Accelerating Particle Transport on GPU - AdePT Experience

Andrei Gheata, Severin Diederichs
and Mihaly Novak
on behalf of the AdePT team

andrei.gheata@cern.ch

High Performance Monte-Carlo Radiation Simulations workshop – Greece 5–6 Mar 2026

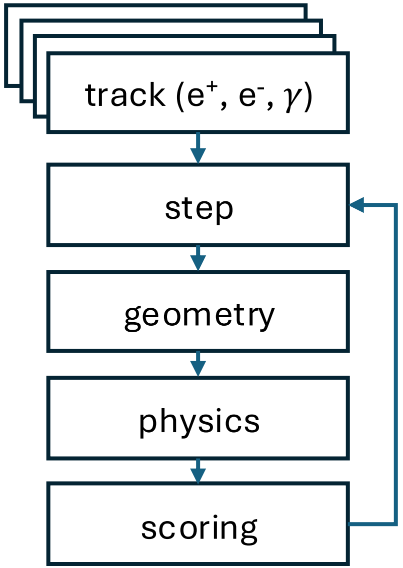
Motivation : MC transport is computationally expensive

Examples across domains:
- HEP detectors
- medical physics
- space radiation

Transport cost dominated by:
- large number of particle steps
- geometry navigation
- electromagnetic interactions
  - In HEP ~50-80%, potential speedup of 2x to 5x

GPU computing offers:
- massive parallelism
- high throughput
- Still, particle transport MC is **NOT** the ideal GPU problem

