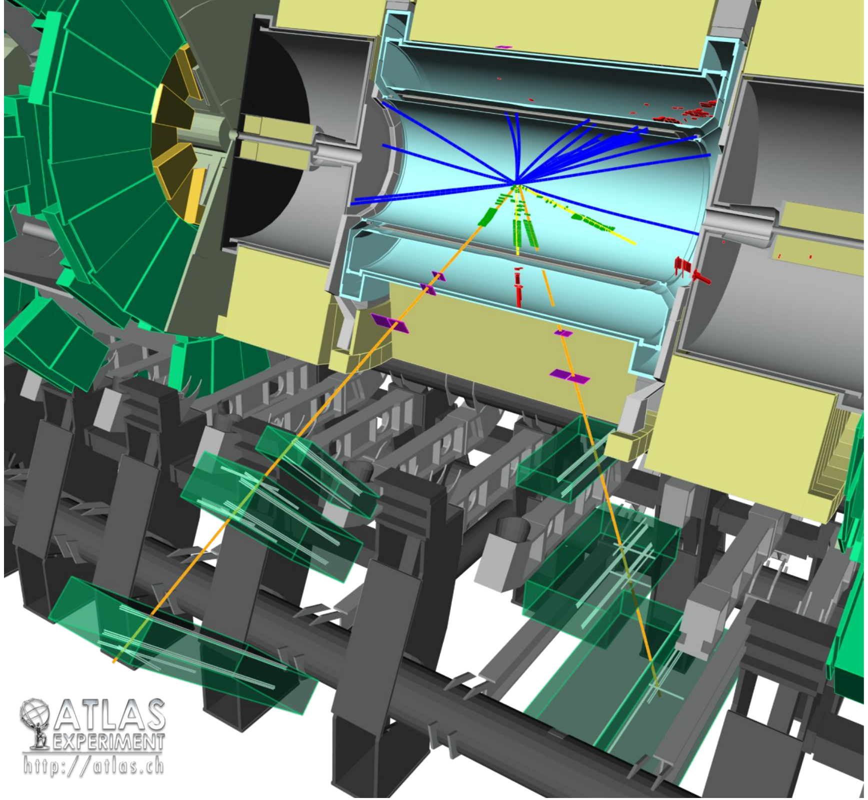
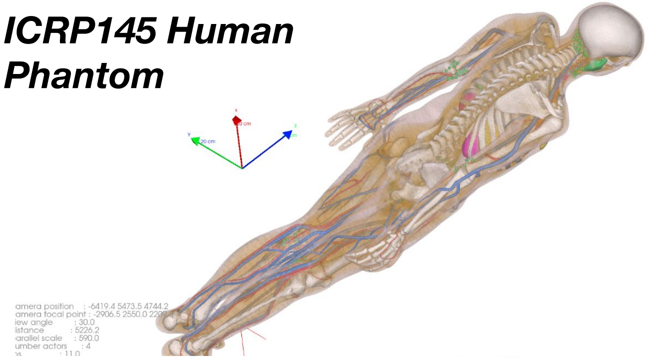
Detector Simulations in Particle Physics

