R&D activities

Overview and plan of work

Witek Pokorski

24.03.2022

Geant4 Technical Forum

R&D overview

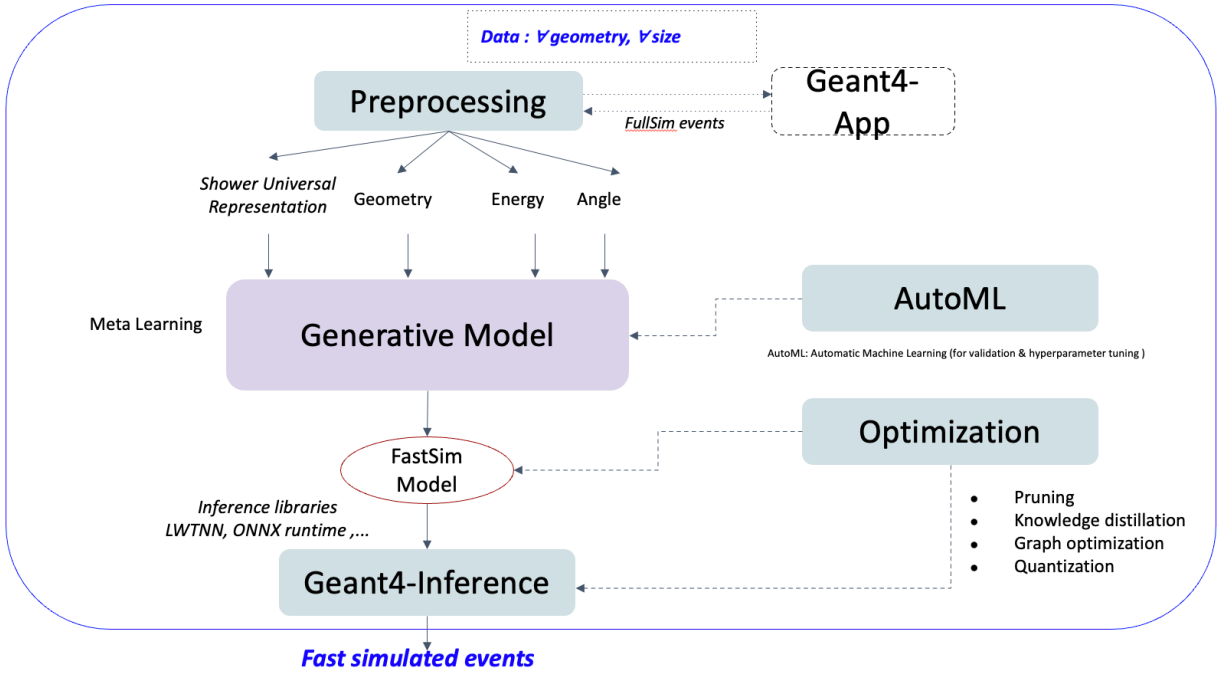
In order to increase the precision and the speed (the amount of produced data) of the simulation, to meet the requirements of the future physics analyses, we advance along the 3 axes:

- improvements to current Geant4 code base
  - incremental improvements, part of each working group
  - some more profound re-designs (like G4HepEm) discussed earlier
- fast simulation techniques
  - ‘classical’ and Machine-Learning based
- exploration of new hardware (GPUs)
  - general transport code prototypes
  - domain specific (like Opticks)

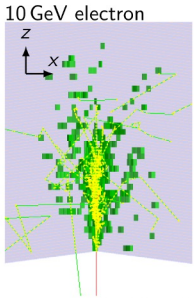
Fast Simulation task : Overview 2021

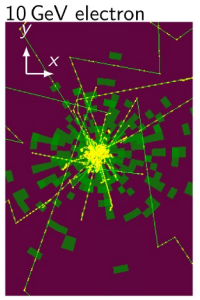
Dalila, Anna

- ML-based fast simulation: from full simulated to ML fast simulated events
- Demonstrated generalization on sample geometries, for different energies and incident angles
- Generalizable and reusable solution: meta-learning based approach provided the possibility to learn from different detector geometries how to fast tune initialization parameters for a new geometry
- LWTNN and ONNX for inference within Geant4
- Optimization techniques to reduce memory footprint of inference
- Automatic Machine Learning (AutoML) for hyperparameter optimization (CERN summer project by Poliana Ferreira)
- New Geant4 example