GPUs : make a step for multiple tracks



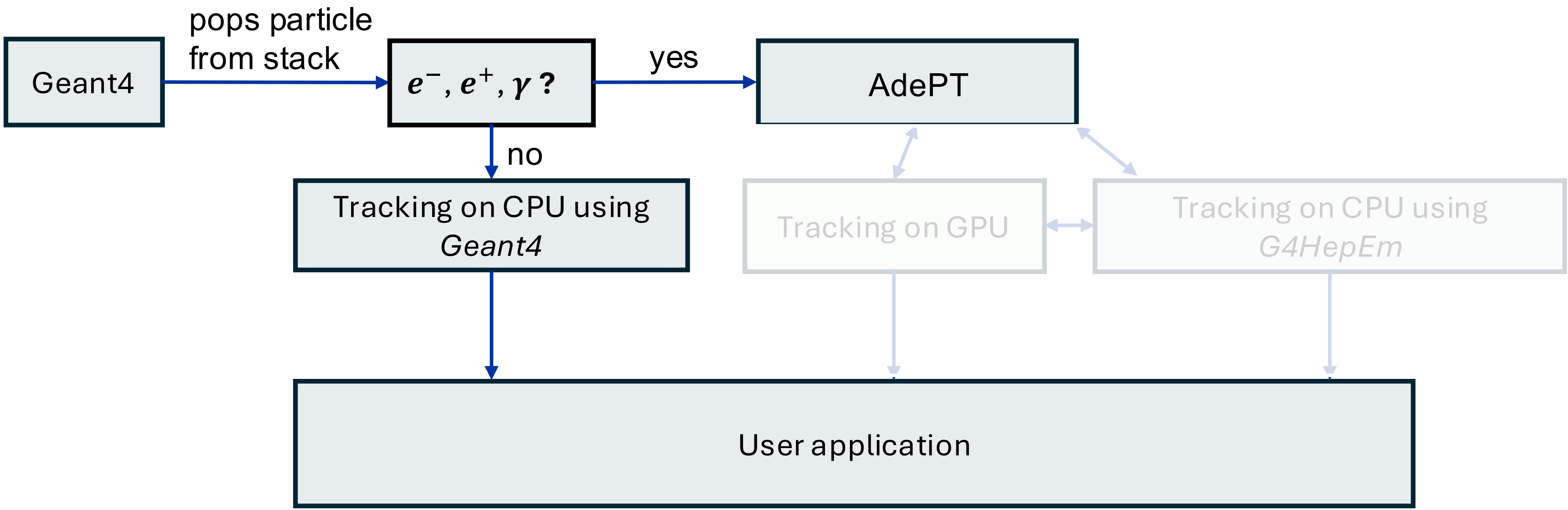
the « hot » loop

AdePT is a lightweight Geant4 plugin

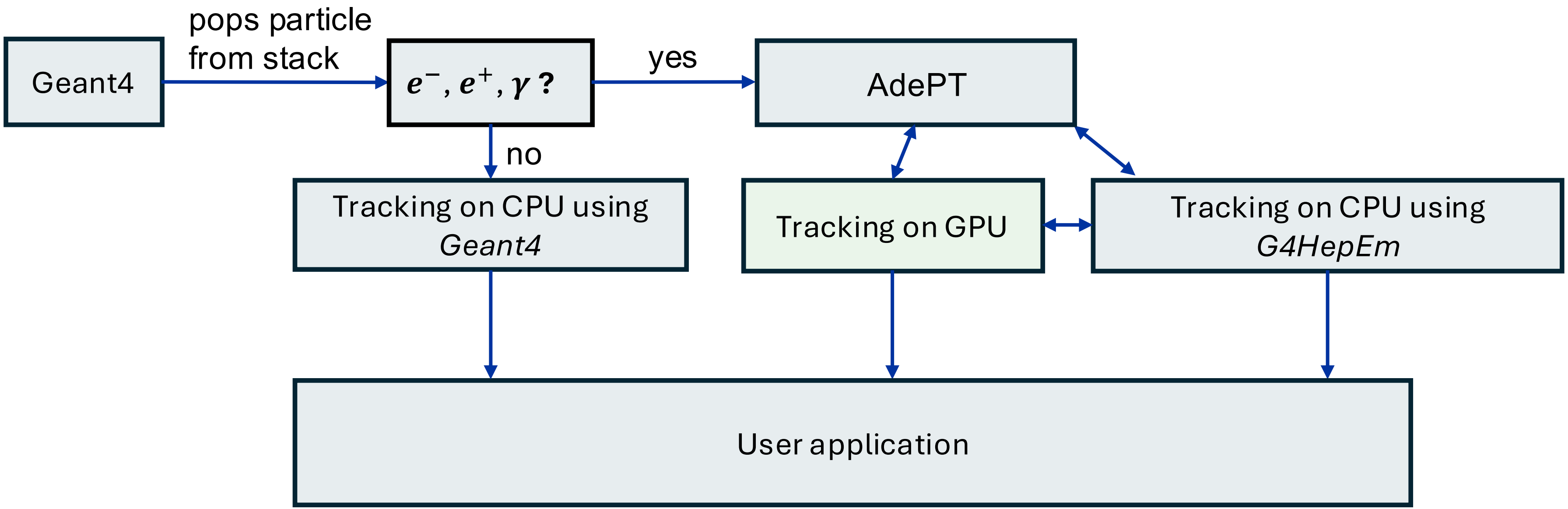
Components:

- Physics: G4HepEm
  - Provides physics functions on GPU and specialized tracking on CPU (gives ~20% speedup over native G4)
- kernel scheduling, particle tracking
- Geometry + Navigation: VecGeom
- Magnetic field: Dormand-Prince integrator
  - Using the **covfie** library for 3D field maps

