Oak Ridge National Laboratory

ORNL is managed by UT–Battelle, LLC for the US Department of Energy

Full-featured Geant4 GPU acceleration using Celeritas

Seth R Johnson

Celeritas code lead

CELERITAS

Celeritas core team:

Elliott Biondo *(ORNL)*, Julien Esseiva *(LBNL)*, Seth R Johnson *(ORNL)*, Soon Yung Jun *(FNAL)*, Guilherme Lima *(FNAL)*, Amanda Lund *(ANL)*, Stefano Tognini *(ORNL)*

Celeritas core advisors:

Tom Evans *(ORNL)*,
Philippe Canal *(FNAL)*,
Marcel Demarteau *(ORNL)*,
Paul Romano *(ANL)*

U.S. DEPARTMENT OF ENERGY

SWIFT-HEP meeting
21 November, 2023

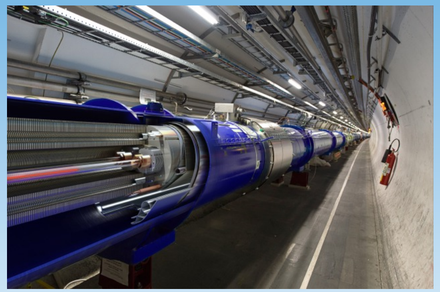
Background

Results

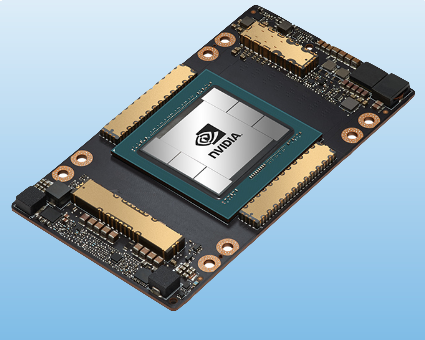
Conclusions

Project overview

- **GPU**-focused implementation of **HEP** Monte Carlo detector simulation
  - **Computing demand:** LHC-HL upgrade
  - **Computing supply:** Paradigm shift in HPC hardware
- Motivated by HL-LHC computational challenges *and* by recent success in GPU MC
  (*Exascale Computing Project [ECP] ExaSMR*)
- **Goal: accelerate production use for LHC Run 4**

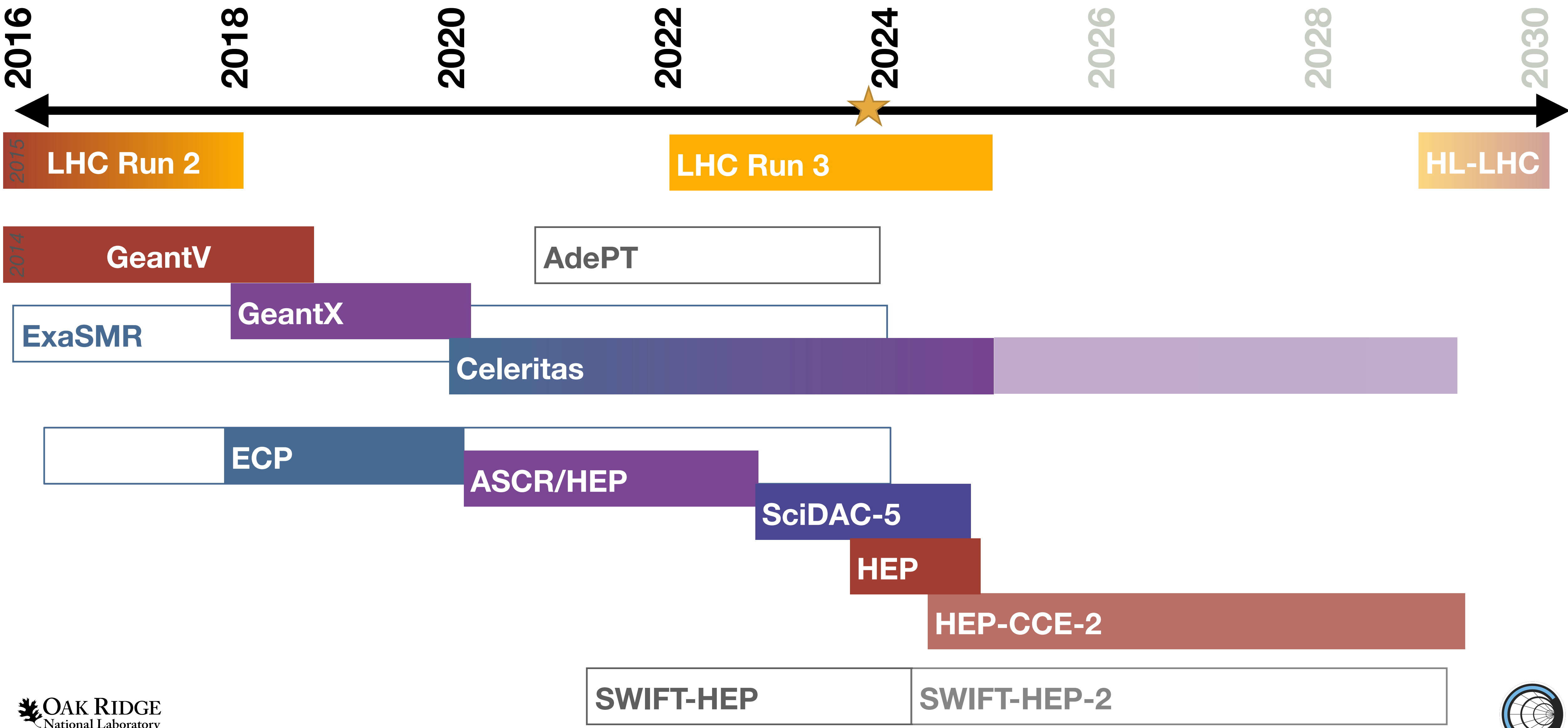


LHC beamline ©CERN



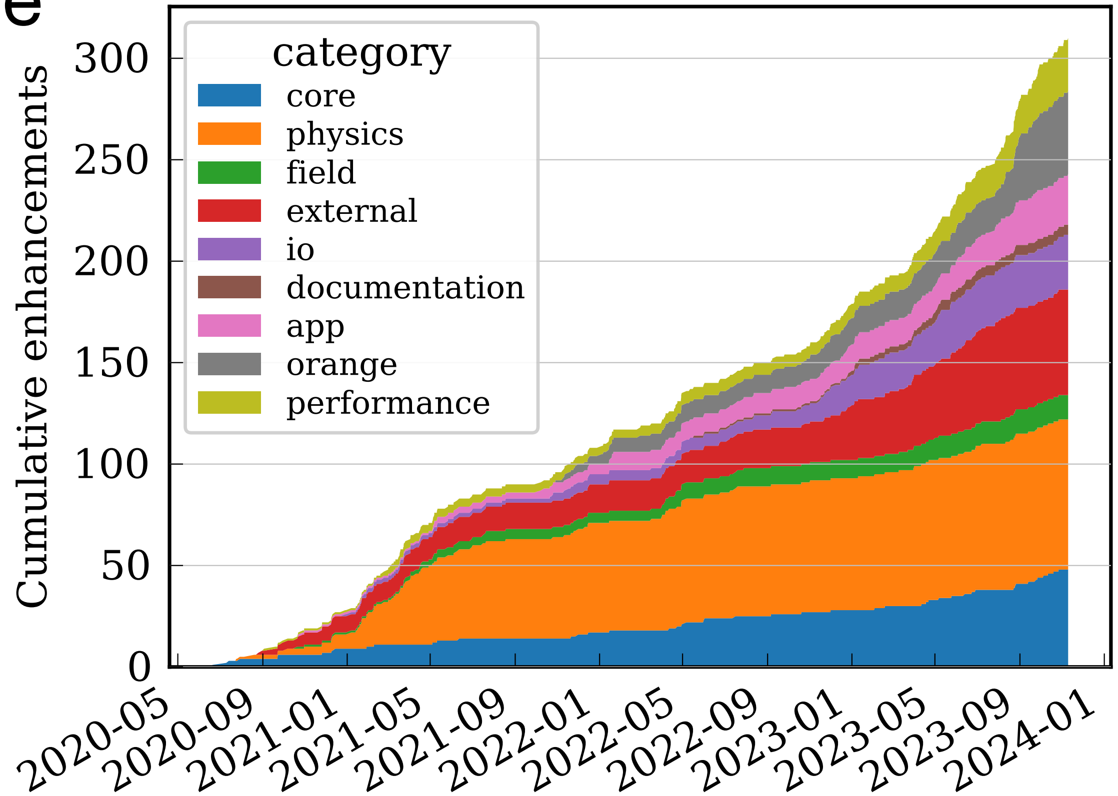
Nvidia A100 GPU (Nvidia)