Par04 example





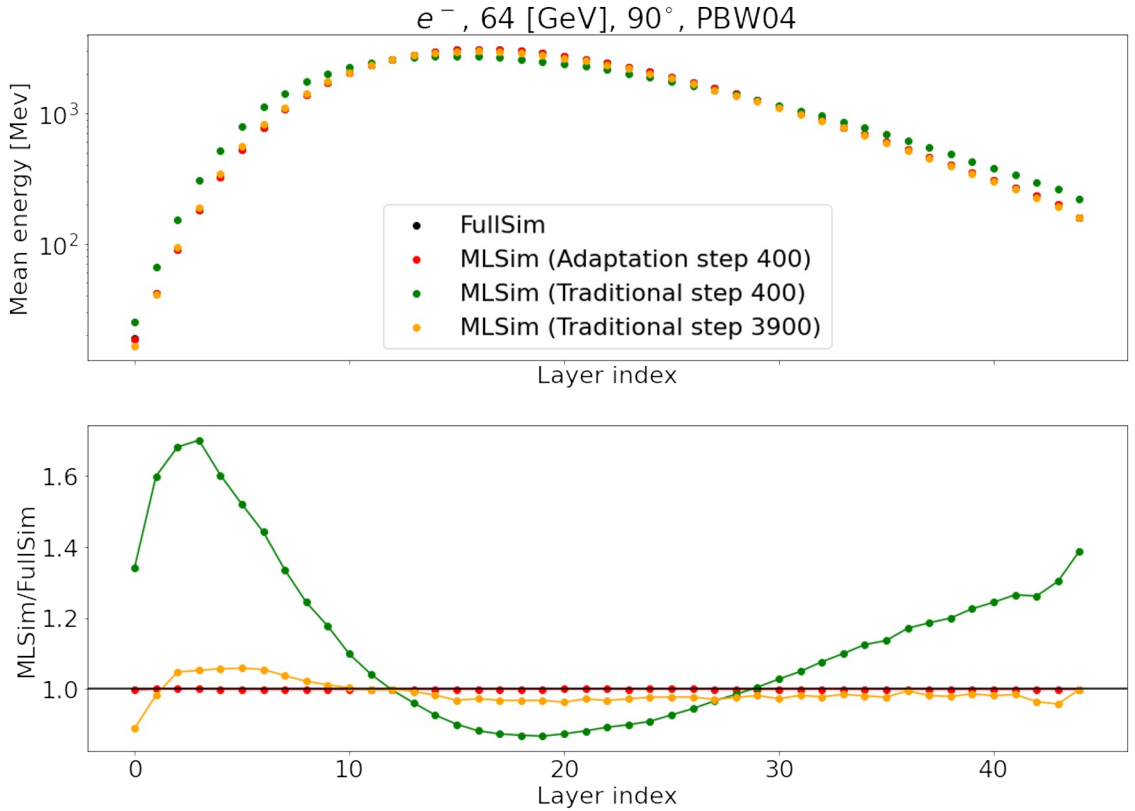
- Fast simulation with ML within Geant4
- New Par04 extended example in Geant4 11.0

  examples/extended/parameterisation/Par04

- Demonstrates how to incorporate inference libraries
  - ONNX Runtime
  - LWTNN
- The ML trained on 2 provided geometries, conditioned on the **energy** and **angle** of the particle
- Example can run full and fast simulation

Name
..
Par04ActionInitialisation.cc
Par04DefineMeshModel.cc
Par04DetectorConstruction.cc
Par04DetectorMessenger.cc
Par04EventAction.cc
Par04EventInformation.cc
Par04Hit.cc
Par04InferenceMessenger.cc
Par04InferenceSetup.cc
Par04LwtnnInference.cc
Par04MLFastSimModel.cc
Par04OnnxInference.cc
Par04PrimaryGeneratorAction.cc
Par04RunAction.cc
Par04SensitiveDetector.cc

Adaptation vs traditional training



Meta learning - Adaptation

- Meta training using geometries & adaptation on a new geometry
- 400 steps of adaptation : 20.48 s

Traditional training

- Training on a single geometry with checkpoint saved every 100 epochs
- 400 steps of training : 1200 s (around 3h for 3900 steps)

