Celeritas: Version 0.1

Seth R Johnson

HPC methods for nuclear applications
Oak Ridge National Laboratory

Celeritas core team:

Philippe Canal *(FNAL)*, Tom Evans *(ORNL),* Soon Yung Jun *(FNAL)*, Guilherme Lima *(FNAL)*, Amanda Lund *(ANL)*, Paul Romano *(ANL)*, Stefano Tognini *(ORNL)*

27th Geant4 Collaboration Meeting
Rennes, France
26 September, 2022

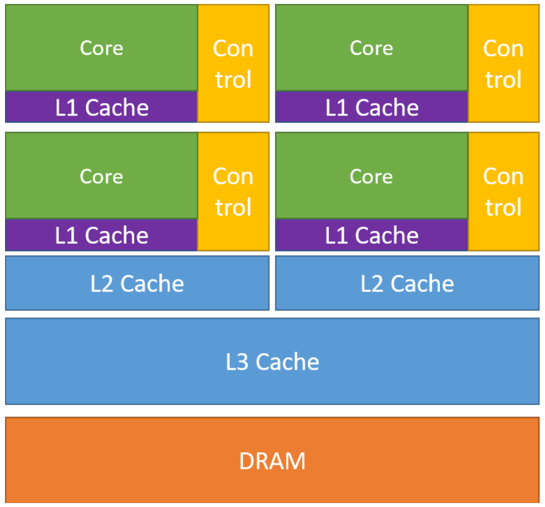
Celeritas project overview

- Motivated by HL-LHC computational challenges *and* by recent success in GPU MC (*ExaSMR*)
- **GPU**-focused implementation of **HEP** detector simulation
  - Physics derived from Geant4 methods and implementation
  - Cross sections, materials, etc. loaded directly from Geant4
- Multi-institution collaboration with external contributors
- Funded through US DOE ASCR/HEP

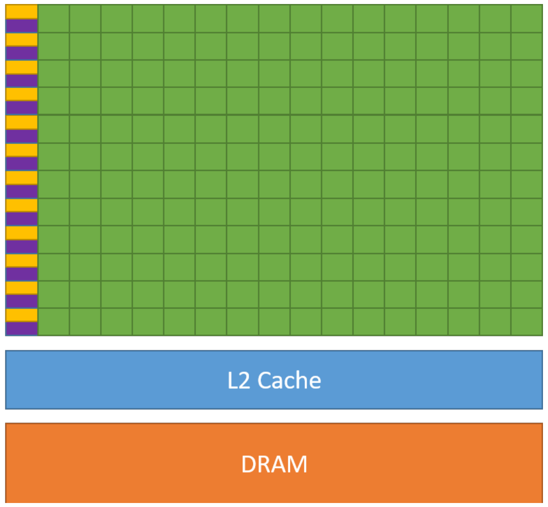
Near-term goal: integrate with Geant4 and experimental frameworks to offload EM tracks to GPU

Hardware considerations

- Modern HPC hardware relies primarily on GPU for computational throughput
  - GP-GPU Conceptualized in early ’00s
  - Very fast and power efficient for “graphics”-like applications
- “Many-core”: massively multithreaded
- Programming models require much more care
  - Not good at flexible/dynamic operations
  - Ideally lots of operations per memory access



CPU



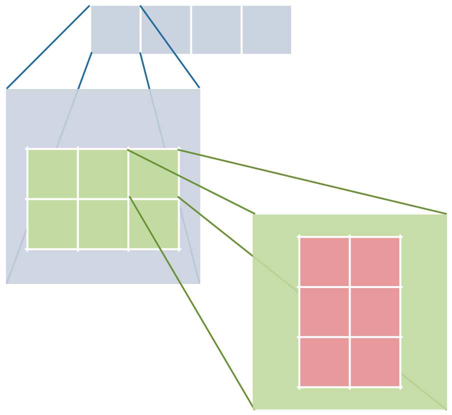
GPU

https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

Challenges

- Execution: divergence and load balancing
  - GPUs want every thread doing the same thing
  - MC: every particle is doing something (somewhat) different
- Memory: data structures and access patterns
  - GPUs want direct, uniform, contiguous access
  - MC: hierarchy and indirection; random access
  - Memory allocation is a particular problem

Structured grid data

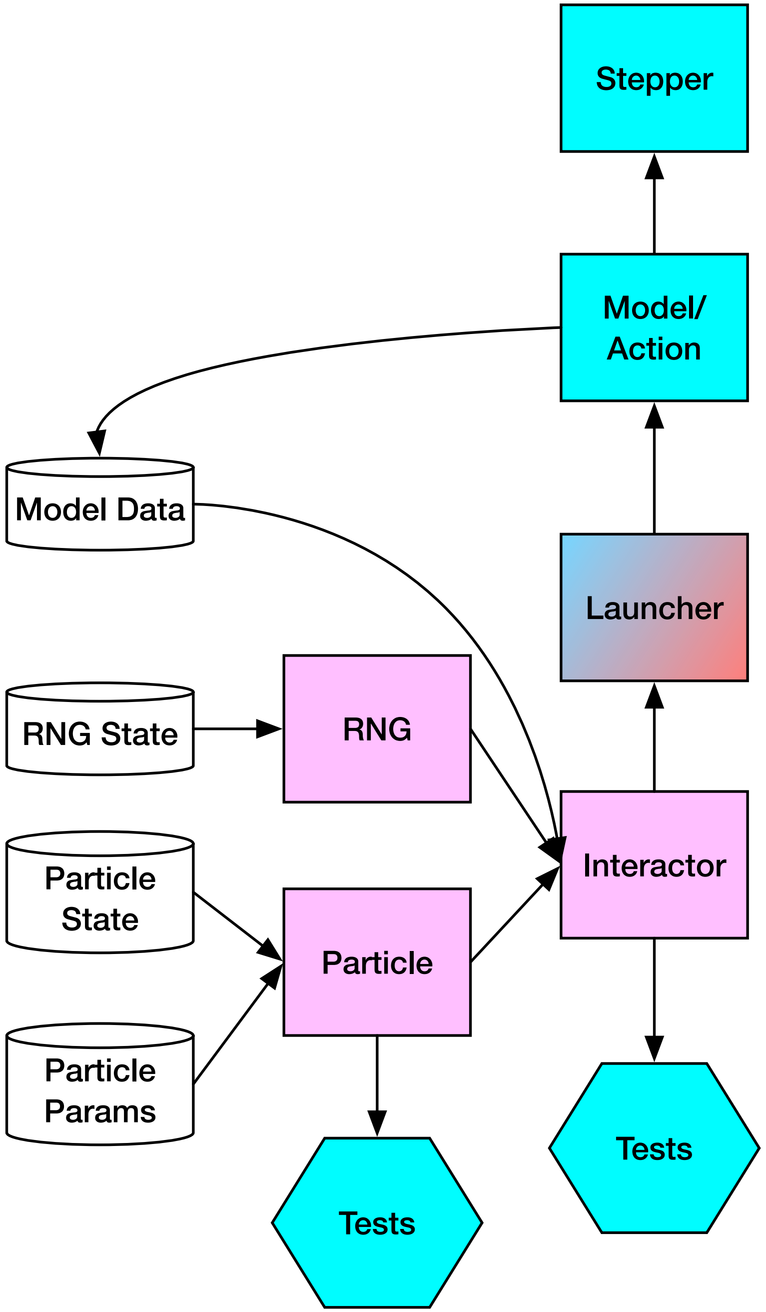


Monte Carlo data

Code design

- Core principles
  - Data-oriented programming (separate data from code)
  - Object-oriented interfaces to data
  - Composition-based objects
  - Revisit legacy design/implementation choices
- Development workflow
  - Extensive unit testing in CPU execution space
  - Some unit testing and more integration testing on GPU
  - In-depth merge request review process
  - Continuous integration

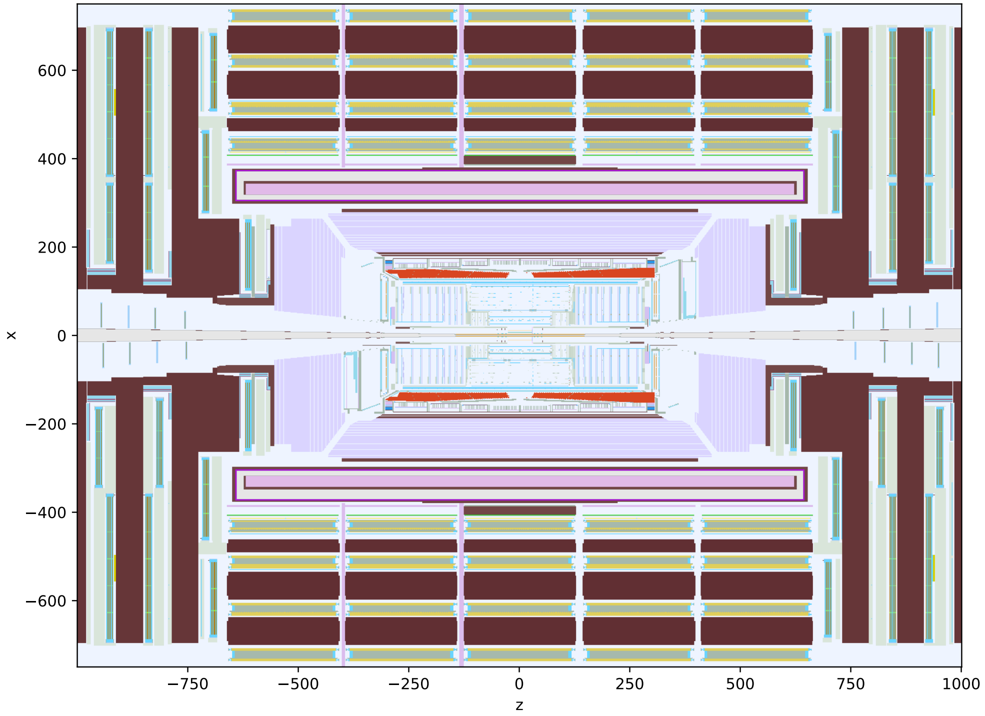
Easily refactored for new architectures, data models, performance