Historical context



Code history and status

- Production-focused scientific software
  - **90%** of source code is reusable library code
  - **1:2** ratio of lines of documentation to code
  - **50k** lines of test code
  - User-extensible physics, particles, detectors, …
  - Code reviews, CI, Doxygen, Sphinx, ...
- Physics is only a part of the work
- Recent focus: Geant4 integration and performance optimizaton



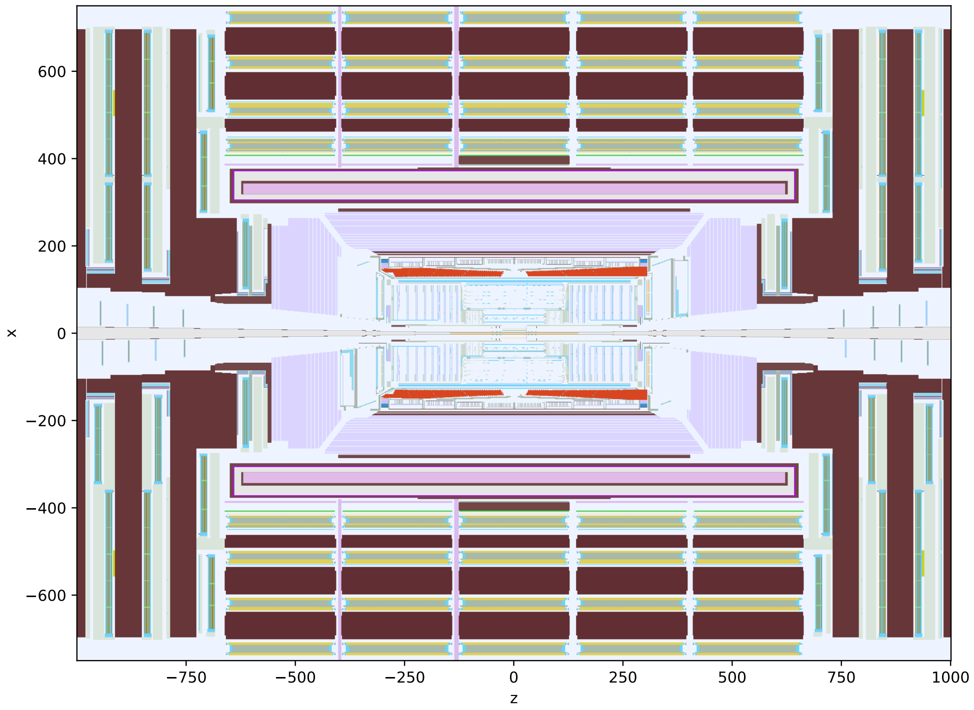
Background

Results

Conclusions

High-level capabilities

- Equivalent to `G4EmStandardPhysics`
  …using Urban MSC for high-E MSC; only γ, e±
- Full-featured Geant4 detector geometries using VecGeom 1.x
- Runtime selectable processes, physics options, field definition
- Execution on CUDA (Nvidia), HIP* (AMD), ***and CPU*** devices

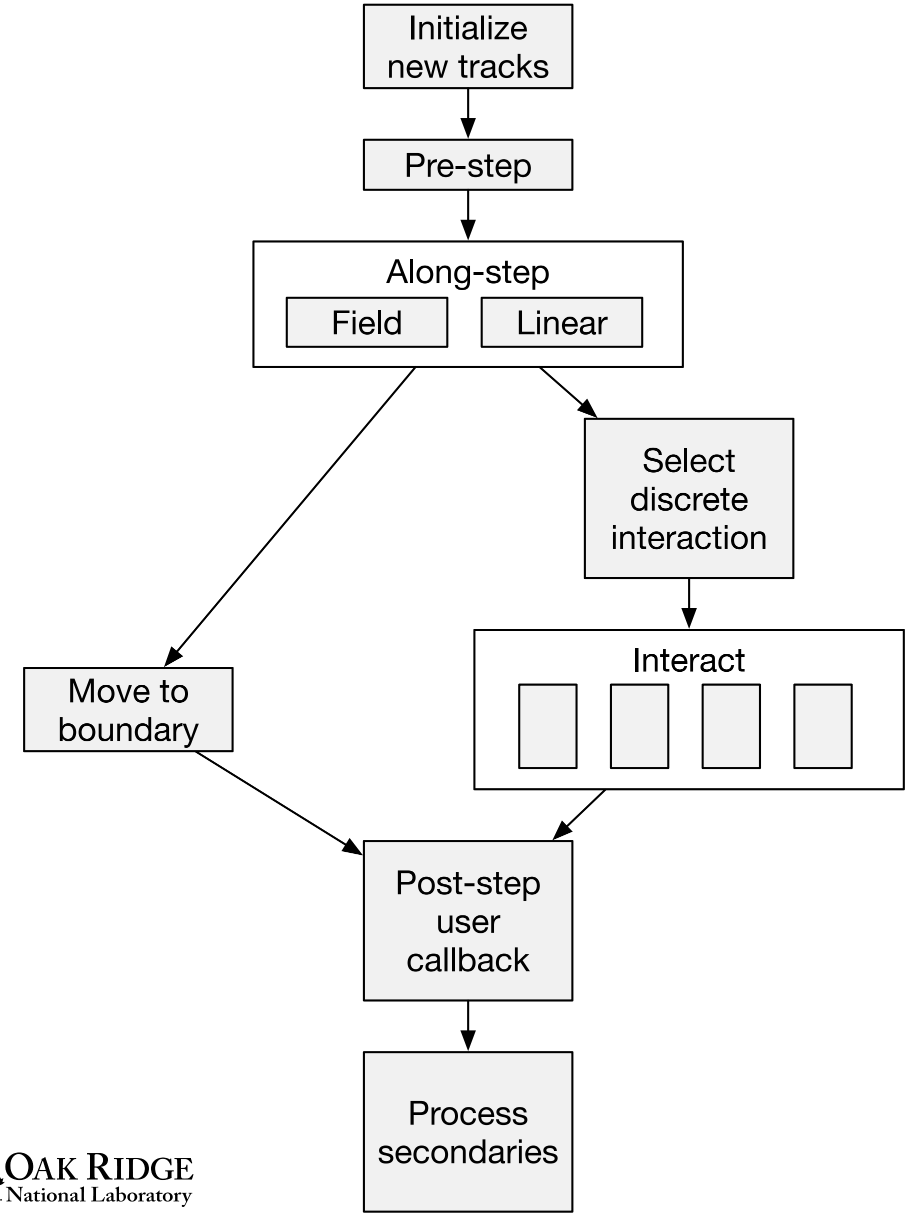


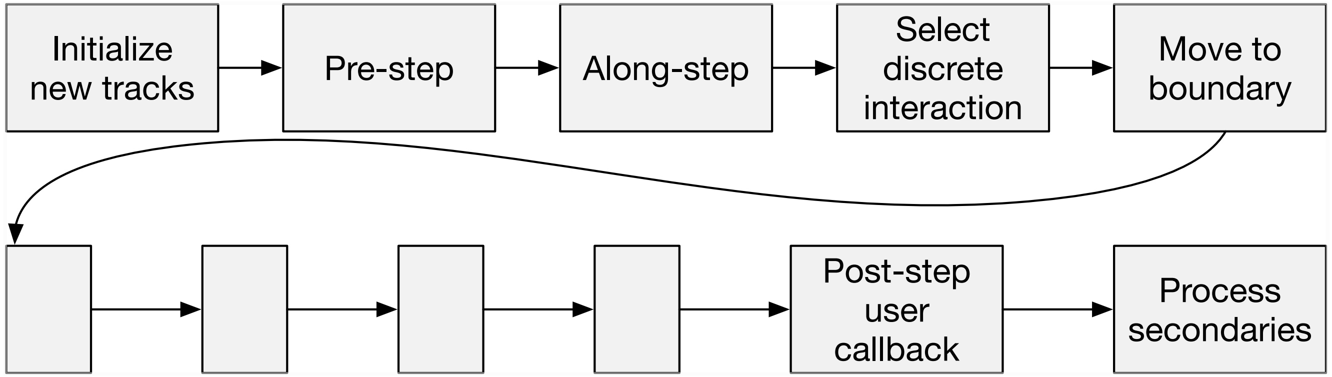
GPU-traced rasterization of CMS 2018

**VecGeom currently requires CUDA: ORANGE navigation required for AMD*

Source: Johnson, S.R. Geant4 Meeting 2023

Stepping loop on a GPU





Topological sort: a loop over kernels

Process large batches of tracks per kernel (10^3–10^6)