Ben Morgan

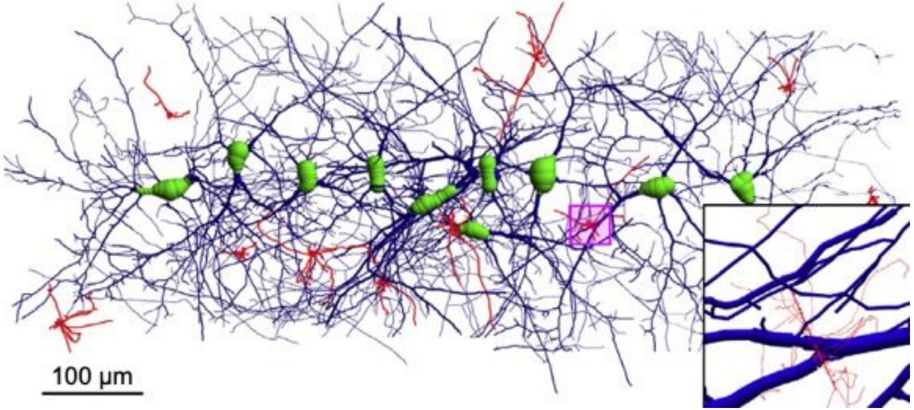
THE UNIVERSITY OF WARWICK



ATLAS Detector



ICRP145 Human Phantom



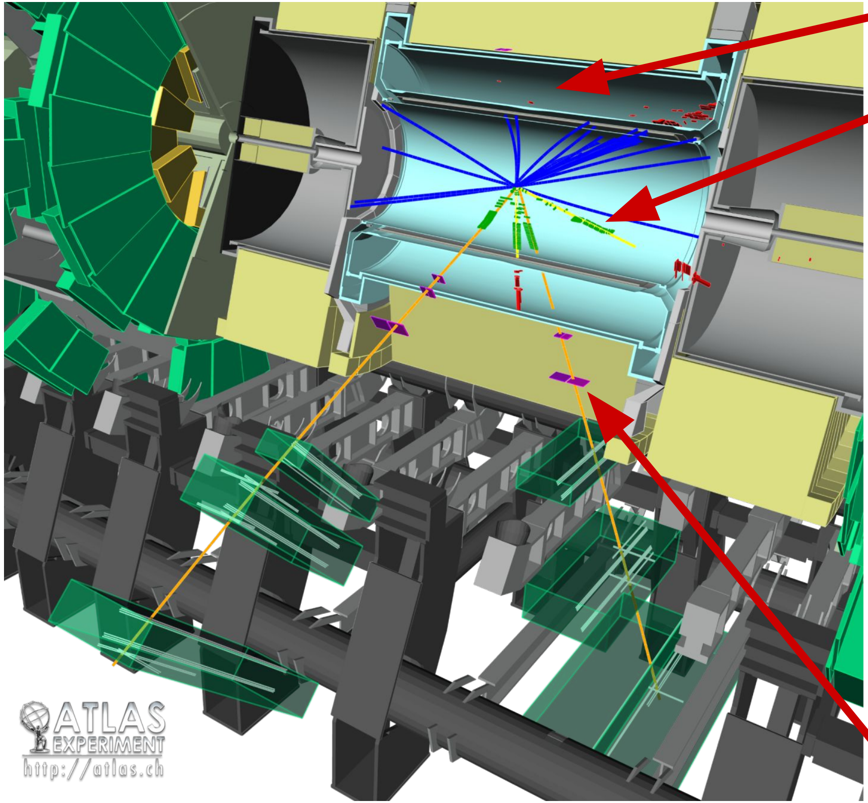
[***Radiation Damage to Neurons***]

Detectors for particle radiation

Scales from LHC to Molecules! Take "Detector" as anything in which particle radiation can interact.

Why Simulate Detectors/Particle Transport?

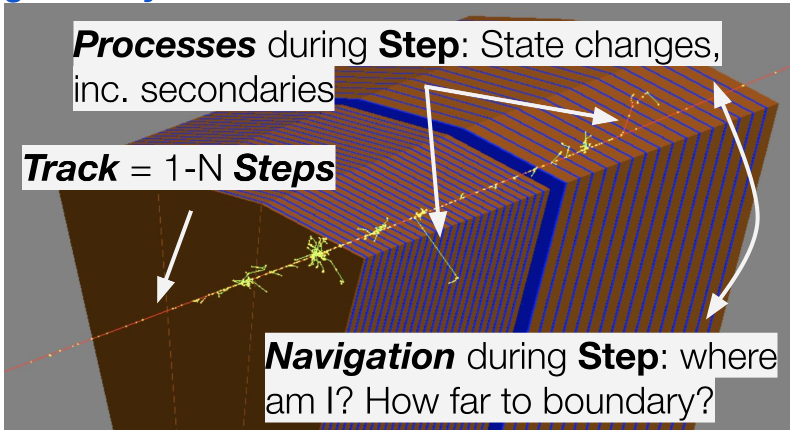
- Design and optimize **new** detectors
  - *Predict response (observables) for signal events…*
  - *… and for backgrounds to signal*
- Model/develop **existing** detectors during operation
  - *Changes in those responses due to modifications of geometry/conditions*
- Simulated data samples an important input to:
  - *Development/training of analyses for physics we want to measure*
  - *To fit/train "fast" simulation models ("fast"=>go direct from particle state to detector response without transport)*
- For simulation to be useful in these areas it must
  - *Generate the needed sample size(s) within available computational budget*
  - *Model the physics at the accuracy/precision required for the study*