OAK RIDGE National Laboratory

Physics and geometry

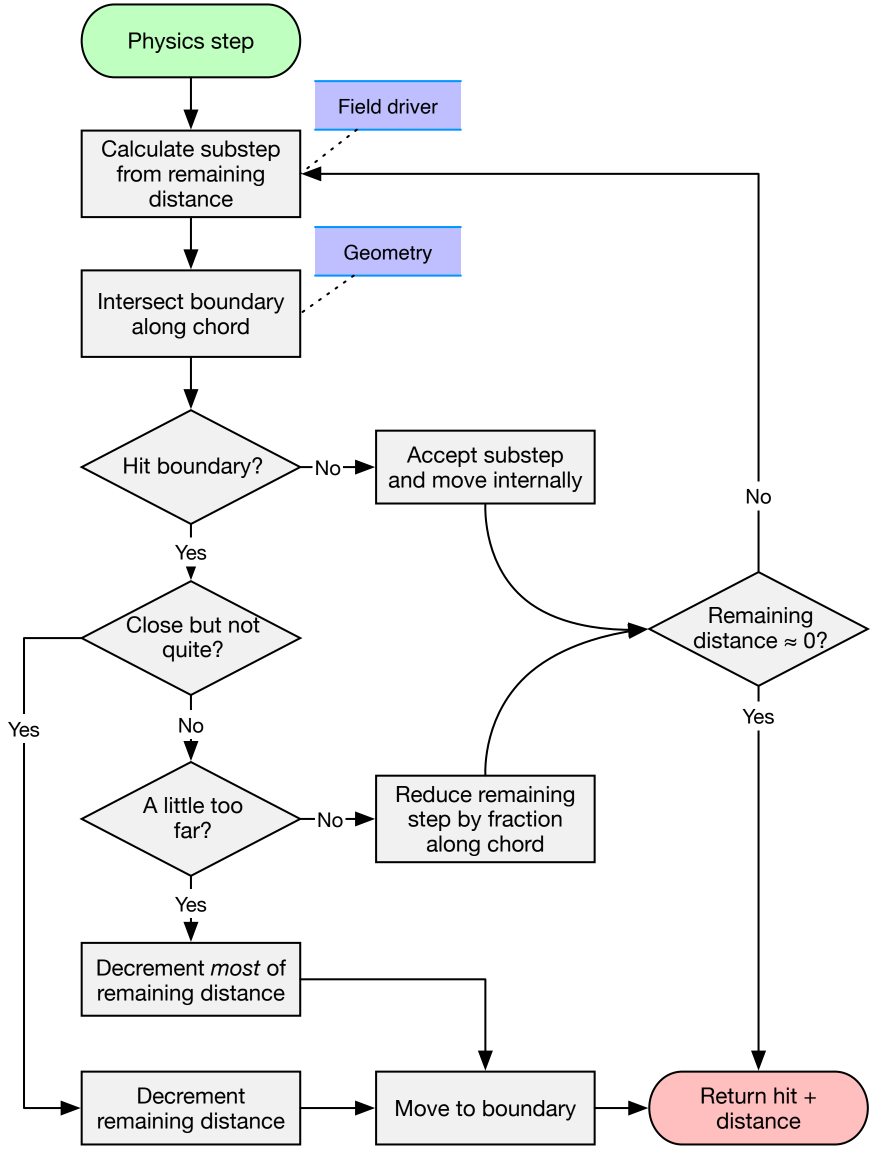
Particle	Process	Model
γ	photon conversion	Bethe–Heitler
	Compton scattering	Klein–Nishina
	photoelectric effect	Livermore
	Rayleigh scattering	Livermore
$e^{\pm}$	ionization	Møller–Bhabha
	bremsstrahlung	Seltzer–Berger
		relativistic
	Coulomb scattering	Urban MSC
e^{+}	annihilation	$\rightarrow(\gamma, \gamma)$
μ	bremsstrahlung	μ brems



- Geant4 Standard EM physics *(verification in progress)*
- VecGeom *(GDML)* + ORANGE *(experimental, AMD/HIP compatible)*

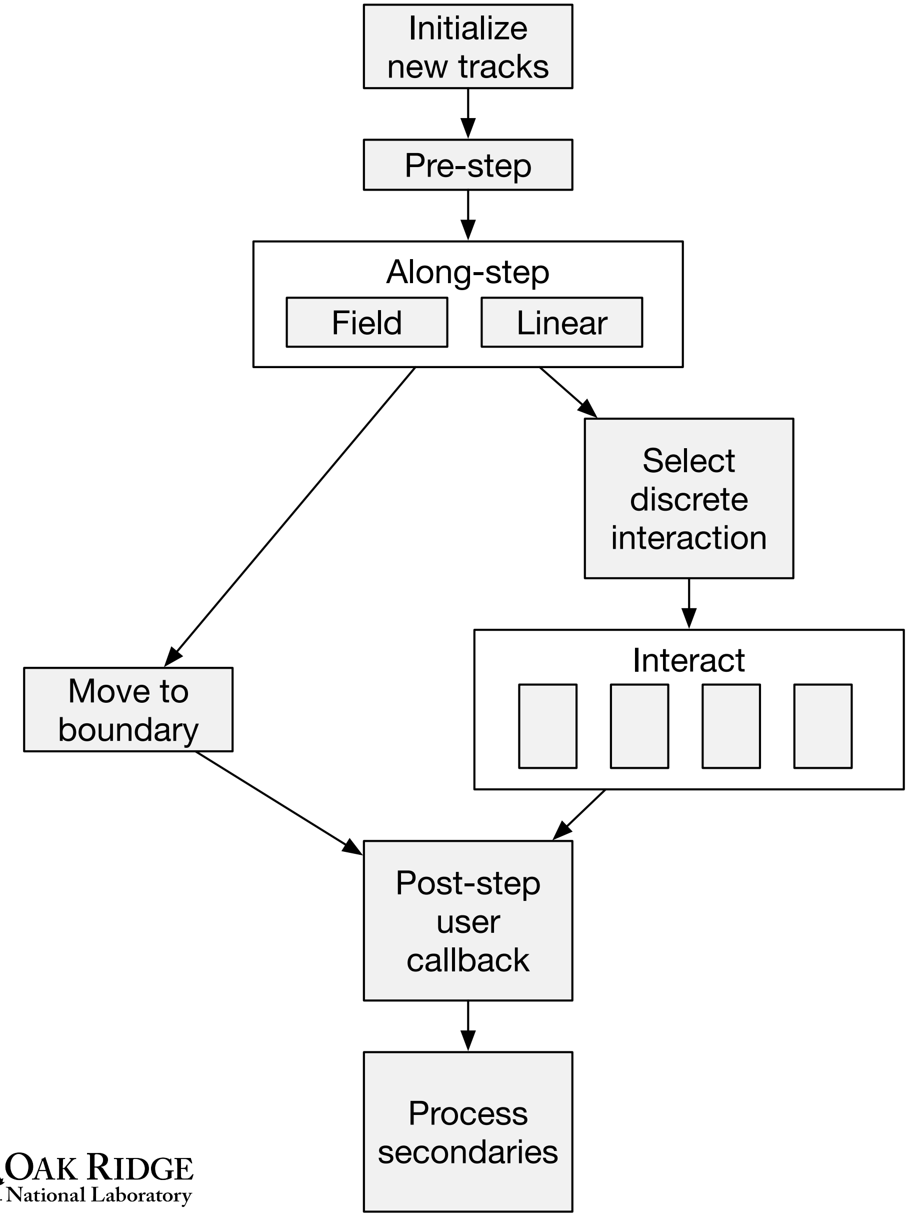
Many differences from Geant4

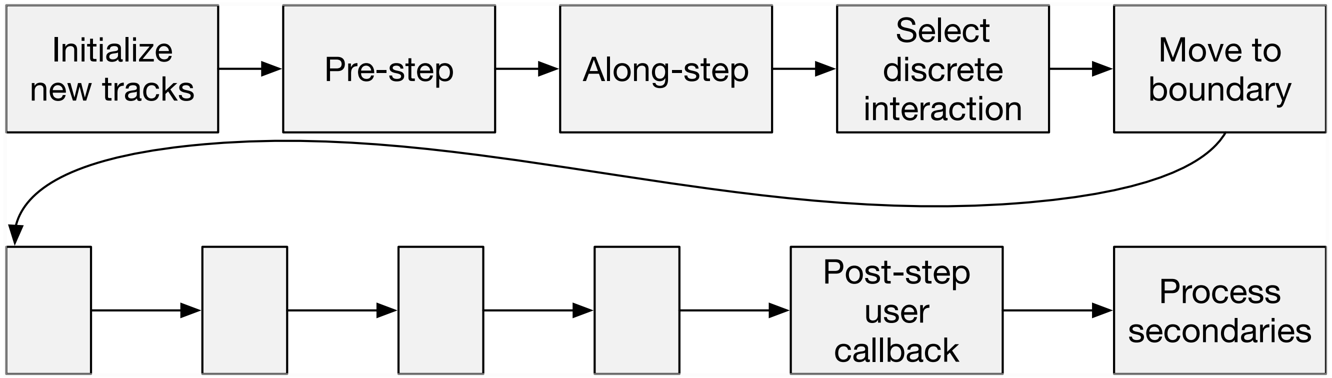
- Code
  - Separate “parameters” and “state”
  - Separate “setup” and “runtime”
  - Type-safe IDs and unit system
- Algorithms
  - XORWOW random number generator
  - Field propagation
  - Event loop
- Physics
  - Cross section interpolation



