Task force for R&Ds

Activities overview and plan of work

Witek Pokorski

Jonathan Madsen

11.03.2021

R&Ds overview

- activities continuing along 3 axis
  - 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)

Anna Zaborowska

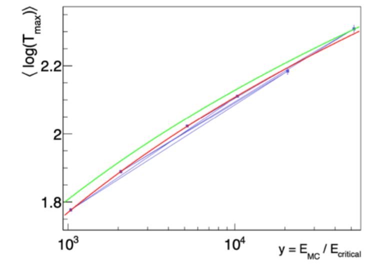
Fast simulation – ‘classical’ parameterization

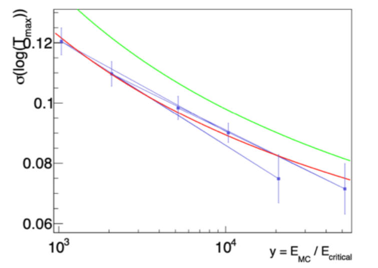
- continuation of tuning of parameters of Gflash-inspired models
  - start of shower dependent tuning
  - transverse shower profile tuning

- automatic tuning tools and generalization procedures

$$f(t) = \frac{((\alpha - 1)\, t)^{\alpha-1}\, (\alpha - 1) \exp^{-\beta t}}{T \cdot \Gamma(\alpha)}$$

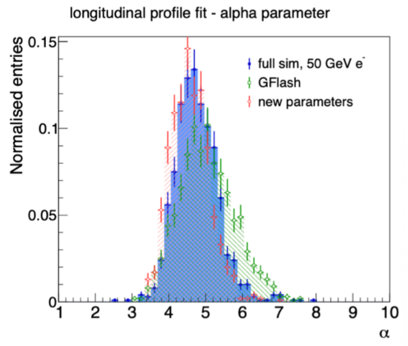
- $\log T$, $\log \alpha$ extracted from single particle longitudinal profiles for $PbWO_4$
- Fitted function (red)
- Original GFlash parameters are plotted in green

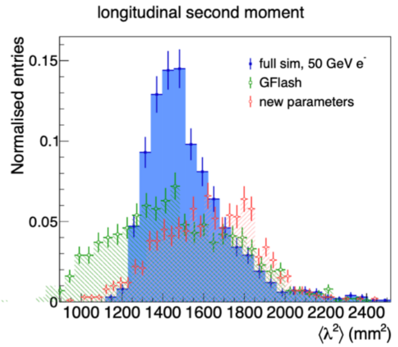


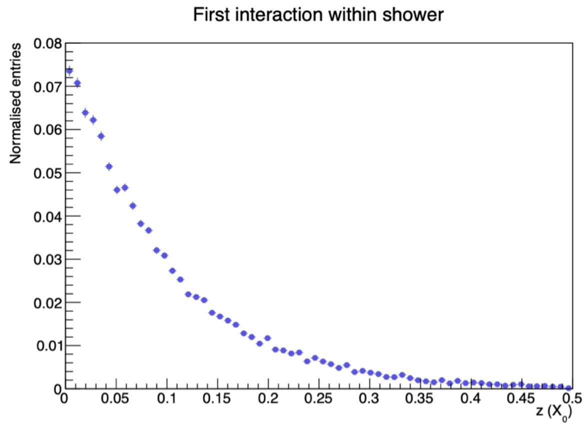


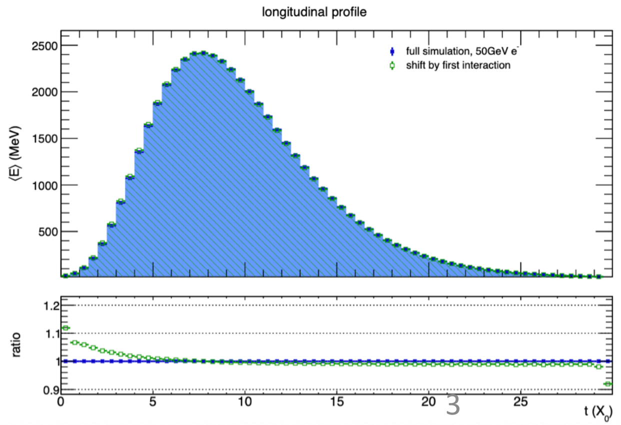
Longitudinal profile not improved immensely - but extracted parameters (T, α), first/second moments are closer to full sim than GFlash.

