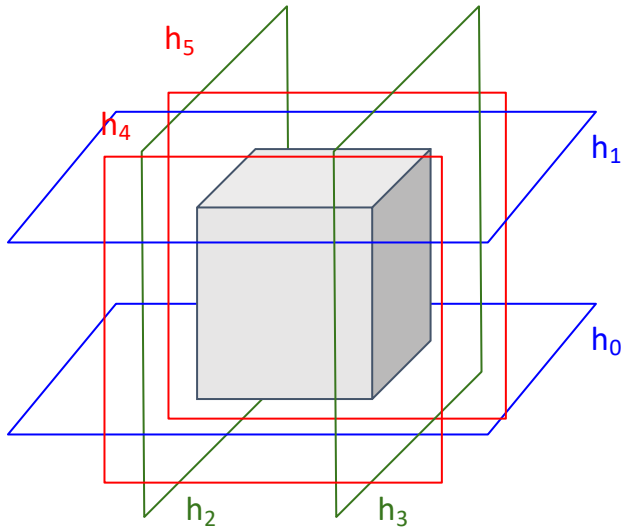
New AdePT Developments

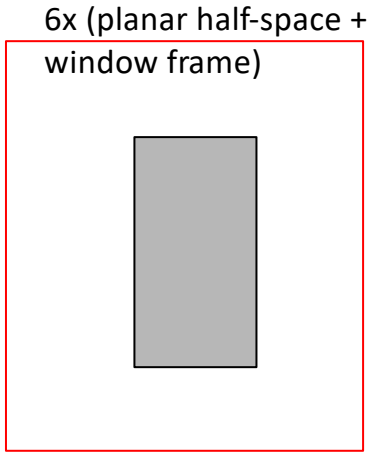


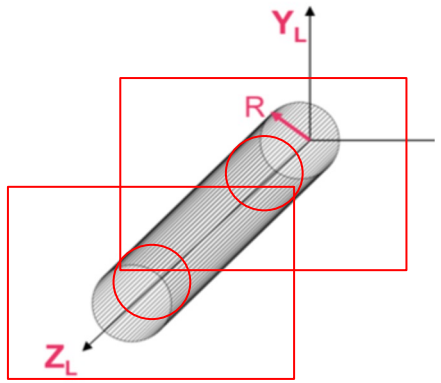
- Re-organisation as a library
  - Major refactoring of the project which will facilitate integration into external projects, maintenance and future development
  - Previously the project consisted in a series of mostly independent examples for the sake of exploration
- Geant4 integration through a custom Tracking Manager
  - Superseding the previously the integration approach using the Geant4 fast simulation hooks
    - custom tracking manager offers more flexibility (including all the previous functionality)
  - From the point of view of a user, AdePT can now be integrated just by registering a Physics Constructor
- Ongoing research into asynchronous scheduling strategies
  - The current strategy for scheduling work on the GPU is not optimal:
    - In certain situations this may even hinder CPU performance
  - Initial results using a new scheduling strategy are promising

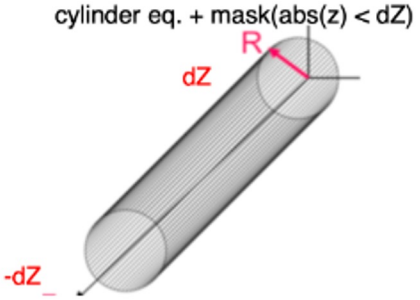
Bounded surface modeling – mitigating the geometry bottleneck

- 3D bodies represented as Boolean operation of half-spaces
  - First and second order, infinite
  - Just intersections for convex primitives
    - e.g. box = h_0 & h_1 & h_2 & h_3 & h_4 & h_5
- Storing in addition the solid imprint (frame) on each surface: FramedSurface
  - The frame information allows avoiding to evaluate the Boolean expression for distance calculations to primitive solids