Field propagator algorithm

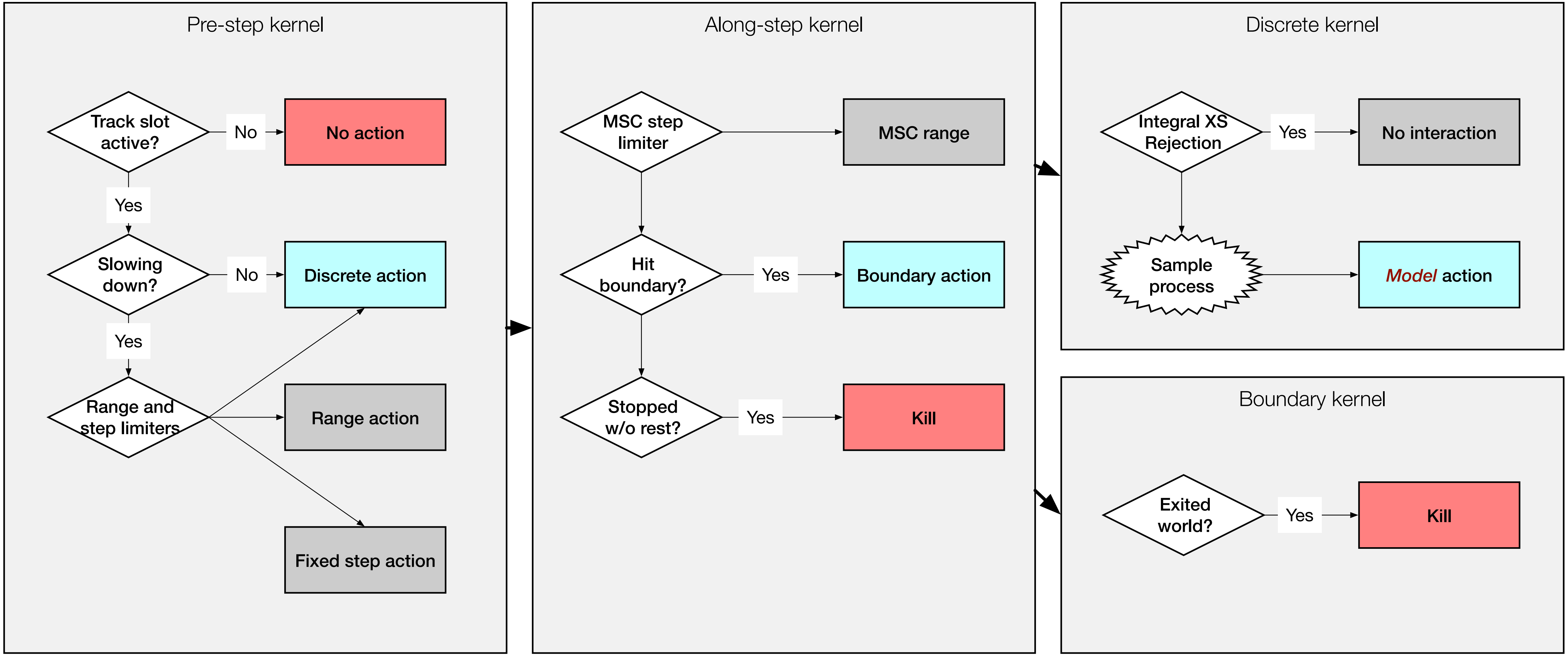
Event loop on a GPU





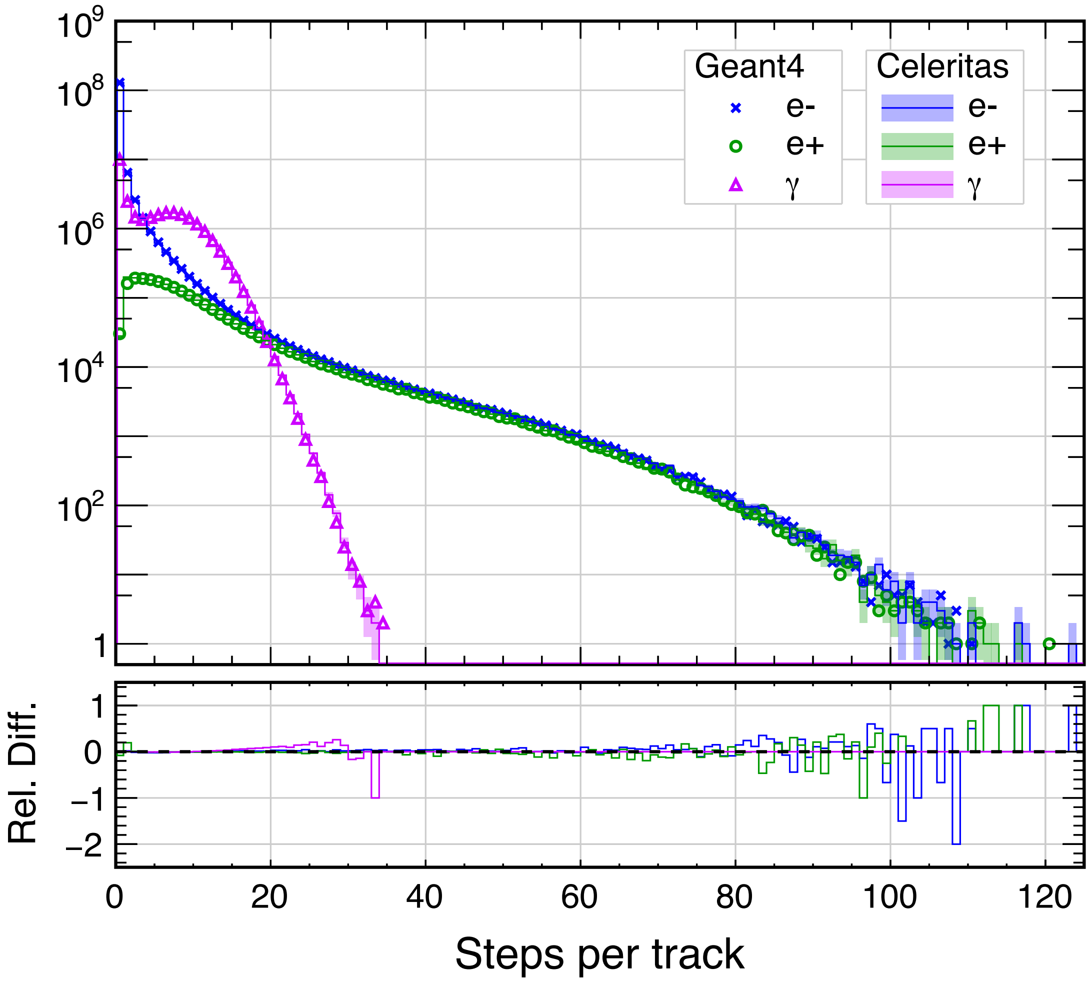
Topological sort: a loop over kernels

Inside a kernel

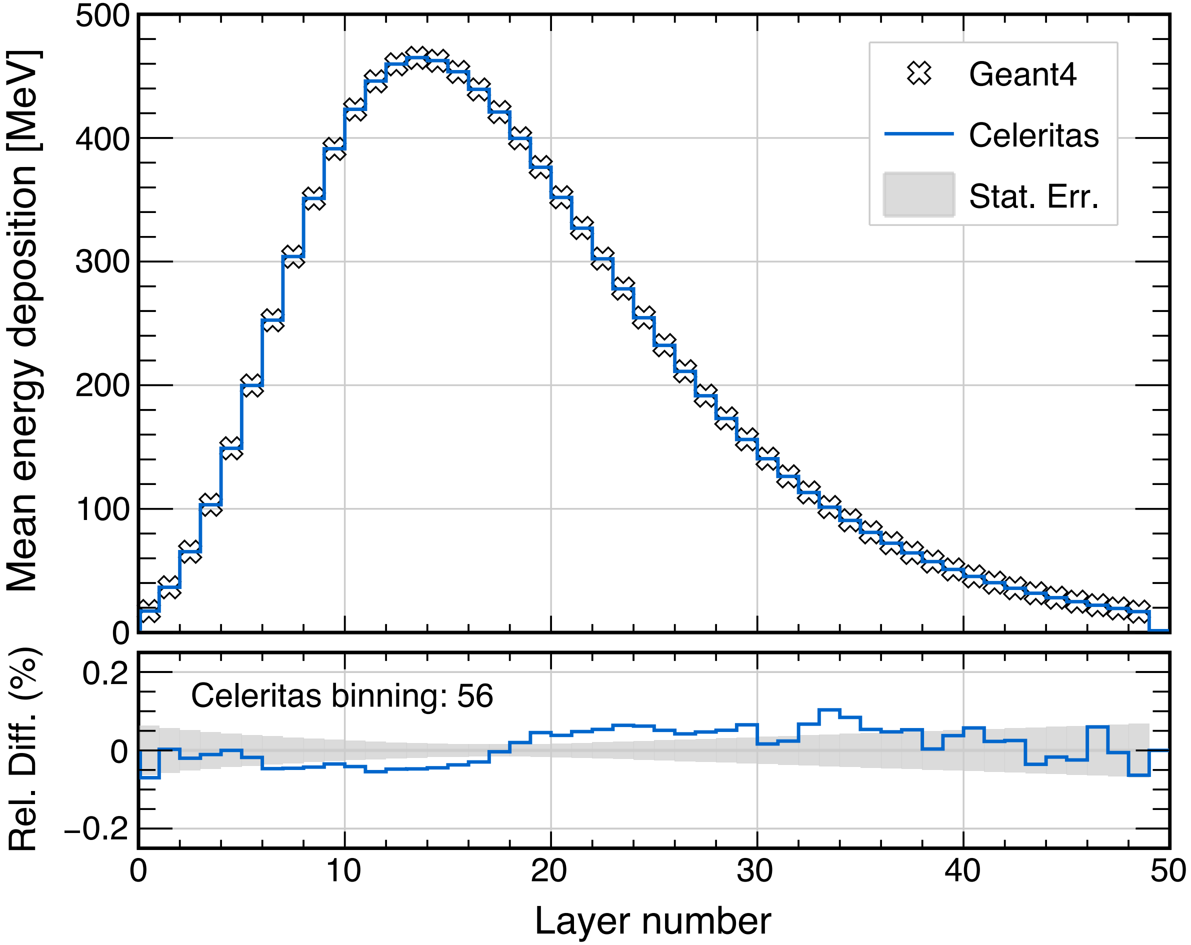


Initial physics code comparison results

- No fields
- No multiple scattering



“Simple CMS” step counts
Isotropic 1 GeV γ



TestEM3 (Pb+LAr) energy deposition
Monodirectional 10 GeV e⁻

Early performance testing

- TestEm3 — simplified calorimeter
  - 50 alternating layers of Pb and lAr
  - 10000 10 GeV electron primaries split between 7 events
- Equivalent configurations of Celeritas/Geant4
  - No magnetic field
  - Disabled multiple scattering, energy loss fluctuations, Rayleigh scattering
  - Excludes initialization time
- No spline interpolation in Celeritas (for now)
  - ~3% performance penalty for Geant4 with spline
  - Compensate by using 8× cross section grid points: <2% slower

Early performance testing (results)

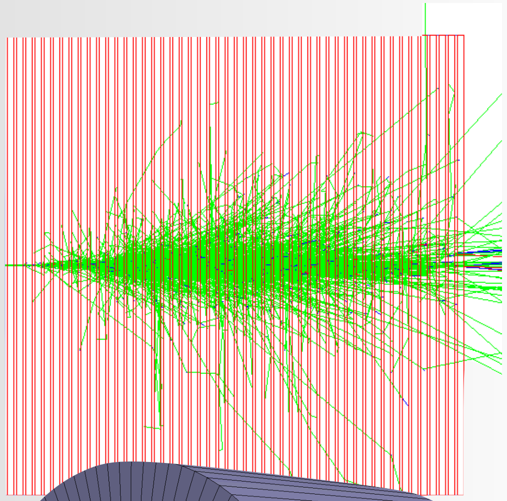
- 1–2 batches of 6 simultaneous runs on Summit (OLCF)
  - CPU (Power9): multithreaded with 7 cores
  - GPU (V100): one CPU plus one GPU
- 30–45× faster with GPUs
  - Apples-to-apples: Celeritas CPU vs GPU
  - Similar order-of-magnitude improvement irrespective of code
  - 210–315 CPU core to GPU equivalence