OAK RIDGE National Laboratory

Source: Johnson, S.R. Geant4 Meeting 2023

Celeritas version `0.4`: Geant4 integration status

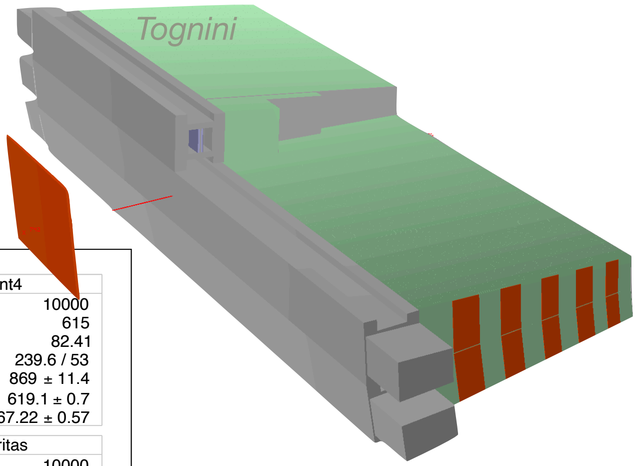
- **Imports** EM physics selection, cross sections, parameters
- **Converts** geometry to VecGeom model
- **Offloads** EM tracks from Geant4
- **Scores** hits to user "sensitive detectors"
  (Copies from GPU to CPU; reconstructs `G4Hit`, `G4Step`, `G4Track`; calls `Hit`)
- **Builds** against Geant4 10.5–11.1

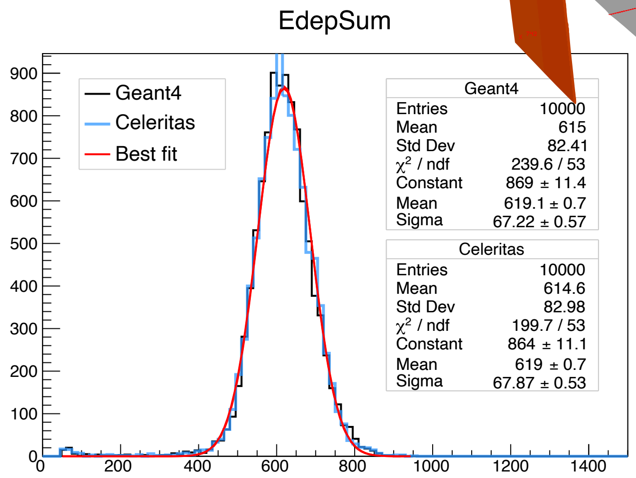
*Celeritas has **production quality** interfaces to simplify user application integration*

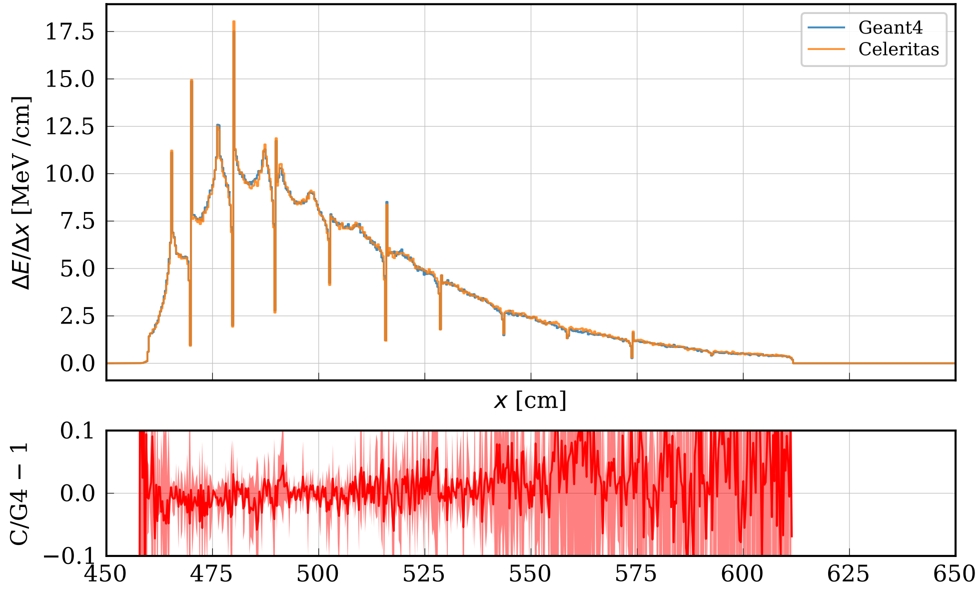
Source: Johnson, S.R. Geant4 Meeting 2023

EM offloading with FullSimLight

- ATLAS FullSimLight: hadronic tile calorimeter module segment
  - 64 segments in full ATLAS, 2 in this test beam
  - 18 GeV π^+ beam, no field
  - FTFP_BERT (default) physics list
    (includes standard EM)
- **~100 lines of code to integrate**
  - Offload e^-, e^+, γ to Celeritas
  - Celeritas reconstructs hits and sends to user-defined `G4VSensitiveDetector`
- Excellent agreement in energy deposition







yz-integrated energy deposition

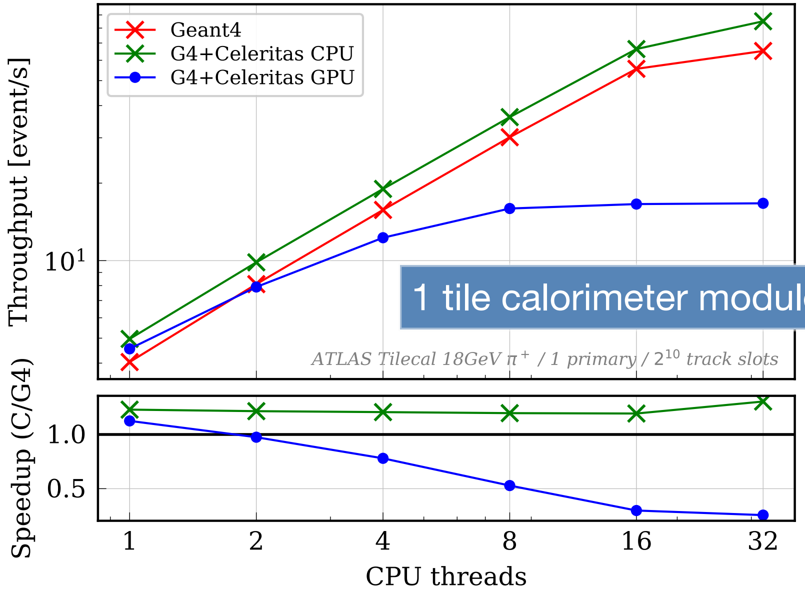
Test problem: Lachnit, Pezzotti; FSL integration: Morgan

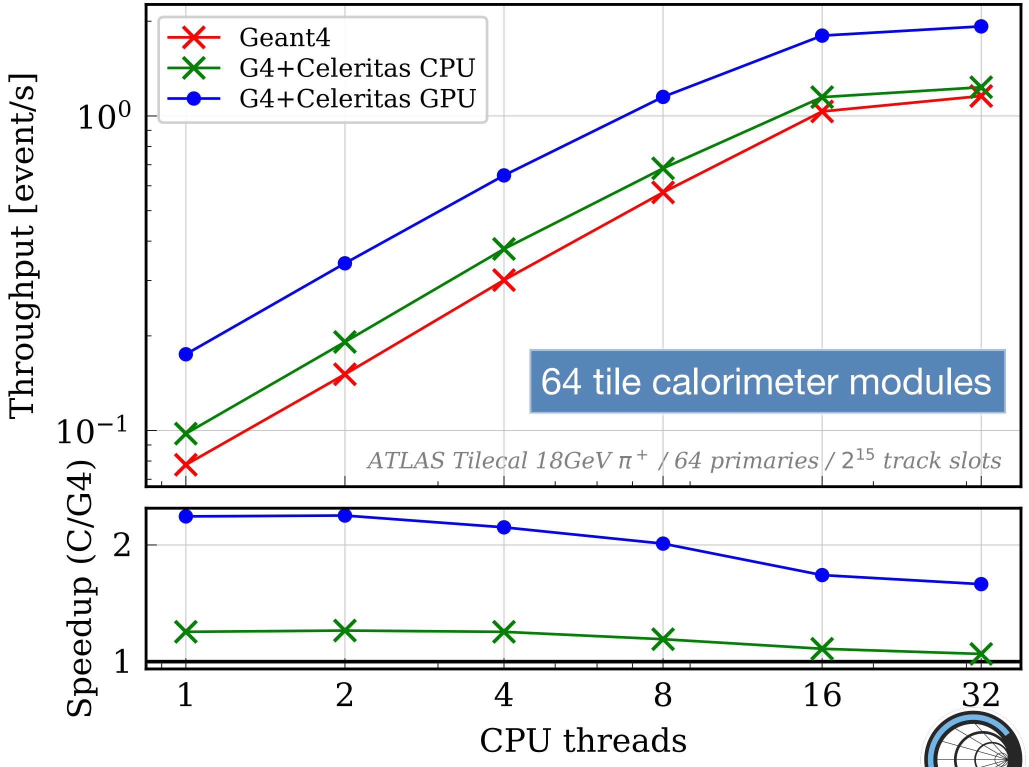
Source: Johnson, S.R. Geant4 Meeting 2023