- full sim of 50 GeV e^- in $PbWO_4$
- GFlash parametrisation in G4
- New parameters









Fast simulation – ML

Dalila Salamani, Anna Zaborowska

1. **Integration of inference into C++**
    a. Provide G4 example extending its simulation facilities to ML-based fast simulation designed with long-term maintainability
    b. [Already started]
        i. FastMLSim class integrated in G4 : loads the saved ML model and simulates energy depositions in 3D coordinates.
        ii. Comparative study of existing tools (LWTNN, ONNX, TMVA, TF light) in terms of supported ML models, stability, memory footprint, inference time ...
2. **Provide detector-agnostic models for easy application and extension to detector geometries facilitating their use by various experiments**
    a. Design a ML model with predefined architecture & condition on generic parameters (energy of the truth particle, η,..) to study and track model's performance on custom detector geometries to evaluate the changes & limitations,..
    b. This allows the easier design of a generic, detector-agnostic ML simulator based on the ML concept of learning to learn fast or "Meta learning".
3. **Validation & optimization of ML models**
    a. Design of generic metrics for validating the ML model performance during the optimization to automatically select the best set of model parameters.

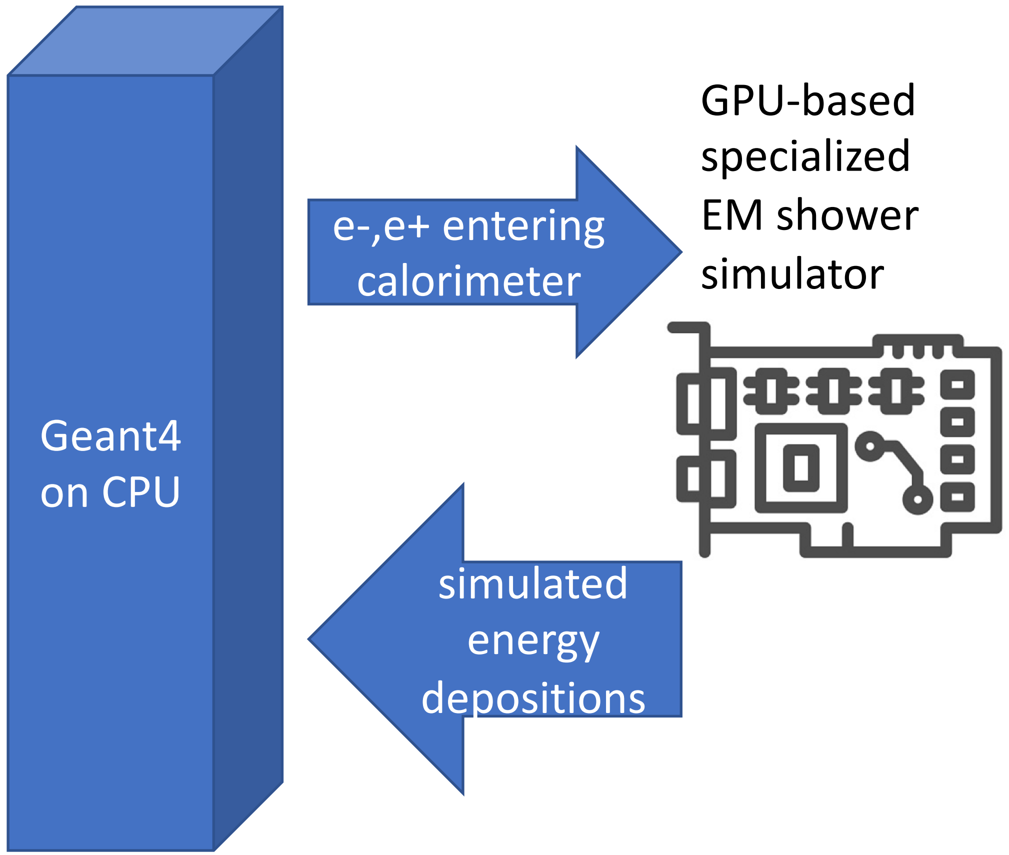
GPU-based simulation

- AdePT – Accelerated demonstrator of electromagnetic Particle Transport
- Celeritas
- Opticks integration