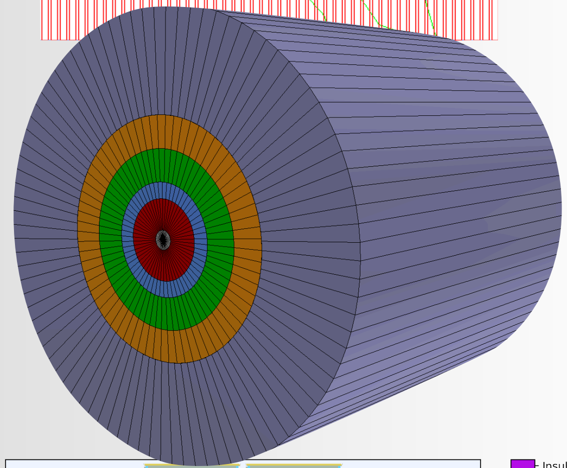
Work rate (events/s)

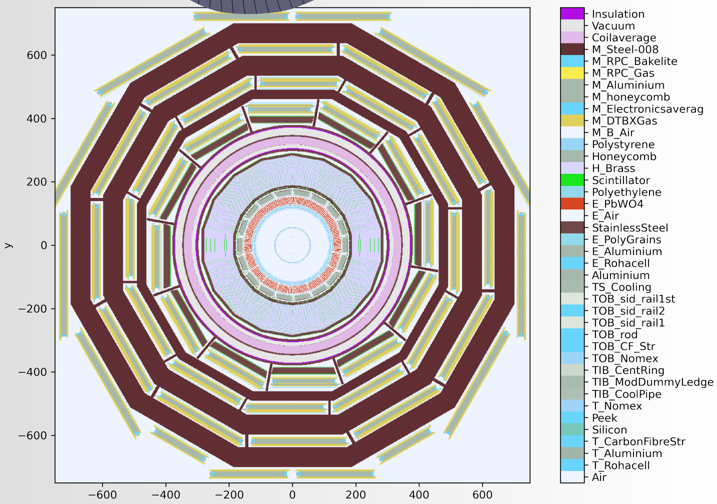
	geo	arch	mean	σ
Geant4 10.7.1	Geant4	CPU	0.24	0.010
Celeritas 8d83ebab (29 Apr 2022)	ORANGE	CPU	0.33	0.003
		GPU	15.09	0.375
	VecGeom	CPU	0.36	0.006
		GPU	11.17	0.075

New regression/timing suite

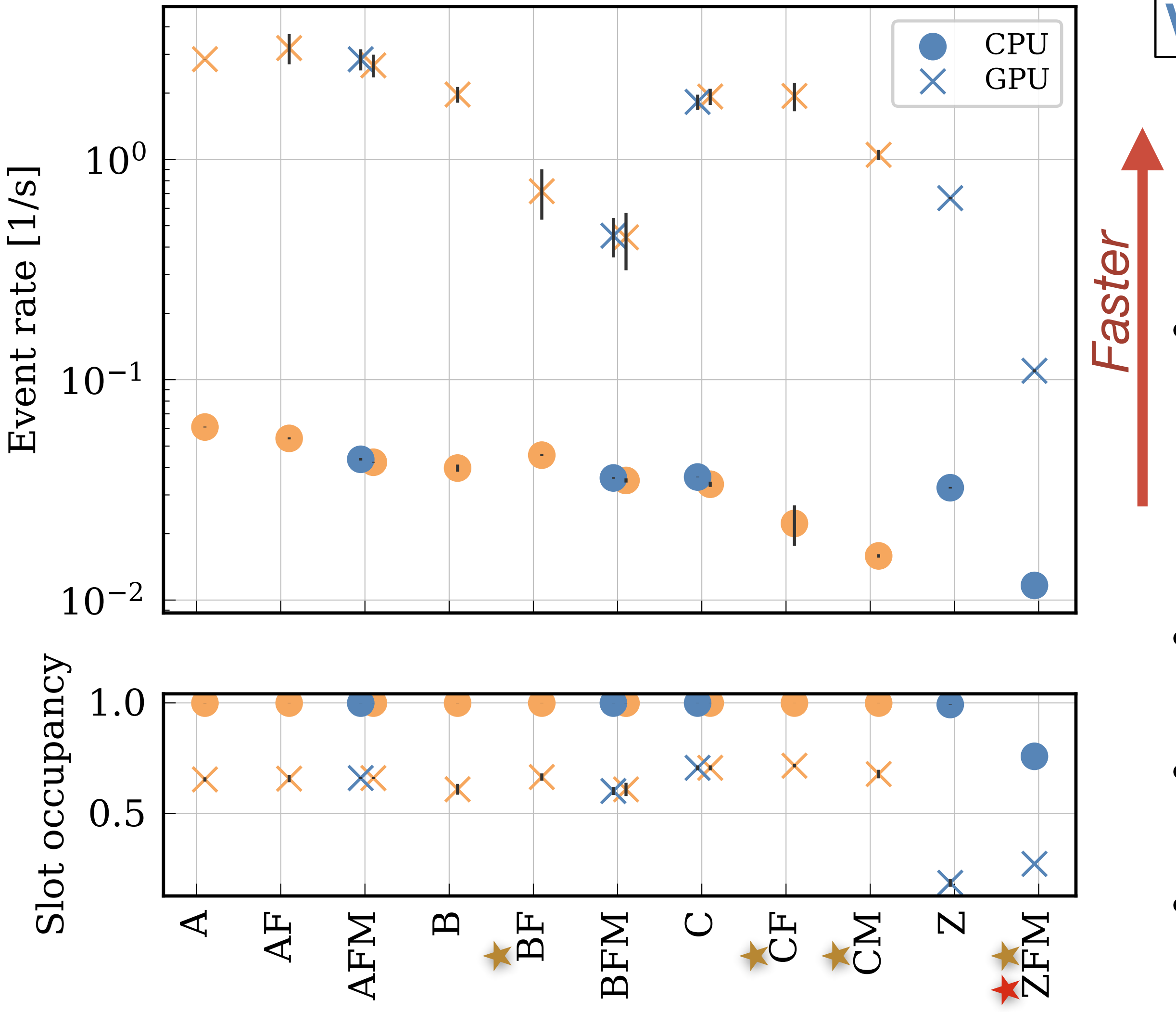
- **1300 10 GeV electrons per event, 7 events per run** (1 per CPU, 7 per GPU)
- **Very preliminary set of problem definitions** *(working with AdePT team to develop)*
- **Not currently optimized** *(more on this later)*
- **Run on single node of Summit** (6 separate runs simultaneously, different seed for each)
- **Initial results are apples-to-apples** (Celeritas only)







New performance results



*unconverged

*some instances fail

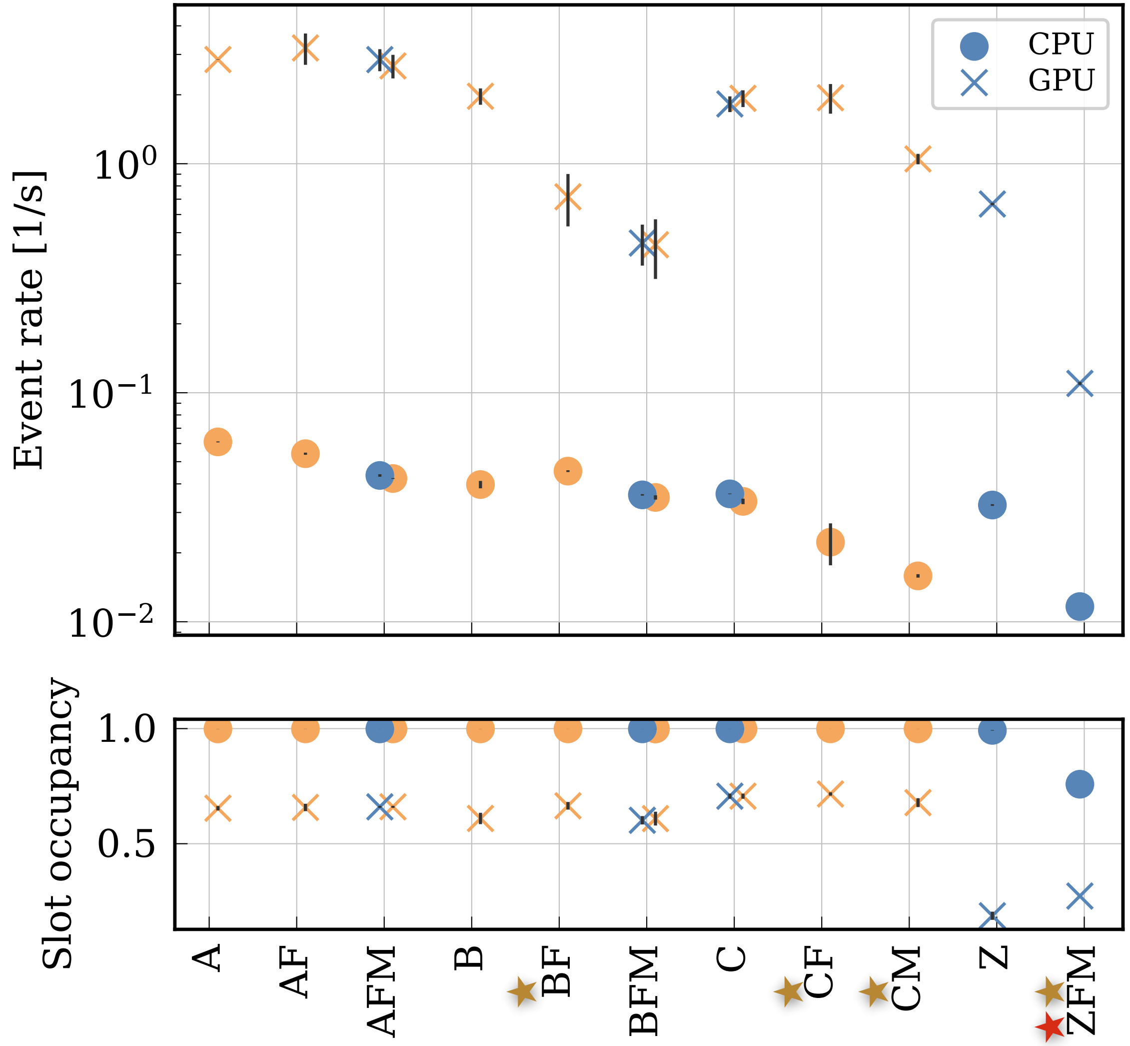
ORANGE
VecGeom

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

	Modifier
F	+field
M	+msc

- Higher “slot occupancy” *(fraction of GPU track slots in use)* → better performance
- MSC slows GPU tracking by 2×
- Occasional tracking failures in field
- ORANGE and VecGeom show approximate performance parity