Oak Ridge National Laboratory

Offload performance results

- 1/4 of a Perlmutter (NERSC) GPU node
  16 cores of AMD EPYC, 1 Nvidia A100
- Time **includes** startup overhead, Geant4 hadronic physics, track reconstruction, and SD callback *(2048 π^+ in all cases)*
- GPU speedup: **1.7–1.9×** at full occupancy
  Using all CPU cores with a single GPU
- CPU-only speedup: still **1.1–1.3×**!
- LHC-scale energy per event (i.e., all 64 modules) is needed for GPU efficiency
- One fast GPU can be shared effectively by full multithreaded Geant4

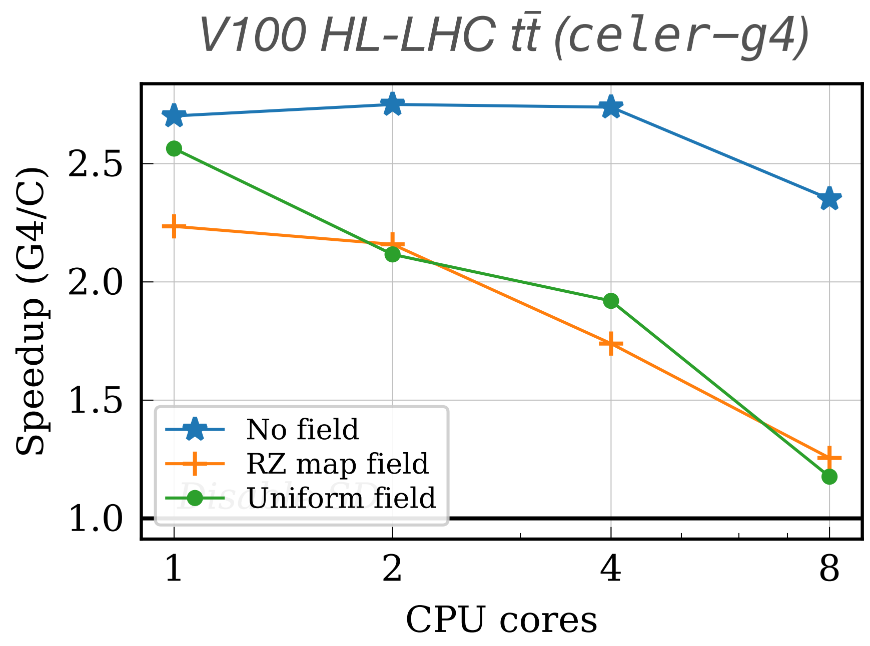


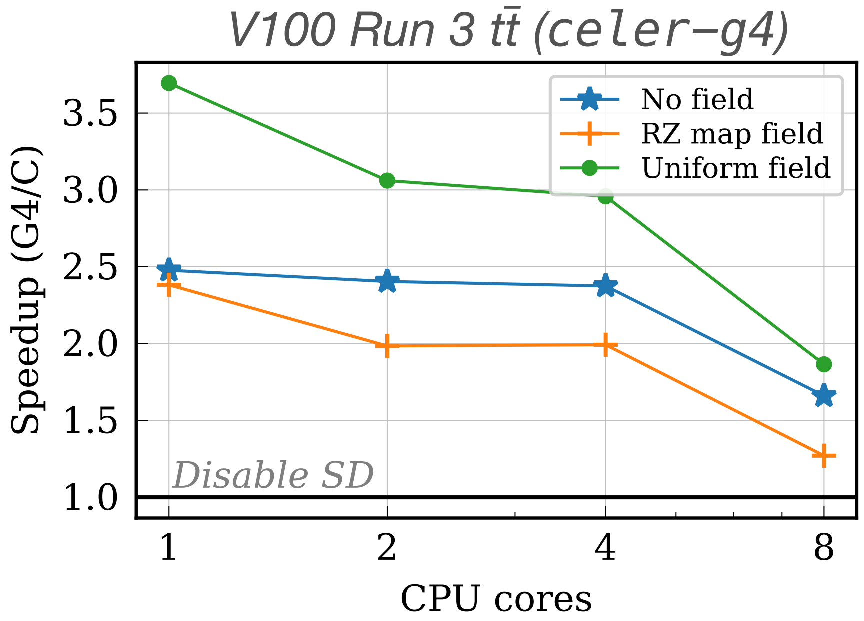


Source: Johnson, S.R. Geant4 Meeting 2023

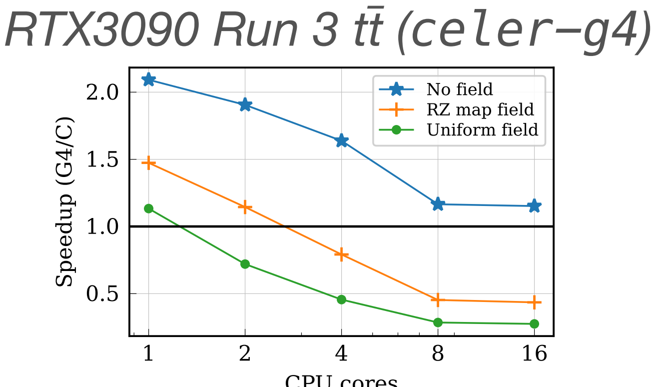
CMS performance results

- Initial CMSSW integration “works”
  - ~500 lines to integrate
  - Performance isn’t comparable due to different physics
- Standalone Geant4 app
  - FTFP_BERT, discretized+interpolated RZ magnetic field
  - Without Celeritas, 8× slower than CMSSW
- Promising results
  - SD reconstruction is <15% of runtime
  - Initial comparison of hits shows good agreement
  - Possibly over factor of 2 speedup in production
- Plenty to investigate
  - Magnetic field driver
  - Single-precision





Hardware: Intel Xeon Gold 6152 CPU 22c 2.10GHz
+ NVIDIA Tesla V100 SXM2 w/32GB HBM2
Input: 8 tt̄ events @ 14 TeV from LHC pp collision



Hardware: Intel Xeon i9-10900K CPU 20c 3.70GHz
+ NVIDIA Tesla RTX3090 w/32GB HBM2
Input: 8 tt̄ events @ 14 TeV from LHC pp collision

Standalone EM performance

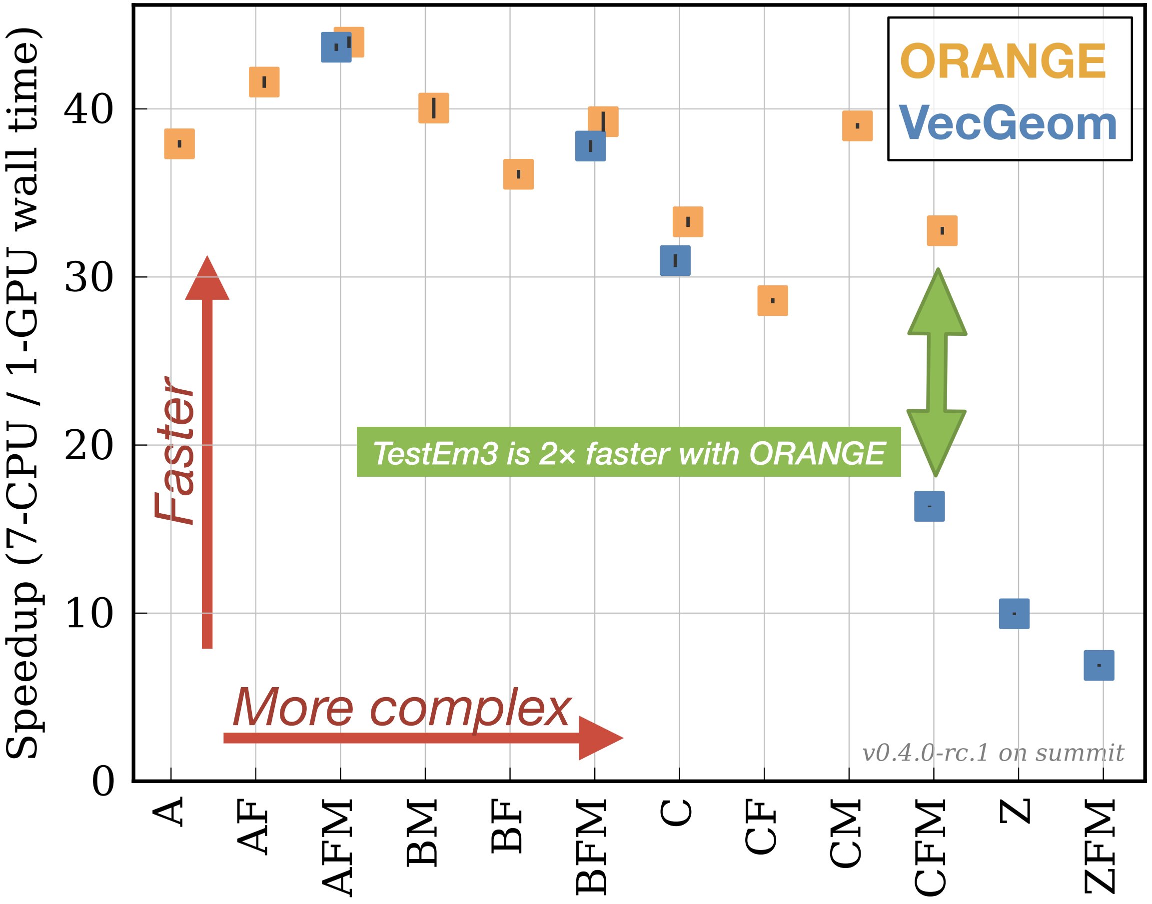
	Problem definition
A	testem15
B	simple-cms
C	testem3
Z	cms2018

