OAK RIDGE National Laboratory

Celeritas for platform portable HEP comparison

Seth R Johnson

Celeritas Code Lead
Senior R&D Staff
Scalable Engineering Applications

CELERITAS

Celeritas core team:

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

Celeritas core advisors:

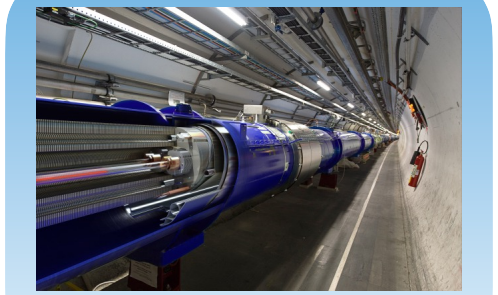
Tom Evans *(ORNL)*, Philippe Canal *(FNAL)*, Marcel Demarteau *(ORNL)*, Paul Romano *(ANL)*

U.S. DEPARTMENT OF ENERGY

HEPiX meeting
6 March, 2024

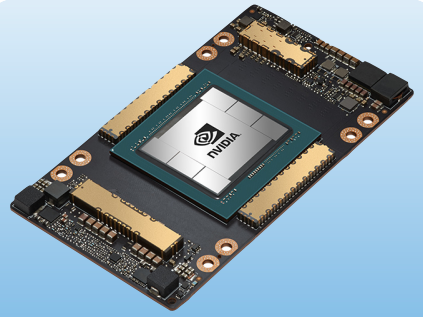
Celeritas project goal

- **Accelerate scientific discovery** by improving detector simulation throughput and energy efficiency for LHC production simulation
  - Long term goal: as much work as possible on GPU
  - Initial funding: focus on EM physics (but keep door open for more!)
- Multidisciplinary approach
  - **Research and develop** novel algorithms for GPU-based Monte Carlo simulation in High Energy Physics
  - **Implement** production-quality code for GPU simulation
  - **Integrate** collaboratively into experiment frameworks