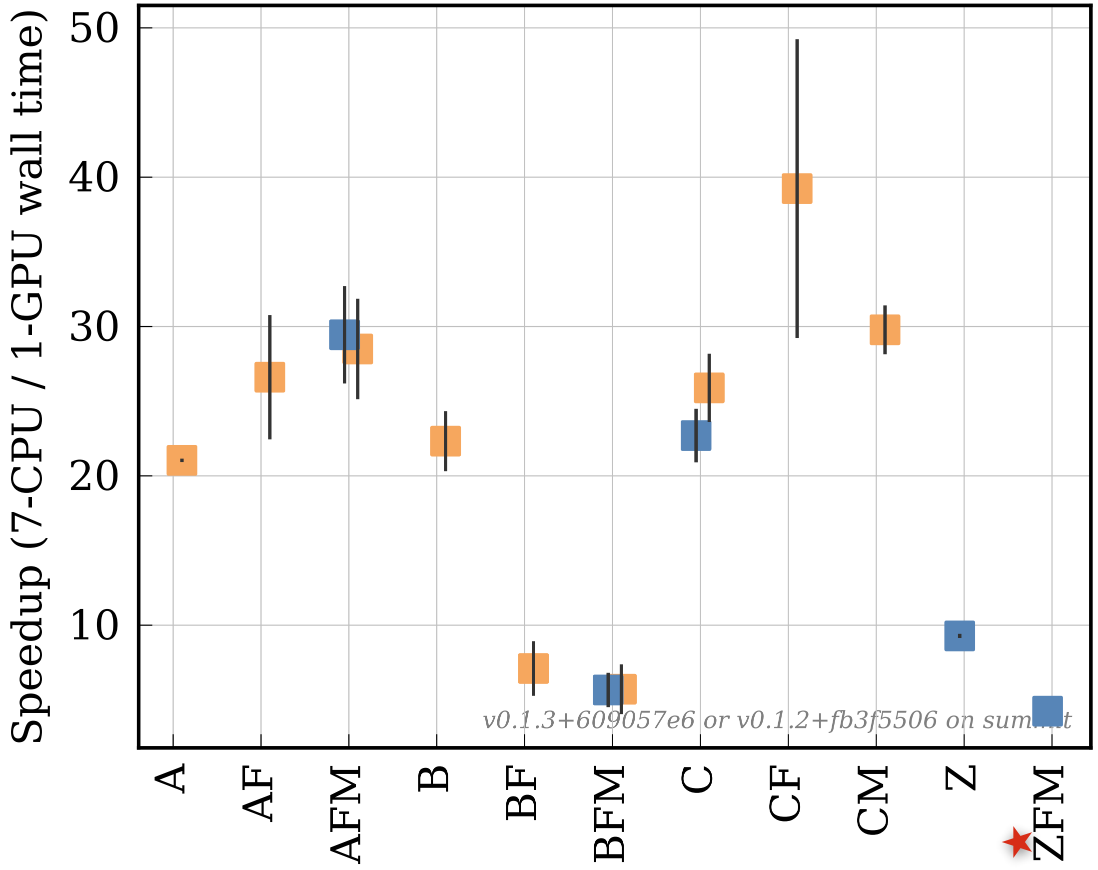
New performance results



ORANGE	
VecGeom	

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

	Modifier
F	+field
M	+msc



*unconverged

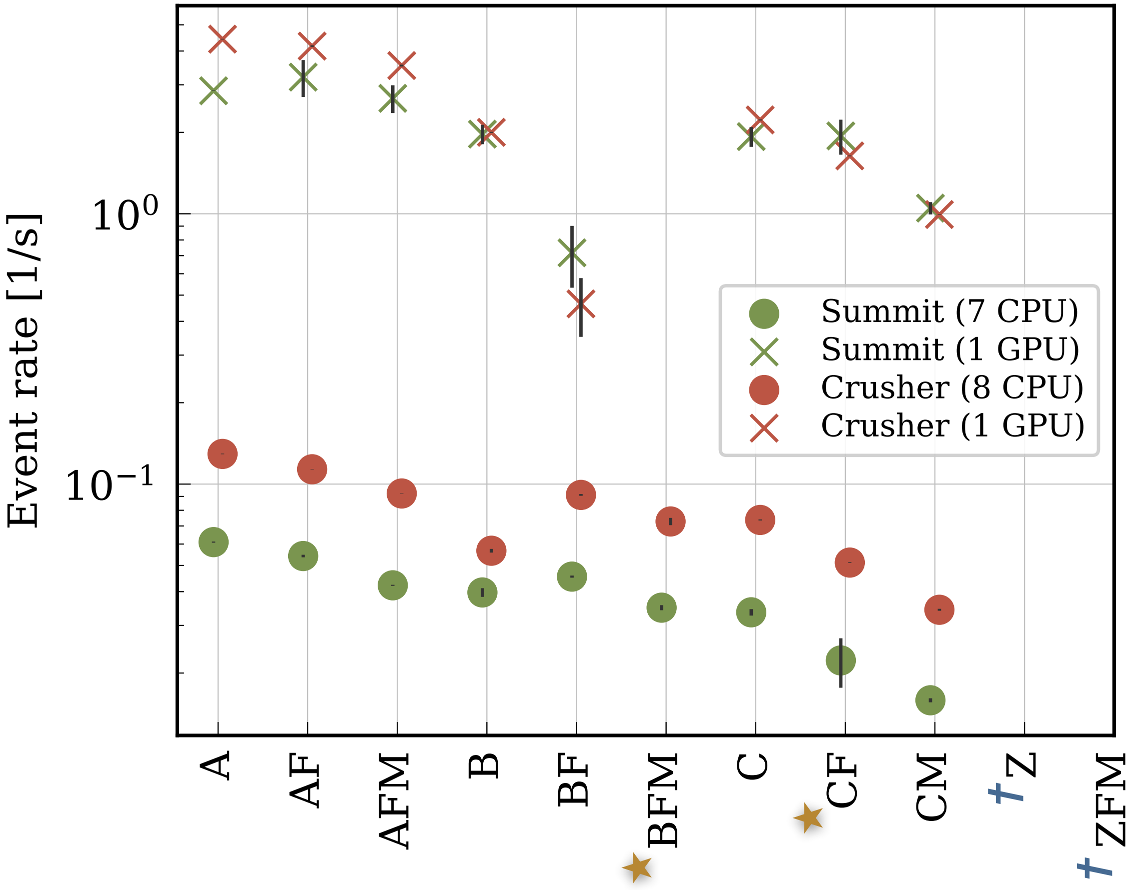
*some instances fail

Multiply speedup by 7× for CPU:GPU equivalence

Crusher (pre-Frontier) performance

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

	Modifier
F	+field
M	+msc



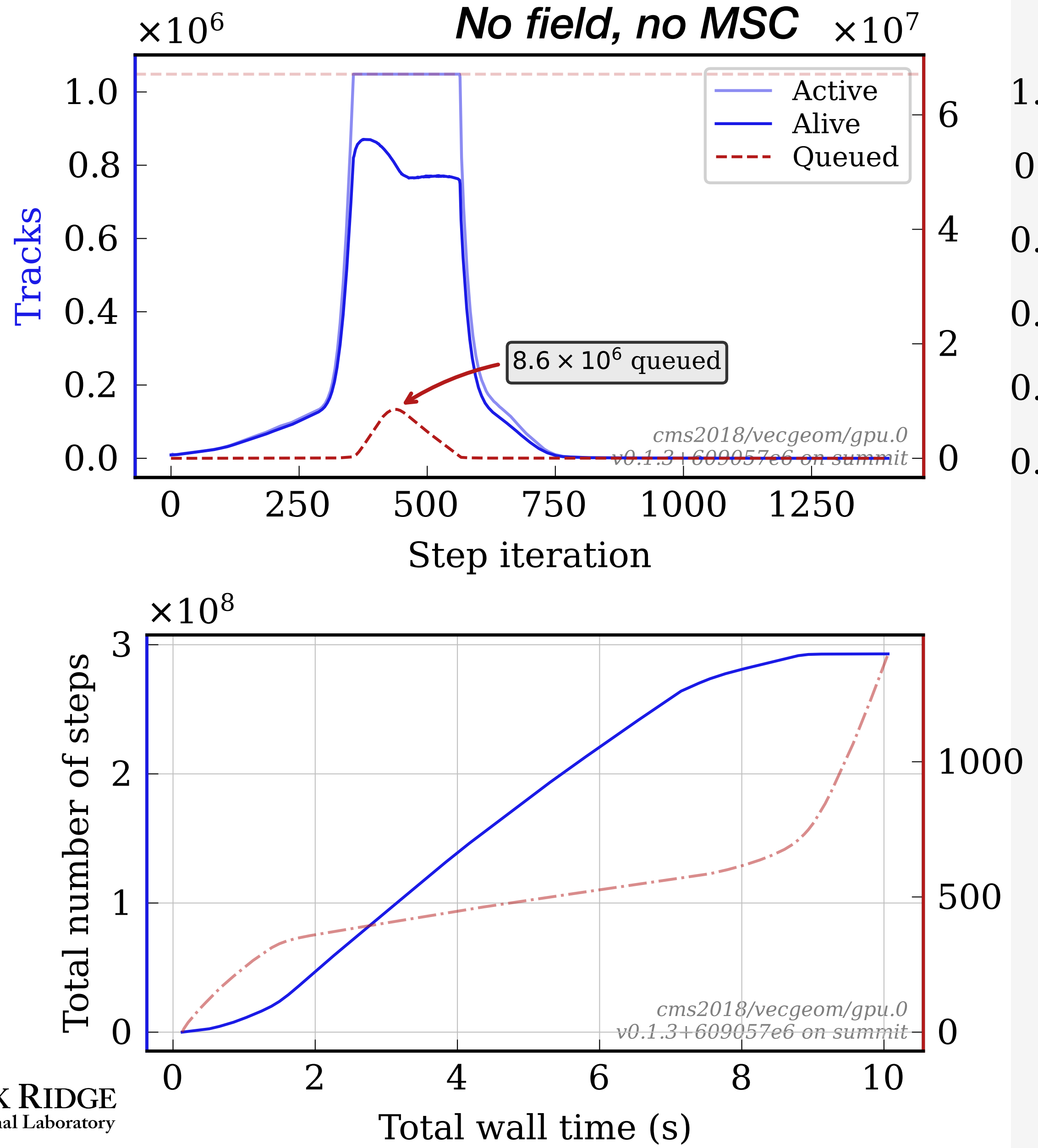
*some instances fail