	Modifier
F	+field
M	+msc

- 1300 × 10 GeV e-, 7 events
- ⅙ Summit node

  1 × Nvidia V100 GPU, 7 × IBM Power9 CPUs
- Celeritas GPU vs CPU

  CUDA (1 CPU thread) vs OpenMP (7 CPU threads)
- "Speed of light": **7–44×**



Multiply speedup by 7× for CPU:GPU equivalence

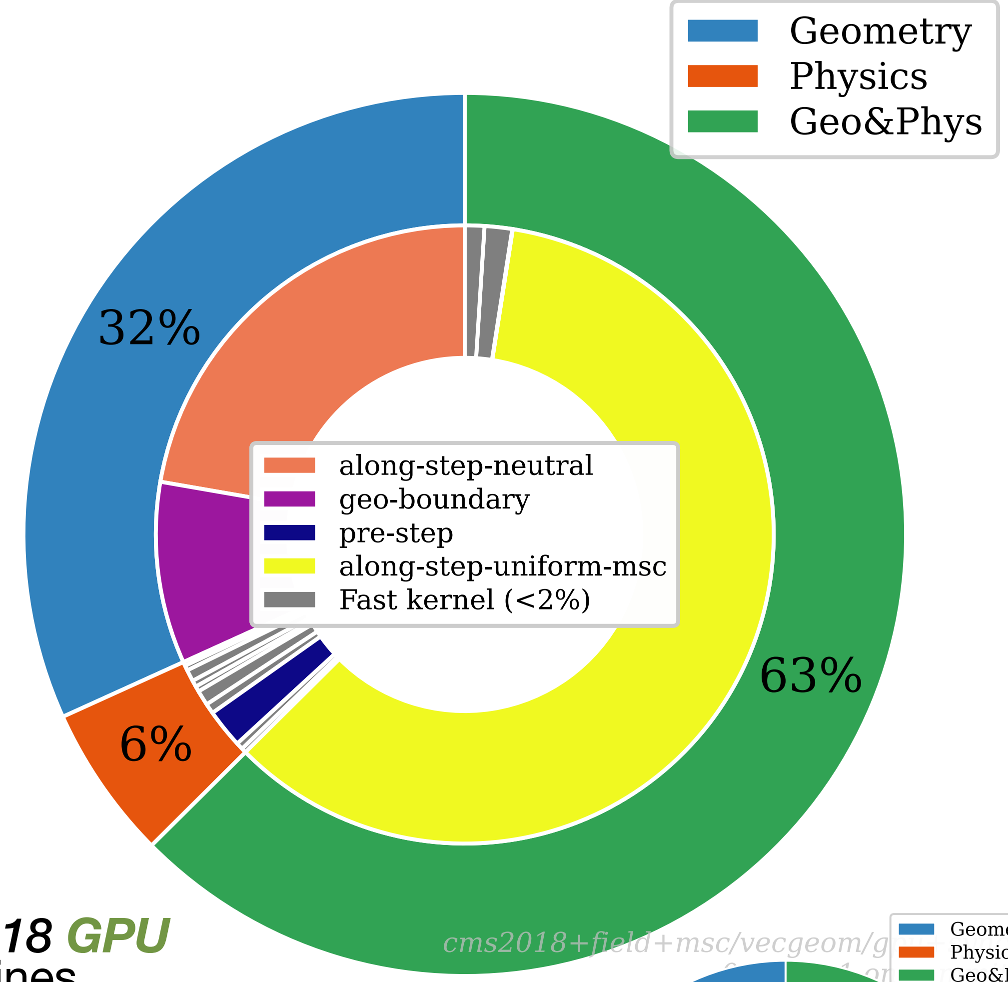
Background

Results

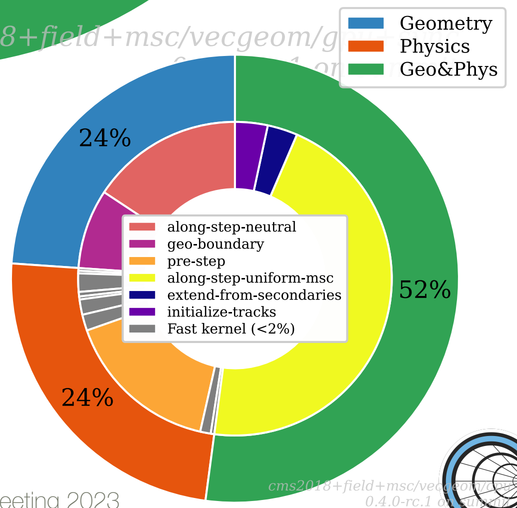
Conclusions

Ongoing work

- Integration
  - CMSSW
  - Athena (ATLAS) framework
- Verification & validation
  - EM test problems
  - CMSSW workflow
  - Benchmark problems with AdePT team
- Optimization and geometry
  - **95%** of standalone runtime in CMS2018 is in geometry routines
  - Performance tuning “knobs” have vast and mostly unexplored parameter space
  - GPU native sensitive detectors
  - ORANGE navigation



CMS2018 GPU



CMS2018 CPU

Source: Johnson, S.R. Geant4 Meeting 2023

Future work

- Optimize magnetic field propagation and geometry
- Validate for use in frameworks (CPU or GPU)
- Evaluate performance on commodity graphics cards
- Implement optical physics for other HEP experiments

Goals for Celeritas GPU EM-only performance
2× per watt vs CPU *(efficiency)*
160× CPU:GPU *(capacity)*

Summary

- Proven real-world performance for GPU EM physics **and** ease of integration for client codes
  - Comparisons with 1 GPU, multicore CPU, against pure Geant4
  - Calorimeter test beam net improvement: **10–30% faster** on CPU, **1.8–2.2×** on GPU (Nvidia A100)
  - CMS Run 3 configuration standalone simulation speedup: **12–87% faster** on GPU (Nvidia V100)
- Anticipated performance even higher
  - Standalone EM problems: **~7–34× faster** (Celeritas CPU vs GPU) on Summit (Nvidia V100) (49–238× GPU/CPU core equivalence)
  - ORANGE vs current VecGeom for TestEM3: **85% faster**

https://github.com/celeritas-project/celeritas

Acknowledgments

Celeritas v0.4 code contributors:

- Elliott Biondo (@elliottbiondo)
- Philippe Canal (@pcanal)
- Julien Esseiva (@esseivaju)
- Tom Evans (@tmdelellis)
- Hayden Hollenbeck (@hhollenb)
- Seth R Johnson (@sethrj)
- Soon Yung Jun (@whokion)
- Guilherme Lima (@mrguilima)
- Amanda Lund (@amandalund)
- Ben Morgan (@drbenmorgan)
- Stefano C Tognini (@stognini)

Past code contributors:

- Doaa Deeb (@DoaaDeeb)
- Vincent R Pascuzzi (@vrpascuzzi)
- Paul Romano (@paulromano)

