OAK RIDGE National Laboratory

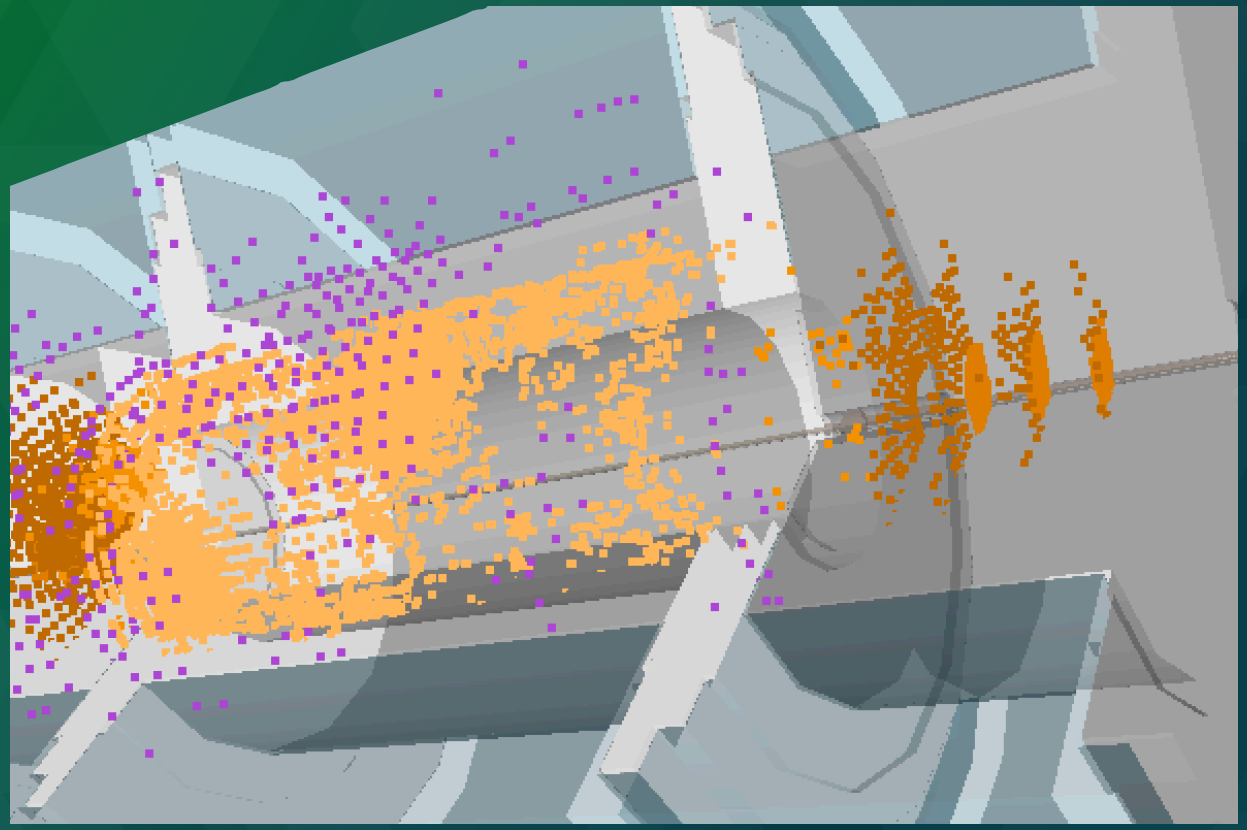
HP MC RAD SIM WORKSHOP

# GPU simulation for space applications

Seth R Johnson

*with*

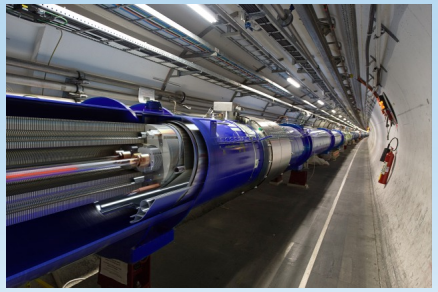
The Celeritas team



U.S. DEPARTMENT OF ENERGY

ORNL IS MANAGED BY UT–BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

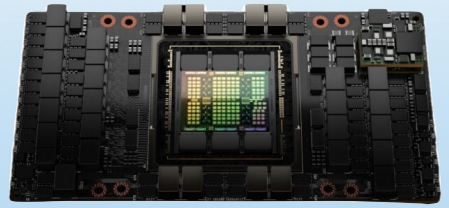
# Celeritas will accelerate HEP discoveries through GPU-powered simulation



*LHC beamline* ©CERN

- **GPU** optimized, CPU reproducible
- **Full-fidelity** Monte Carlo detector simulation
- **Automated** Geant4 integration (geometry, physics, hits)
- **Open scientific software** with active, collaborative development