Seamless offload of electrons, positrons, and gammas

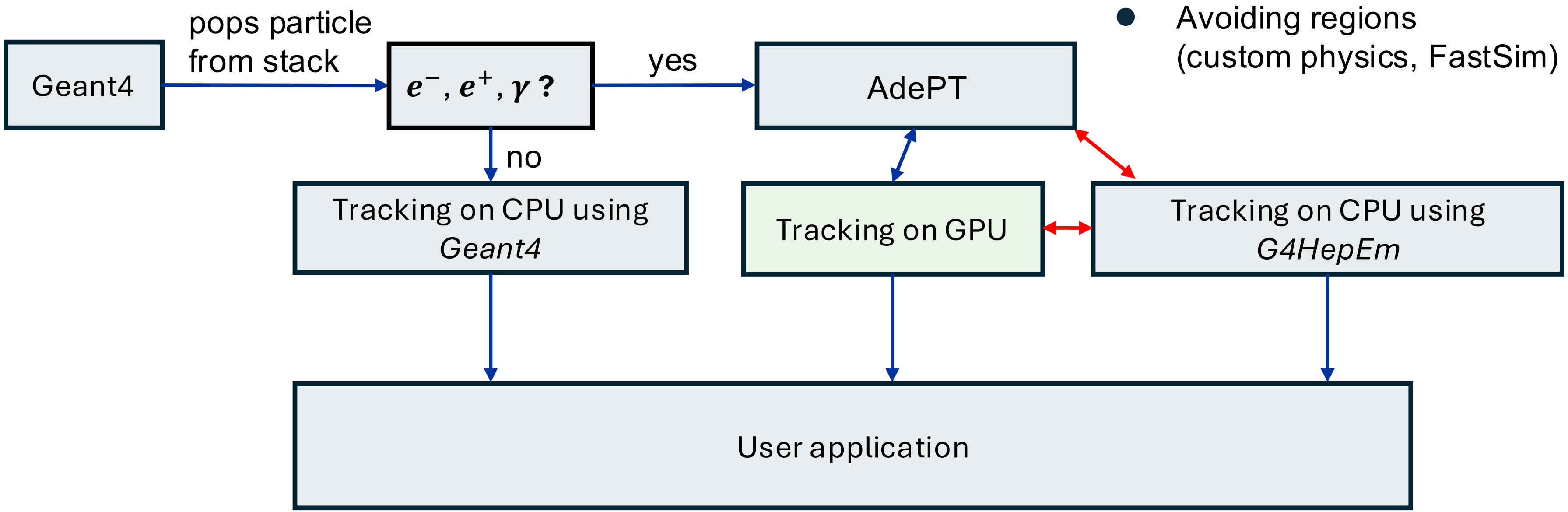


Seamless offload of electrons, positrons, and gammas

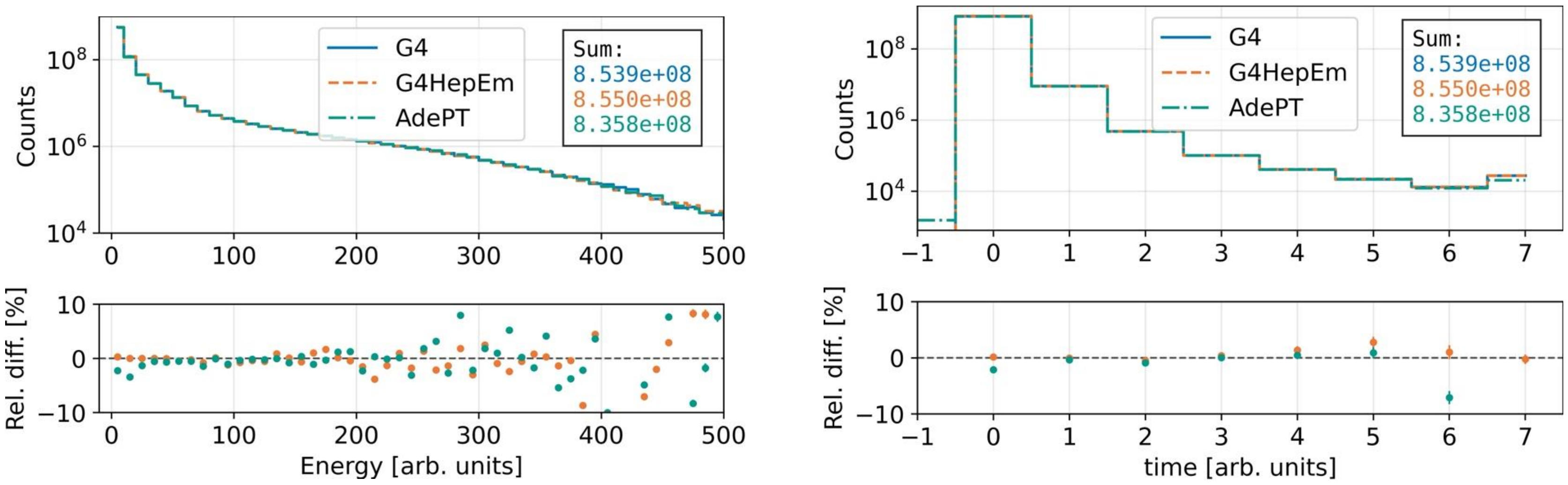


Seamless offload of electrons, positrons, and gammas

Fallback to G4HepEm:
- Nuclear processes
- Avoiding regions (custom physics, FastSim)



Status: can run close-to production simulations for LHCb, CMS, and ATLAS on GPU