OLCF: This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

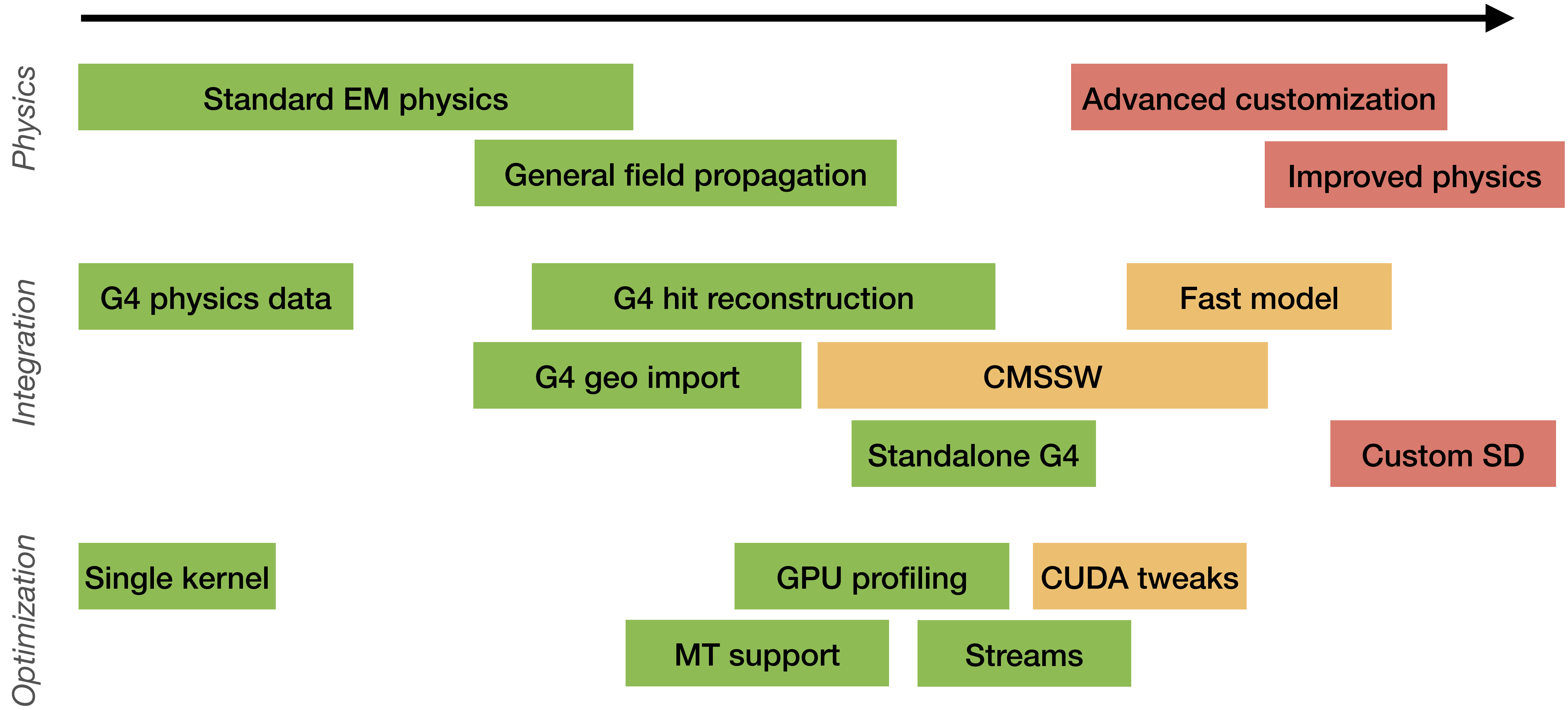
SciDAC: This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics, Scientific Discovery through Advanced Computing (SciDAC) program.

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.

https://github.com/celeritas-project/celeritas

Backup slides

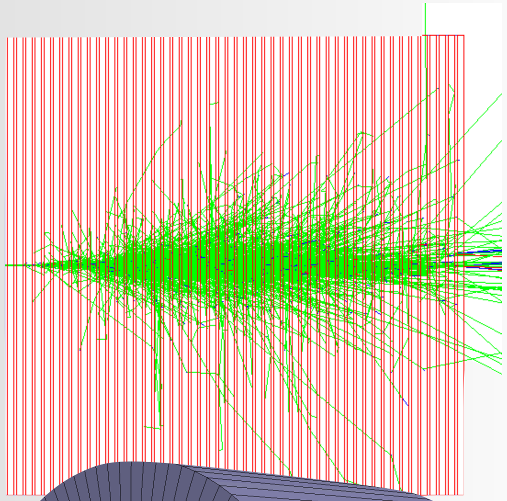
Qualitative timeline

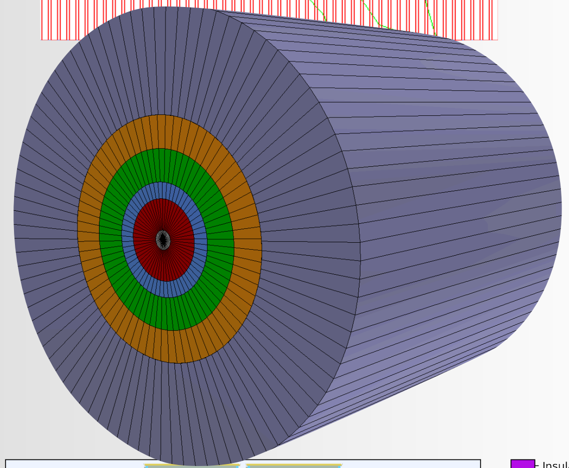


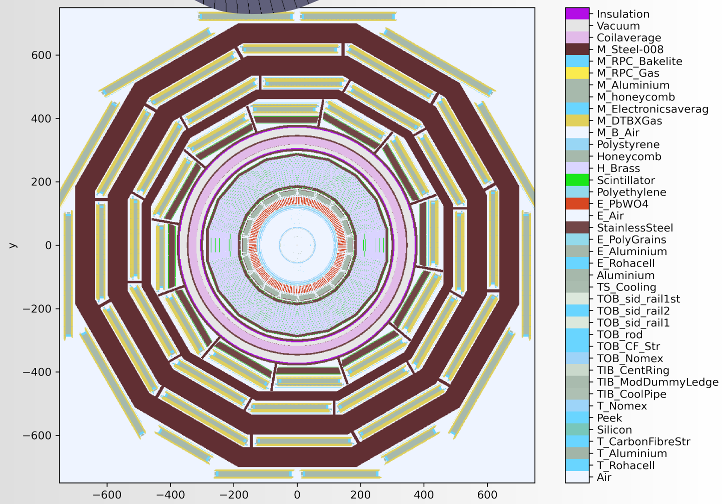
Regression/timing suite

- Run on single node of Summit at full capacity
  - 6 separate runs simultaneously (different seed for each)
  - Each run: 7 CPU (OpenMP) vs. 1 GPU (+1 CPU)
  - Demonstrate performance “loss” by neglecting GPU resources
- 1300 × 10 GeV e^- per event, 7 events per run
- Preliminary set of problem definitions

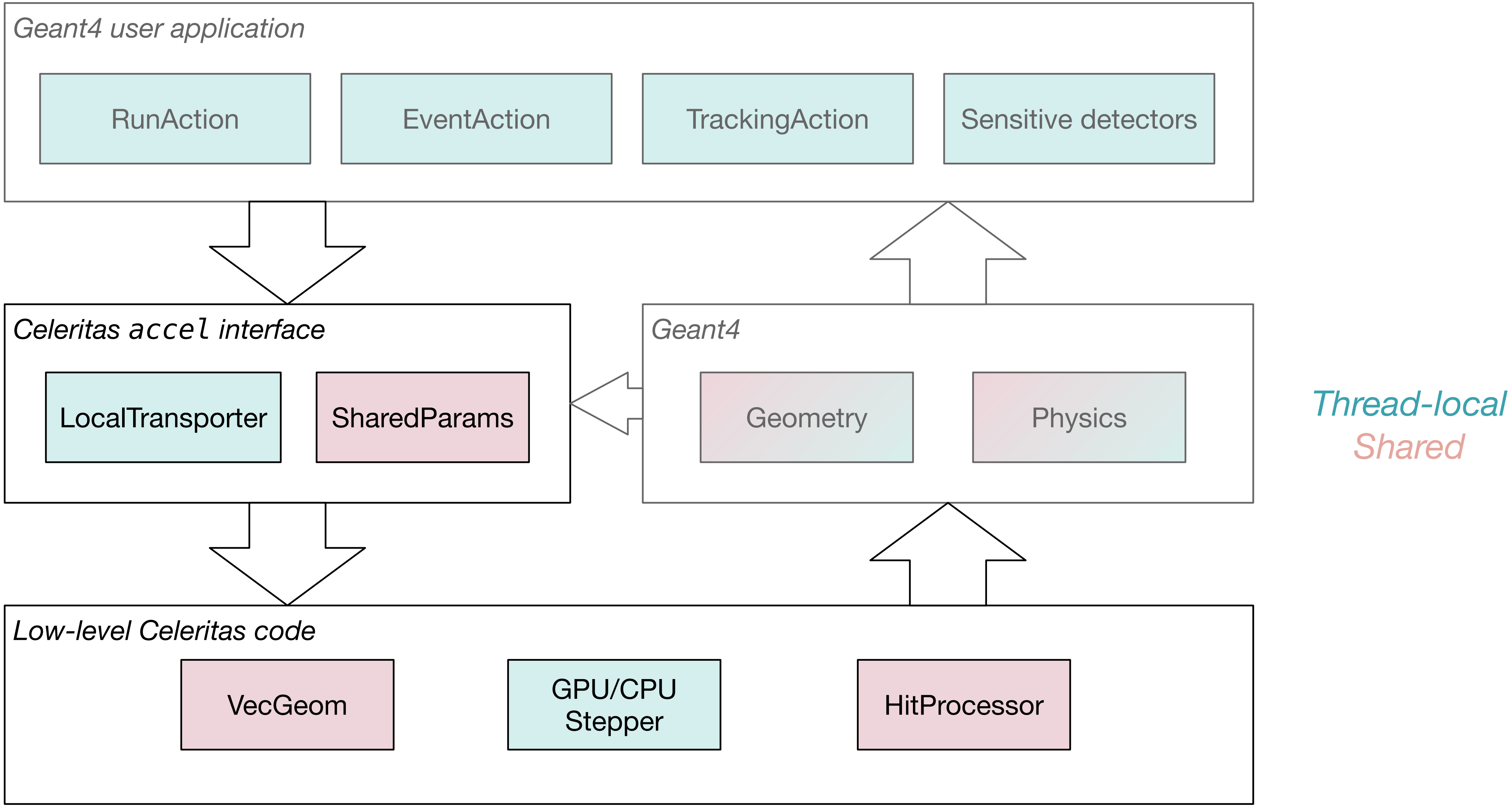
  (working with AdePT team to develop)
- Initial optimizations
- Initial results are apples-to-apples