CELERITAS



*Nvidia H100 GPU @Nvidia*

*Informal core collaborators*

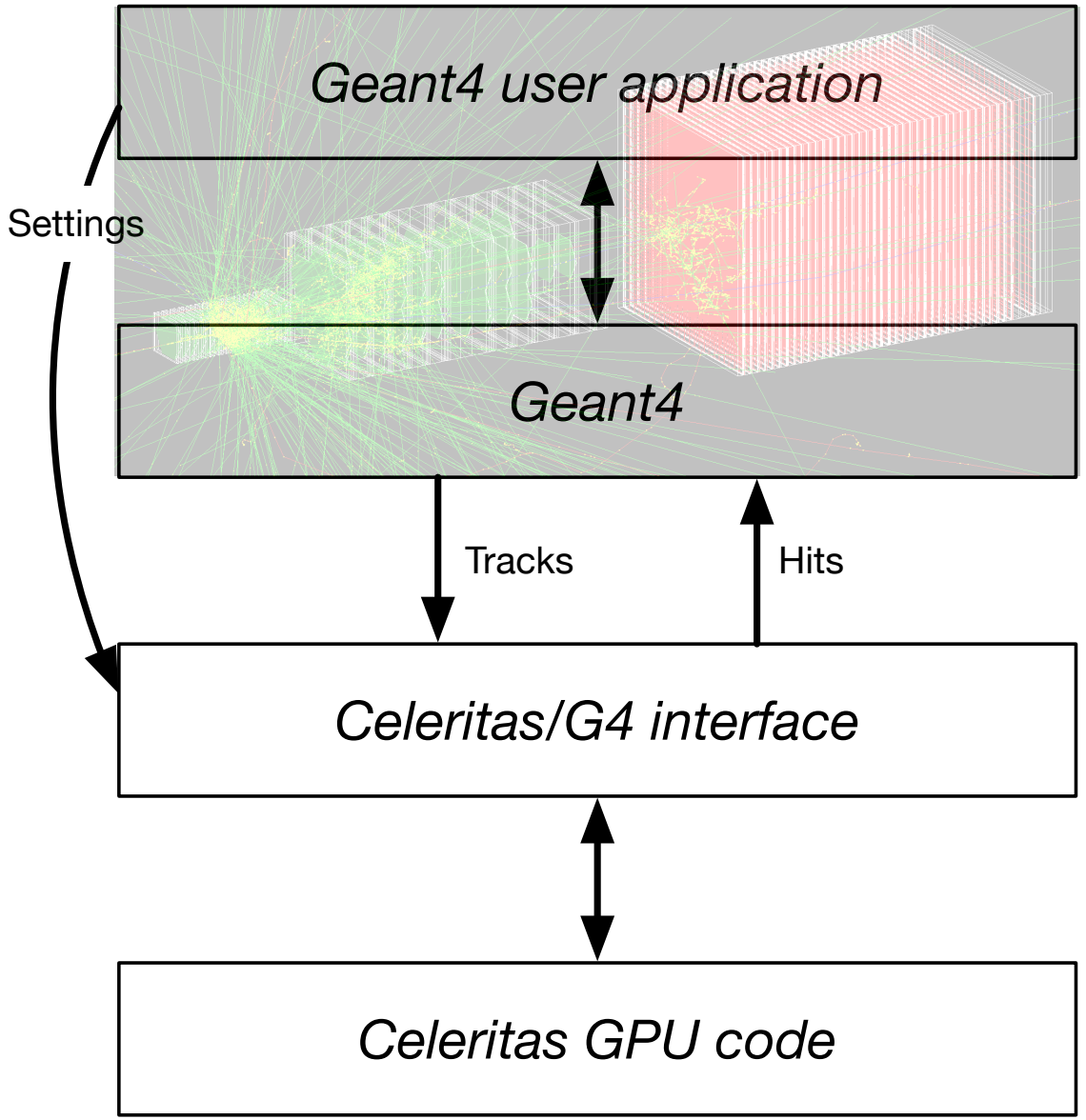
# Motivated by computing need *and* GPU availability

- High-fidelity Monte Carlo detector simulations required for:
  - Detector design optimization and calibration
  - Systematic uncertainty characterization
  - Background subtraction
  - Event reconstruction
- Flagship experiments massively increase sensitivity and data taking
  - LHC (energy frontier): increasing luminosity and pileup
  - DUNE (intensity frontier): unprecedented neutrino hit rates
  - LZ (cosmic frontier): rare-event searches
- Hardware landscape evolution is optimizing for AI/ML
  - Higher-memory GPUs (**good** for bandwidth-intensive Monte Carlo applications)
  - Lower-precision arithmetic (**bad** for general scientific applications)
  - **Training lower-order models still requires high-fidelity input datasets**

# Core capabilities

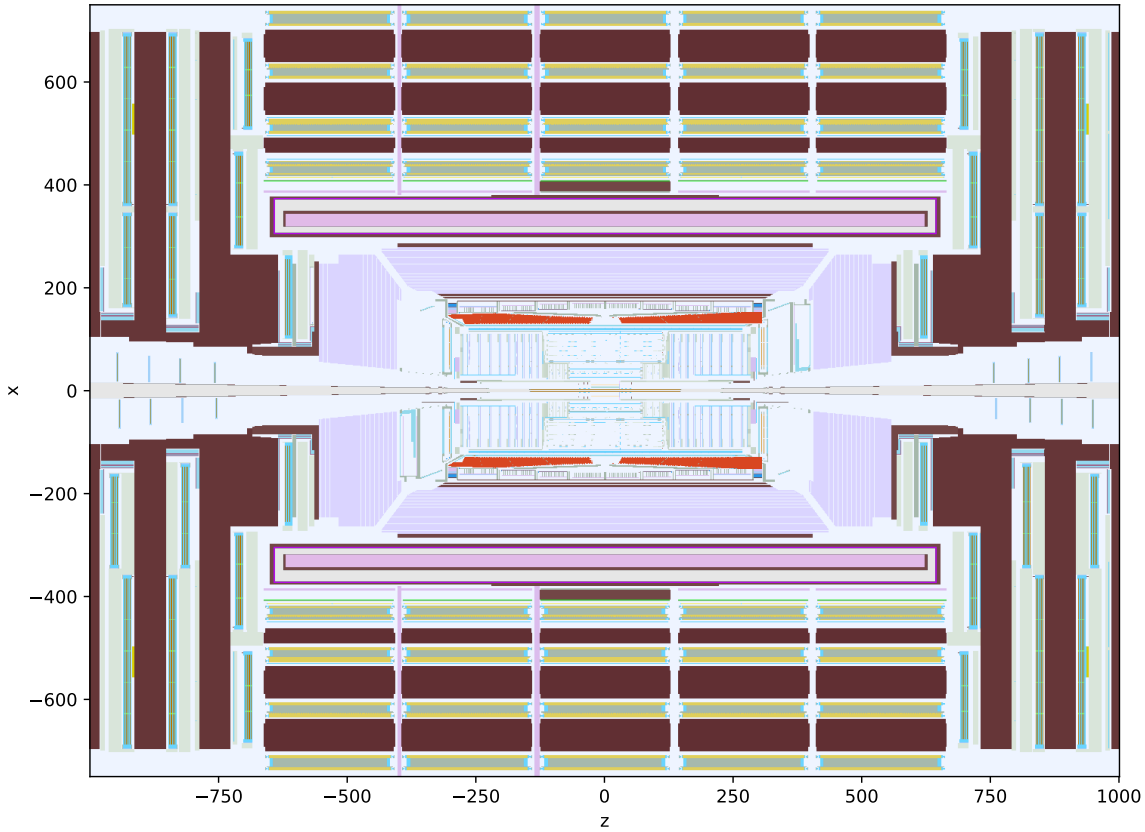
# Geant4 integration

- ~20 lines to integrate into user apps
  - Supports Geant4 10.5–11.3
  - Integrates as a plugin to G4-based simulators DD4HEP, LArSoft
- Celeritas directly imports G4 data
  - Physics options and cross sections
  - Geometry translation to VecGeom/ORANGE
- $e^-$, $e^+$, γ sent to Celeritas (GPU)
  - Tracking manager or user action offload
  - Tracks queued till buffer capacity or end of event
- Reconstituted Geant4 steps *(energy deposition, MC truth data, etc.)* sent back to user-defined detectors from GPU



# High-level capabilities initially targeted LHC simulation

- Celeritas physics is equivalent to `G4EmStandardPhysics`
- Full-featured Geant4 detector geometries using VecGeom 1.x (solid) and 2.x (surface)
- Runtime selectable processes, physics options, arbitrary field definition
- Execution on CUDA (Nvidia), HIP* (AMD), ***and CPU*** devices