LHC beamline ©CERN

CELERITAS



Nvidia A100 GPU @Nvidia

OAK RIDGE National Laboratory

Background

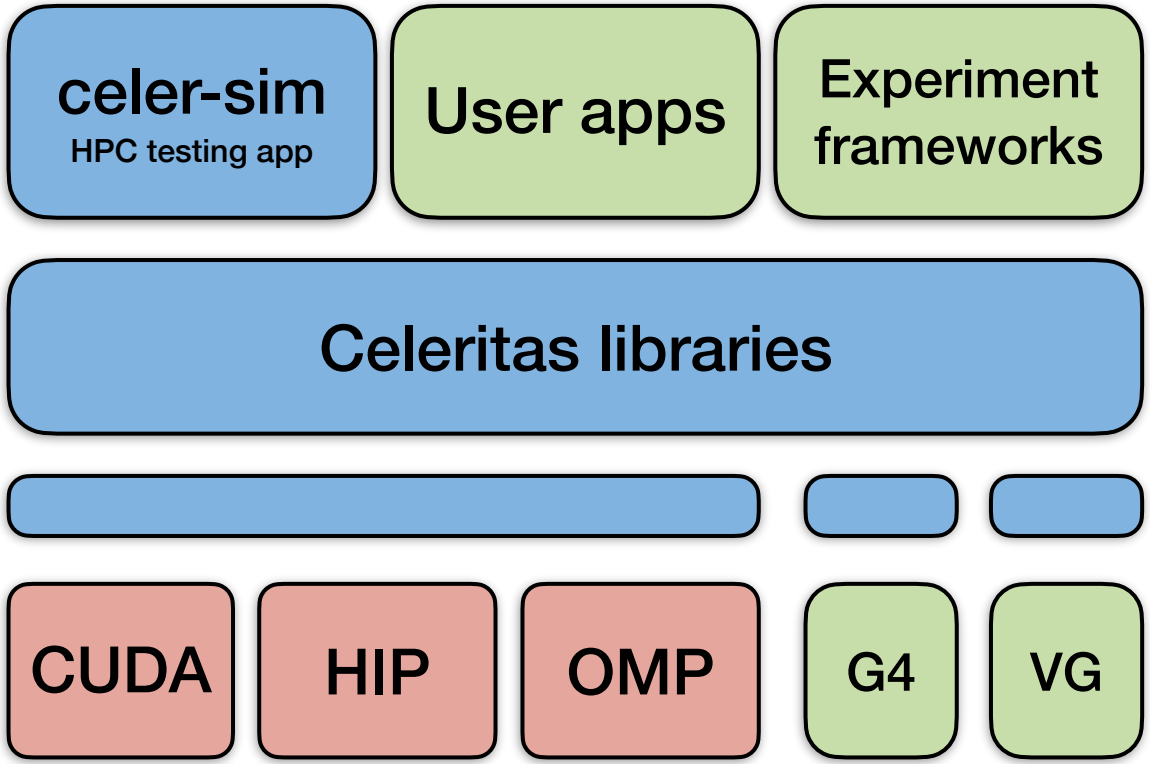
Methods

Results

Conclusions

Software stack

- Every dependency is optional
  - Needed for testing on oddball HPC systems
  - Minimum useful simulation requires Geant4 or ROOT for physics data input
  - HEP detector geometry requires VecGeom
- In-house data model supports platform performance portability
  - Most code is just C++ (<1% is CUDA/HIP)
  - Data management layer abstracts memory location and transfer
  - “Launcher” helper abstracts code execution
  - Plan this year to add Intel via DPC++

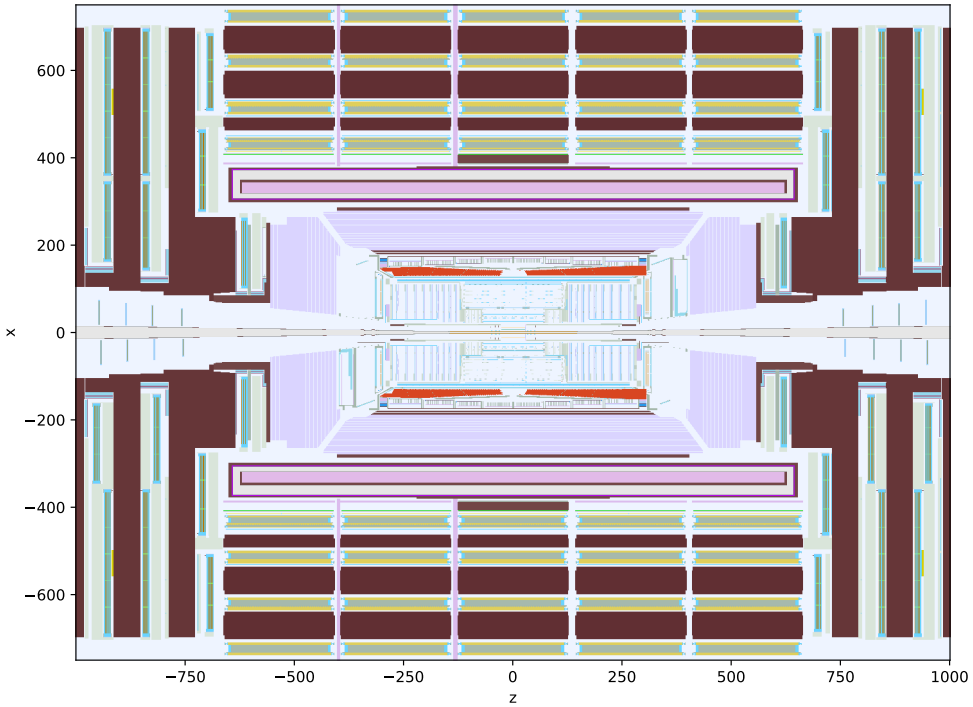


Optional dependencies:
- nlohmann/json
- HepMC3
- googletest

Documentation:
- Doxygen
- Sphinx
- Breathe

High-level capabilities targeting LHC simulation

- Equivalent to `G4EmStandardPhysics`
  …using Urban MSC for high-E MSC; only γ, $e^{\pm}$
- 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