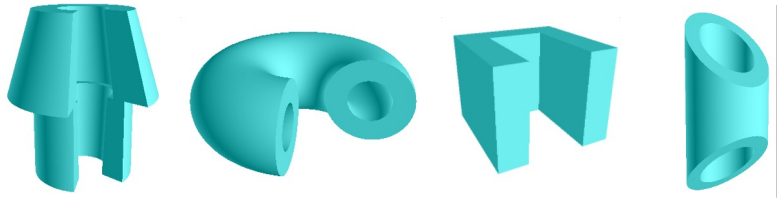


Objectives for the surface model

- Portable GPU-friendly header library
  - Algorithmic part independent on the backend, compilable with any native/portability compiler
  - Headers templated on the precision type to allow for a single-precision mode
  - Reduced set of simple surface/frame algorithms (vs. ~20 primitive solids now)
    - Reducing divergence and register usage on the GPU
- Automatic conversion from VecGeom transient solid model
  - Transparent creation of the data structures and copy to GPU
  - Preserve awareness of the Geant4 containment feature as powerful optimization
- Target: code simplification compared to the solid model
  - No virtual calls, no recursions, more work-balanced
  - Better device occupancy and kernel coherence

Recent developments in the surface model

Added additional shapes:
polycone, simple extruded, cut tube, trapezoid, torus
(+ enabling reflected solids)



: all required solids implemented

: missing generic trapezoid
(to be added soon)

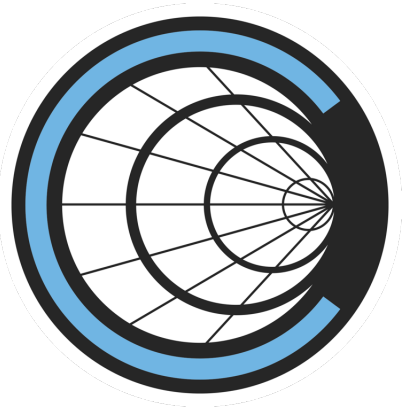
Current status: debugging CMS (handling of overlaps and other surprises)

Next:

- Adding BVH accelerating structure to surface model
- Performance measurements, testing calorimeters (ATLAS EMEC, CMS HGCAL, LHCb ECAL), adding possibly missing solids of those

Celeritas project goal

- **Accelerate scientific discovery** by improving LHC detector simulation throughput and energy efficiency
  - Long term goal: as much work as possible on GPU
  - Initial funding: focus on EM physics (but keep door open for more!)
- Jointly funded by US DOE ASCR and HEP
  - **Research and develop** novel algorithms for GPU-based Monte Carlo simulation in High Energy Physics
  - **Implement** production-quality code for GPU simulation
  - **Integrate** collaboratively into experiment frameworks



Celeritas core team:

Elliott Biondo *(ORNL)*,
Julien Esseiva *(LBNL)*,
Seth R Johnson *(ORNL)*,
Soon Yung Jun *(FNAL)*,
Guilherme Lima *(FNAL)*,
Amanda Lund *(ANL)*,
Ben Morgan *(U Warwick)*,
Stefano Tognini *(ORNL)*

Celeritas core advisors:

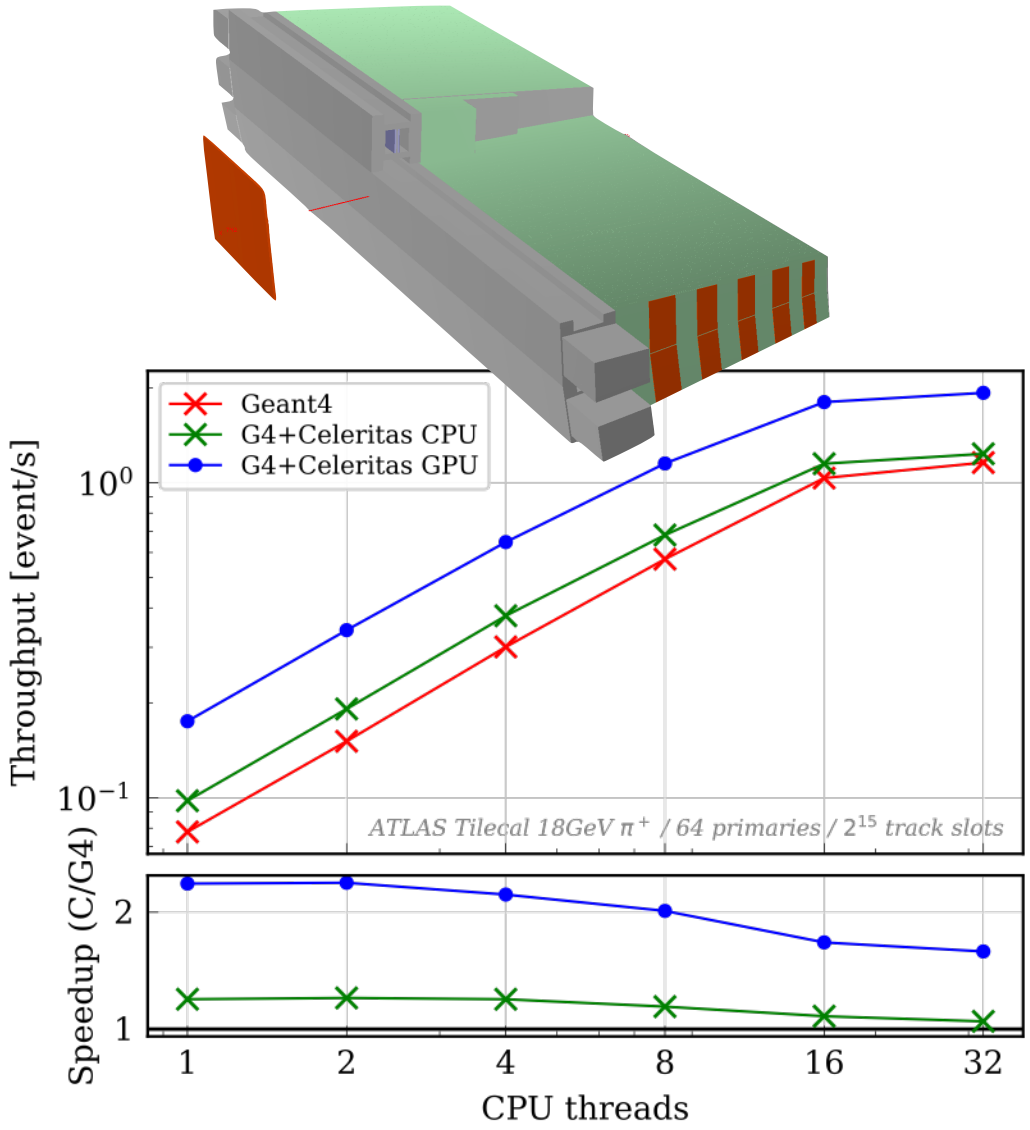
Tom Evans *(ORNL)*,
Philippe Canal *(FNAL)*,
Marcel Demarteau *(ORNL)*,
Paul Romano *(ANL)*





Celeritas 2023 accomplishments

- Characterized GPU EM performance
- Developed library for automated, user-friendly integration into Geant4 apps
  - Loads user geometry, physics (cuts), settings
  - Sends hits from GPU through layer to user SDs
  - Demonstrated with ATLAS FullSimLight, CMSSW, standalone test apps
  - *Full support* for offloading EM tracks to GPU or CPU
- Added features for robust G4 integration
  - Optimized performance for multithreading
  - Fully reproducible results on GPU
  - G4VFastSimulationModel and G4VTrackingManager



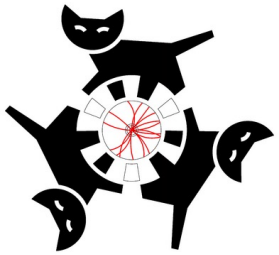
18 GeV π+ test beam; Geant4 hadronics
See Geant4 Meeting results for further details

Test problem: Lachnit, Pezzotti; FSL integration: Morgan

Celeritas status by the numbers

100	**lines of code**	to integrate Celeritas into a FullSimLight tile calorimeter test application, with no modifications to Geant4
1.8×	**full-simulation speedup**	including hadronics and SD hits, by using 1 Nvidia A100 with 16 AMD EPYC cores for the ATLAS test beam application *[NERSC Perlmutter]*
2–20×	**throughput**	when using Celeritas on GPU (compared to Geant4 MT CPU) for EM test problems *[NERSC Perlmutter]*
4×	**performance per watt**	for TestEM3 (ORANGE geometry) using Celeritas GPU instead of Geant4 CPU *[NERSC Perlmutter]*