CERN EP/SFT, IT/SC, UK ExCALIBUR-HEP

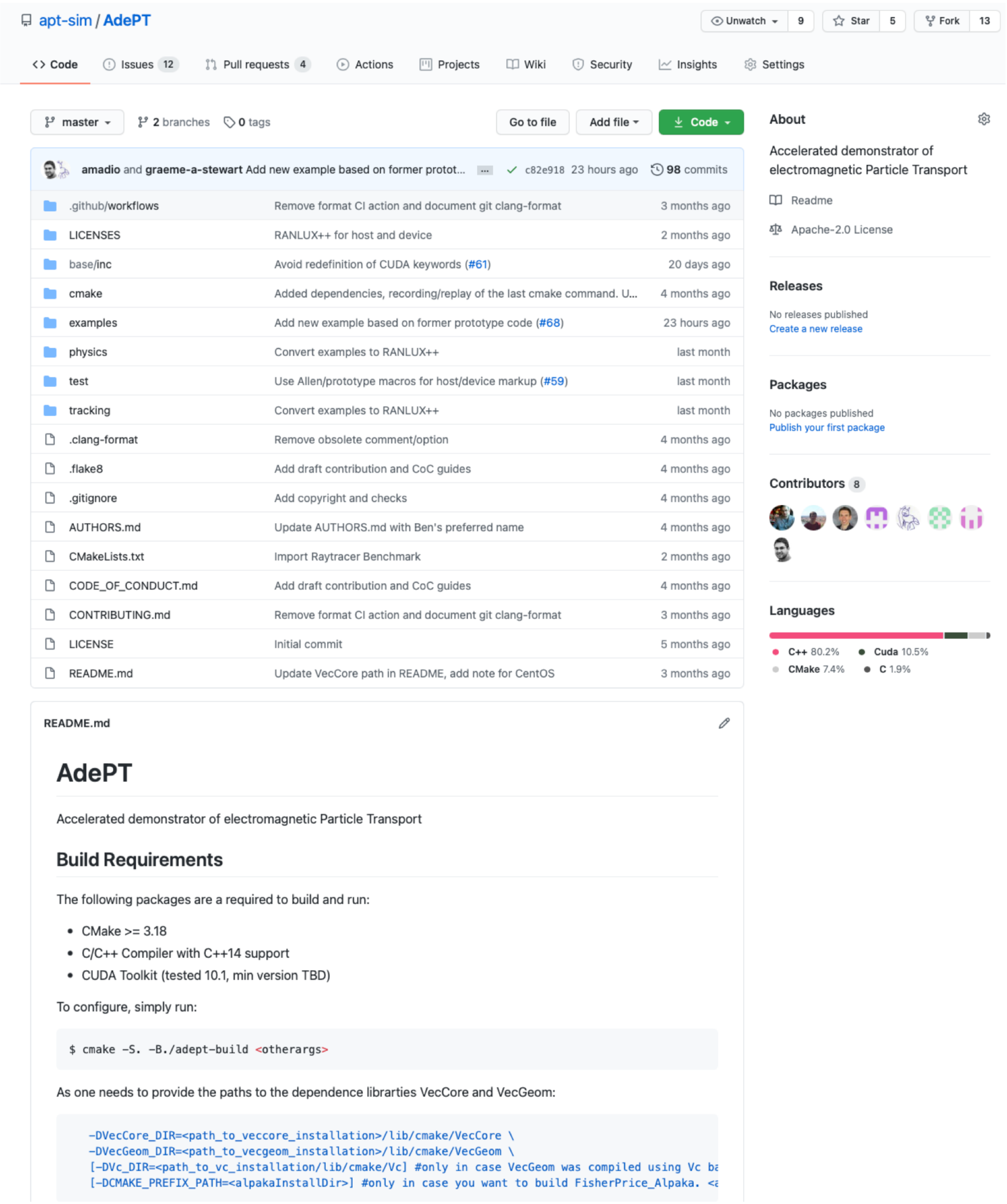
AdePT – ideas and goals

- significant amount of work needs to be performed in one go on the device due to the high cost of transferring data between CPU and GPU
- decided to focus on prototyping specialized GPU code to perform electromagnetic shower simulation in a calorimeter
    - specialized set of physics models and geometry
    - pre-defined scoring capabilities
- Geant4 would off-load simulation of EM showers to the GPU library
    - similar concept to 'fast-simulation' processes, but doing full simulation on GPU



AdePT status

- started with CUDA utilities for track data handling
- implemented several toy examples of increasing complexity
  - selecting ’physics process’ based on random number
  - dummy energy loss and pair production processes as kernels consuming queues of particles
  - running ‘shower’ of particles
- integrated geometry (VecGeom)
  - several improvements to VecGeom code to make it GPU-aware (and efficient)
    - implemented new way of handling navigation states as indices
- implemented first version of magnetic field propagator
- interfaced to G4HepEm library
  - example with VecGeom geometry, constant magnetic fields, G4HepEm physics processes implemented



AdePT goal for 2021

- develop a demonstrator of EM calorimeter simulation on GPU with as many realistic components as possible
  - LHC calorimeter-like geometry
  - main (all relevant) EM physics processes
  - magnetic field (as LHC detectors)
  - calorimeter-specific scoring (as expected by the rest for the event processing)

- perform first assessment of possible speed-up with respect to equivalent CPU-based simulation

AdePT road map

March 2021
Example demonstrating interfacing to G4HepEm library and tracking in constant magnetic field implemented (done).

April 2021
Example allowing physics validation between AdePT and Geant4 implemented (equivalent to TestEM3).

May 2021