†requires VecGeom

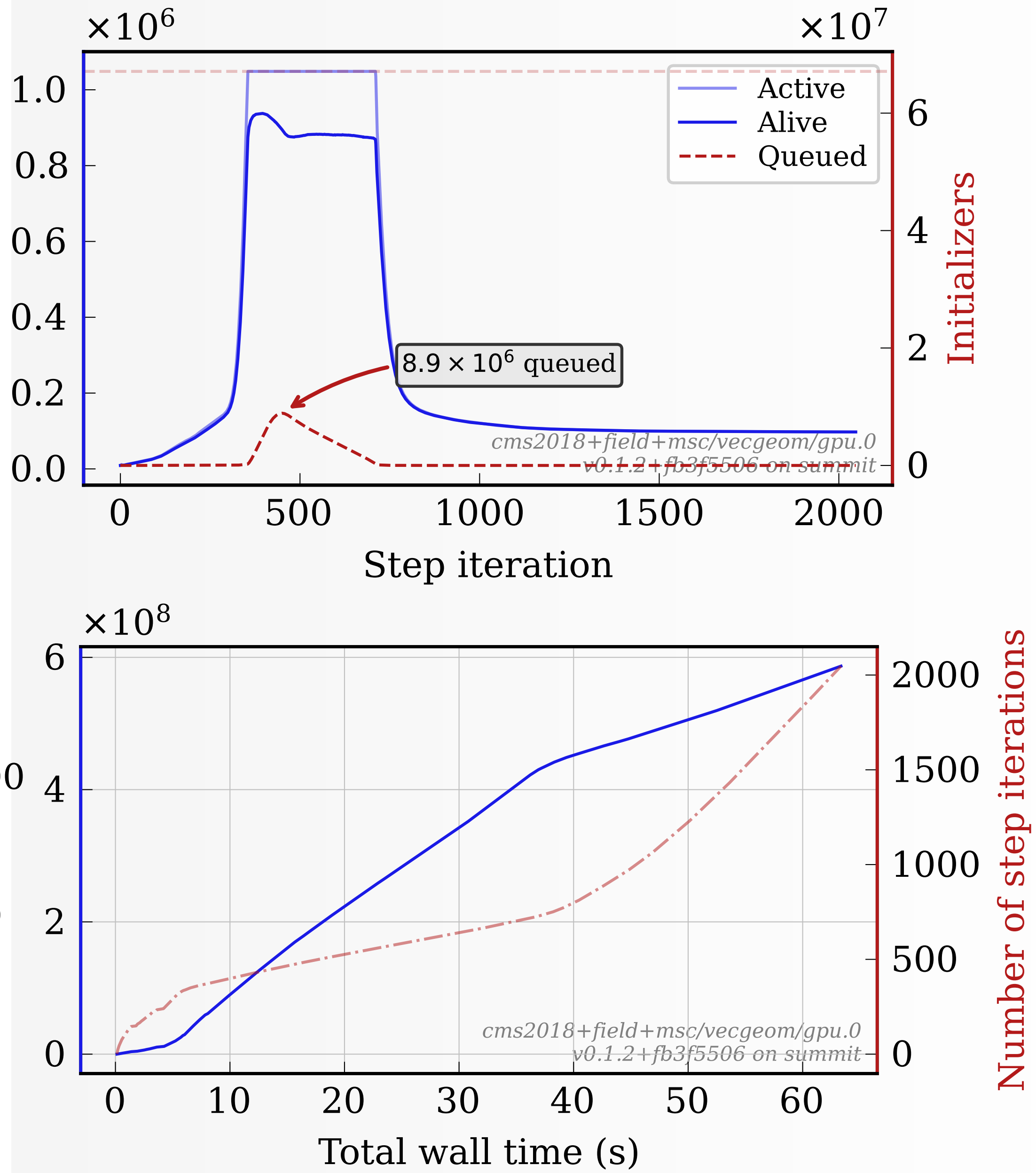
- CPUs and GPUs are faster than those on Summit
- All ORANGE geometry
- Field propagation failures are being fixed
- No AMD-specific optimizations have been made

Oak Ridge National Laboratory

CMS GPU step performance

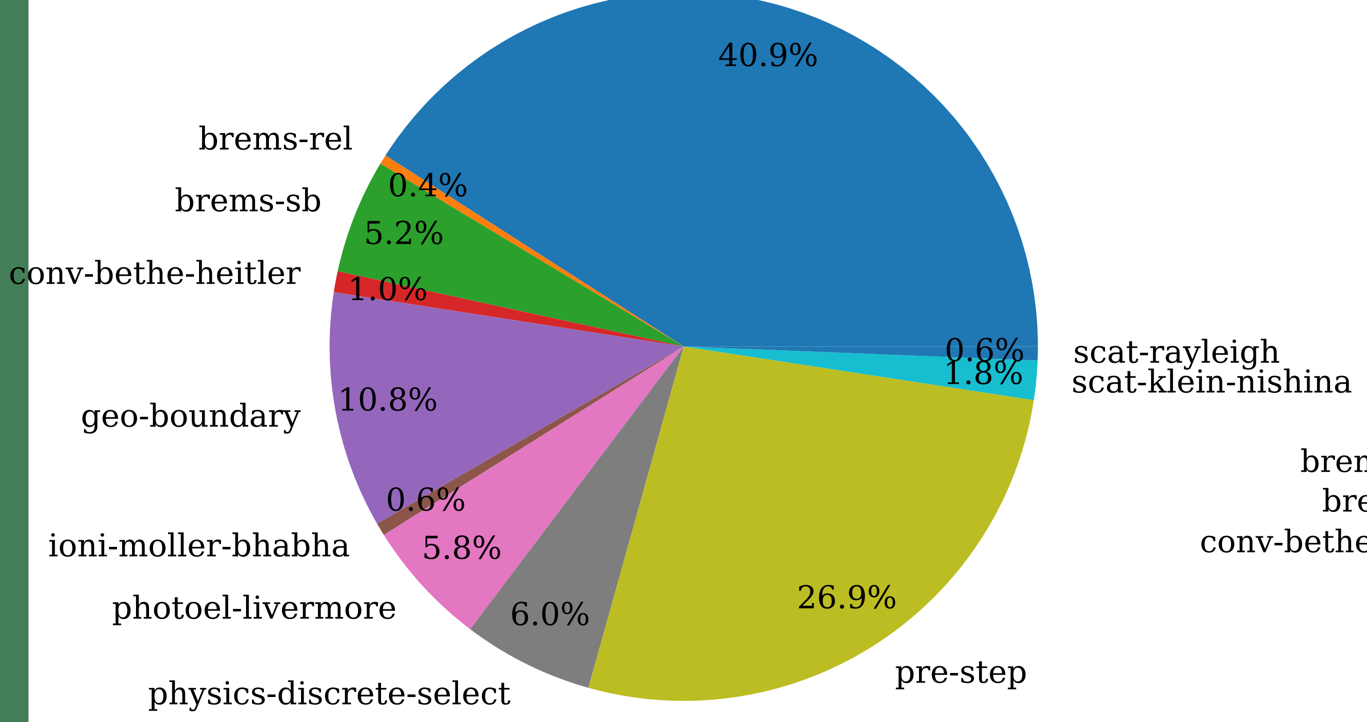
1T mag field along Z, Urban MSC



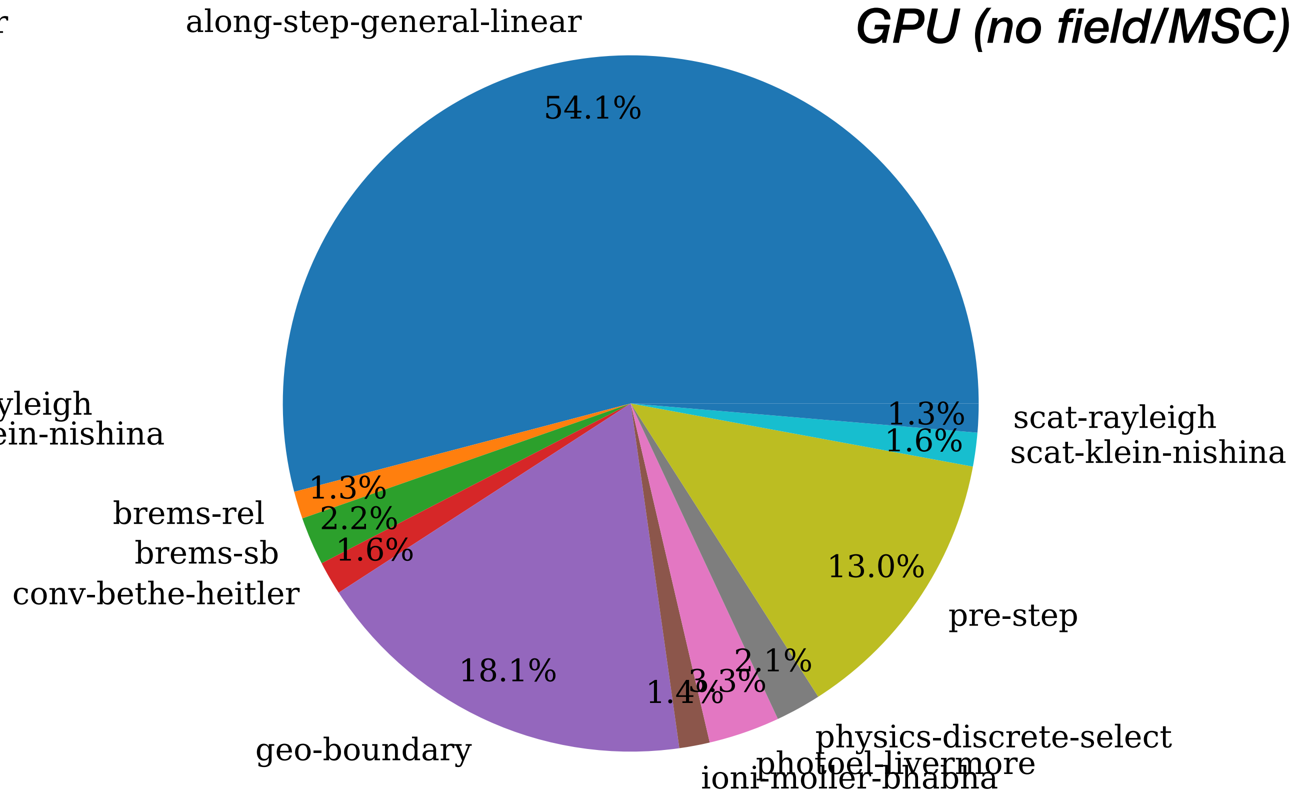


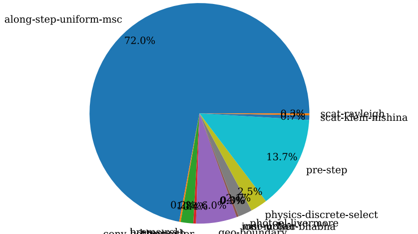
Fractional kernel time (CMS2018)