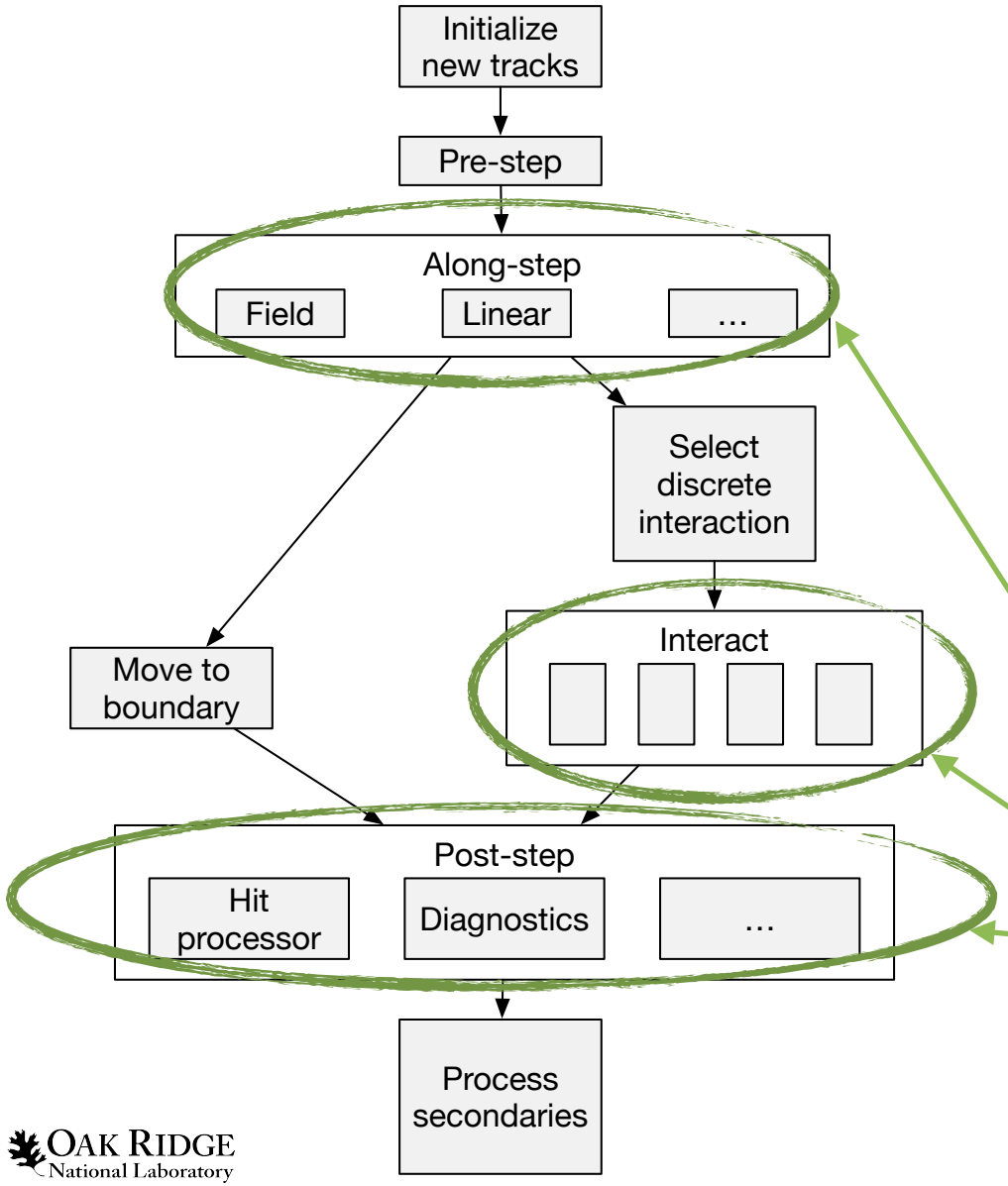


GPU-traced rasterization of CMS 2018

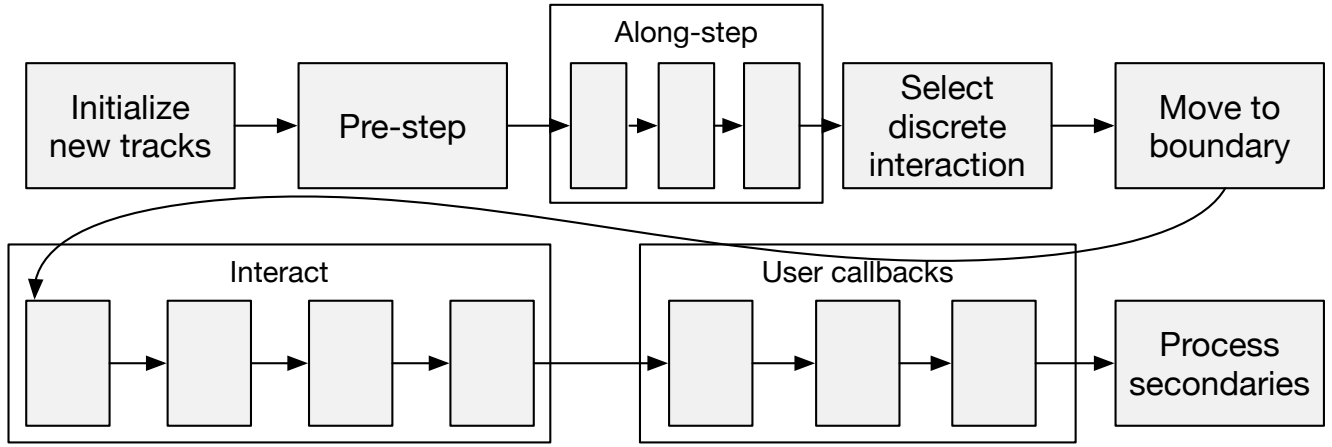
**VecGeom currently requires CUDA: ORANGE navigation required for AMD*

Source: Johnson, S.R. Geant4 Meeting 2023

Stepping loop on a GPU



Process large batches of tracks through all kernels (10^3–10^6)



Topological sort: a loop over kernels

Using many small kernels improves extensibility

Source: Johnson, S.R. Geant4 R&D Summary 2023

Celeritas/Geant4 integration

- **Imports** EM physics selection, cross sections, parameters
- **Converts** geometry to VecGeom model without I/O
- **Offloads** EM tracks from Geant4
  *(Via `G4UserTrackingAction`, `G4VFastSimulationModel`, or **`G4VTrackingManager`**)*
- **Scores** hits to user "sensitive detectors"
  (Copies from GPU to CPU; reconstructs `G4Hit`, `G4Step`, `G4Track`; calls `Hit`)
- **Builds** against Geant4 10.5–11.2 with no patches required