Fast Simulation task : Plan of work 2022

- ML-based fast simulation
  - Ongoing work on validation for use on realistic geometries (ATLAS, FCC), with geometry-independent scoring
  - Integration of key elements of Par04 with key4HEP framework
  - Systematic studies on training
  - Testing and development of optimized data pipelines (eg . Kubeflow)
  - Studies of memory footprint optimization strategies (inference)
  - Extension of Par04 (adding new inference libraries such as LibTorch and TFLite, inference on GPUs)
- Classical parametrisation
  - Follow LHCb's feedback on their usage of fast sim hooks in Geant4 (in Gaussino)
  - Revisit GFlash implementation and continue to work on automated tuning
- Collaborations & publications
  - Public release of the data produced with Par04 (CERN open data or Zenodo)
  - Involved in CaloChallenge
    - In collaboration with *David Shih, Michele Faucci Giannelli, Gregor Kasieczka* and *Ben Nachman.*
    - A community challenge, based on a common dataset, for developing and benchmarking different approaches to fast calorimeter simulation
    - Proposal in ML4Jets 2021, Results will be published in the next ML4Jets (January 2023)
  - Community effort (paper) to prepare a fast calo sim benchmark
  - Open Data Detector - contribute to calorimeter & propose it as a base of the benchmark

AdePT – Accelerated demonstrator of electromagnetic Particle Transport

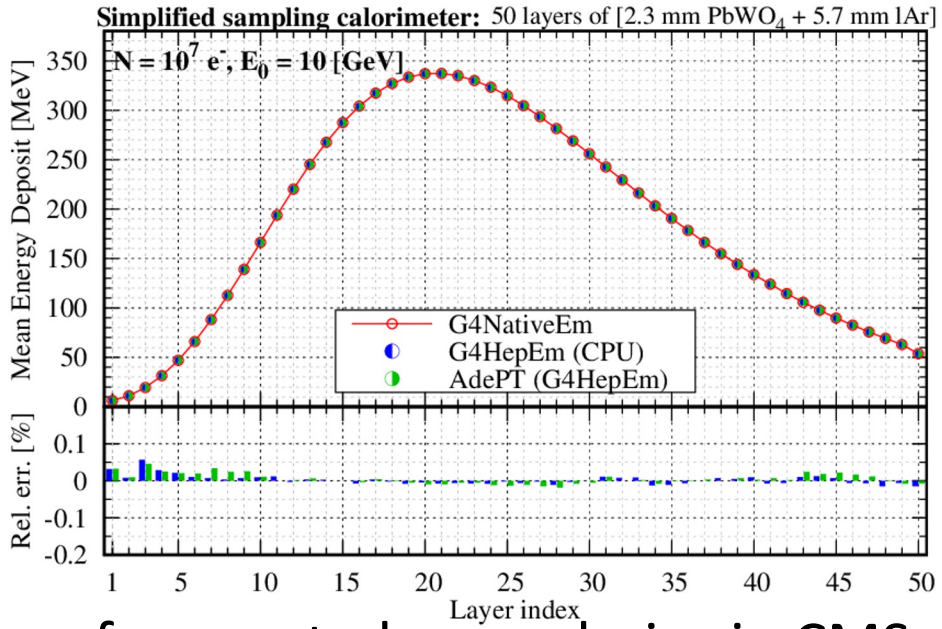
- Project started over a year ago, trying to address the inability to use GPU cards for detailed simulation of collider experiments
  - seeking to answer the questions
    - Can we transform HEP particle transport to be efficient-enough on GPU?
    - How much effort will it take to create a production-level tool ?
    - What level of changes would be required to port key elements of the user code of production experiment simulation?
  - some initial ideas shown last year at the Geant4 collaboration meeting
- GitHub repository
  - 13 contributors so far

Major achievements

- 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.

