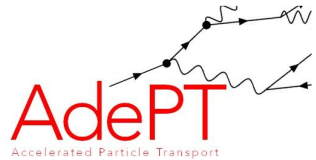


AdePT status

Accelerated demonstrator of electromagnetic Particle Transport

Guilherme Amadio (CERN), John Apostolakis (CERN), Predrag Buncic (CERN), Gabriele Cosmo (CERN), Daniel Dosaru (EPFL), Andrei Gheata (CERN), Stephan Hageboeck (CERN), Jonas Hahnfeld (CERN), Mark Hodgkinson (Sheffield University), Benjamin Morgan (Warwick University), Mihaly Novak (CERN), Adrian Antonio Petre (ISS Bucharest), Witold Pokorski (CERN), Alberto Ribon (CERN), Graeme A Stewart (CERN), Pere Mato Vila (CERN)

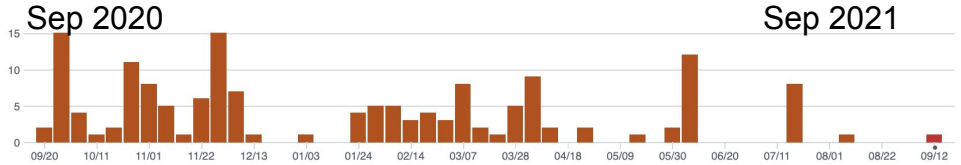
26th Geant4 Collaboration Meeting
Sep 16, 2021

andrei.gheata@cern.ch

R&D for EM physics transport simulation on GPU

- Project started one year ago, trying to address inability to use GPU cards for detailed simulation of collider experiments
  - Transforming simulation code to be more GPU-friendly, see how far we can go…
  - Some initial ideas shown last year
- Now close to having a complete prototype for e^+, e^- and γ shower simulation
  - Completing the set of EM interactions with the last models supported by G4HepEm
  - Magnetic field propagation in detector geometry
  - Code producing user `hits` data transferred from the GPU back to the host
- Support for both GPU standalone and hybrid modes
  - Integrating with Geant4 CPU workflow, `stealing` particles and offloading to GPU
- Optimizing performance on realistic use-cases
  - To understand usability and realistic benefits for the current workflows

The AdePT project



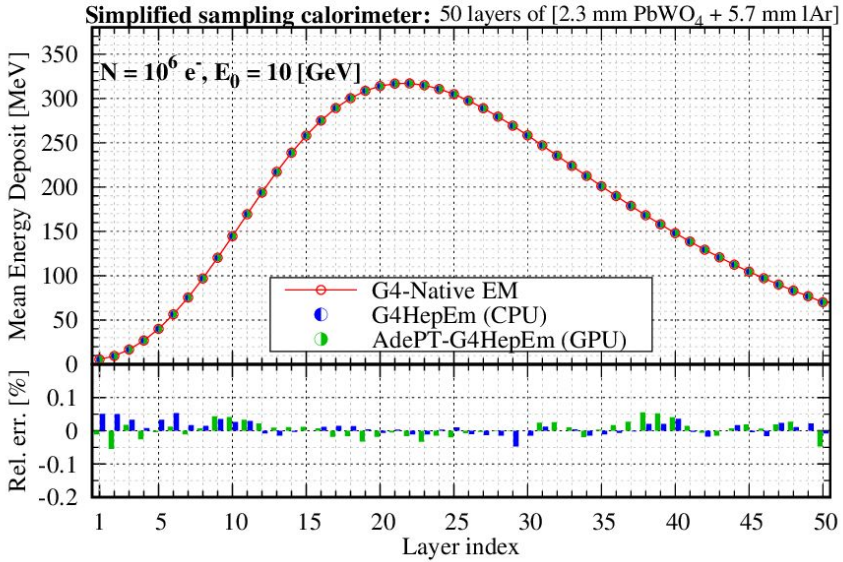
- GitHub repository
  - Started in September 2020, 11 contributors so far
- Strategy: evolve to a prototype based on gradually more complex examples
  - Core types/macros/abstractions: ***CopCore***
  - External physics: ***G4HepEm*** and geometry: ***VecGeom***
- One year project roadmap
  - Moving from a toy version to a real prototype implementation
  - **Sep 2021:** First complete AdePT simulation running, including all EM interactions, tracking in non-constant magnetic field (optionally), sensitive detector functionality
  - **~ Jan 2022:** HSF meeting to plan next steps and seek convergence of efforts on a common project
  - Some delay, but really close to these objectives now