Geant4 (MT) example implemented to allow calling AdePT like a fast simulation process, delegating the simulation of the whole EM shower coming from a particle entering the calorimeter volume and then transferring back the energy depositions.

June 2021
First version magnetic field GPU propagator for non-constant field implemented.

September 2021
First complete AdePT simulation running (on realistic setup like CMS calorimeter), including all EM interactions, tracking in non-constant magnetic field, sensitive detector functionality.

November 2021
First stage of optimization and performance assessment completed.

Early 2022
Community meeting discussing the results and planning a possible community wide-project.

Celeritas project objective

Deliver a GPU-accelerated particle transport application for HEP detector simulations

1. Why is this capability needed?
   - Current high-fidelity, time-dependent, detector energy deposition simulations will not scale to proposed 10× luminosity increase in 2025-2026
2. What are the technical capabilities and opportunities needed for breakthroughs in the detector mod-sim area?
   - Efficiently use leadership class hardware (GPUs) to increase particle tracking throughput with concurrent improvements in I/O and post-processing analysis
3. How is the current project different from previous efforts and how will it enable those breakthroughs?
   - Accelerator technology (GPUs) has reached maturity
   - Monte Carlo transport applications have demonstrated performance on these architectures in ECP (exascaleproject.org) and CASL (casl.gov) for science campaign level simulations
   - Through ECP, we have access to a complete ecosystem of high performance libraries and tools
4. What is the long-term strategy for integration with Geant4?
   - Use Celeritas to offload EM physics in a standard Geant4-constructed application
   - Use Celeritas as part of a broader LHC workflow for complete detector simulation
   - Combinations of both approaches should be possible