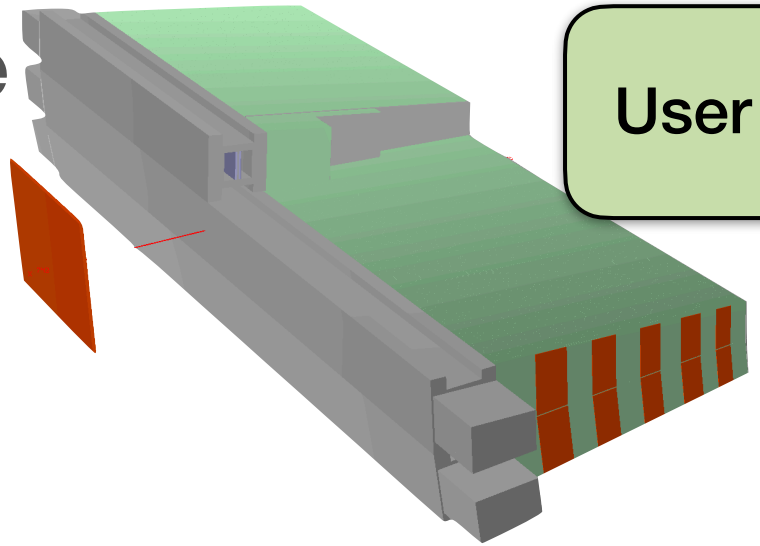
Background

Methods

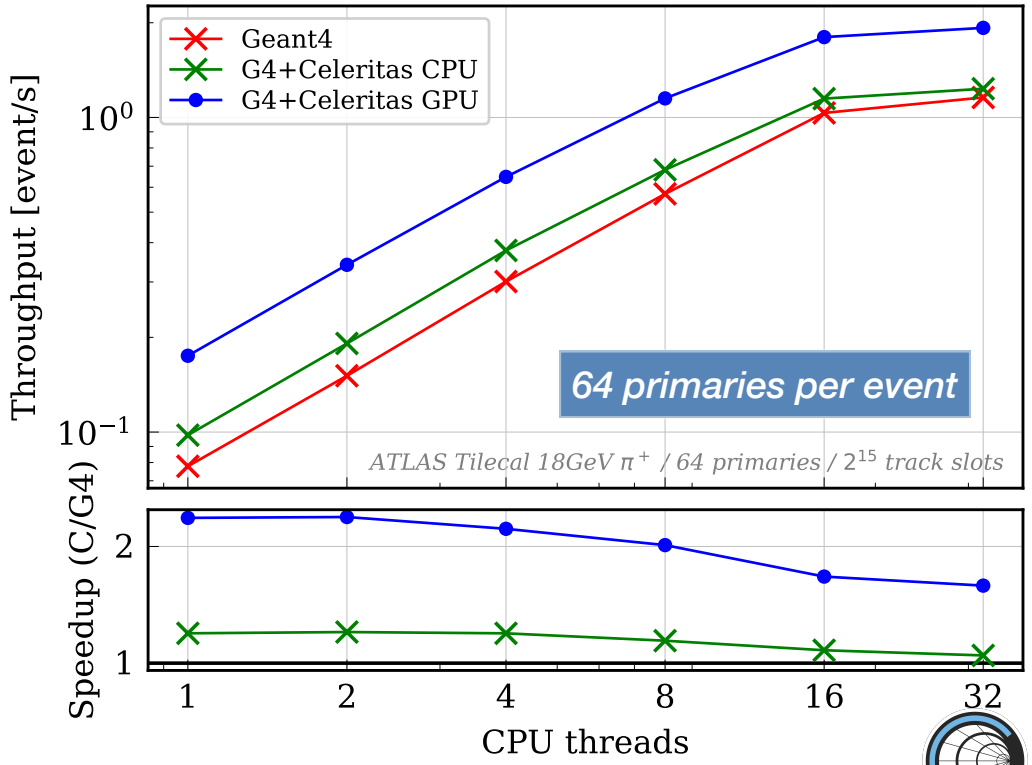
Results

Conclusions

ATLAS Tile Calorimeter performance



- Example of combined GPU/CPU comparison
- 1/4 of a Perlmutter (NERSC) GPU node
  16 cores of AMD EPYC, 1 Nvidia A100
- GPU speedup: **1.7×** at full occupancy
  Using all CPU cores with a single GPU
- CPU-only speedup: **1.1–1.3×**
- Theoretical maximum speedup: **2.2–2.5×**
  Instantly killing e-, e+, γ when born
- **One GPU is effective with many-CPU Geant4**