AdePT ingredients: G4HepEm physics

- G4HepEm: compact library of EM processes for HEP
  - Covers interactions of $e^-/e^+/\gamma$, including multiple scattering (not yet integrated into AdePT)
  - Separation between data initialization (relies on Geant4) and computation of cross-sections and final state sampling during run-time (standalone, independent of Geant4)
- Design of library very supportive for heterogeneous simulations
  - Implicit support in the interfaces: Standalone functions without global state
  - Explicit support for transfer of physics tables and other data structures to GPUs
  - Makes it possible to re-use > 95% of the code from G4HepEm for GPU shower simulation
- For details, see presentations by M. Novák in December, at Geant4 Technical Forum in March, and recent status update
  - Also talk in the Kernel parallel session and in Plenary 3 next Wednesday

G4HepEm validation

- Previously 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)



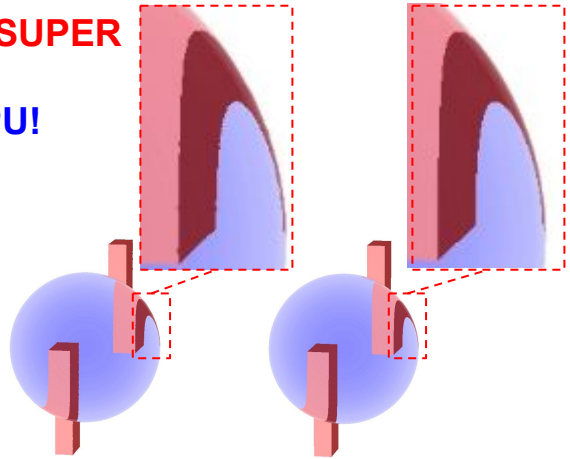
AdePT ingredients: VecGeom geometry

- Using VecGeom as GPU-aware library describing the detector geometry
  - Same CPU and GPU algorithms compiled separately for host and device
  - Geometry data re-constructed on GPU based on the transient data on the host
  - Navigation layer customized for GPU use
- Improving gradually GPU support
  - Developed custom optimised navigation state, single-precision support
  - Moving to a simple looper to an optimized BVH navigator (see talk of G.Amadio)
  - Adopting modern CMake GPU support
  - See talks in the Geometry working session
- Moving forward: specializing the VecGeom GPU navigation support
  - Important step towards portability and performance

Performance impact of single-precision

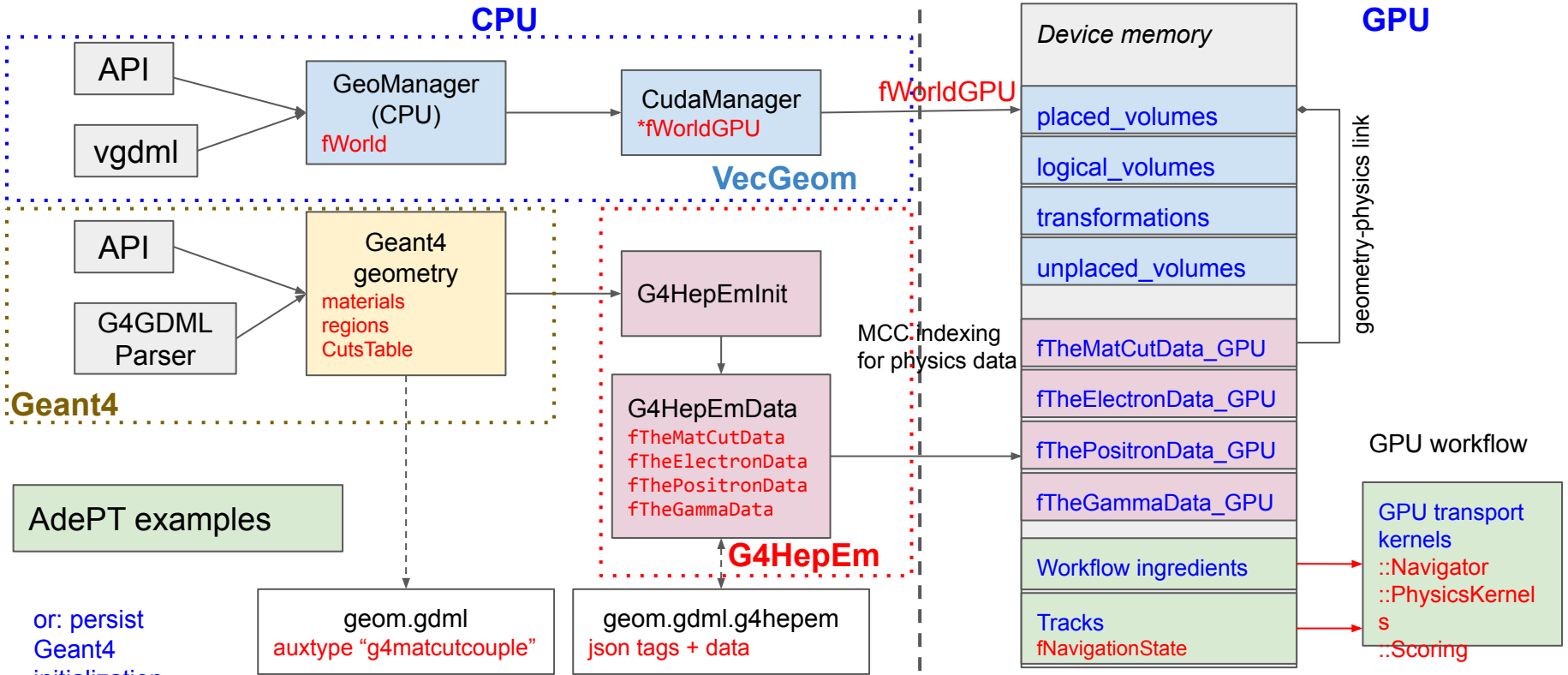
- Tested impact on performance in the AdePT examples
  - After doing several fixes for importing VecGeom precision type and using it in the navigators
    - LoopNavigator (simple looper for daughters), and BVHNavigator
- RaytraceBenchmark example (using BVHNavigator)
  - Reading a GDML file and modeling reflections/refractions and specularity
  - Validated by the output image
    - Very simple geometry: ~ 7.5% speedup
    - Complex geometry (trackML): ~ 44% GPU, ~ **13.6% CPU!**
- Physics-enabled GPU examples
  - Exa9 + trackML + LoopNavigator: ~ 2.8x speedup
  - Exa11 + trackML + BVHNavigator: ~ 30% speedup