*GPU-traced "material scan" of CMS 2018 demonstrating geometry integration*

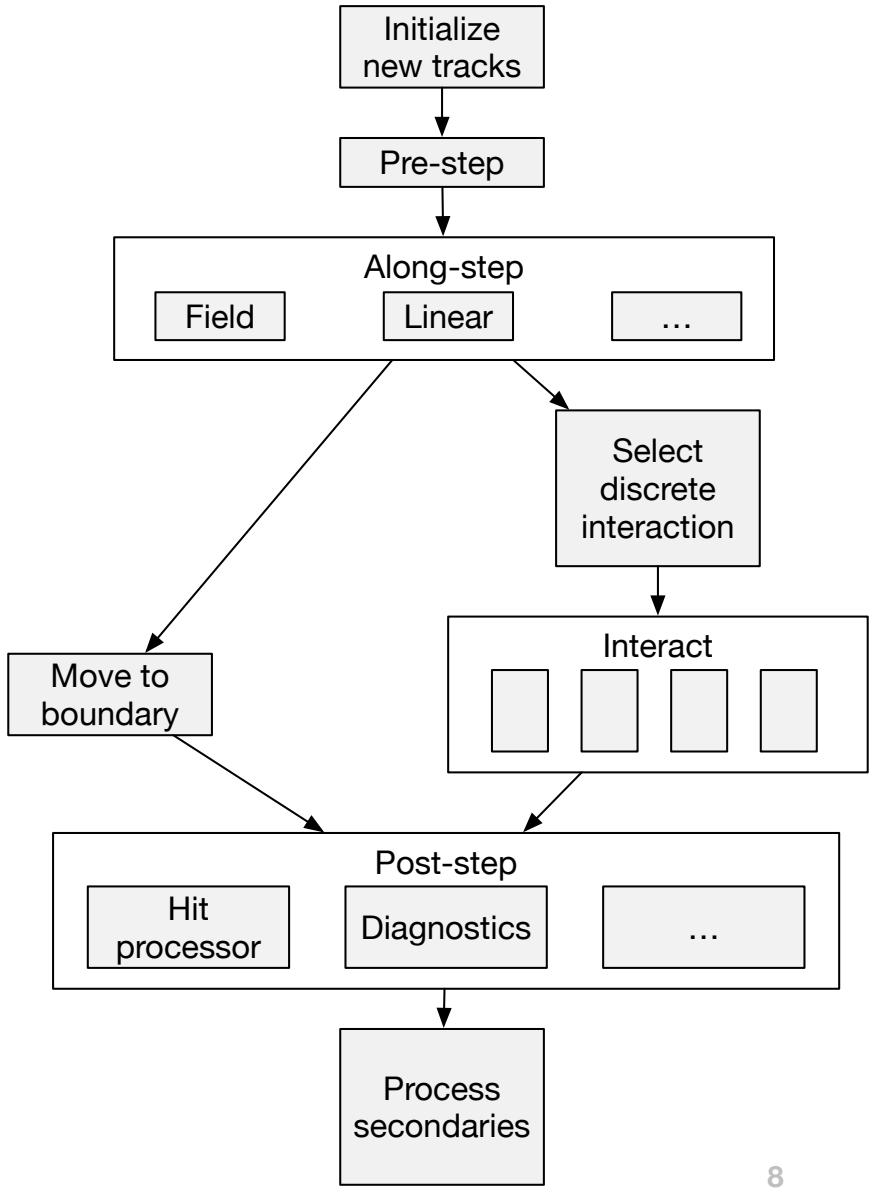
**VecGeom currently requires CUDA: navigation required for HIP*

# Designed to support multiple use cases

| Capability | Use case | Implementation |
|---|---|---|
| **Execution hardware** | Traditional hardware, debugging | CPU |
| | Single-thread G4 with spare CPU cores | OpenMP |
| | Nvidia GPU | CUDA |
| | AMD GPU | ROCm |
| **Geometry** | Photon tracking, reactor physics | ORANGE |
| | LHC detectors | VecGeom |
| | Geometry comparison | Geant4 |
| **Front end** | Local app, grid execution | Geant4 tracking manager |
| | Framework plug-in | DD4HEP, LArSoft |
| | HPC execution | Standalone app |
| **Unit system** | LHC detectors | CLHEP |
| | Photon tracking, reactor physics | CGS |

# Framework for arbitrary extensibility

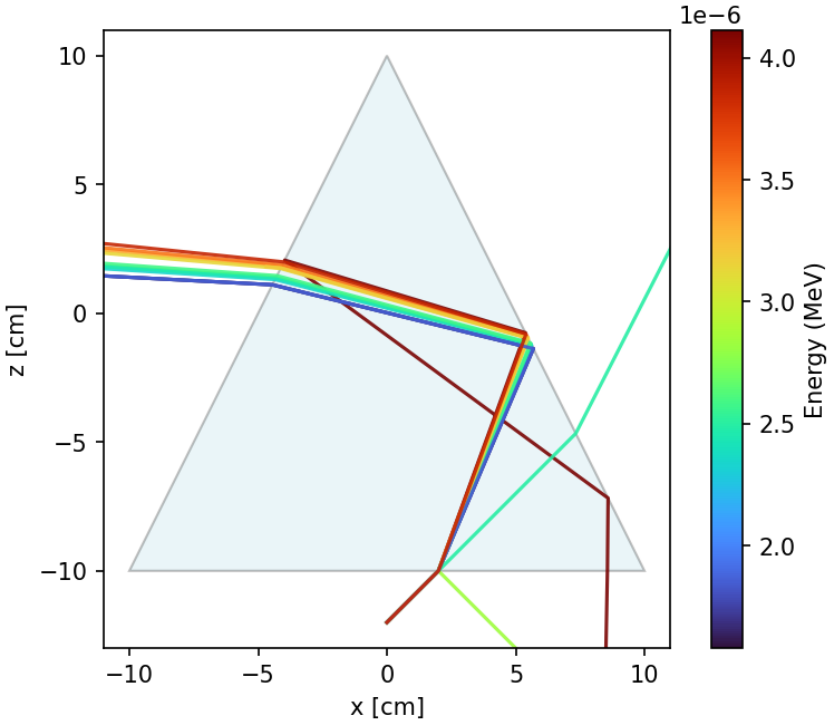
- "Actions" allow runtime GPU flexibility
  - Physics models: each operates on a subset of tracks
  - Propagation: different kernels for charged, neutral tracks
  - Sensitive detector "hit" callbacks
  - Diagnostics about slot occupancy, interactions, …
- Each action has *auxiliary per-track state*
  - Separate state group per "stream" (usually CPU thread)
  - Useful for buffers, persistent state, ...
- Actions/aux data can be added by:
  - Developers to provide new physics and capabilities
  - User for custom physics, "stepping action", and beyond