Example: AdePT reproduces LHCb physics. Small differences are due to different magnetic field integrator.
Close-to : except current MC truth implementation in LHCb, depending on step sequence assumptions

Performance results in Gauss

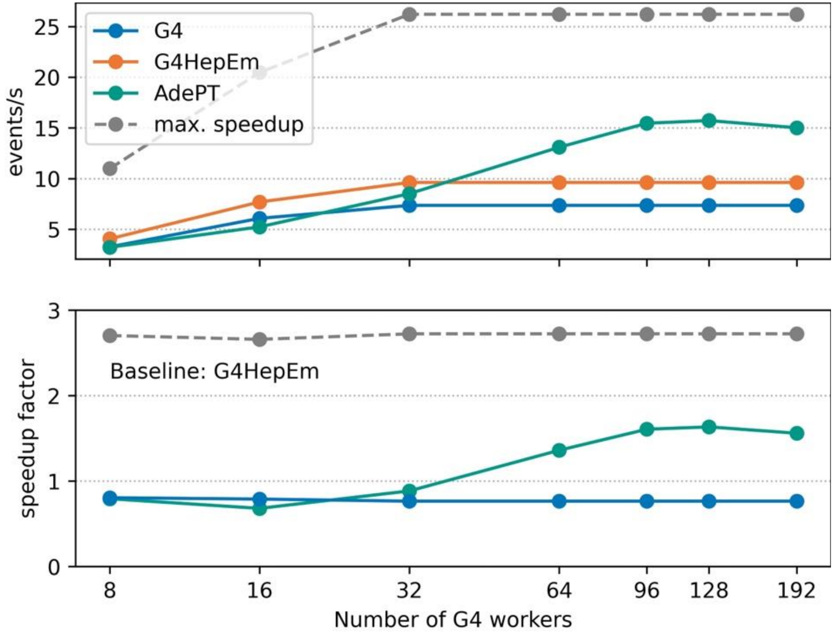
Hardware:
GPU: Nvidia **RTX4090**
CPU: AMD Ryzen 9 **16 cores**