CPU (no field/MSC)

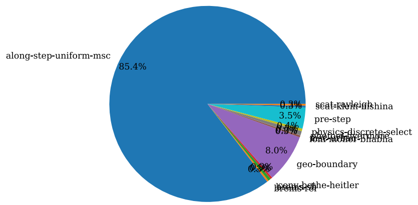


GPU (no field/MSC)





CPU (field+MSC)



GPU (field+MSC)

Lazy and (probably) inefficient choices for now

- Same “field propagator” and MSC code path for neutral tracks
  (multiple along-step kernels switching on type/energy)
- Always launching threads for all track slots on every kernel
  (partition tracks based on action and launch smaller grids)
- All primaries start simultaneously
  (wait until “peak” of shower before starting a new event)
- No inter-thread cooperation for physics/geometry
  (parallelize work by launching additional threads)

Increase maximum track count
Cache log-energy
Add launch bounds
Use `ldg` for improved caching
Improve memory layout
Increase kernel occupancy

Additional optimizations to do

Key areas of continuing work

- Physics ***validation*** (physics models, progression problems, experiment-specific)
- Experiment ***integration*** (Acceleritas + direct)
- Performance ***experimentation*** (there's a long list)
- International ***collaboration*** (AdePT, VecGeom, CMS, ORANGE)

Conclusions

- Celeritas is a new *specialized* detector simulation code
- Current test problems show **~10–30× performance boost** using GPUs on Summit (70–210× GPU/CPU core equivalence)
- Laundry list of fixes, features, validation, optimization to do
- Version 0.1.3 now available (install from source or Spack)

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

Acknowledgments

Celeritas v0.1.0 contributors:

- Philippe Canal (@pcanal)
- Doaa Deeb (@DoaaDeeb)
- Tom Evans (@tmdelellis)
- Seth R Johnson (@sethrj)
- Soon Yung Jun (@whokion)
- Guilherme Lima (@mrguilima)
- Amanda Lund (@amandalund)
- Ben Morgan (@drbenmorgan)
- Vincent R Pascuzzi (@vrpascuzzi)
- Paul Romano (@paulromano)
- Stefano C Tognini (@stognini)

ECP: This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation’s exascale computing imperative.

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.

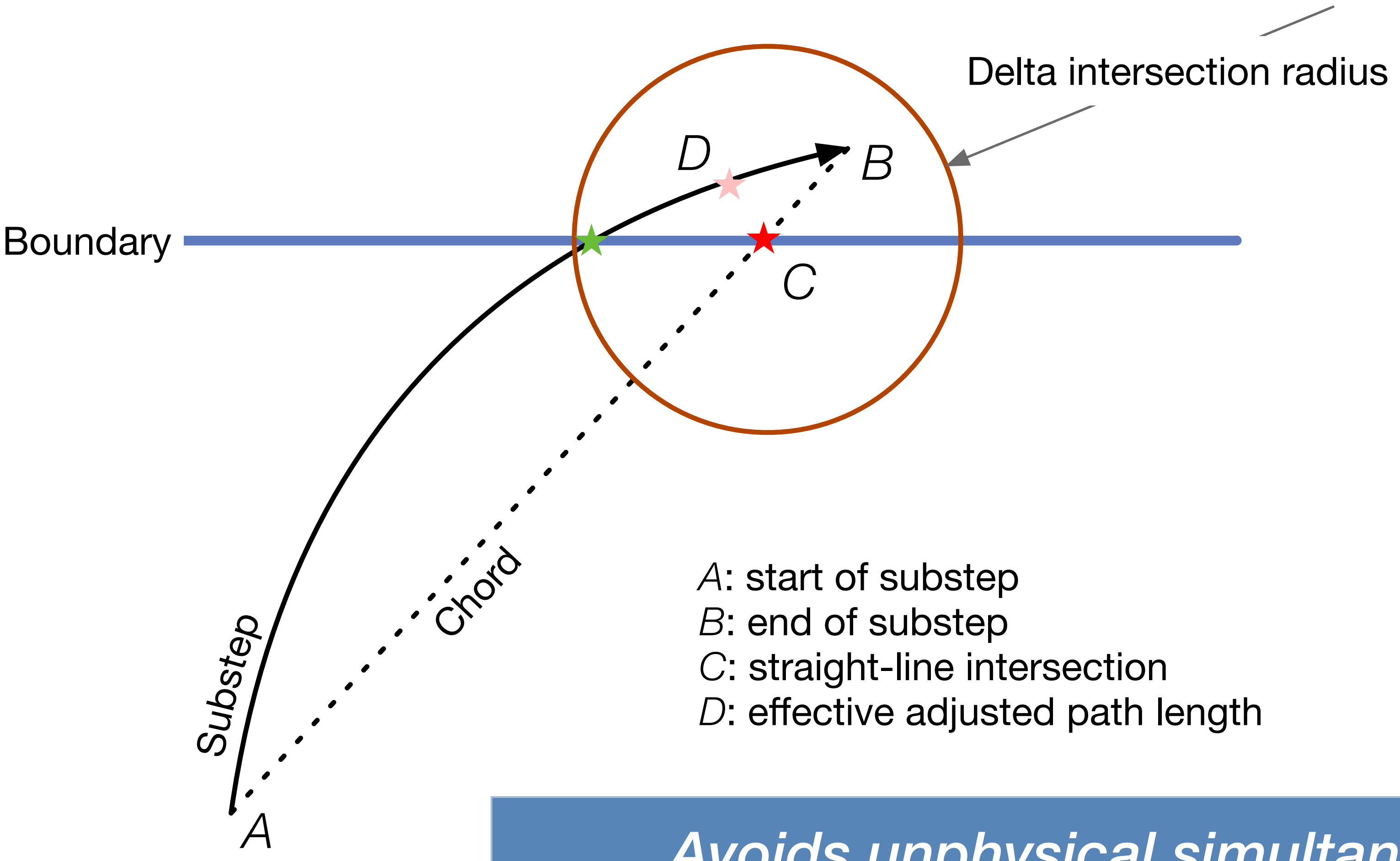
https://github.com/celeritas-project/celeritas

Backup slides

Full benchmark results

problem	geo	arch	steps/primary	time/primary	time/step	steps	time	unconv	occ
cms2018	vecgeom	cpu	7.155E+04	2.372E-02	3.316E-07	2.931E+08	97.2	0.0	0.99
cms2018	vecgeom	gpu	3.222E+04	1.154E-03	3.582E-08	2.932E+08	10.5	0.0	0.19
cms2018+field+msc	vecgeom	cpu	5.243E+05	3.722E-01	7.099E-07	2.147E+09	1524.5	181427.5	1.00
cms2018+field+msc	vecgeom	gpu	6.437E+04	7.005E-03	1.088E-07	5.858E+08	63.7	97231.0	0.27
simple-cms+field+msc	vecgeom	cpu	6.193E+04	3.126E-02	5.047E-07	2.537E+08	128.0	1.5	0.13
simple-cms+field+msc	vecgeom	gpu	2.765E+04	2.166E-03	7.834E-08	2.516E+08	19.7	15.2	0.12
simple-cms+msc	orange	cpu	6.125E+04	1.875E-02	3.062E-07	2.509E+08	76.8	0.0	1.00
simple-cms+msc	orange	gpu	2.757E+04	3.917E-04	1.421E-08	2.509E+08	3.6	0.0	0.61
testem15	orange	cpu	5.003E+04	1.253E-02	2.504E-07	2.049E+08	51.3	0.0	1.00
testem15	orange	gpu	2.252E+04	2.695E-04	1.196E-08	2.049E+08	2.5	0.0	0.65
testem15+field	orange	cpu	5.008E+04	1.403E-02	2.802E-07	2.051E+08	57.5	0.0	1.00
testem15+field	orange	gpu	2.254E+04	2.504E-04	1.111E-08	2.051E+08	2.3	0.0	0.66
testem15+field+msc	orange	cpu	5.008E+04	1.808E-02	3.610E-07	2.051E+08	74.1	0.0	1.00
testem15+field+msc	orange	gpu	2.254E+04	2.658E-04	1.179E-08	2.051E+08	2.4	0.0	0.66
testem15+field+msc	vecgeom	cpu	5.008E+04	1.757E-02	3.509E-07	2.051E+08	72.0	0.0	1.00
testem15+field+msc	vecgeom	gpu	2.254E+04	2.825E-04	1.253E-08	2.051E+08	2.6	0.0	0.66
testem3-flat	orange	cpu	1.024E+05	2.257E-02	2.204E-07	4.193E+08	92.4	0.0	1.00
testem3-flat	orange	gpu	4.608E+04	3.824E-04	8.298E-09	4.193E+08	3.5	0.0	0.71
testem3-flat	vecgeom	cpu	1.024E+05	2.158E-02	2.108E-07	4.193E+08	88.4	0.0	1.00
testem3-flat	vecgeom	gpu	4.608E+04	4.298E-04	9.328E-09	4.193E+08	3.9	0.0	0.71
testem3-flat+field	orange	cpu	1.024E+05	3.082E-02	3.009E-07	4.195E+08	126.2	0.0	1.00
testem3-flat+field	orange	gpu	4.612E+04	4.073E-04	8.831E-09	4.197E+08	3.7	0.0	0.71
testem3-flat+msc	orange	cpu	1.362E+05	4.760E-02	3.495E-07	5.578E+08	195.0	0.0	1.00
testem3-flat+msc	orange	gpu	6.129E+04	7.658E-04	1.249E-08	5.578E+08	7.0	0.0	0.68