RTX 2080 SUPER



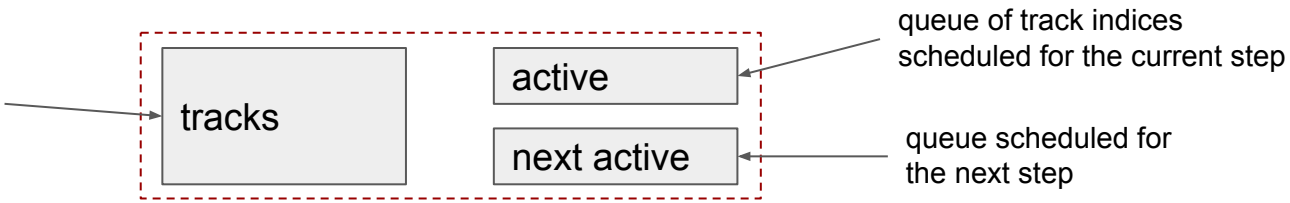
andrei.gheata@cern.ch

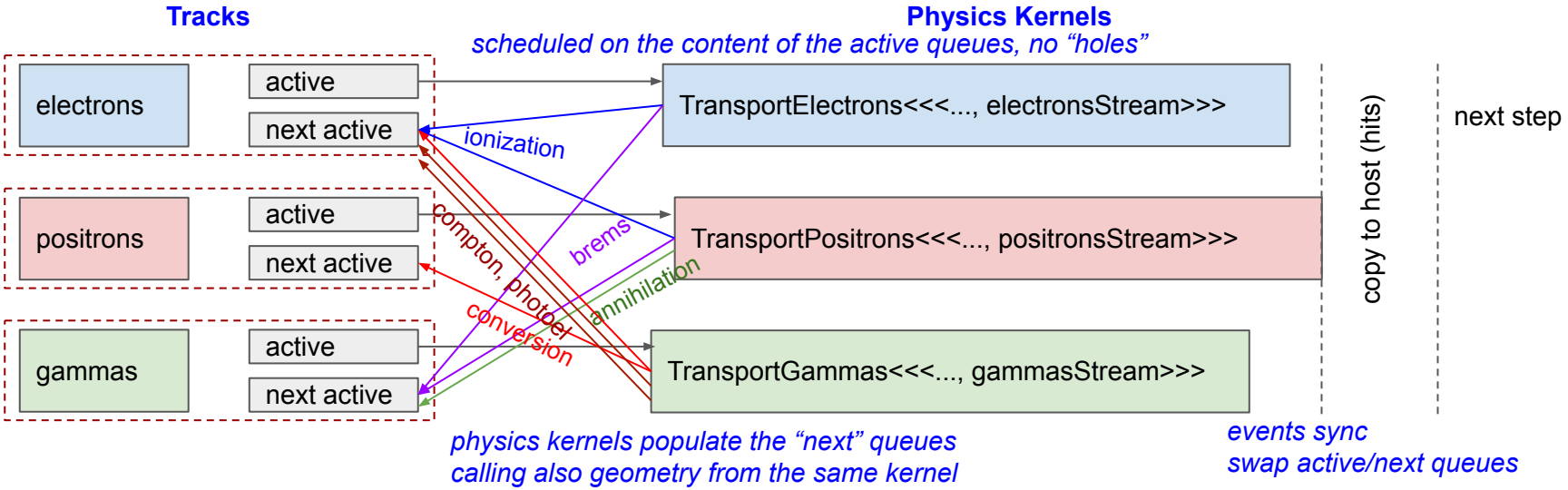
The AdePT “cookbook”