Max. speedup factor: **2.7x**

Current speedup:
1.6x at 128 threads

Oversubscription (4x)
needed to fill the GPU

Strong scaling 20k min bias events



Performance results in Gauss

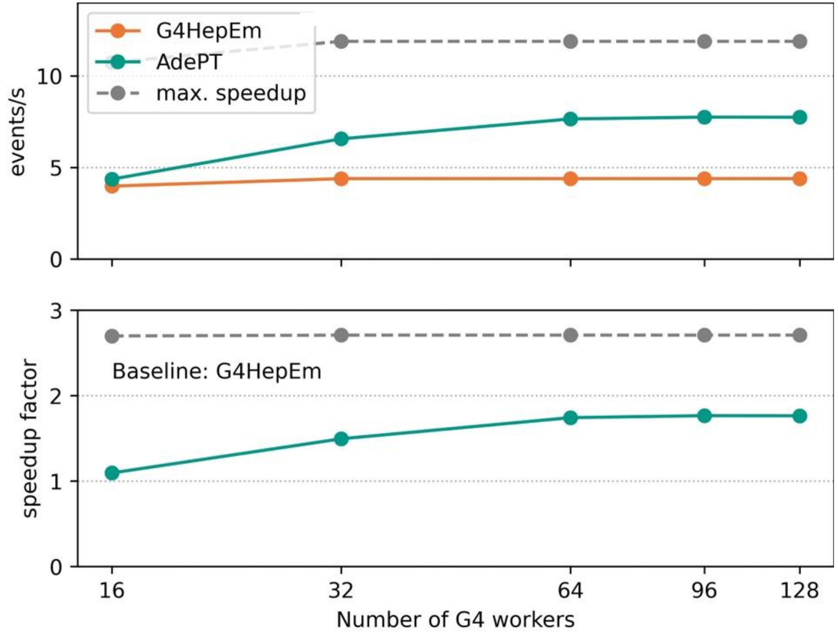
Hardware:
GPU: 4x **A100**
CPU: AMD EPYC 7763 (64 cores)
partitioned into 4 slots

Max. speedup factor: **2.7x**

Current speedup:
1.8x at 96 threads

Oversubscription (3x)
needed to fill the GPU

Strong scaling 10k min bias events for 1 slot



G4HepEm

- Complete coverage of EM physics
  - Including sampling for lepto/gamma-nuclear, deferring the secondary generation to the corresponding Geant4 models
- optimized for performance and portability
- can be used to accelerate Geant4 out of the box also on CPU
  - ~20% in average for the HEP case
- reproduces Geant4 electromagnetic physics, currently focusing on **standard configurations** used in HEP.
  - Currently no high-precision support (option4)

G4HepEm extension

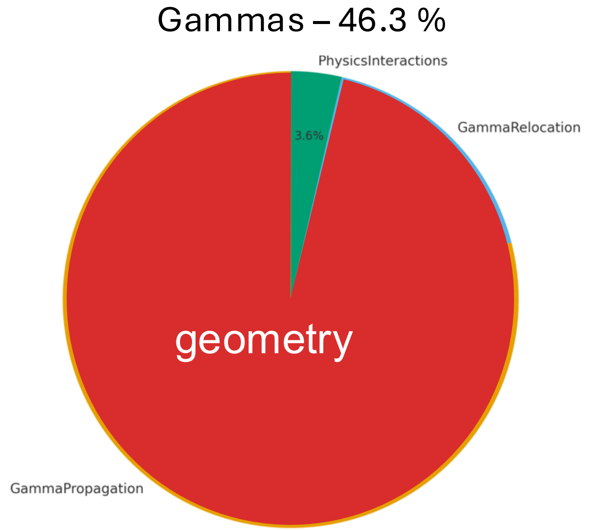
- **emstandard_option0/1/2**
  - Different options can be enabled per region in the same setup base on a flag
  - Extension to support high precision electron transport possible, but currently not a priority
- Other extensions also possible/foreseen, some with high impact on GPU
  - optical photons
  - low energy neutron elastic scattering
- A focused discussion with the main developer (M. Novak) on the space applications requirements is advisable