Celeritas team represents cross-discipline group from ECP and HEP projects

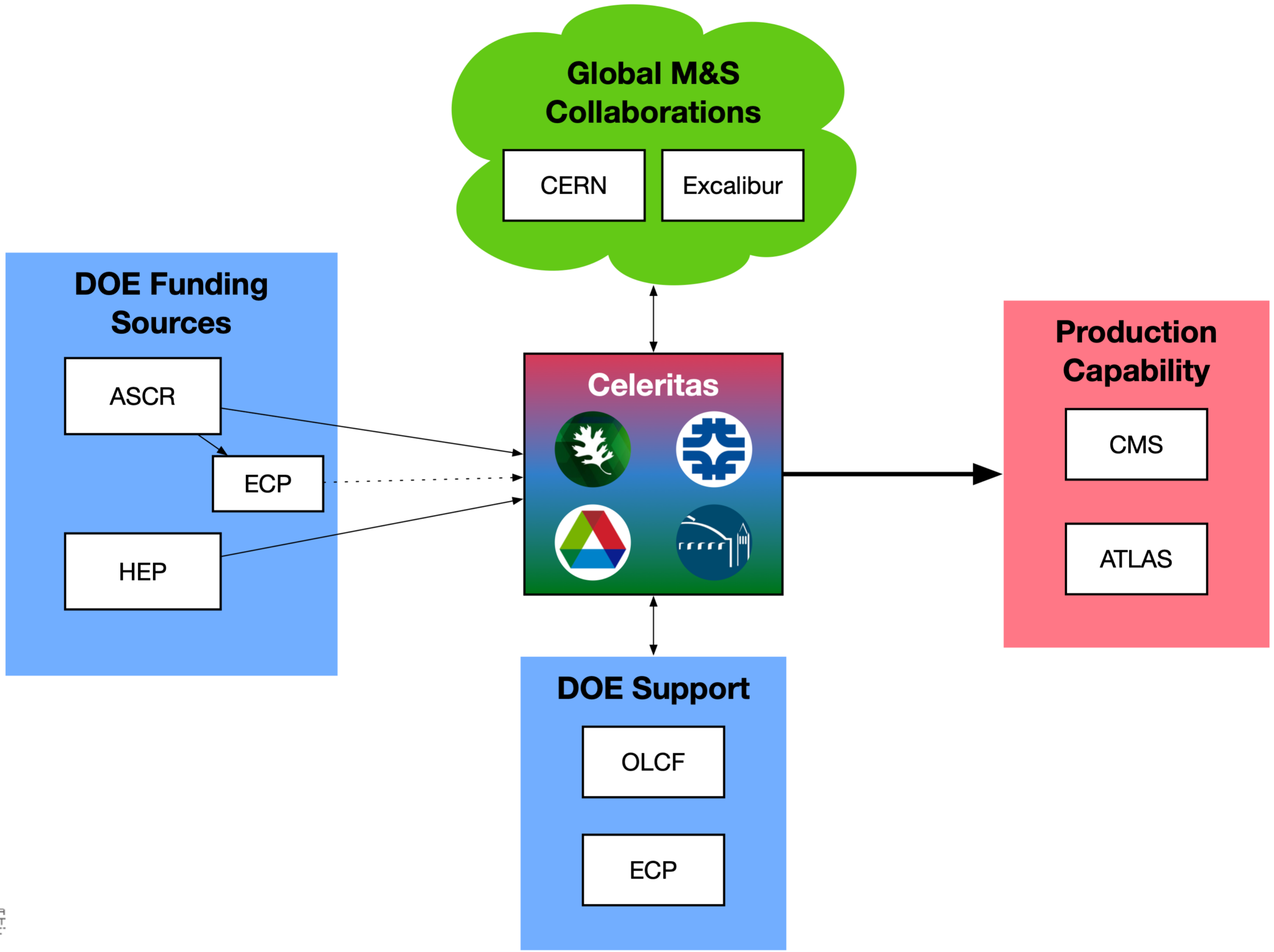
- ANL—Particle Transport Group
  - Amanda Lund
- Fermilab—Physics and Detector Simulation Group
  - Philippe Canal, Soon Yung Jun, Guilherme Lima
- LBL—ATLAS Group
  - Vincent Pascuzzi (PD)
- ORNL—HPC Methods and Nuclear Applications Group
  - Tom Evans (**PI**), Seth Johnson (**Code Lead**), Stefano Tognini (PD)
- Most of these staff are leveraged through other project funding

Argonne NATIONAL LABORATORY

Fermilab

OAK RIDGE National Laboratory

BERKELEY LAB


Global M&S Collaborations
CERN
Excalibur
DOE Funding Sources
ASCR
ECP
HEP
Celeritas
Production Capability
CMS
ATLAS
DOE Support
OLCF
ECP