AdePT recipes: stepping loop workflow

- contiguous
- next slot atomic
- no slot reuse
- different for e+/e-/gamma





More recipes: AdePT Geant4 integration

- Integrate AdePT GPU workflow with standard Geant4 simulation on CPU
  - One of the main goals of the AdePT prototype
  - We target a Geant4 plug-in to offload work to GPU for increased throughput and improved energy/cost efficiency
    - If proven efficient, this can become a plug-in for offloading part of Geant4 simulation on device
- Inserted in the Geant4 stepping loop using the *FastSimulationModel* interface
  - Intercepts and removes from stack all EM particles entering the region(s) where the “model” triggers
  - Buffering and showering them in AdePT (now synchronous)
  - Return escaping particles and hits to CPU
  - Flushing the buffer until empty before ending the Geant4 event
- Detailed showcase of the recipe in Witek’s presentation

AdePT: evolving via examples

- Set of documented examples demonstrating different features
  - Starting with preliminary embryonic ideas implementing a Fisher-Price-like workflow
  - Adding new components in the stepping loop: pRNG, geometry, physics, magnetic field
  - Testing or demonstrating features: geometry navigation via ray-tracing, SoA vs. AoS, G4HepEm persistency, Alpaka, ...
  - Evolving the workflow: more steps per kernel, parallel relocation, multiple feeding threads
- More complete examples validating against Geant4
  - Measuring and optimizing performance
  - Common components refactored as services: navigation, field, physics
    - Recent examples share many commonalities
- Targeting a complete prototype integrated with Geant4
  - Realistic experiment geometry and physics settings + some basic scoring

Advanced examples

- Sampling calorimeter example (TestEm3)
  - Constant field on/off, Physics = ALL except MSC + photoelectric, Geant4 and GPU versions
  - Quite similar throughput for Geant4 MT on AMD Ryzen 9 3900 (12C/24T), versus AdePT on GeForce RTX 2070 SUPER after some performance improvements (J. Hahnfeld talk)
  - MT example shows limited benefits in the current workflow due to bottleneck in feeding concurrently input particle buffers
- Generic geometry with generic scoring example (CMSapp, currently PR)
  - Field on/off, Physics = ALL except MSC + photoelectric, AdePT version only for now
  - Doing initialization of Geant4, G4HepEm and VecGeom data structures based on GDML info
  - Generic scoring in all volumes, in the future only for sensitive volumes
  - Example recipe will be adopted in the integration prototype

Porting AdePT to oneAPI

- oneAPI - unified programming model by Intel
  - Uses DPC++ as programming language, implementing SYCL
- oneAdePT - port to oneAPI of an AdePT snapshot
  - core utilities, magnetic field, RNG, G4HepEM
  - Attempt to use legacy CUDA code compiled in VecGeom
- Many obstacles for migrating CUDA to DPC++ code
  - SYCL limitations in calling virtual functions or function pointers, non-const globals, support for std:: math functions, support for CUDA compiled libraries, documentation
- Triggered investigations and work in VecGeom
  - Non-virtual dispatch and CUDA compilation using clang, deeper restructuring needed
- Calling into CUDA libraries now possible, but examples on GPU still crash
  - First compile CUDA code to LLVM bitcode, then follow an obscure 10-step manual linking procedure

In preparation

- Interfacing the last physics interactions available in G4HepEm
  - Adding multiple scattering will probably change dramatically the picture
  - Testing with CMSapp example for both Geant4 and AdePT
- Finalizing the Geant4 AdePT integration
  - Now including MT support, buffering and flushing sequence
  - Understand *G4VTrackingManager* interfaces usability for GPU integration
- Putting together the complete CMS example using the integration
  - Most ingredients already integrated, but for a simpler setup
  - Performance study in standalone and combine workflows

Overlook

- Converging to a prototype combining complete particle transport features
  - Most functionality demonstrated in examples, combined CPU-GPU workflow for a realistic LHC geometry being developed
- R&D work and improvement of components
  - Several improvements and adaptations in VecGeom: navigation and state, optimizers, single-precision support
  - Support for EM physics in a GPU-friendly way
  - RNG optimisation, magnetic field support, workflow optimization
- We can answer the question “can it work?”, now assessing the “how efficient?” part
  - Only a realistic full prototype can give us an unbiased measure of performance