Physics in AdePT: G4HepEm

- Validated simulation results on GPU against Geant4
  - Agreement at per-mill level in the mean energy deposit (shown below) and other quantities (number of secondaries, number of steps, charged track length)

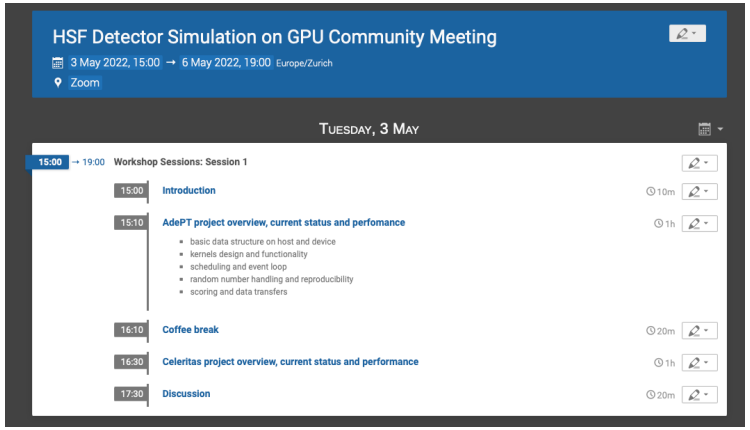


  - Compared number of generated secondaries in CMS detector (wider range of materials)

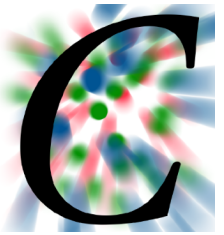
AdePT prototype plan for 2022

- completion of the AdePT prototype
  - validation of scoring in CMS (and LHCb) geometry
  - implementation of interface to HepMC to run with Pythia events
  - implementation of tracking in non-constant magnetic field
  - R&D on the new geometry model
  - debugging, validation, benchmarking, tuning, performance assessment, identification of bottlenecks in the workflow and possible solutions

- organization of the community meeting 3-6 May
- exploration of possible synergy with Celeritas project for the continuation of GPU R&D



Celeritas: motivation and scope



Philippe Canal *(FNAL)*,
Tom Evans *(ORNL)*,
Seth Johnson *(ORNL)*,
Guilherme Lima *(FNAL)*,
Amanda Lund *(ANL)*,
Soon Yung Jun *(FNAL)*,
Vincent Pascuzzi *(BNL)*,
Stefano Tognini *(ORNL)*

- US DOE effort to accelerate full-fidelity MC detector simulation for LHC on GPU
  - Insufficient CPU resources for LHC requirements
  - Motivated by successful GPU refactor of nuclear reactor MC transport (ExaSMR) via the Exascale Computing Project
- Goal: full standard physics in time for run 4
  - Standard EM physics (e-, gamma, e+) currently
  - VecGeom for CUDA detector navigation