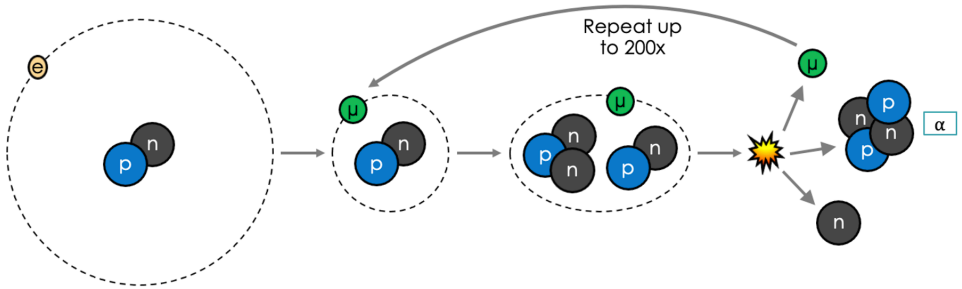



***Not just single-purpose, hardcoded physics***

# GPU physics capabilities beyond LHC

- Optical physics for DUNE experiment and SiPM detector R&D
- Neutron elastic scattering and inelastic XS calculations
- Muon EM physics, decay, and muon-catalyzed fusion
- Surface-based geometry representation (ORANGE)



# Physics capabilities

| Electromagnetic | | |
|---|---|---|
| **Particle** | **Process** | **Model(s)** |
| $\gamma$ | Photon conversion | Bethe-Heitler |
| | Compton scattering | Klein-Nishina |
| | Photoelectric effect | Livermore |
| | Rayleigh scattering | Livermore |
| $e^{\pm}$ | Ionization | Moller-Bhabha |
| | Bremsstrahlung | Seltzer-Berger, relativistic |
| | Multiple scattering | Urban (extended to high E) |
| | Coulomb scattering | Wentzel |
| $e^{+}$ | Pair annihilation | Positron to two gammas |
| $\mu^{\pm}$ | Bremsstrahlung | Bremsstrahlung |
| | Pair production | Muon pair production |
| | Multiple scattering | Urban |
| | Ionization | Bethe-Bloch |
| $\mu^{+}$ | Ionization | Bragg |
| $\mu^{-}$ | Ionization | ICRU73QO |
| | Muon-catalyzed fusion | Knaian/Lynch/Tripathy |

| Optical physics |
|---|
| Cherenkov |
| Scintillation |
| WLS |
| Rayleigh scattering |
| Absorption |
| Mie scattering |
| GLISUR, UNIFIED |

| Additional physics |
|---|
| Neutron elastic scattering |
| Particle decay |
| X-ray/auger cascade |

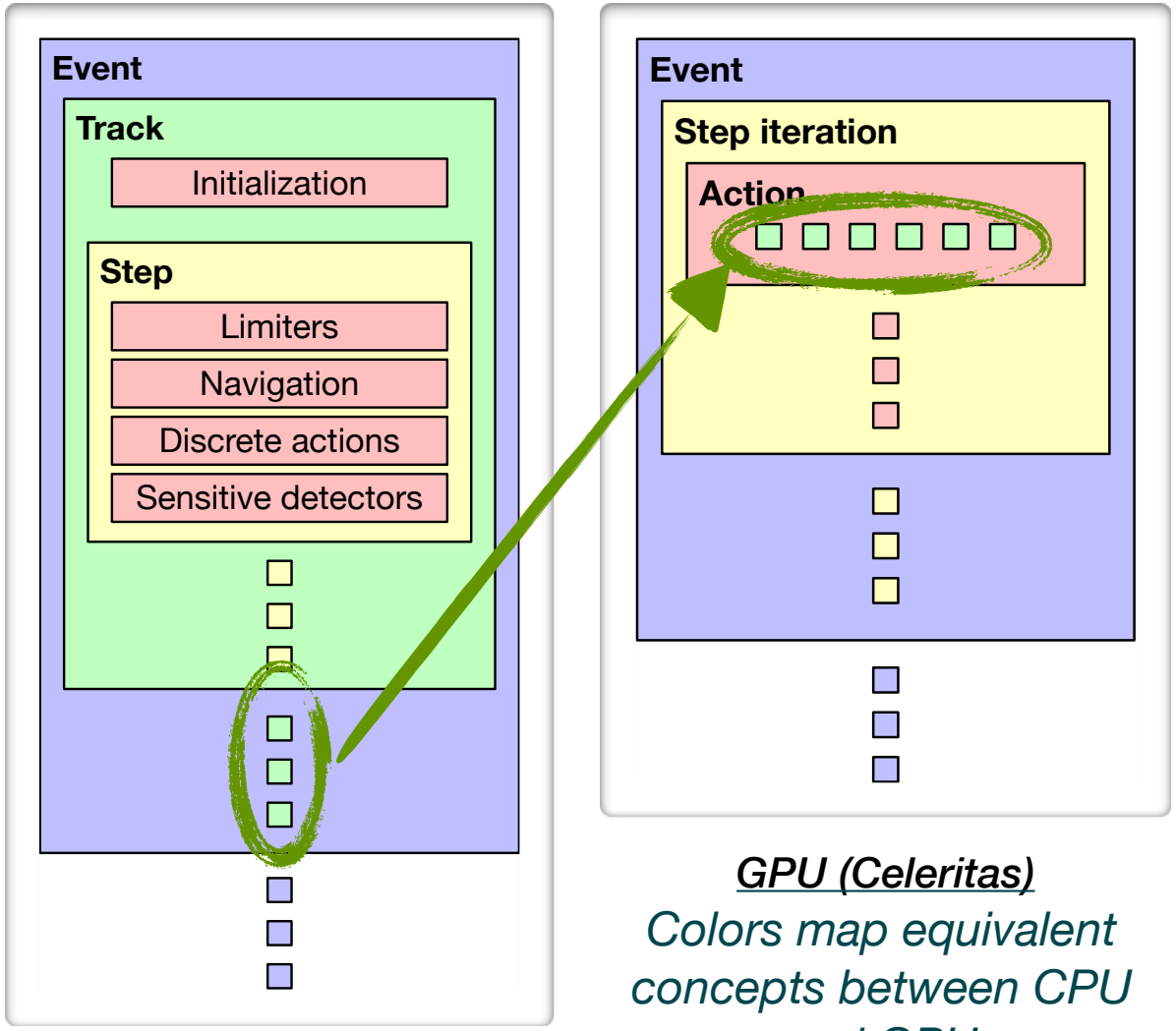
# Magnetic field propagation

- Composition based: **P○S○I○E○F**
- Numerous built-in options to select from
  - Field representations: nonuniform cylindrical, uniform cartesian
  - Equations: Lorentz (magnetic)
  - Integrators: Runge–Kutta 4 and Dormand–Prince RK5(4)7M
- Template interface allows trivial extension
  - Electric fields
  - Custom analytic field representations