Geant4 interface library



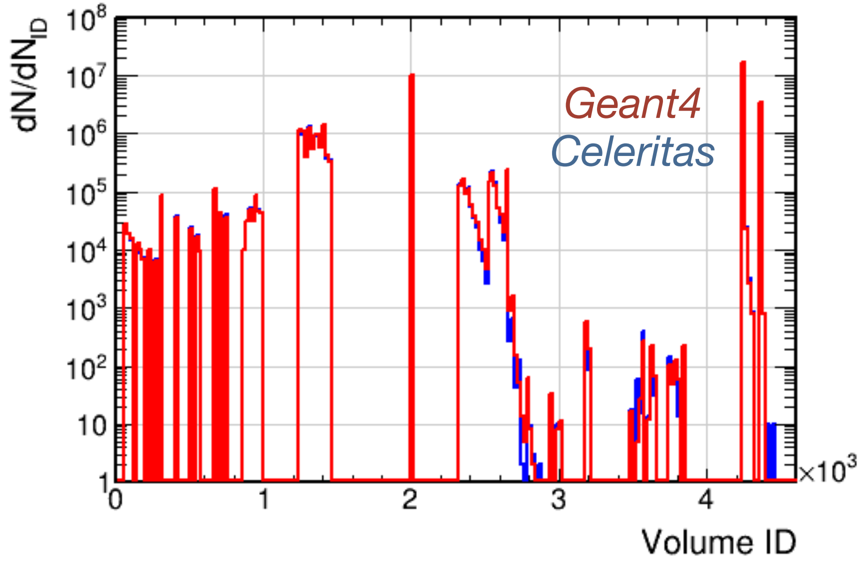
https://celeritas-project.github.io/celeritas/user/index.html

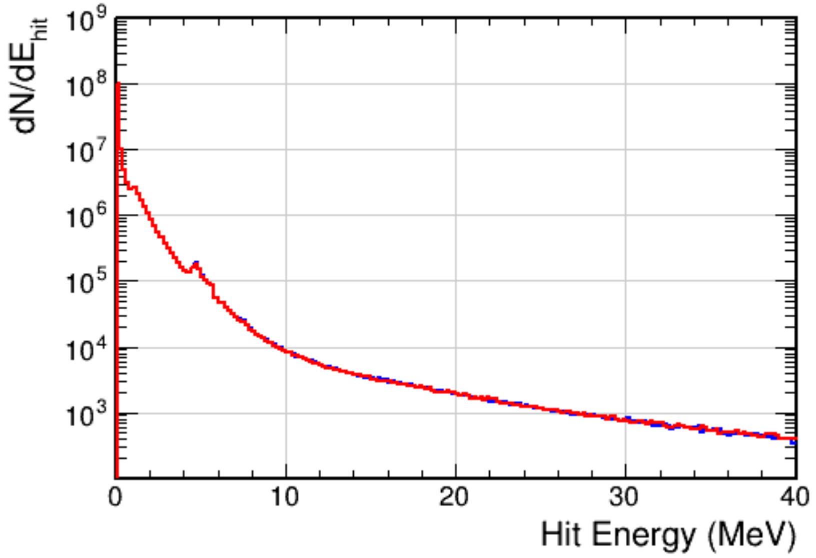
Johnson, Seth R. "Celeritas v0.2: Offloading EM tracks to GPU from Geant4," 21 Feb 2023.

Initial CMSSW integration

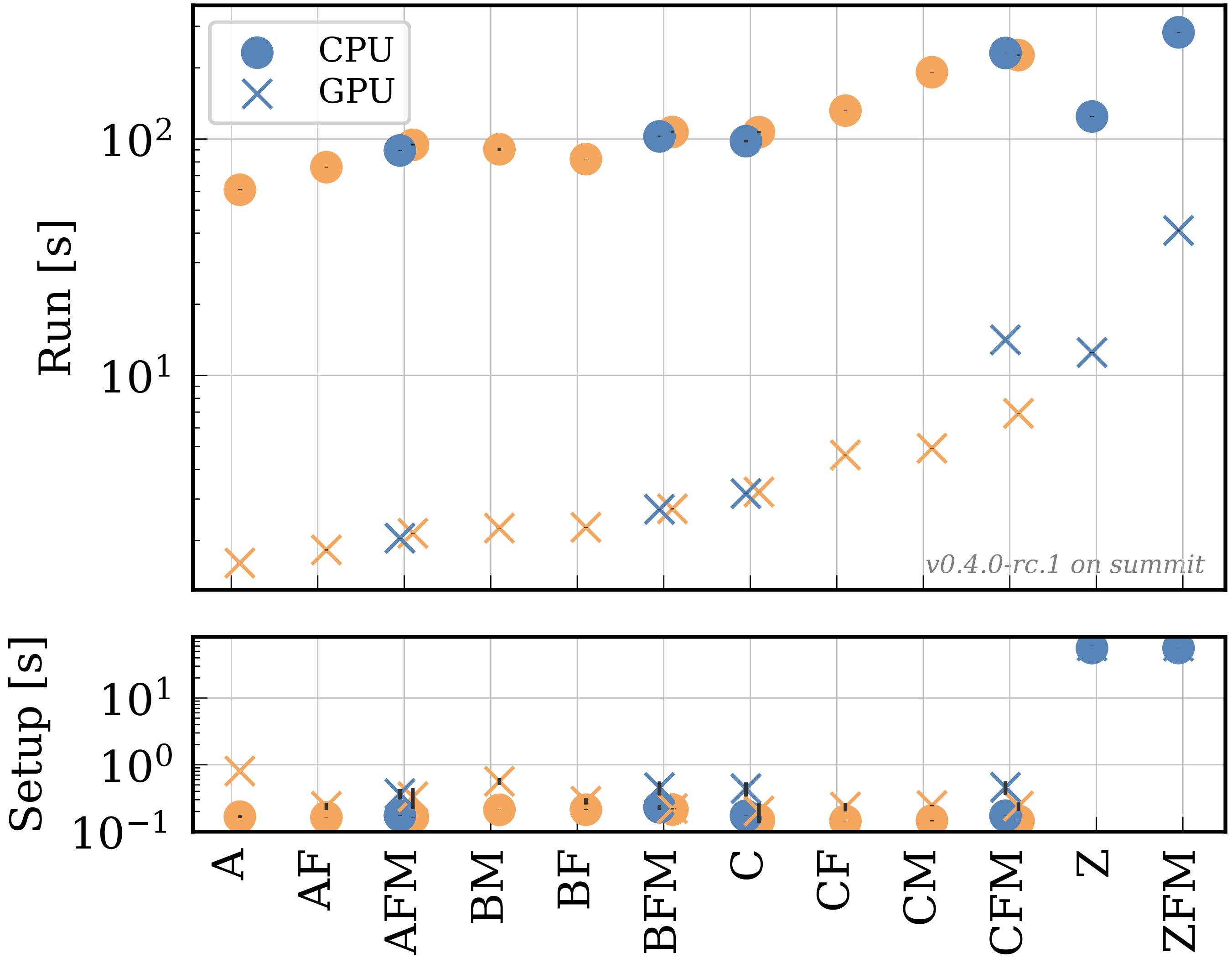
- ~500 lines of code to integrate
  - Offload e^-, e^+, γ to Celeritas
  - R–Z field map preprocessed for Celeritas
  - Celeritas reconstructs hits and sends to CMSSW SDs
  - No support for MC truth or track-level granularity
  - CMSSW has numerous fine-grained tweaks to physics/ propagation compared to default EM
- Initial “fair” performance comparison
  - Current approach: export CMS geometry and detectors to GDML, run through standalone Geant4+Celeritas app
  - 8 CPU+1 GPU standalone simulation: **17–87% faster**
  - **Theoretical** max speedup in framework: 230%