Models for the detector Geometry/Materials/Fields

Models for Particle properties, Processes

Stepping algorithm for Monte Carlo transport through geometry with interactions



Scoring when particles traverse active regions of detector (position, time, energy deposition, …)

Detector Simulation in Two Images

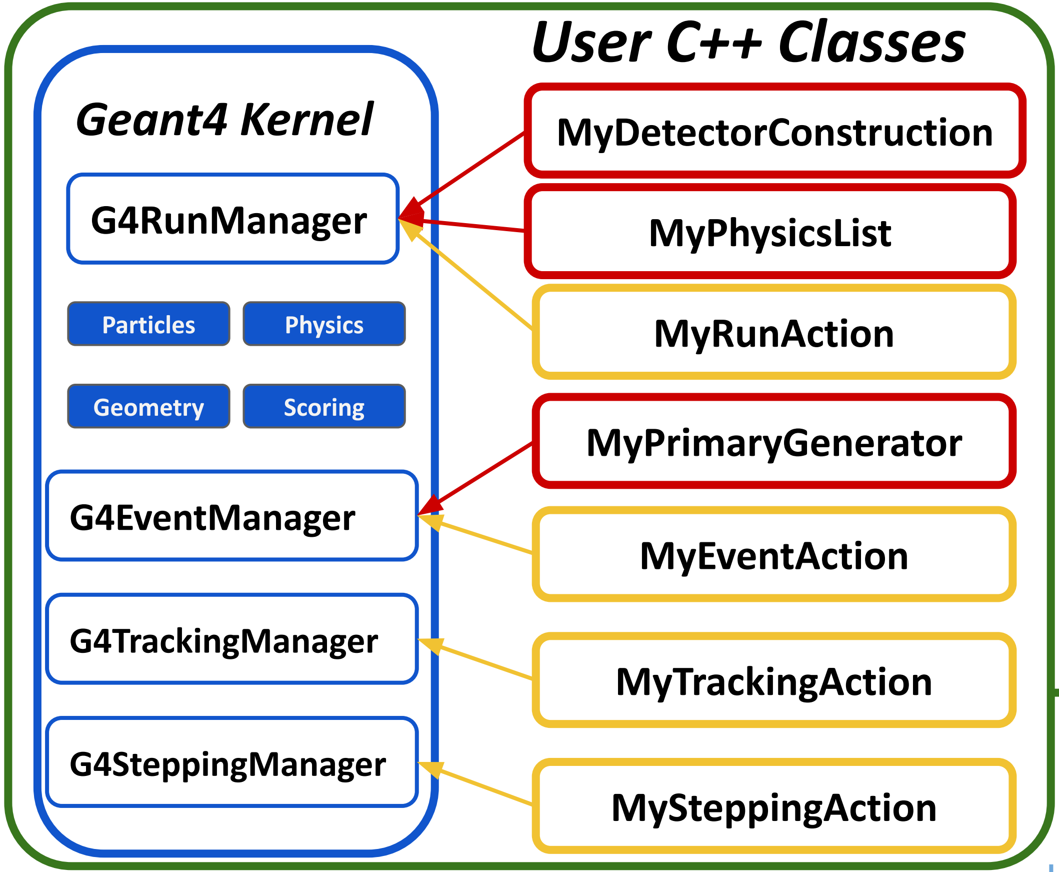
Each track is independent and does not influence transport of any other

Geant4: HEP's Primary Detector Simulation Code

GEANT4
A SIMULATION TOOLKIT

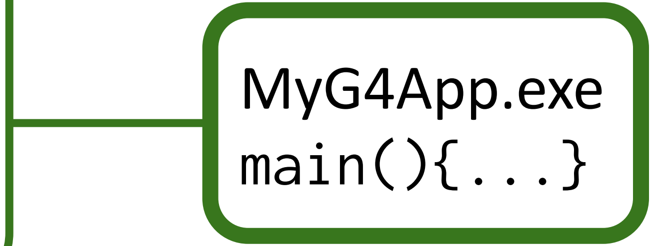
- *R&D in mid 1990s, first release in 1998 with 24/11 minor/major releases*
  - *O(800K) lines of C++ (17), supported on Linux/macOS/Windows*
- *Other codes like Fluka, MCNPX, EGSnrc, also used, but in more limited or specialized applications*

NB: "Geant4" not "Geant" as the "4" is part of the name, not version.
"Geant" is also not an acronym, despite what Wikipedia may say!