| Operator | Input | Output |
|---|---|---|
| **F**ield | $\boldsymbol{x}$ | $\boldsymbol{B}$ |
| **E**quation of motion | $\boldsymbol{x}$, $\boldsymbol{p}$, $\boldsymbol{B}$ | $\boldsymbol{x}'$, $\boldsymbol{p}'$ |
| **I**ntegrator | $\boldsymbol{x}$, $\boldsymbol{p}$, $h$ | $\boldsymbol{x}^*$, $\boldsymbol{p}^*$, $e$ |
| **S**ubstepper | $\boldsymbol{x}$, $\boldsymbol{p}$, $s$ | $\boldsymbol{x}^*$, $\boldsymbol{p}^*$, $s^*$ |
| **P**ropagator | $\boldsymbol{x}$, $\boldsymbol{\Omega}$, $E$, $s$ | $\boldsymbol{x}^*$, $\boldsymbol{\Omega}^*$, $s^*$ |

# Core algorithm for simulation: stepping loop

- Dependency between **steps** and independence of **tracks** allows *loop interchange*
- Synchronization point at each **event**
  - **Compatible with Geant4 and experiment frameworks** (require one event per thread)
  - Some framework code *(e.g., sensitive detectors)* assumes serialization of **tracks** as well
- Polymorphic **action** applies to a *vector* of tracks, not just a single track



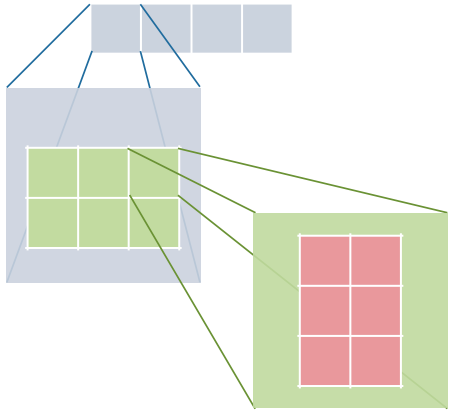

***CPU (Geant4)***

***GPU (Celeritas)***

*Colors map equivalent concepts between CPU and GPU*

# Adding a new process/model on GPU

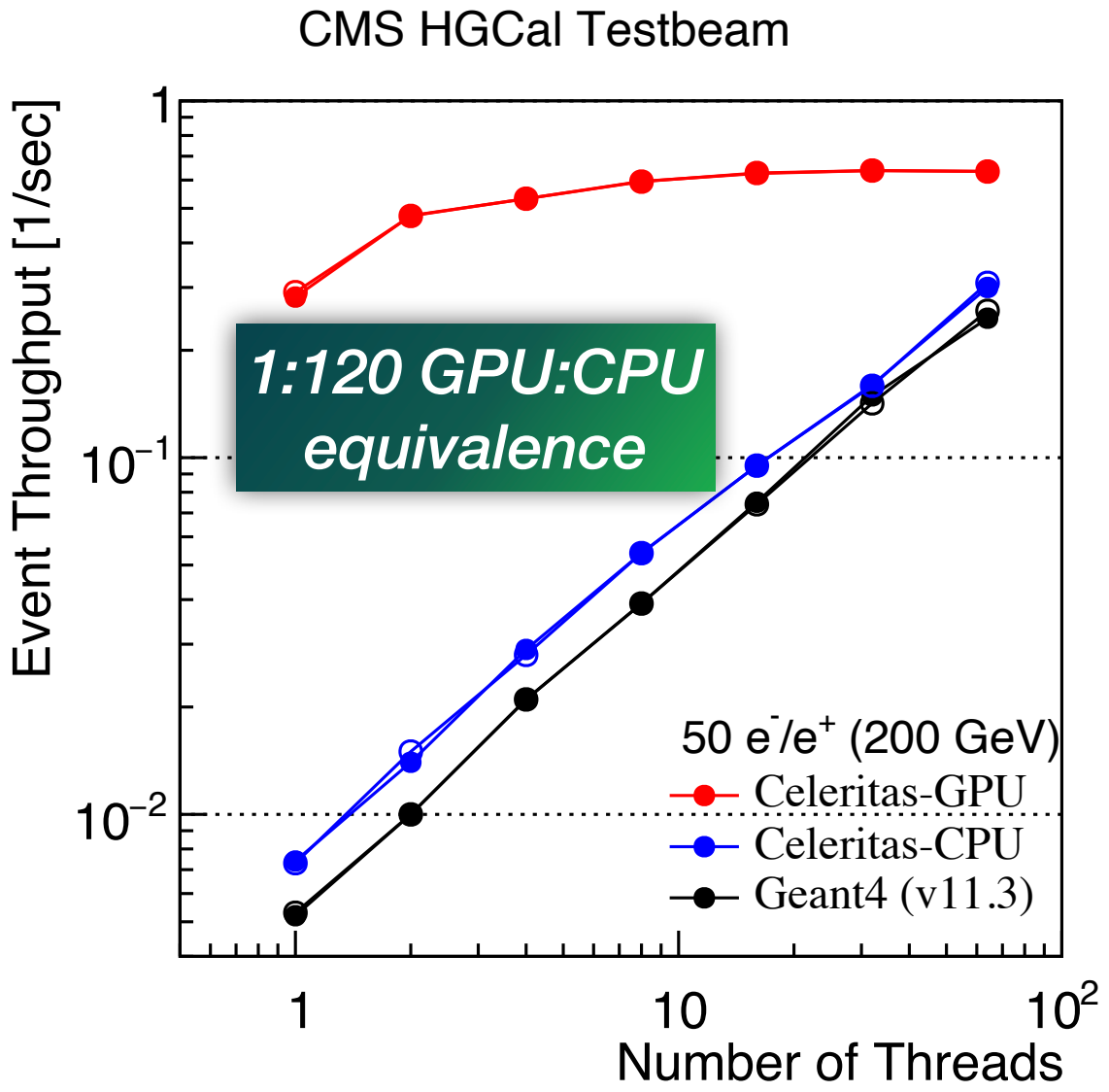
- Geant4 code is *not* compatible with GPU
  - All memory lives on CPU, most functions incompatible with “device” decorator
  - Single local CPU thread per process instance mixes *physics parameters* and *track state*
  - Heavy use of virtual functions and dynamic memory allocation
- But cross sections and algorithms usually are
  - Much data can be loaded directly from Geant4 physics processes
  - Celeritas has tools to set up for host/device execution
  - “Interactor” encapsulates post-step “do it” with track state change
- Use Celeritas “views” for data access
  - Local *track state* data indexes into global *physics parameter* data
  - Type-safe integer IDs rather than pointers
  - IDs: particle, model, process, volume, material, element, isotope, …