($t\bar{t}$ events, CMS Run3 geometry, tuned physics, full fidelity magnetic field)





Regression problem run time

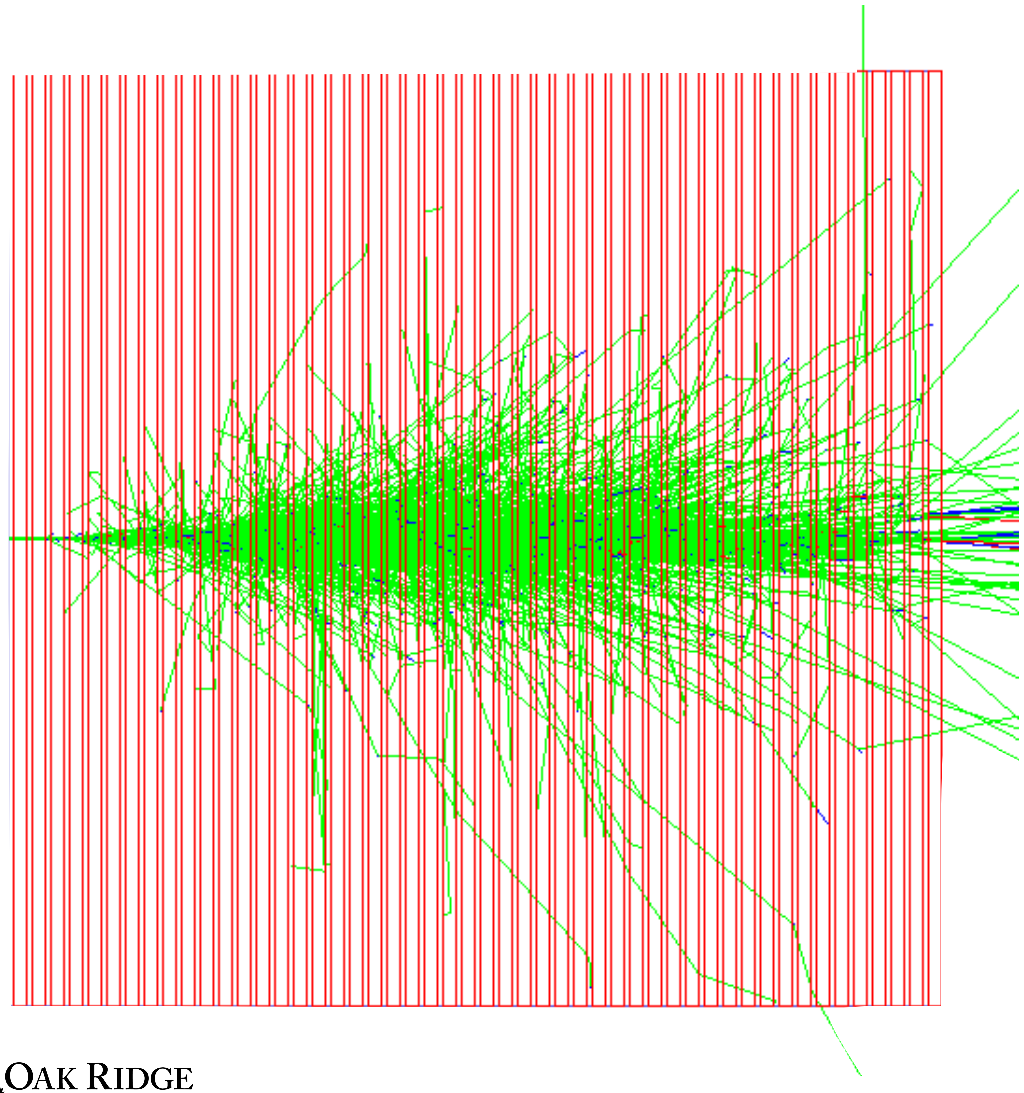


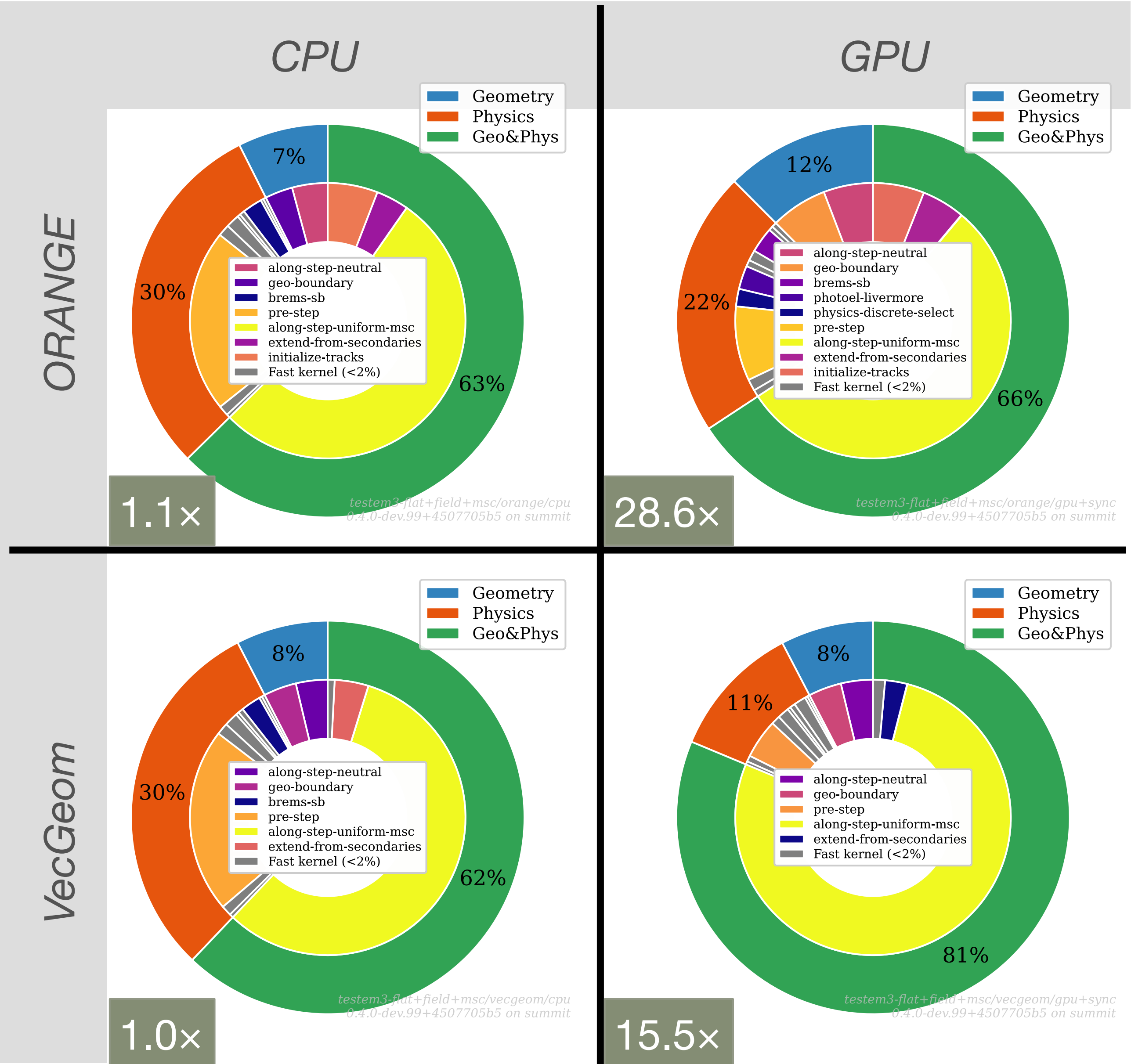
ORANGE
VecGeom

	Problem definition
A	testem15
B	simple-cms
C	testem3
Z	cms2018

	Modifier
F	+field
M	+msc

TestEM3 performance





Stargazing

- Interest in GPU HEP is growing!