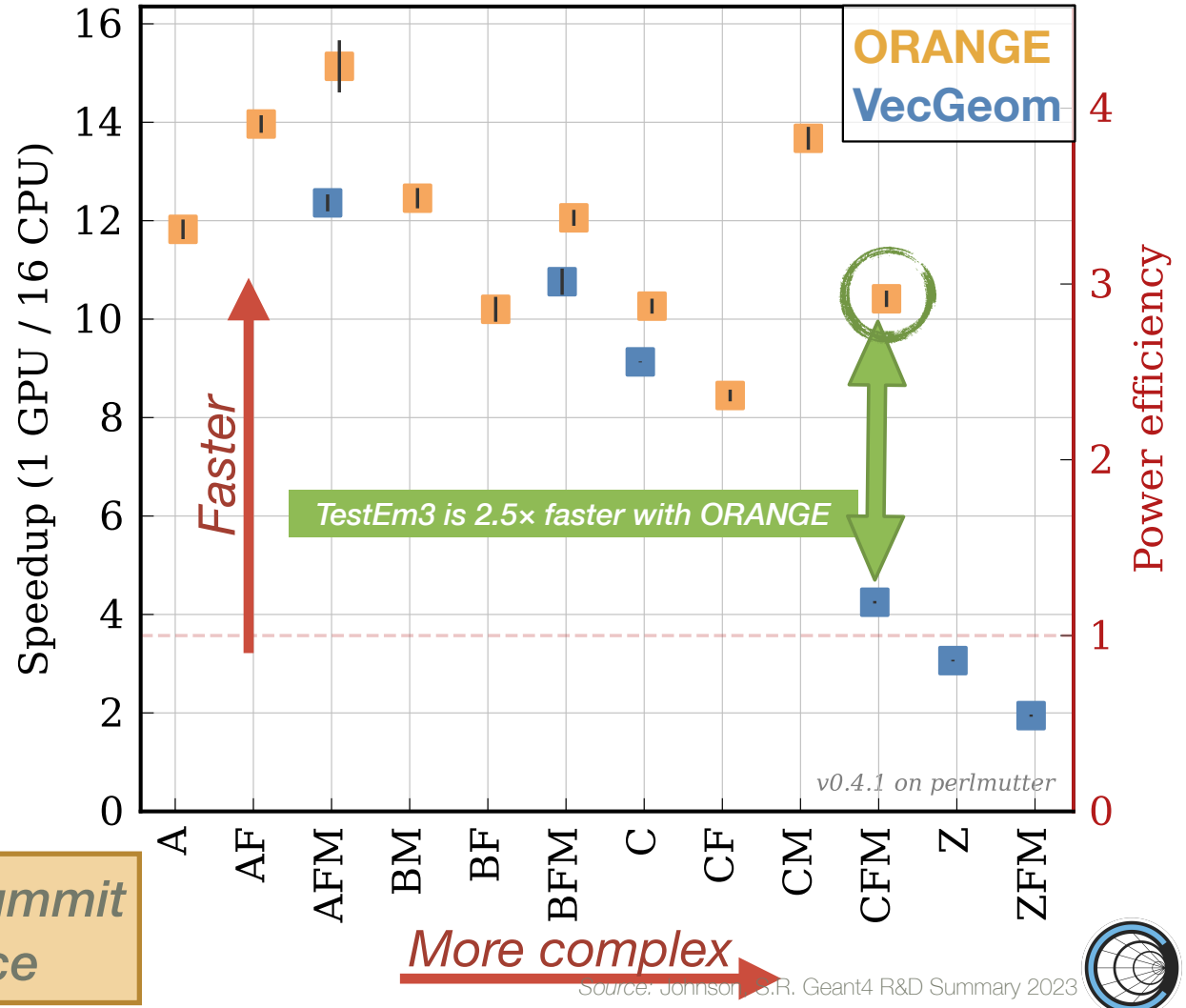


Standalone EM performance

Modifier		Problem definition	
		A	“infinite” medium
		B	simple-cms
F	+field	C	idealized calorimeter
M	+msc	Z	cms2018

celer-sim
Performance testing app

- 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:* **50–94% loss** if GPUs are ignored
  - *Efficiency:* up to **4×** performance per watt



