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

Celeritas and an HPC perspective of detector simulation R&D

Seth R Johnson

Celeritas Code Lead
Scalable Engineering Applications

CELERITAS

Celeritas core team:

Elliott Biondo *(ORNL)*, Julien Esseiva *(LBNL)*, Hayden Hollenbeck *(UVA)*, 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

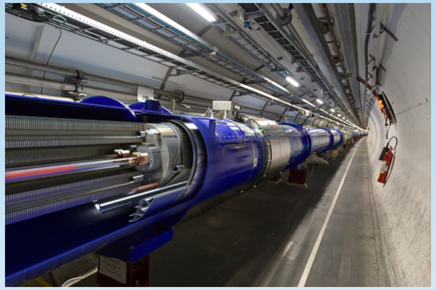
ESPPU Input
22 November, 2024

Celeritas: overview

Using next-generation computing to accelerate HEP's hardest simulations

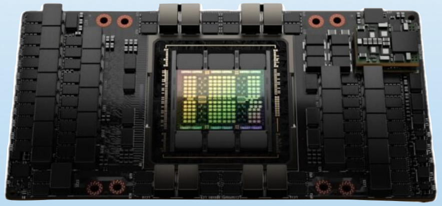
- **GPU** optimized, CPU reproducible
- **Full-fidelity** Monte Carlo detector simulation
- **EM physics** and soon muons, optical photons
- **Automated** Geant4 integration (geometry, physics, SD)
- **Open source** and actively seeking collaborators

https://celeritas.ornl.gov/



LHC beamline ©CERN

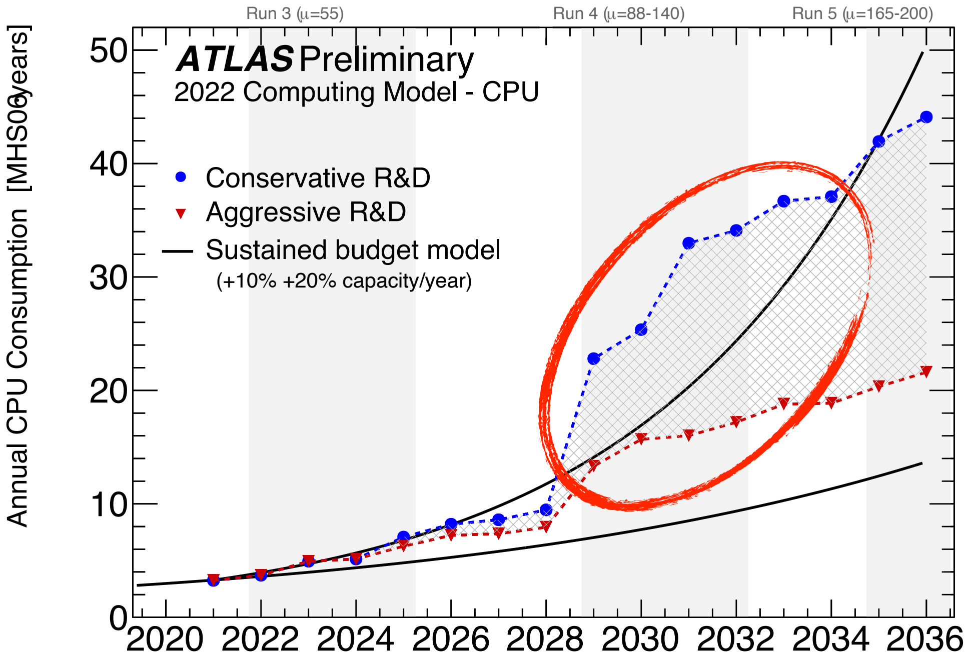
CELERITAS



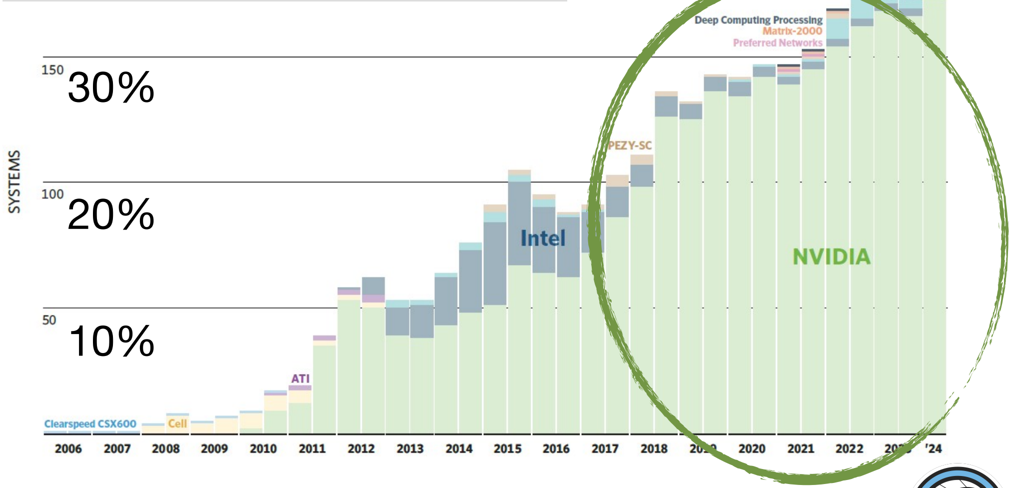
Nvidia H100 GPU @Nvidia

Motivation: HEP×HPC

- **Demand** in high energy physics (HEP) is rapidly increasing
  - **≥10× full** simulation from LHC-HL upgrade; more needed for DUNE, XLZD, EIC, …
  - AI/ML-based **fast** simulation methods will need massive training data, **ideally generated on GPU**
- **Supply** from high performance computing (HPC) is fundamentally changing
  - “Heterogeneous” architectures **dominate** HPC
  - Scientific codes can run more efficiently 🌱⚡ on GPU
    *e.g., Perlmutter reports 5× average energy efficiency**
  - Demand for AI/ML training and models **will accelerate** deployment of GPUs for scientific computing