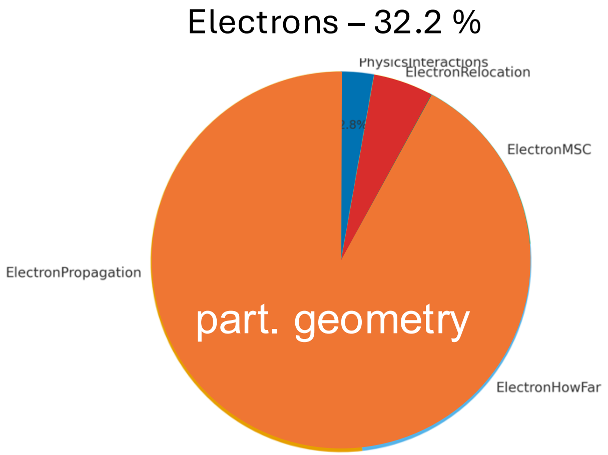
VecGeom

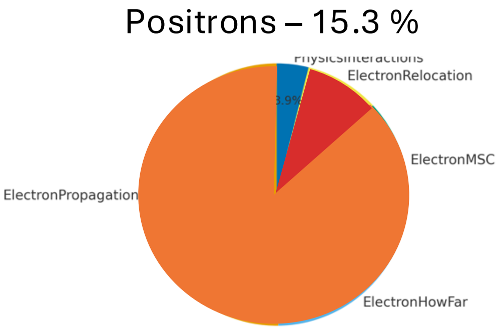
- Supports:
  - primitive solids
  - Boolean solids
  - tessellated solids
- Runs on both CPU and GPU (CUDA), and as backend for Geant4 solids
  - Being restructured focusing on GPU
- Geometry is currently the main bottleneck in complex HEP GPU simulations

Geometry is a bottleneck in complex GPU simulation

~50% pure geometry kernel
~40% partial geometry kernel







This involves propagation in magnetic fields in a HEP setup, the piechart may look quite different for space applications

Tessellations

- Tessellations are faceted representations of objects (triangles/quads)
  - Coming usually from CAD-based modeling
  - A « fast » path from engineering to simulation models
    - **used** in fast detector prototypes in HEP (e.g for FCC), **avoided** in production
    - **base-line** description in space/medical applications
- Tessellations are slow on CPU, but also on GPU
  - Even accelerated, they scale at best like *log(N)*
  - If accuracy is a « must » (option 4), non-planar surfaces must have fine grain tessellations
    - O(100K) facets or more
  - General advice we give for HEP application: avoid whenever possible
- **We are planning optimizing tessellations for GPU in VecGeom**

Space Radiation Applications: Transport Requirements

Application	Particles	Energy range	Transport characteristics
TID	e^-, p	keV – 500 MeV	large volumes, dose scoring
TNID / DDD	p, n, ions	100 keV – 500 MeV	nuclear interactions, secondaries
Internal Charging	e^-	keV – MeV	deep penetration, thin layers
SEU	p, ions	MeV – GeV	rare events, microscopic targets

- Complex CAD geometries (geometry navigation critical in many cases)
- Low-energy electron transport important
- Hadronic transport important for some applications

Does it makes sense to accelerate on GPUs?

- On you to probe, but **yes**, the EM component in many space applications is important (in particular, if bound to option4)
  - If EM is dominant, the theoretical speed-up can be (tens of) factors
- **AdePT can be used for fast testing**
  - Easy integration in your physics list
    - Still to understand requirements and adapt to a specific application
  - To check if scoring is compliant with parallel tracking
    - In general, friendly for pure accumulators
  - Can give you a first estimation on the potential gain
    - Even if not supporting **emstandard_option4**
- Priority candidates: **TID, Internal charging**

Conclusions

- Simulating on GPUs makes probably sense in at least a few space applications
  - A « potential benefit » prioritized list can be derived
  - Conditions and requirements need to be clarified first
  - Geometry becoming a bottleneck may depend on the electron transport settings
- AdePT is NOT Geant4 in full generality, but can be easily tried out
  - Using lower precision electron transport to do a first assessment
  - Possibly adapting models to stricter precision requirements in future
  - **We are very happy to assist if you need any advice/help with the integration**
- Gains may come from multiple optimizations, including or not GPU
  - Limit the use of tessellations
  - Tune the physics lists and cuts, combine GPU with biasing/reverse models

Questions (to you)

- What is the particle « profile » (species, energy ranges) in your specific applications
  - In particular the ratio EM/hadronic in the CPU budget
- What is the weight of scoring in the transport budget ?
  - Is it caching step information (is it parallel tracking friendly) ?
- Is ***emstandard_option4*** required in ALL regions ?
  - e.g. result of studies of impact of physics lists & cuts on the main observables
- Do you use large tessellations and are they hierarchical ?
- A more focused discussion among experts may be needed.