Previous versions of this slide used Summit which has slower CPU performance

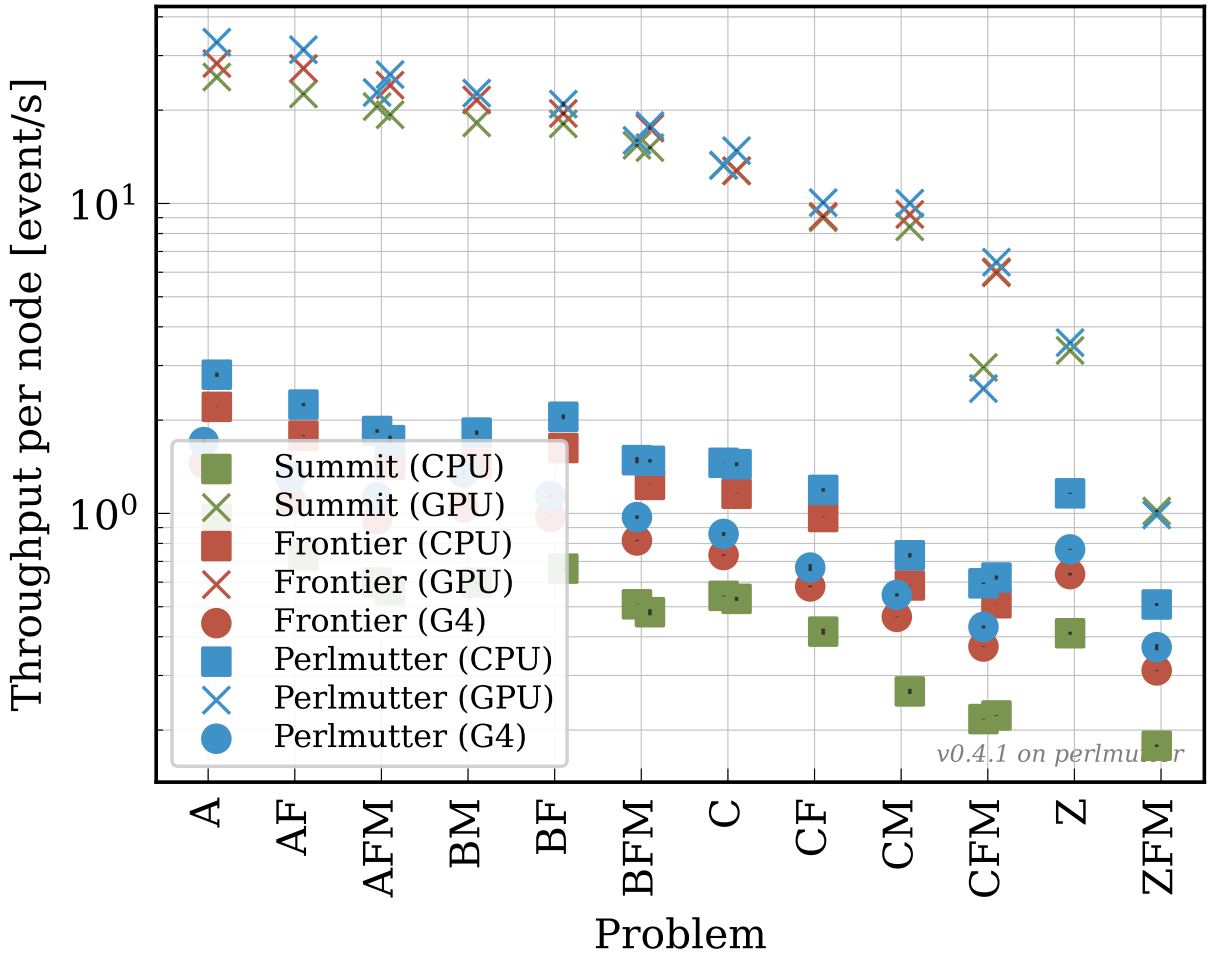
Source: Johnson S.R. Geant4 R&D Summary 2023

Figure of merit: throughput

- **Cross-platform comparison**
- GPUs *cannot* be ignored if present
- AI/ML "revolution" guarantees more coprocessors at all scales

Per-node stats for DOE supercomputers

Machine	Arch	Card	TDP (W)	Cores*	Cards
Summit	CPU	IBM Power9	190	‡22	2
	GPU	Nvidia V100	250	80	6
Perlmutter	CPU	AMD EPYC 7763	280	64	1
	GPU	Nvidia A100	250	108	4
Frontier	CPU	AMD EPYC 7453	225	‡64	1
	GPU	AMD MI250x	500	220	†4