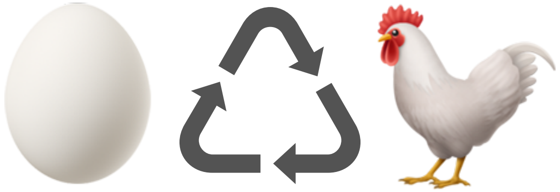
ATLAS Collaboration: CERN-LHCC-2022-005; LHCC-G-182



Top500 Supers: This Is Peak Nvidia For Accelerated Supercomputers, Timothy Prickett Morgan, May 13, 2024.

*https://blogs.nvidia.com/blog/gpu-energy-efficiency-nersc/

Prerequisite: GPU availability



Note: *focus is on GPUs due to hardware development trends*

- **Capacity:** how to **utilize existing** resources with GPUs
  - Online computing hardware during shutdown periods (CMS high-level trigger farm)
  - Multidisciplinary institutional purchases (US DOE computing facilities)
- **Efficiency:** whether to **purchase new** GPUs
  - Requires accurate electrical power measurement of real-life hardware options
  - Lifetime analysis incorporating hardware capital investment, power usage, facility space

Preliminary educated guesses:
- **Lots of “free” hardware to use**
- **GPU accelerators are good investments**

Explore innovative GPU/HPC methods

- *Requirement:* Maximize utilization of hardware resources*

  **2019 recommendation is focused on SIMD*

- *Challenge:* Changing technology** uses heterogenous architectures

  ***GPUs are also a moving target as they shift from graphics→science→AI*

- *Recommendation:* Develop new GPU- and HPC-optimized methods

  ✓ New algorithms targeted at track-parallel full-fidelity simulations†
  *†and integrate **support in experiment frameworks** which often assume track serialization!*

  ✓ Accelerate compute-intensive EM physics, optical physics

  ✓ Machine learning (ML)-based fast simulations, **preferably using data generated on device**

  ▪ Distributed-memory cooperation to reduce I/O bandwidth (HPC)

  ▪ “Oversampling” (multiple simulated responses per event) to maximize GPU parallelism

And improve availability of HPC workflows

Improve physics and framework validation

- *Requirement:* Experiments need improved physics models
- *Challenge:* Production simulations **must validate*** new code

  High-level validation is* ***expensive* *in computing time and personnel qualifications*
- *Recommendation:* Adopt software engineering best practices
  - Add unit testing and verification for low-level components and physics models
  - Automate low-level validation problems for accepting new physics
  - Adopt agile programming techniques to reduce time between testing and deployment
  - Improve modularity of physics (Geant4 models are *not* “a la carte”)

GPU “offload” is effectively a new physics model
(Celeritas, AdePT, Opticks)

Reduce simulation costs with automated biasing

- *Requirement:* Background calculations needed by experiments with increasing sensitivity and complexity
  - Natural radiation (earth, cosmic): low energy/rare event experiments (dark matter, 0νββ)
  - Beam-induced background: EIC, muon collider
- *Challenge:* Need to reduce (not just accelerate) simulation compute time
  - Manual cutoffs and weights used to reduce neutron simulation time in ATLAS
  - Manual geometric splitting/cutoffs in CMS
  - Manual per-region secondary production/tracking cutoffs
- *Recommendation:* Automated biasing/variance reduction (VR) methods
  - Extend well-studied VR methods[1] that focus on time-independent neutral particles
  - ML/genetic algorithms/... to explore parameter space for splits/cutoffs?

[1] Royston, K, T Evans, S Hamilton, and G Davidson. 2023. "Weight Window Variance Reduction on GPUs in the Shift Monte Carlo Code." In *ANS MC2023-The International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. ANS M&C Topical Meeting. Niagra Falls, Ontario, Canada.

High-level suggestions (pontification)

- Capitalize on widely developed tech *(follow the money)*
  - Once upon a time, most computing was science & engineering; now it's TikTok (AI)
  - Explosion in development of open source software: TikTok begets RapidJSON
  - Phase out HEP-specific software if a more popular alternative exists: **focus investment on domain-specific requirements**
- Increase common software infrastructure inside HEP *and* outside it
  - Key4hep, DD4HEP are great examples of unifying future frameworks
  - Scientific software (SSW) ecosystem has resulted from improved open source model
  - Universal tools **increase diversity** by lowering barrier to entry from other fields

Purpose	HEP solution	SSW
Package management	ATLAS/CMS externals	Spack
Analysis & plotting	ROOT	Jupyter, Python, R
Virtual file system	CVMFS	Docker, Ceph

Questions/pillorying?