Celeritas capabilities and plans

June 2020–June 2021	July 2021–June 2022
✓ Basic infrastructure: GPU material/particle/physics data ✓ GPU transport loop with single process and material, secondary production, limited validation ✓ VecGeom tracking on GPU ➢ Multi-material, multi-process, multi-particle demonstration ➢ Magnetic field propagation	➢ Complete standard EM physics, including multiple scattering • Prototype performance portability • Prototype integration with experiment frameworks

EXASCALE COMPUTING PROJECT

Opticks/G4Opticks

Hans Wenzel

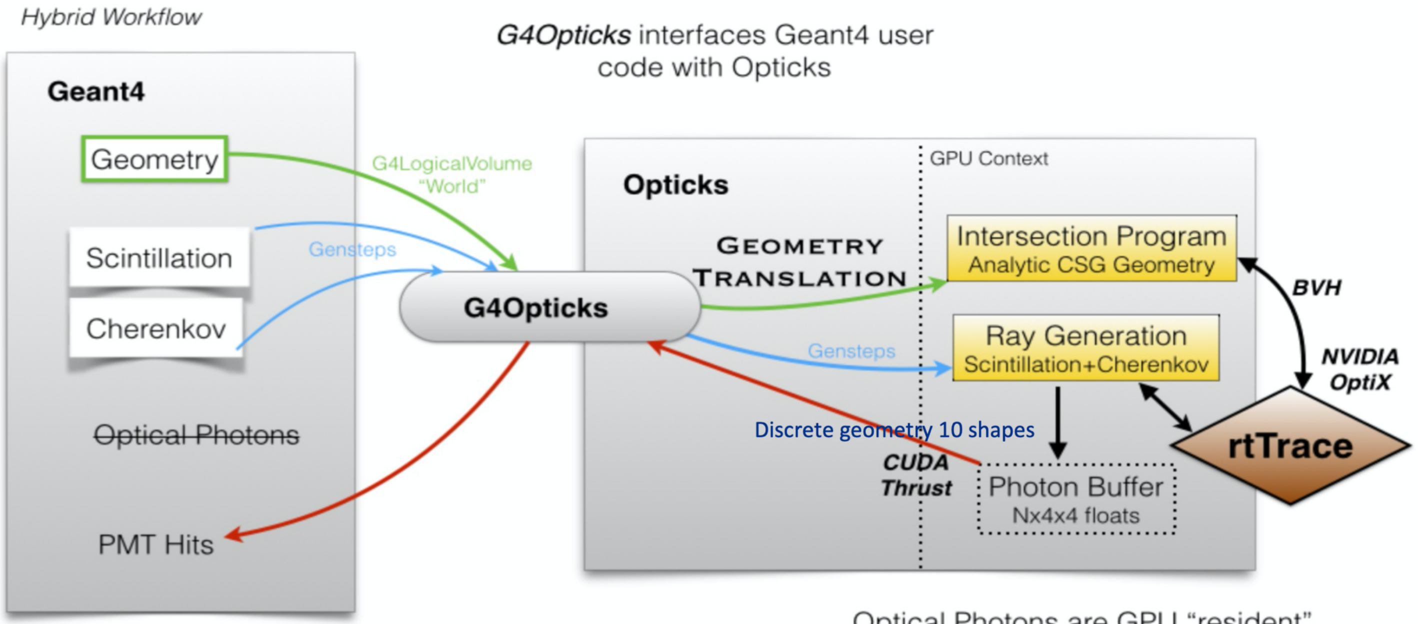
Opticks is an open source project that accelerates optical photon simulation by integrating NVIDIA GPU ray tracing, accessed via NVIDIA OptiX.

Developed by Simon Blyth:

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

G4Opticks: interfaces Geant4 user code with Opticks. It defines a hybrid workflow where generation and tracing of optical photons is offloaded to Opticks (GPU) at end of event, while Geant4(CPU) handles all other particles.