Mandatory *Models*
- **Geometry, Materials, Fields**
  - **Scoring (Active volumes)**
- **Physics Processes/Models**
- **Primary Particle Source(s)**

Optional *Actions*
- **Custom control of steps, tracks, events, runs**

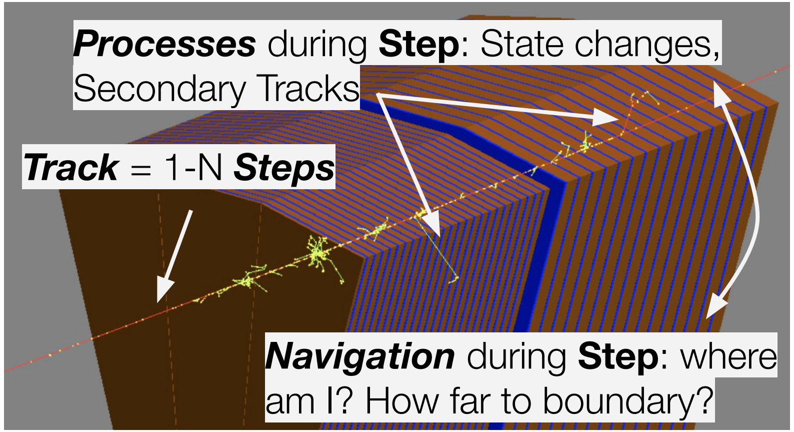


Geant4 in Brief

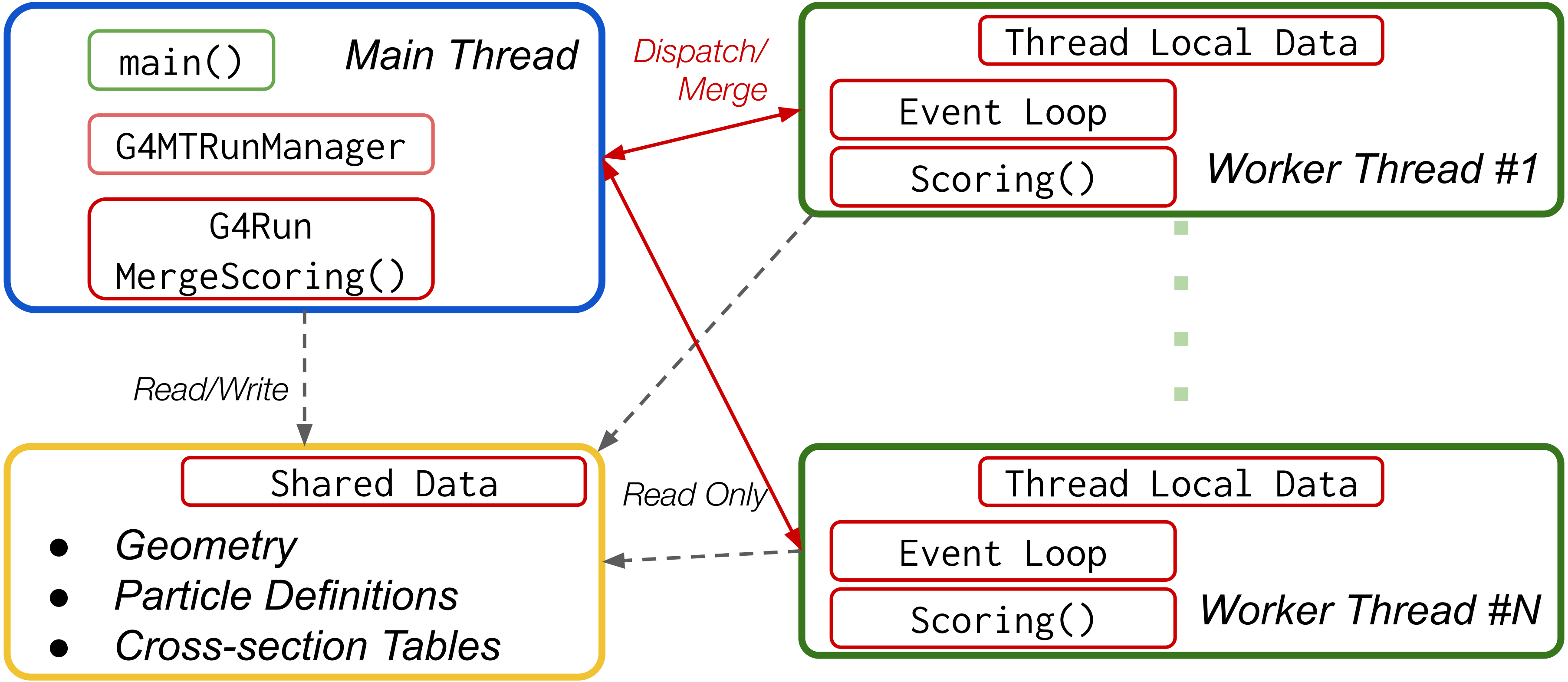
A *general purpose* **C++ Monte Carlo simulation toolkit** for elementary particles passing through and interacting with matter