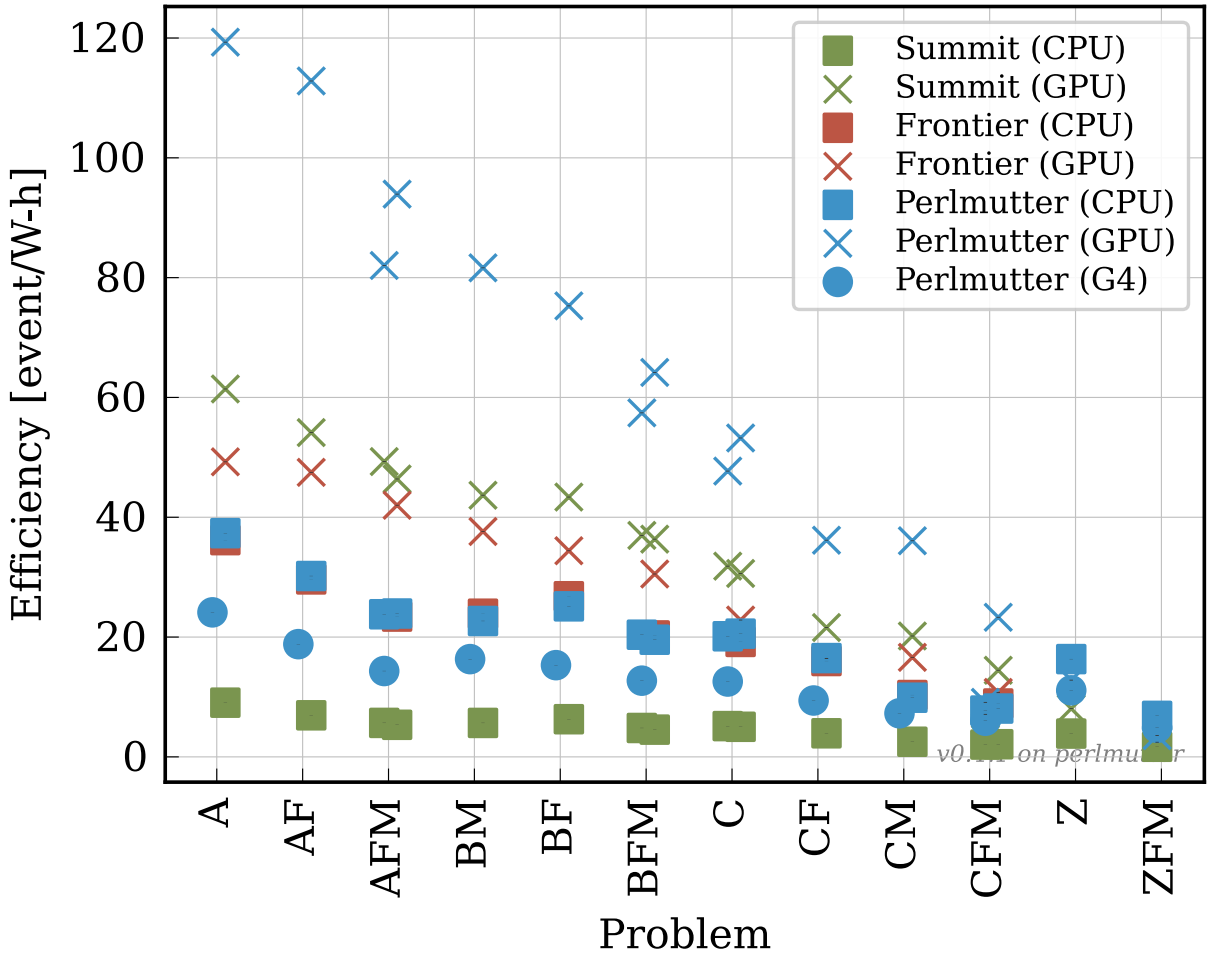
EM only, no SDs

*or SMs;
†Each card has 2 GPUs
‡One core reserved per GPU

Source: Johnson, S.R. Geant4 R&D Summary 2023

Figure of merit: efficiency

- **Estimated** using reported Thermal Design Power (TDP) and Celeritas throughput
  - Likely *conservative* based on nvidia-smi readings, since we use only a fraction of GPU hardware
  - **Really excited to work with your group on potential real-life power draw!**
- GPU consistently shows higher energy efficiency 🌱
- A100:EPYC price is ~4× 💸



EM only, no SDs

Background

Methods

Results

Conclusions

Resources and installation

- Detailed installation, usage, development documentation
  - User documentation: celeritas-project.github.io/celeritas/user/
  - Doxygen: celeritas-project.github.io/celeritas/dev/
  - Github: github.com/celeritas-project/celeritas
  - Slack workspace, or AdePT/Celeritas Mattermost
- Installation
  - Automatic CMake-based detection of available options
  - Full spack environment definition and installable package
  - Docker image with dependencies (used for CI)
  - Rigorous CI matrix gives us confidence in build/test on new platforms

Continuing work

- Concluding efforts funded by SciDAC V
  - EM physics validation
  - Integration with experiment frameworks
- Now officially funded by HEP-CCE (US DOE program)

  High Energy Physics - Center for Computational Excellence
  - Optical physics for LZ, LEGEND, JUNO, etc.
  - GPU-optimized, platform portable computational geometry for HEP
  - Implementation and testing on **Intel GPUs**
  - **Platform portability and energy efficiency are central to this program**

Next steps

- Identify multiple testing hardware
- Configure software stack
- Determine test problems, targeted run time
- Perform cross-hardware efficiency comparisons?
- Incorporate into HEPScore benchmarking?

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