Figure from Simon's presentation



Optical Photons are GPU "resident", only hits are copied to CPU memory

Plans with respect to evolving G4Opticks/Opticks:

- Use the same implementation of the scintillation process on CPU and GPU, use the same optical properties.
- Implement Wave Length Shifting (WLS).
- Achieve true concurrency by using G4Tasking by J. Madsen. (Geant4 $>$ 10.7)

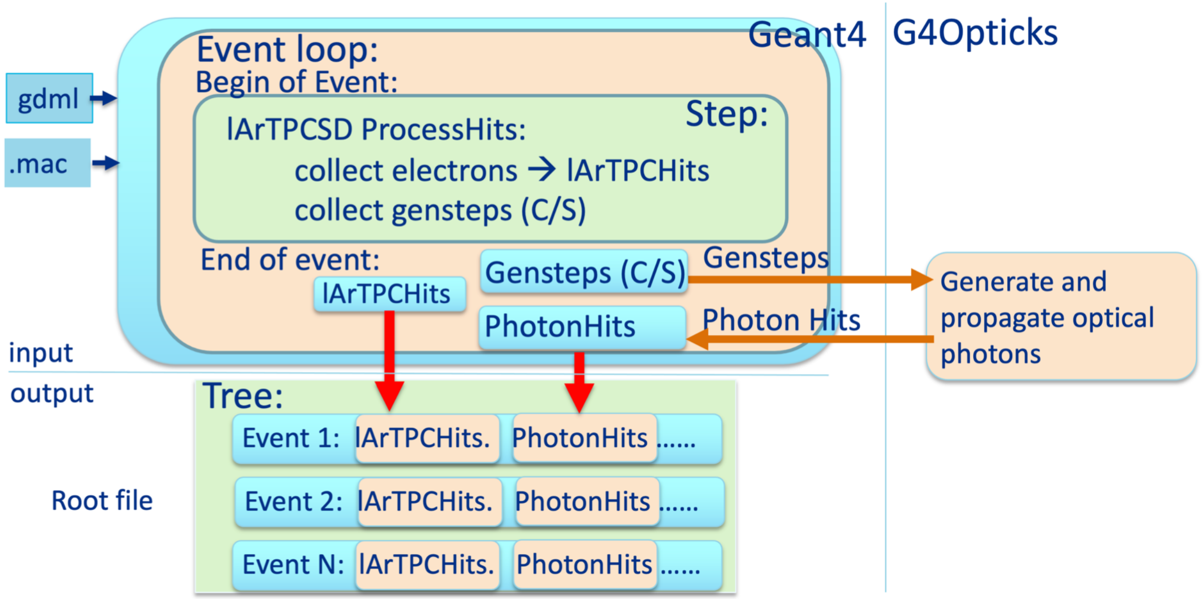
Hans Wenzel Fermilab

G4OpticksTest:

- Geant4 Application demonstrating the use of the G4Opticks hybrid workflow, works with Geant4 10.6.xx (requires minor patches) and 10.7.xx
  - Code: https://github.com/hanswenzel/G4OpticksTest

Features are:

- Uses Geant4 to collect Scintillation and Cerenkov Gensteps. The harvesting is done in sensitive Detectors(SD) (RadiatorSD/lArTPCSD).
- At runtimes allows to select Opticks/Geant4 optical physics to generate and propagate optical photons, returns Photon-Hits.
- 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, DRCalorimeter,....
  - Assigning step-limits to logical Volume
  - Assigning visualization attributes.
- Uses G4PhysListFactoryAlt (R. Hatcher) to define and configure physics at runtime.
- Uses Root IO to provide persistency for Hits.