Geant4: History-based Monte Carlo

- 1-N ***Runs***
  - *1 **Run** is 1-M **Events***
    - *1 **Event** has 1-P **Tracks***
      - *1 **Track** has 1-Q **Steps***
- Each **Track** may produce 0-S secondary Tracks, e.g. ionization
  - ***Amount of work per event is not known up front***
- Each Track stepped until it is killed or leaves the geometry
  - *Secondaries pushed onto Stack*
- Tracks are processed ***sequentially***




```
foreach 1...N events
  GeneratePrimaries                > Initializes primary tracks
  StackTracks                        and stacks them
  while TrackStack not empty       > Loop over tracks
    track = PopTrack
    while track is Alive           > Stepping loop
      StepTrack                    > Navigation, Physics, State Changes
      StackTracks                  > Stack any secondaries
    endwhile
  endwhile
endforeach
```



CPU Parallelism in Geant4

Parallelize over ***Events***, each thread processes N events, tracks/steps in each still processed ***sequentially***

Scopus   Search   Lists   Sources   SciVal   Create account   Sign in

17,468 documents have cited:

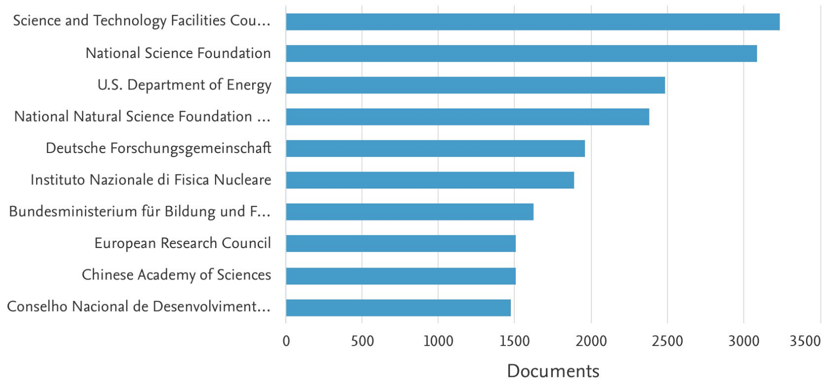
GEANT4 - A simulation toolkit

Agostinelli S., Allison J., Amako K., Apostolakis J., Araujo H., Arce P., Asai M., (...), Zschiesche D.

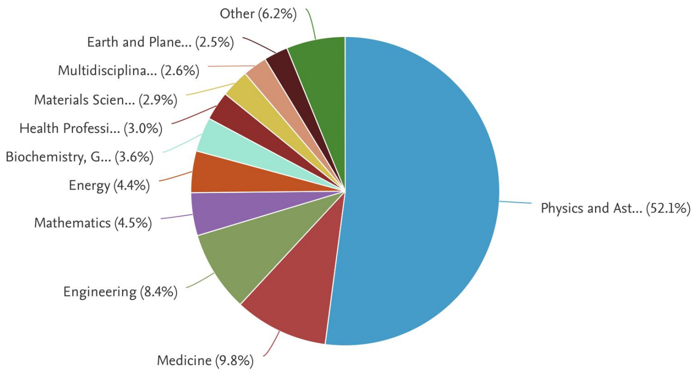
(2003) Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 506 (3) , pp. 250-303.