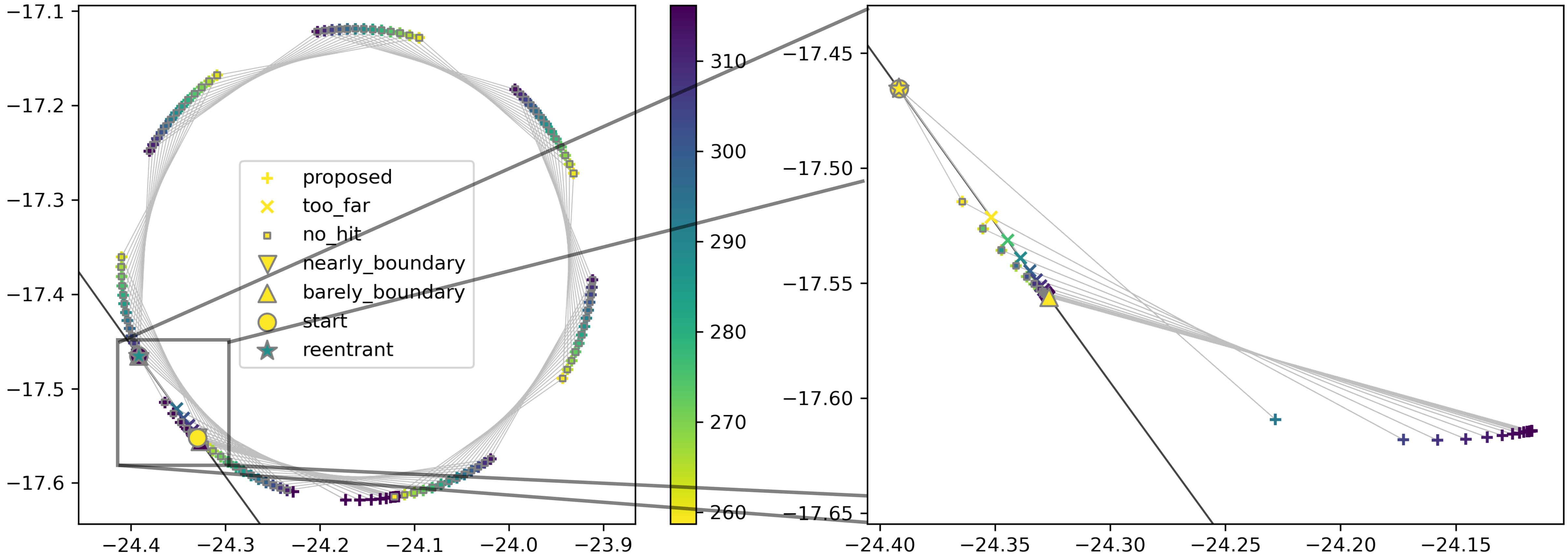
Field propagator near-miss



A: start of substep
B: end of substep
C: straight-line intersection
D: effective adjusted path length

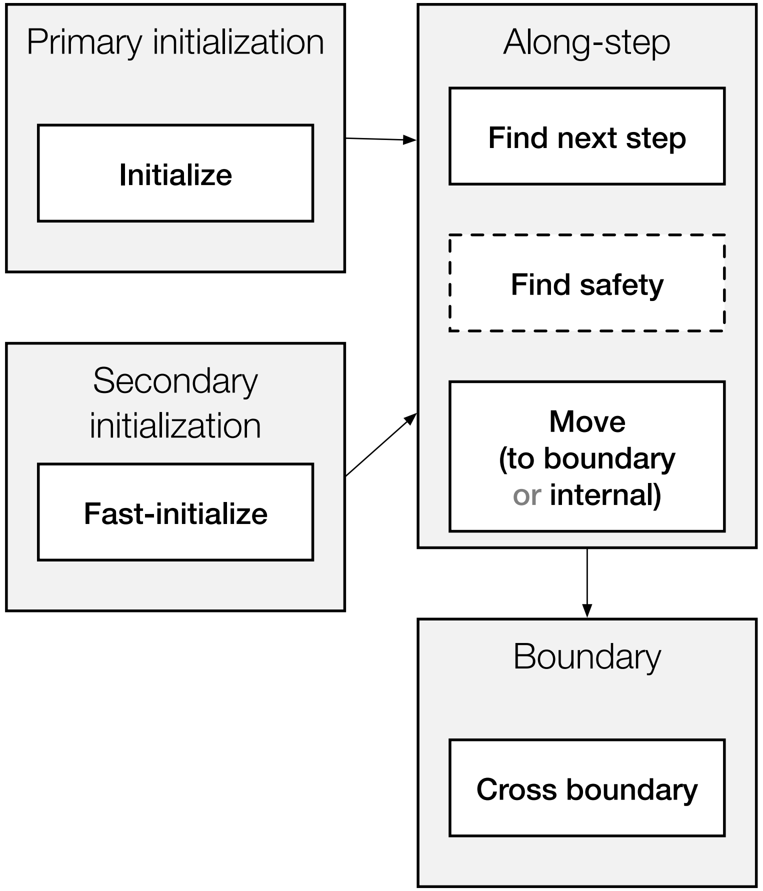
Avoids unphysical simultaneous interaction & boundary crossing

Field propagation edge case

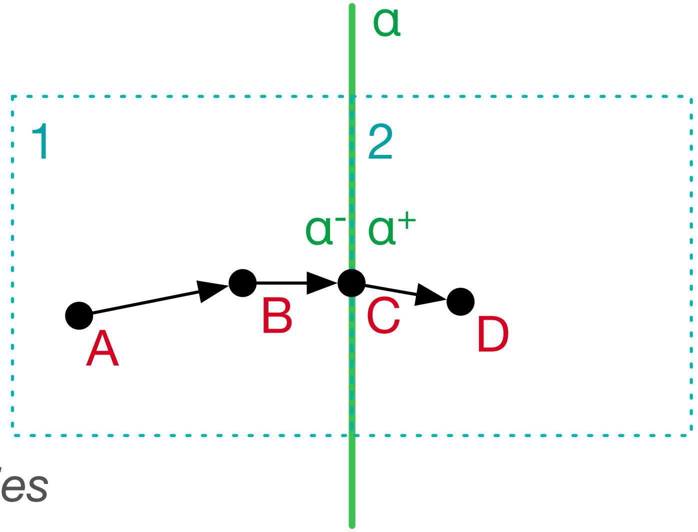


ORANGE surface-based tracking methodology

Celeritas geometry interface

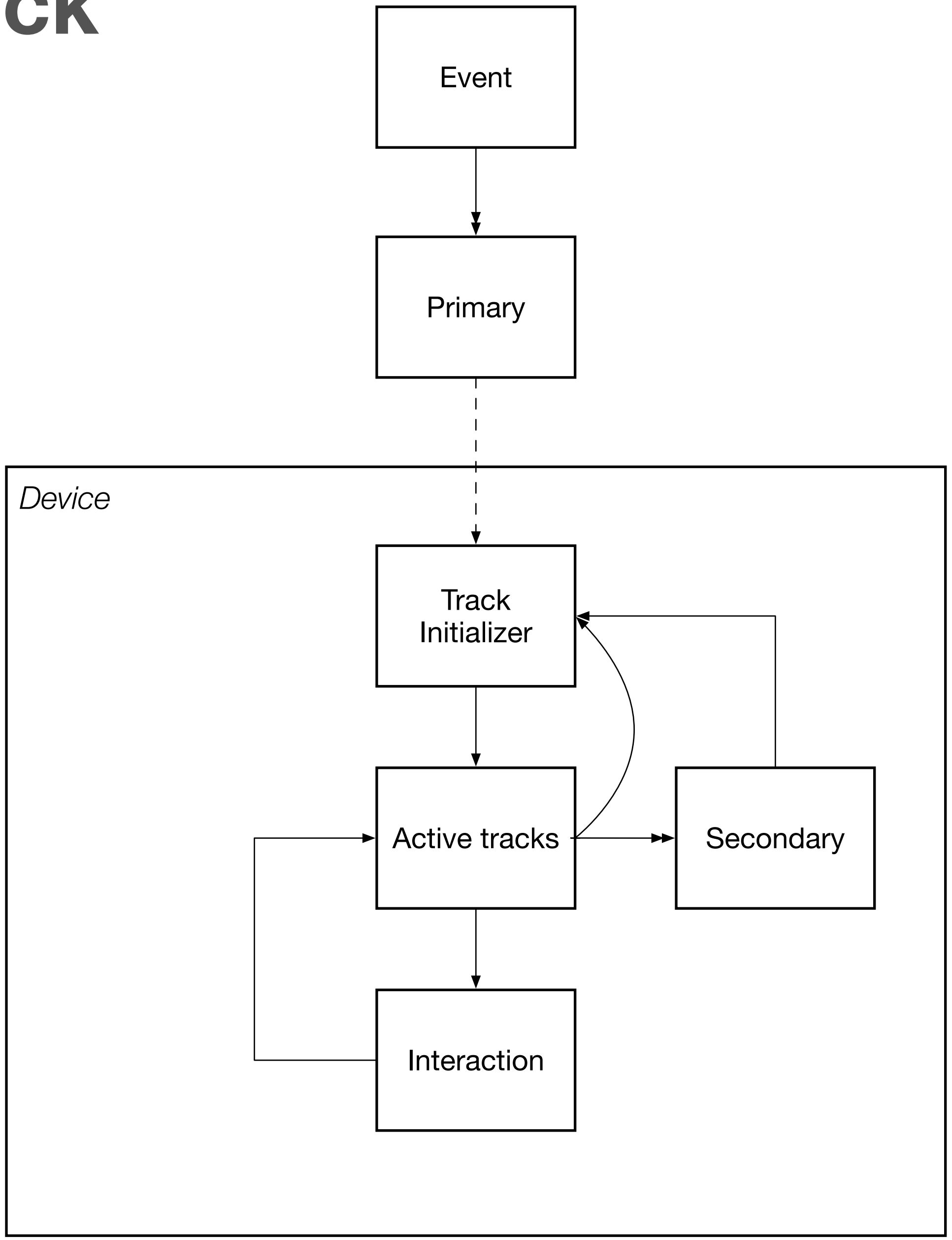


	Position	Volume	Surface+Sense
(input)	A	—	
Initialize	A	1	—
Find step	A	1	—
Move internal	B	1	—
Move to bdy	C	1	α inside
Cross bdy	C	2	α outside
Move internal	D	2	—



*exact handling of direction changes on boundaries

Life cycle of a track



OAK RIDGE National Laboratory