Celeritas v0.4: https://celeritas-project.github.io/celeritas/



Opticks/G4CXOpticks/CaTS: Full Integration of Opticks with Geant4

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 (Calorimeter and Tracker Simulation) is a flexible and extend-able framework. With respect to Opticks it interfaces Geant4 user code with Opticks and defines a hybrid workflow where generation and tracing of optical photons is optionally offloaded to Opticks (GPU) using the G4CXOpticks interface, while Geant4(CPU) handles all other particles.

CaTS based on legacy version of Opticks (based on Optix 6) was included in Geant4 11.0 as an advanced example.

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

Opticks has been completely reengineered by Simon Blyth migrating to OptiX7 which is significantly different from OptiX6 and made a complete rewrite of Opticks necessary. The other benefits of the refactoring process for Opticks are:

- It is more modular and easier to implement extensions.
- It uses less libraries.
- There is a clear distinction between CPU and GPU codes.

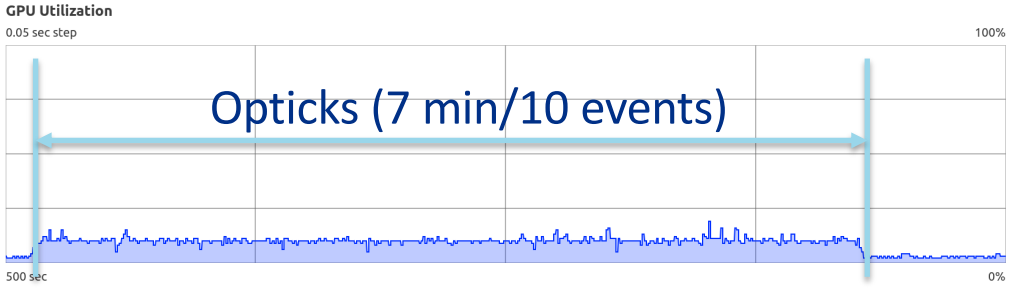
Status

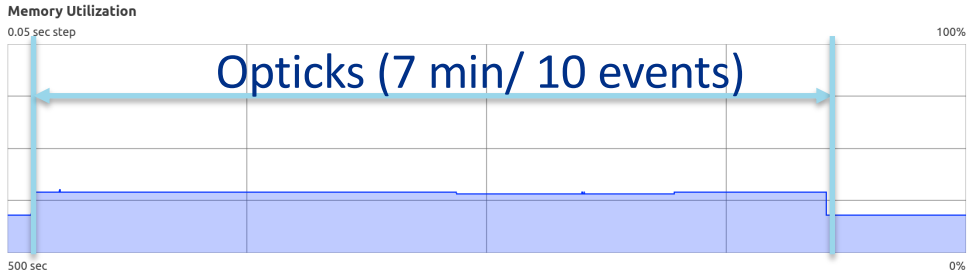
The new Opticks (v.0.2.7, with NVIDIA OptiX7) has been fully functional (stable) since January 2024.
New Opticks APIs are tested and successfully integrated with a modified workflow of CaTS (v2.0.3).
. https://github.com/hanswenzel/CaTS (v2.0.3). Physics validation and benchmarking of the new Opticks are ongoing!

With the legacy version of Opticks (based on NVIDIA Optix 6) we achieved speed up in the order of a few times 10^2, this depends strongly on detector geometry, hardware and settings.
So far with the new version of Opticks (same computing hardware and detector geometry, 2 GeV e-shower) we found:

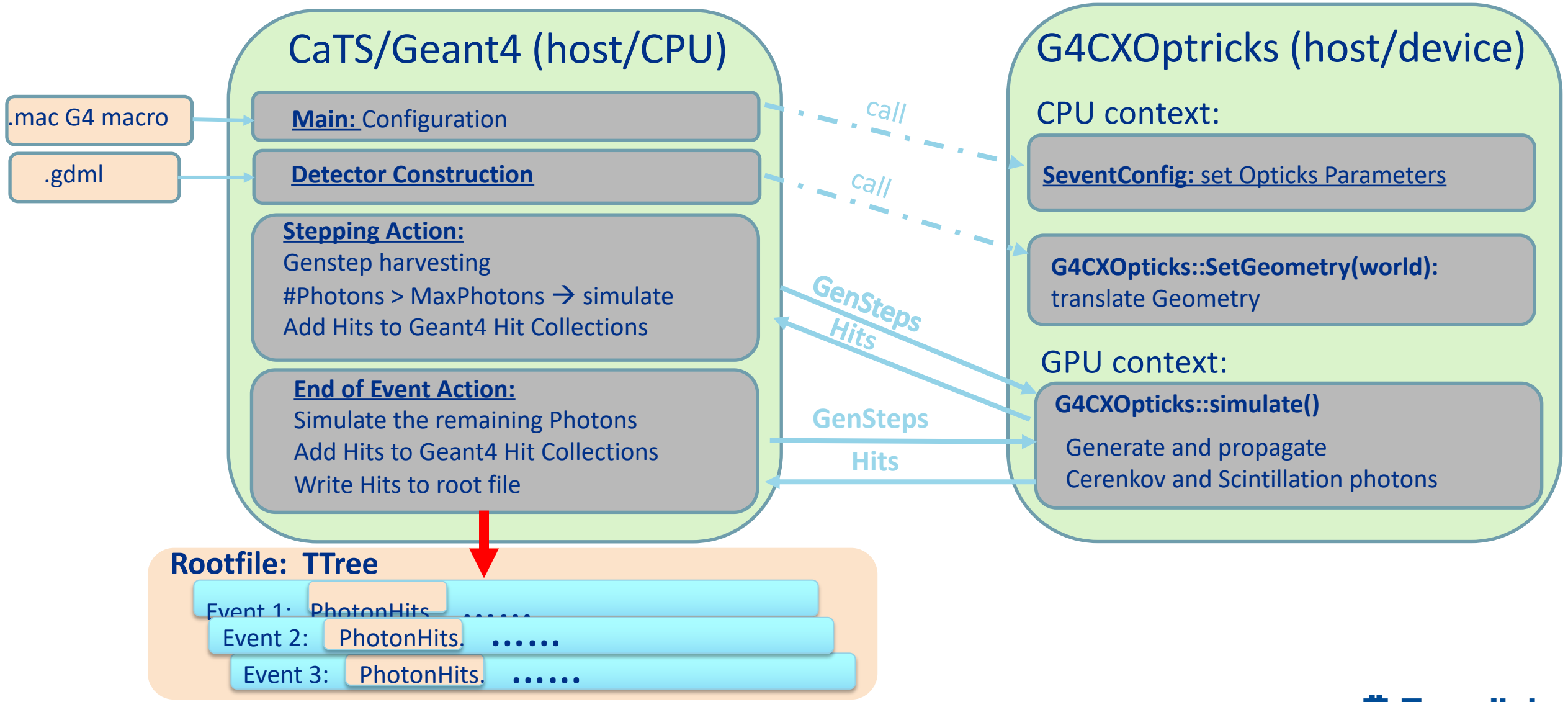
- Compared to the previous version, the speed up by GPU over CPU is only a factor 10 times compared to several 10^2 times previously. We found the GPU (computing and memory) resources are underutilized.
- the optimization of the GPU kernel launch promises to improve the performance significantly.

The work integrating Opticks with the liquid argon TPC software framework (LArSoft) is ongoing.





CaTS workflow using the new version of Opticks based on OptiX®7:



Conclusion

- generic ML fast simulation models start to be applied in the experiments
- interesting results from new ML models

- significant progress in GPU protypes
  - ready do start testing full integration with experimental frameworks
  - new surface model ready very soon
- very productive and interesting assessment of GPU-based prototypes
  - the coming year will provide us with new performance numbers after removing the geometry bottleneck

- CaTS moving to new version of Opticks based on NVidia OptiX7