Documents by funding sponsor

Compare the document counts for up to 15 funding sponsors.



Documents by subject area



Geant4 - A simulation toolkit

NIM A, vol 506(3), pp250-303, 2003

Significant use across **many** research areas, considered **mission critical for HEP**

2,046 documents have cited:

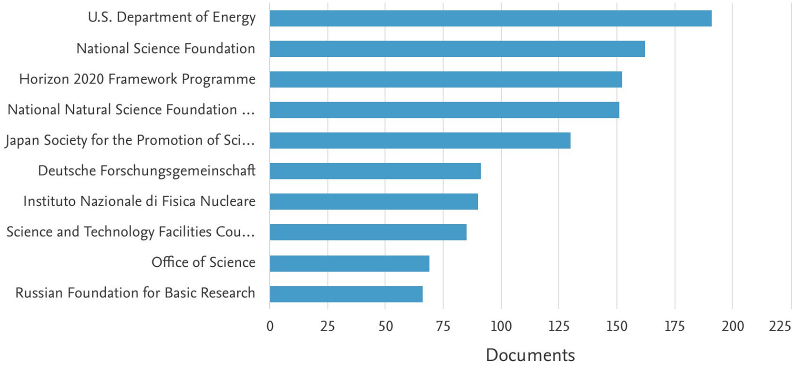
Recent developments in GEANT4

Allison J., Amako K., Apostolakis J., Arce P., Asai M., Aso T., Bagli E., (...), Yoshida H.

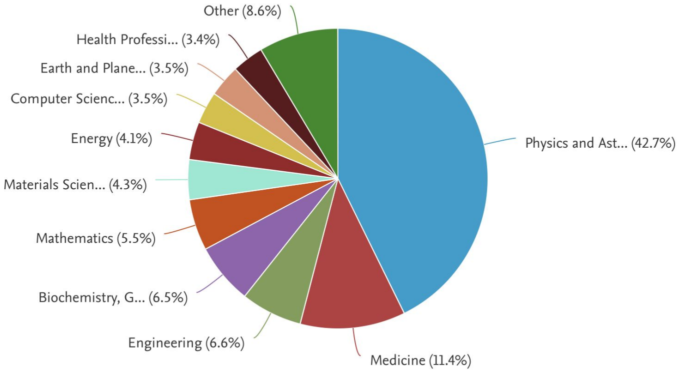
(2016) Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 835 , pp. 186-225.

Documents by funding sponsor

Compare the document counts for up to 15 funding sponsors.



Documents by subject area



Recent developments in Geant4

[NIM A, vol 835, pp 186-225]

Reflects major upgrades for **multithreading** and **capability** of version 10/11 series