Argonne NATIONAL LABORATORY

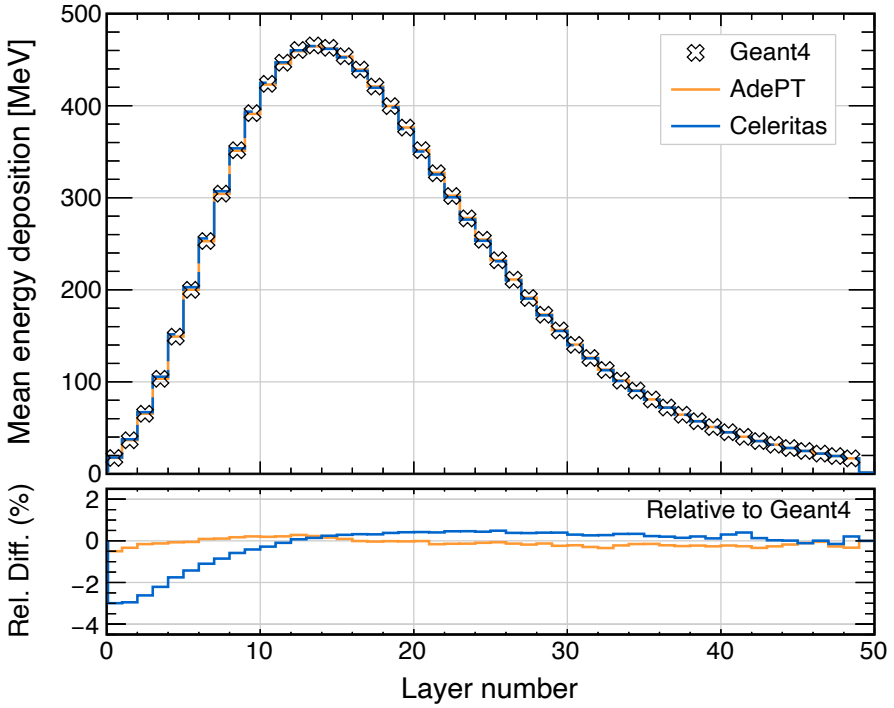
Fermilab

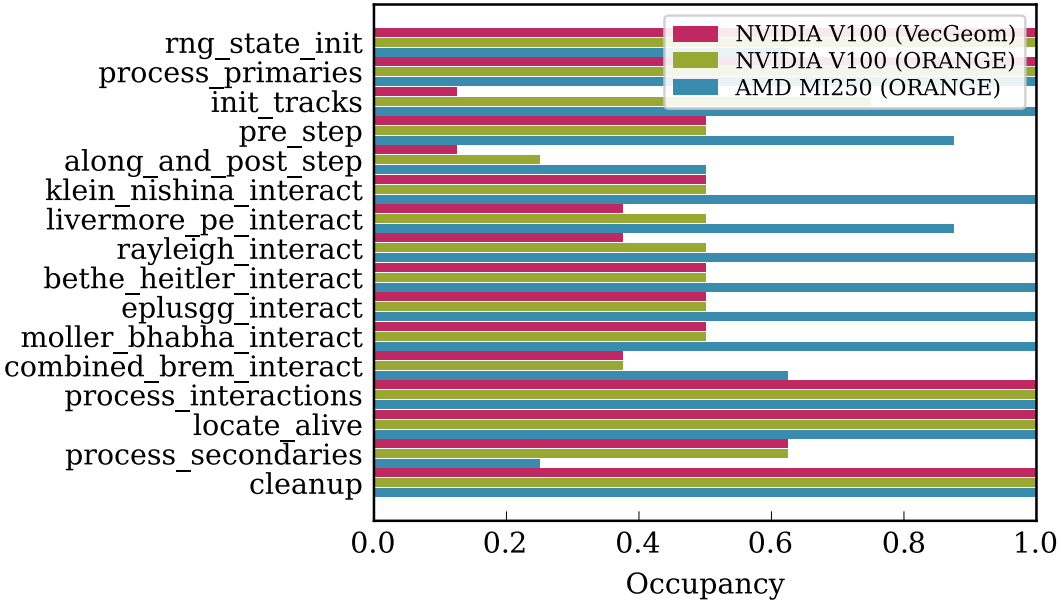
OAK RIDGE National Laboratory

Brookhaven™ National Laboratory

Celeritas: current results

- TestEm3 (100k particles; 1GeV e^-; 50×0.8 cm lAr/Pb slabs)
- Summit: GPU to CPU-only speedup per node: **33×**
  - 6 NVIDIA V100 GPUs and 42 Power9 CPU cores
  - Celeritas with VecGeom: CUDA vs OpenMP
- Agreement with Geant4 to ~2% for this test problem
  - Physics validation in progress
- Platform-portable geometry (ORANGE)
  - Verified on AMD GPUs (MI250x, ROCm 4.5)
  - Performance testing in progress
- Ongoing multi-institutional effort to codify performance and accuracy benchmarks for GPU development





Celeritas: near-term plans

- FY2022:
  - EM physics verification
  - Performance portability
  - *Acceleritas*: Geant4 task framework for dispatching EM physics to Celeritas
- FY2023:
  - Scoring system for high-performance workflow integration
  - Hadronic EM physics
  - *Acceleritas* testing/verification

Physics	Process	Particles(s)
EM	photon conversion	γ
	pair annihilation	$e^{\pm}$
	photoelectric effect	γ
	ionization	charged leptons and ions
	bremsstrahlung	charged leptons and hadrons
	Rayleigh scattering	γ
	Compton scattering	γ
	Coulomb scattering	γ
	multiple scattering	charged leptons, hadrons
	continuous energy loss	charged leptons, hadrons and ions
Decay	two body decay	$\mu^{\pm}$, $\tau^{\pm}$, hadrons
	three body decay	$\mu^{\pm}$, $\tau^{\pm}$, hadrons
	n-body decay	$\mu^{\pm}$, $\tau^{\pm}$, hadrons
Hadronic	nucleon-nucleon	p, n
	hadron-nucleon	hadrons
	hadron-nucleus	hadrons
	nucleus-nucleus	hadrons