Plan to make it a Geant4 advanced example

Performance (preliminary):

Hardware:

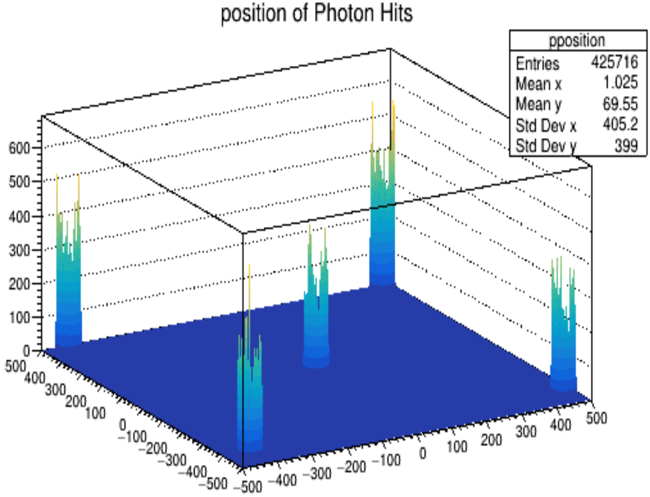
CPU	Intel(R) Core i7-9700K 3.6GHz 32 GB memory.
GPU	GeForce RTX 2070 CUDA Driver Version /11.0 CUDA Capability: 7.5 VRAM: 7981 Mbytes Cores: 2304

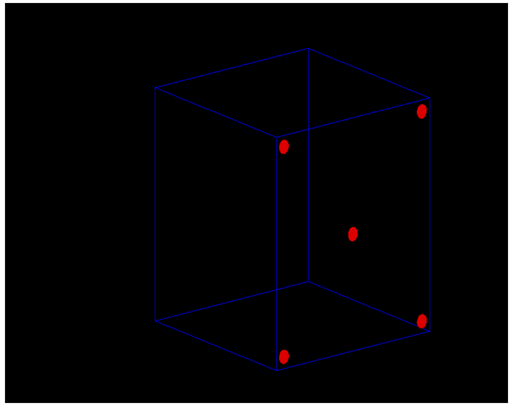
Timing results (Geant4 10.6.p03, includes RootIO):

Geant4 optical physics	85.6 sec/evt
Opticks	1.84 sec/evt
Opticks RTX enabled	0.57 sec/evt

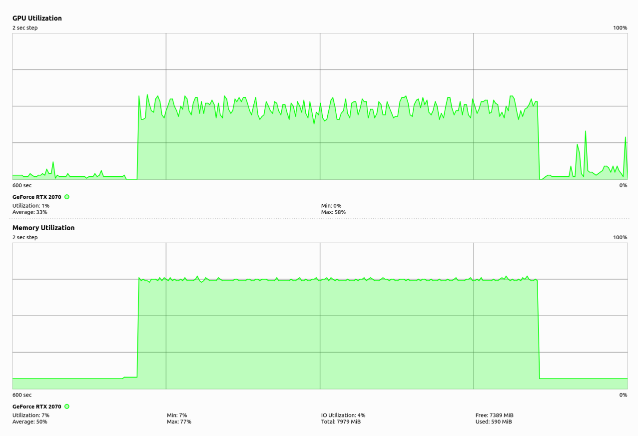
Geant4/Opticks: 85.6/0.57 = 150 x overall speed up.
RTX hardware acceleration: 1.84/0.57= >3 X speed up.

Simple Geometry:
Liquid Argon: 1 m^3
5 photo detectors (red)
photon yield: 50000 γ/MeV single
1GeV muon

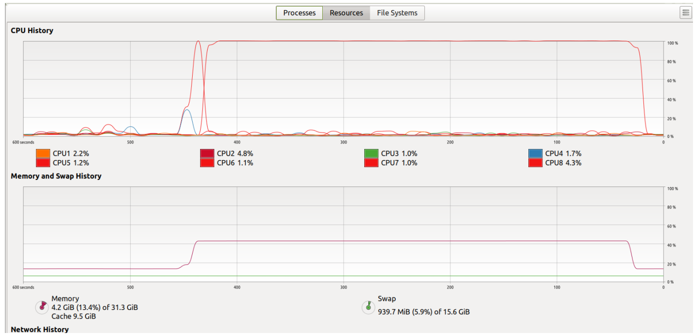




GPU and VRAM usage:



CPU and main memory usage



Hans Wenzel Fermilab

Summary

- several activities in the area of compute accelerators and Machine Learning
  - focusing either on specific domain or generic
- we expect two important demonstrators to be delivered in 2021
  - generic ML-based fast simulation tools
  - GPU-based HEP simulation prototypes
- outcome of those prototypes will set the direction for further R&D work