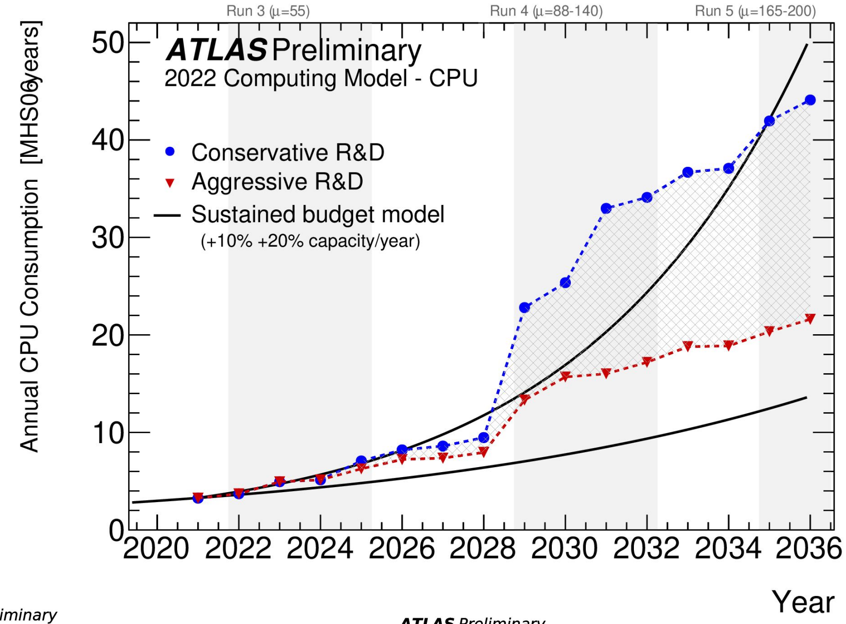
Geant4 Development Principles

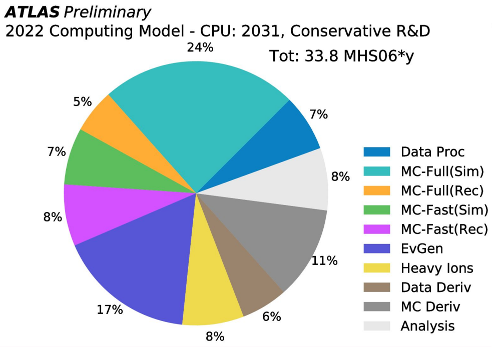
- Provide excellent **support** to users and their requirements
- Maintain the toolkit for **long-term stability and sustainability**
  - *Flexible and easy **integration** with a broad range of user/project codes*
  - *Recognizing **>10yr timescales** needed by experiments/projects*
- Improve the **precision and energy range of physics models**
- Improve the **performance** of the toolkit

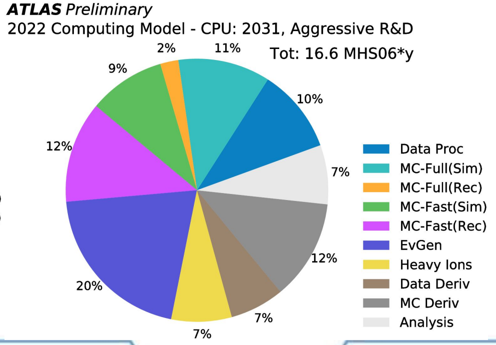
- Together, recognize the **mission critical nature of Geant4 for HEP**
  - *Very careful to not introduce breaking changes without consultation across breadth of user communities*
- **Improving performance** cannot be at cost to other three items, and vice versa, but it is **perhaps the most important task for HEP** at present!

Compute Budgets and Simulation Performance

- Particle physics, not just LHC, needs large amounts of simulation to support precision measurements over next 10-15yr
- Simulation already major fraction of the overall compute budget in HEP, even in aggressive scenarios
- **Major R&D efforts ongoing on increasing simulation throughput using a range of methods**







Simulation R&D Strands

"Fast"

- Use parametric or generative ML models instead of full stepping transport loop and secondary production
- Go directly from track (type, energy, direction) to detector response
- Good overview of current/future approaches in this CHEP Plenary
- Still require full simulation at some level though to supply training data!

"Full"

Optical Photons

- Key bottleneck in simulations needing them due to sheer number per typical event
- Photon transport == Raytracing, many efforts over the years to using GPU/graphics approaches
- Opticks: Offload photons form Geant4 to NVidia OptiX on GPU
    - Migration to Optix7
    - SWIFT-HEP project (UoM) contributed to work, including hackathon
- Mitsuba 3 Renderer: UoM R&D in SWIFT-HEP

Electromagnetic Processes

- Main bottleneck in HEP simulations due to number of primaries/N-aries in particle showers in calorimeters, steps in relatively dense/complex geometries
- Needs full charged particle transport, but physics models amenable to computation on GPUs
    - Mostly simple
    - Few branches
    - Only three particle types

Detector Simulation on GPUs R&D @ June 2023

- **AdePT Project (CERN-SFT)**
  - *https://github.com/apt-sim, Latest Status @ CHEP 2023*
- **Celeritas Project (ECP: ORNL, FNAL, Argonne, LBL)**
  - *https://github.com/celeritas-project, Latest Status @ CHEP 2023*
- *ExaCALIBUR-HEP and SWIFT-HEP projects contributing to both*
- **Vecgeom/ORANGE Surface Based Geometry (CERN, Celeritas/ORNL)**
  - *https://gitlab.cern.ch/VecGeom/VecGeom (See surface_model branch)*
  - *https://github.com/celeritas-project/celeritas/tree/develop/src/orange*
  - ***ExaTEPP is contributing here (see Peter's presentation tomorrow)***
- Regular working and strategy meetings between projects since ~2020, knowledge sharing building on prior years of work/experience
- **Can only give a higher level overview today!**