Opticks/G4Opticks/CaTS

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

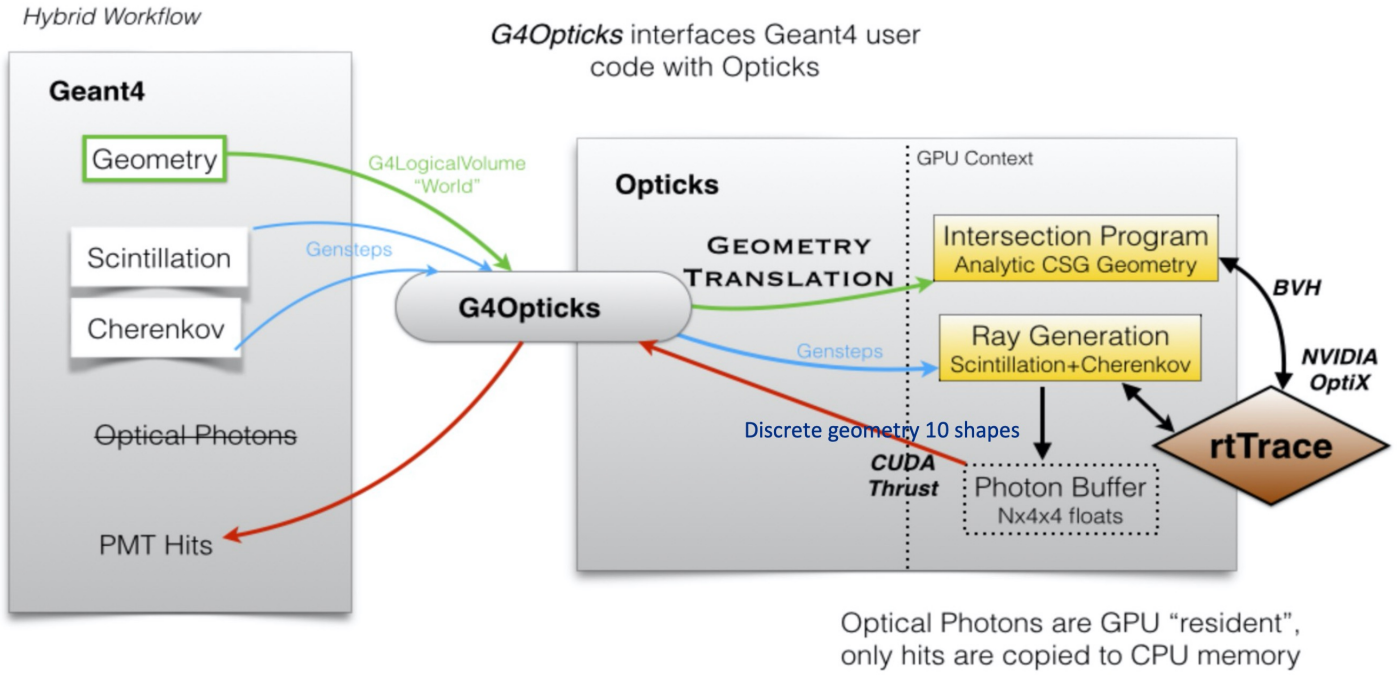
Developed by Simon Blyth:

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

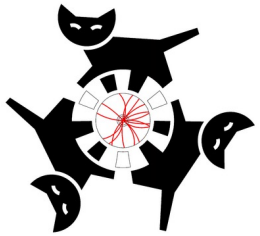
CaTS: interfaces Geant4 user code with Opticks. It defines a hybrid workflow where generation and tracing of optical photons is offloaded to Opticks (GPU) at the end of the event, while Geant4(CPU) handles all other particles. CaTS was included in Geant4 11.0 as an advanced example:

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

Figure from Simon's presentation

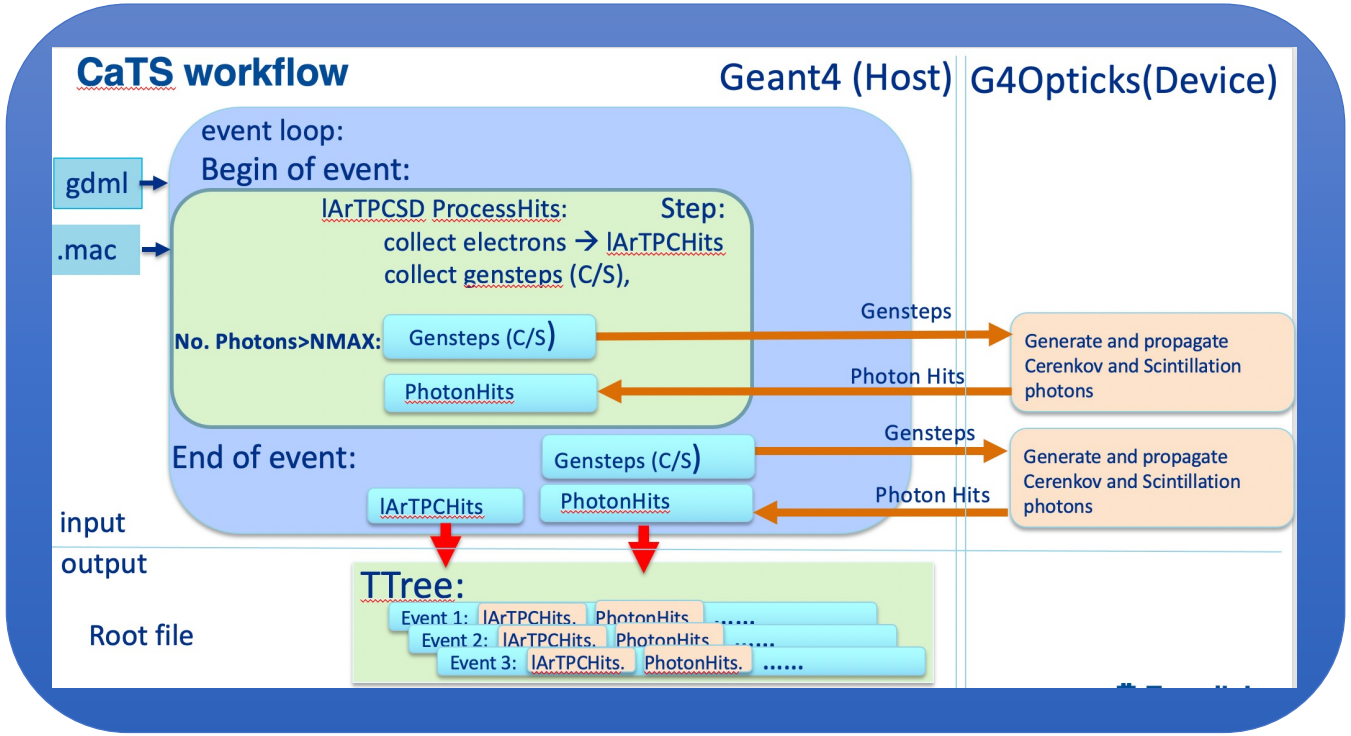


CaTS: advanced Geant4 example

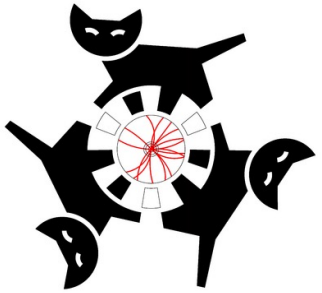


- Uses Geant4 to collect Scintillation and Cerenkov Gensteps. A Genstep collects all the data necessary to generate Cerenkov/Scintillation photons on the GPU. The harvesting is done in Sensitive Detectors(SD) (RadiatorSD/lArTPCSD). The number of photons to be generated is calculated by Geant4 and constrained to be identical whether one uses the Geant4 optical physics or G4Opticks.
- At runtime allows to select between Opticks and Geant4 optical physics to generate and propagate optical photons.
- 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, TrackerSD, CalorimeterSD, DRCalorimeterSD, PhotonSD….
  - Assigning step-limits to logical Volumes.
  - Assigning production Cuts by regions.
  - Assigning visualization attributes.
- 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.



Plans



G4Opticks/Opticks:

- Use the same implementation of the scintillation process on CPU and GPU, use the same optical properties/keywords.
- Implement Wavelength shifting process (WLS).

CaTS:

- Achieve true concurrency by using G4Tasking.
- 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.

Summary

- activities in the area of compute accelerators and Machine Learning continuing with important achievements
  - first prototypes of GPU based EM particle transport for realistic geometry (CMS)
  - integration of OptiX-based acceleration in Geant4 optical physics
  - first Geant4 example of ML-based fast simulation inference integration
- community meeting on GPU-based simulation (3-6 May) to discuss the current status and future directions
- CaloChalenge – testing and validating Machine Learning-based fast simulation