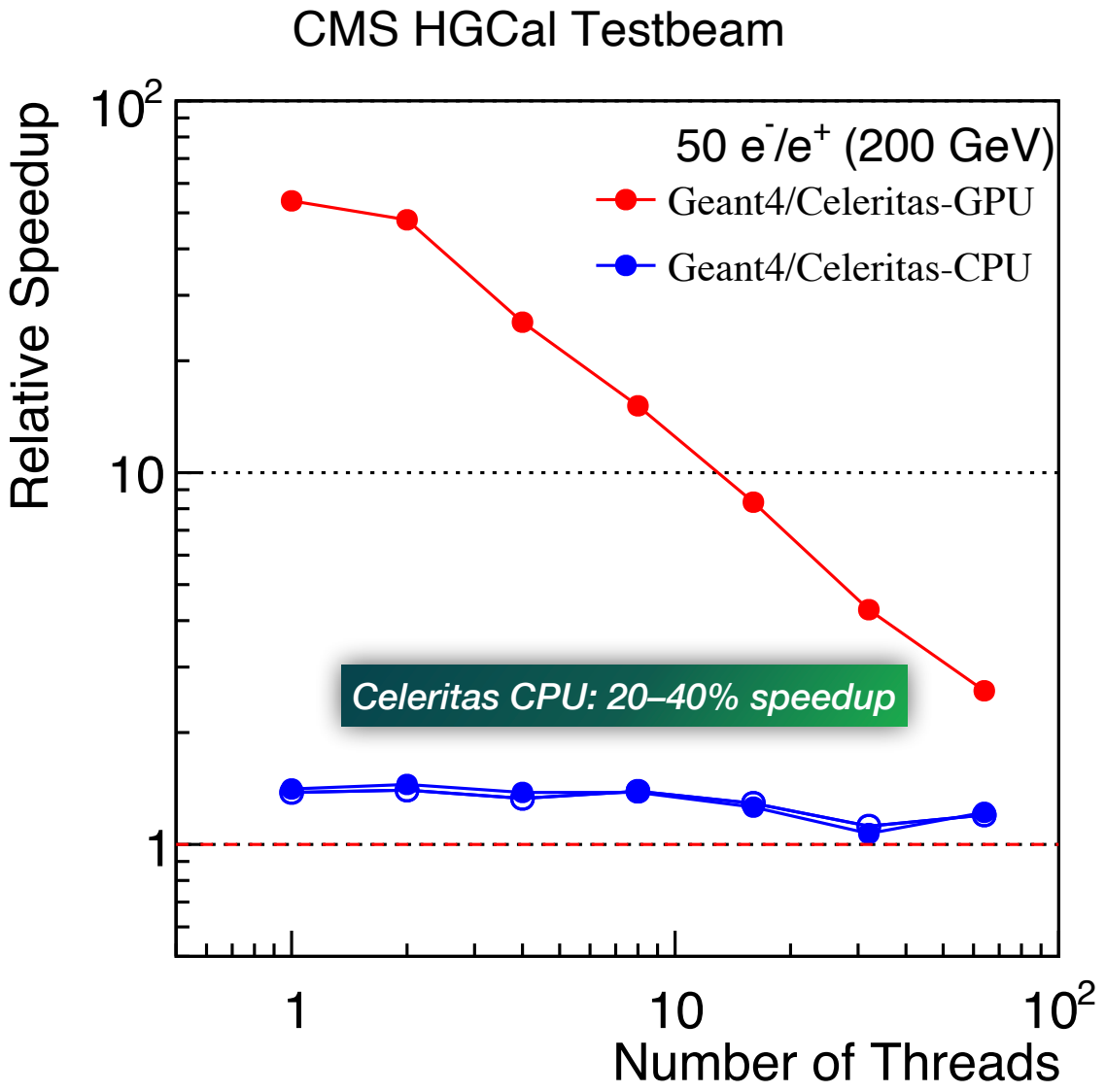


*Monte Carlo data*

# Results

# GPU performance via Geant4 app



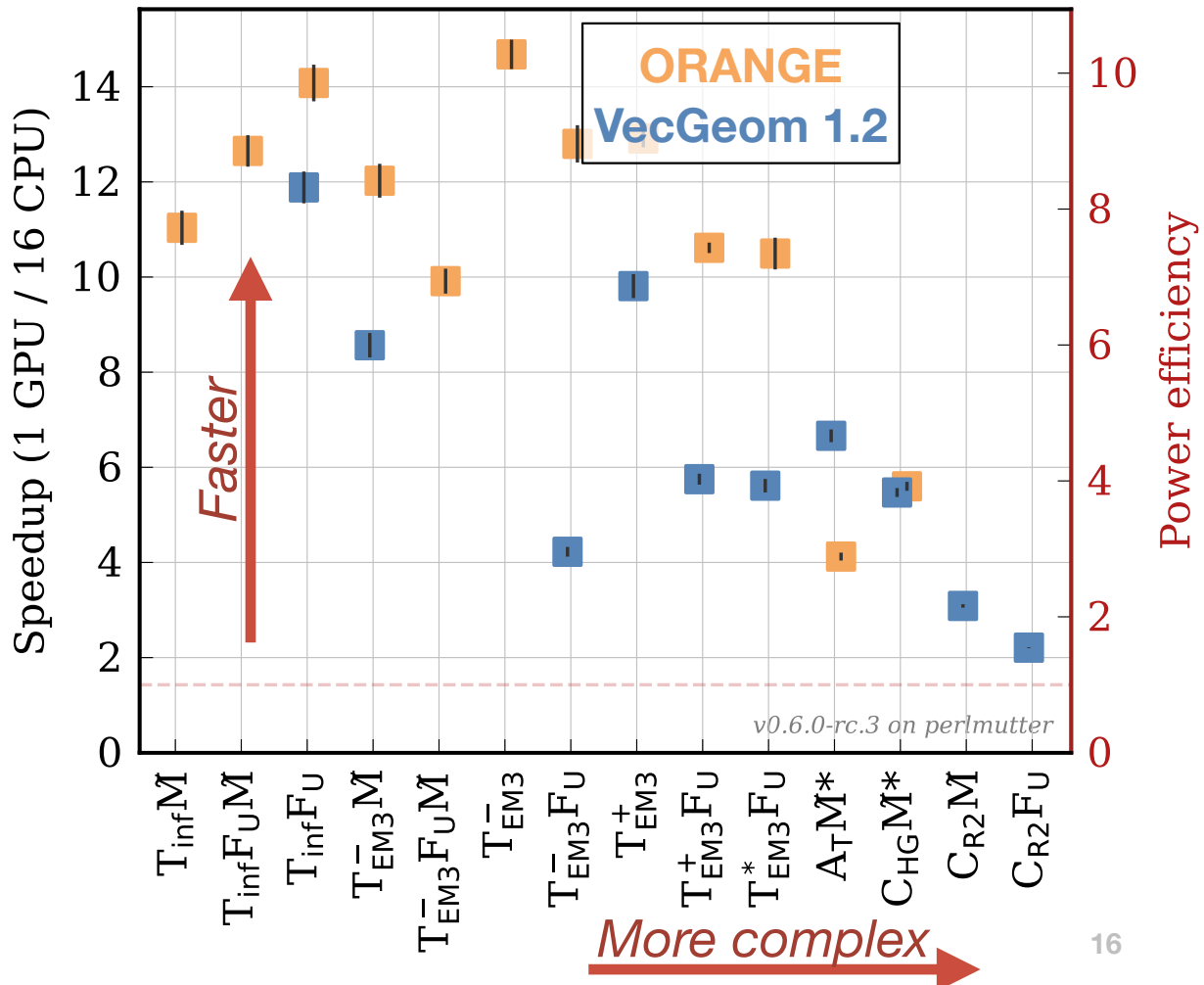



# Standalone EM performance

- Maximum performance test
  - Pure EM problem, no Geant4 driver, no SDs
  - All primaries share a single GPU stream
  - CPU reference performance is **also** Celeritas
- LHC-scale simulations on DOE LCF
  - 1300 × 10 GeV $e^-$, 16 events
  - ¼ Perlmutter node (NERSC)
    *1 × Nvidia A100 GPU, ¼ × 64-core AMD EPYC 7763*
  - Celeritas GPU vs CPU
    *CUDA (1 CPU thread) vs OpenMP (16 CPU threads)*
- **Key metrics favor GPU**
  - *Capacity:* **55–93% loss** if GPUs are ignored
  - *Efficiency:* up to **10×** performance per watt
    - Real-world A100 power consumption: ~100W
    - Plus 1 CPU core to drive: ~5W

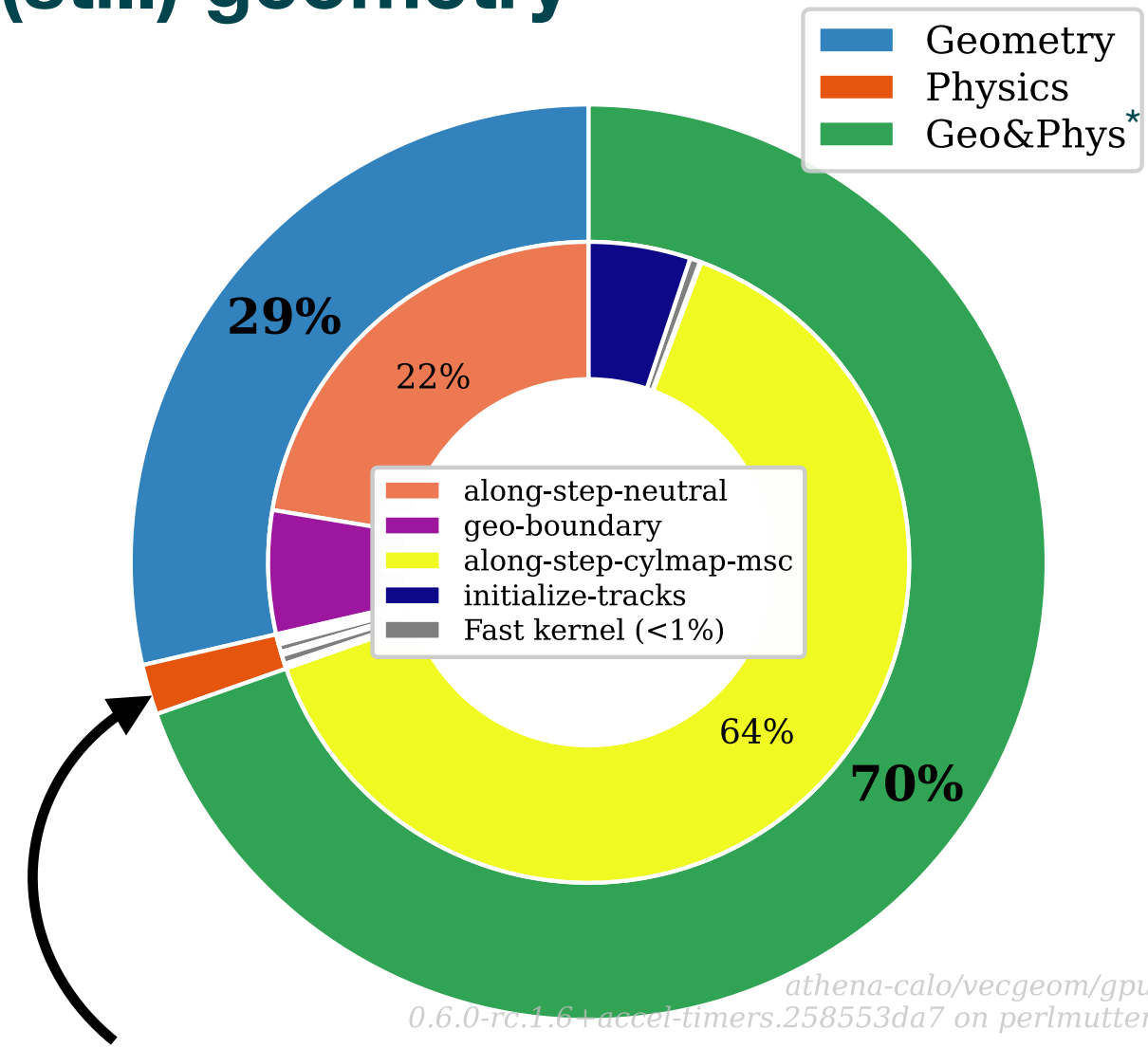
| *Problem definition* | |
|---|---|
| $T_{inf}$ | "Infinite" medium |
| $T_{EM3}$ | Idealized calorimeter |
| $A_{TC}$ | ATLAS TileCal |
| $C_{HG}$ | CMS HGCal |
| $C_{R2}$ | CMS Run 2 (2018) |

| *Modifier* | |
|---|---|
| $F_U$ | uniform 1T field |
| $\tilde{M}$ | no MSC |



# EM performance bottleneck is (still) geometry

- Example shown with VecGeom
  - ATLAS EM calorimeters (tile, end cap, barrel)
  - Full-fidelity magnetic field
  - 36.5K unique volumes, 1.84M total elements
- Sub-optimal on GPU
  - Underlying problem: **random access, lots of data**
  - Double-precision required for accuracy
    - $>10^7$ relative scale: μm steps, 10 m detector
    - $\sqrt{1-z^2}$ in direction change, quadratic equations
    - Accumulation of very small mag. field steps
- Performance optimization in progress:
  - Field propagation and MSC algorithms
  - VecGeom integration
  - ORANGE development
- **CAD-based geometry known to be well suited for GPU**



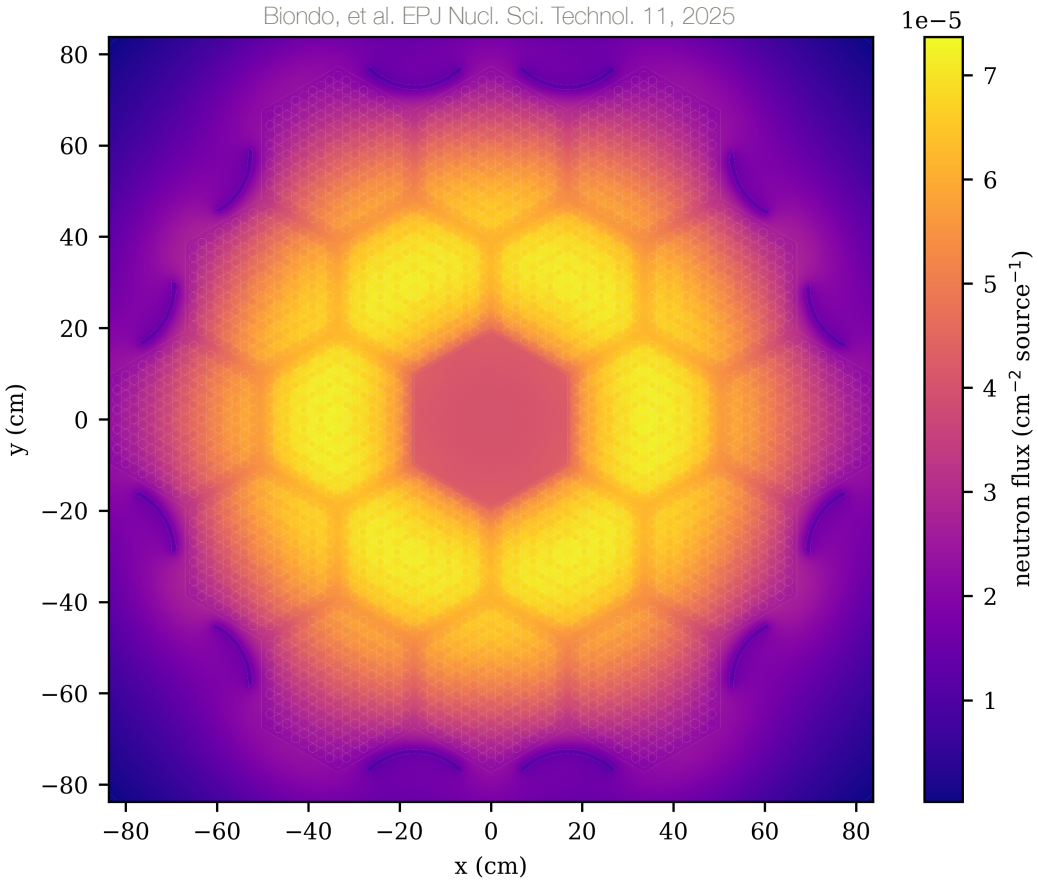

*Physics models are 1% on GPU*

*G&P (field propagation, MSC safety) uses both geometry and physics

# Discussion

# Some use cases will be more performant on GPU

- GPU generally has **good weak scaling**
  - More histories → more work to do → fuller pipeline
  - More histories → better statistics
- Integral results are desirable for HPC
  - MPI allows in-memory reduction
  - Reduced I/O bandwidth
- **Dosimetry applications** will do well
  - Celeritas includes basic integral energy deposition tallies
  - Plans for general on-device scoring with MPI reductions *(celeritas#1752)*
  - Shift MC GPU application demonstrated 2–5× on-node speedup on Frontier




Neutron flux using Shift with Celeritas ORANGE geometry on GPU

# Celeritas is extensible to space applications

- New particles are trivial
  - Protons and neutrons coexist with electrons, gammas in particle state vector
  - Data imported through Geant4 class interfaces
- New physics is feasible but will require work: some easy, some hard
  - Easy to add new models to the stepping loop
  - Expanded EM physics needed for low-energy physics: probably easy
  - Some hadronic physics is in place
- **GPU is inherently less flexible than CPU**
  - But Celeritas framework tries to maximize flexibility using CPU-based building blocks
  - Some applications show superior performance per watt
  - **If the hardware is there, we should use it**

*Celeritas v0.7 code contributors:*

- Seth R. Johnson (@sethrj)
- Amanda Lund (@amandalund)
- Elliott Biondo (@elliottbiondo)
- Julien Esseiva (@esseivaju)
- Hayden Hollenbeck (@hhollenb)
- Philippe Canal (@pcanal)
- Stefano Tognini (@stognini)
- Rashika Gupta (@Rashika Gupta)
- Loren Schwiebert (@LSchwiebert)
- Soon Yung Jun (@whokion)
- Greg Davidson (@davidsgr)
- Ben Morgan (@drbenmorgan)
- Owen Strong (@osanstrong)
- Lance Bullerwell (@lebuller)
- Guilherme Lima (@mrguilima)
- Sakib Rahman (@rahmans1)

***SciDAC***: This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics, Scientific Discovery through Advanced Computing (SciDAC) program.

***NERSC***: This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award HEP-ERCAP-0023868.

***HEP-CCE***: This work was supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, High Energy Physics Center for Computational Excellence (HEP-CCE) under B&R KA2401045.

*Celeritas core team:*

- Elliott Biondo *(ORNL)*
- Greg Davidson *(ORNL)*
- Julien Esseiva *(LBNL)*
- Rashika Gupta *(ORNL)*
- Hayden Hollenbeck *(UVA)*
- Seth R Johnson *(ORNL)*
- Soon Yung Jun *(FNAL)*
- Guilherme Lima *(FNAL)*
- Amanda Lund *(ANL)*
- Ben Morgan *(U Warwick)*
- Sakib Rahman *(BNL)*
- Stefano Tognini *(ORNL)*

*Celeritas core advisors:*

- Tom Evans (ORNL)
- Philippe Canal (FNAL)

*Past code contributors:*

- Paul Romano (@paulromano)
- Vincent R. Pascuzzi (@vrpascuzzi)
- Tom Evans (@tmdelellis)
- Vidor Heli Lujan Montiel (@VHLM2001)
- Damien L-G (@dalg24)
- Doaa Deeb (@DoaaDeeb)
- Andrey Prokopenko (@aprokop)
- Shane Hart (@hartsw)
- Peter Heywood (@ptheywood)

***Code*** ***Documentation*** ***Publication***