Objectives of AdePT and Celeritas

- ***Understand usability of GPUs for general particle transport simulation problem***
  - *Prototype e+/e−/γ shower simulation on GPU, evolve to realistic HEP use-cases*
  - ***Focus on EM physics** given known computational cost in HEP workflows, prior knowledge of applicability of physics models on GPU (hadronic physics more complex!)*
- ***Implement GPU-targeted components for physics, geometry, magnetic field, with data models and workflow***
  - *Integrate components in a **hybrid** CPU-GPU Geant4 workflow*
  - *Offload tracks to GPU/CPU when preconditions like particle type or geometric region met*
  - *Most realistic short-term objective to allow rapid testing/use in existing experiment code*
- **Ensure correctness and reproducibility**
  - *Validate GPU-only, CPU+GPU off/onload against pure CPU Geant4*
- **Understand bottlenecks and blockers limiting performance**
  - *Feasibility and future effort required for efficient simulation workflows on GPU*
- **Celeritas also have a longer term objective to include full hadronic physics**

Development Approaches

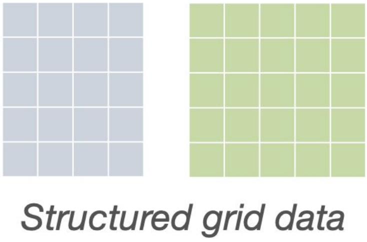
- **AdePT:**
- Implement features as new examples
  - *Flexibility to explore different directions, optimizations*
- Common functionality built up from successful/existing features
  - *Core types, helper functions*
  - *Geometry: **VecGeom library***
  - *Physics: G4HepEM library*
- Portability (non-CUDA) not a major priority in current phase
  - *VecGeom blocker: only supports CUDA*

- **Celeritas:**
- Ground-up approach, core principles
  - *Data-oriented programming*
  - *Composition-based objects*
  - *Revisit older design/impl choices*
- Common functionality:
  - *Geometry: **VecGeom or ORANGE***
  - *Physics: Data imported from Geant4, models reimplemented for GPU*
- C++-only execution code, allowing portability to HIP(*) and others
  - *(*)Only for ORANGE geometry*

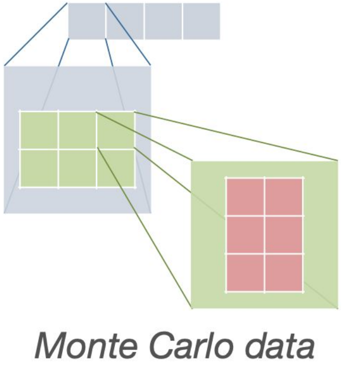
- **Different approaches with active collaboration allows broad range of designs to be explored in R&D phase - with eventual aim to converge on common solutions**

Challenges for Monte Carlo on GPUs

- **Execution**: divergence and load balancing
  - *GPUs want every thread doing the same thing*
  - *MC: every particle is doing something (somewhat) different, may create new work (secondary particles)*

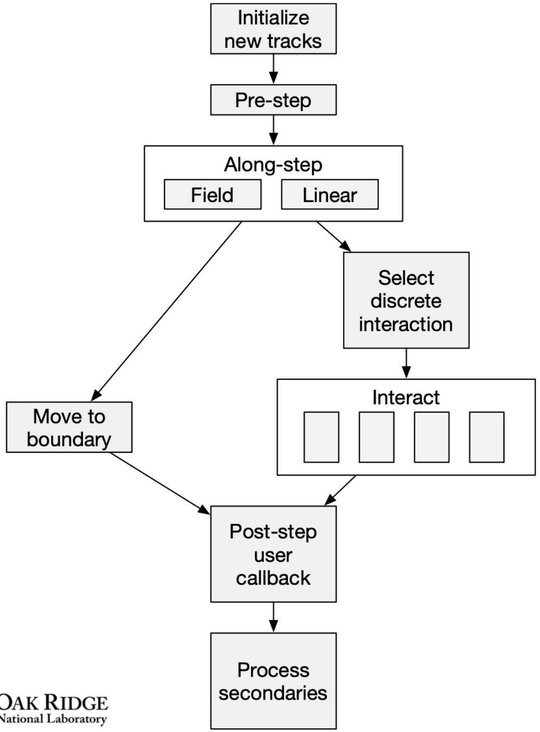


- **Memory**: data structures and access patterns
  - *GPUs want direct, uniform, contiguous access*
  - *MC: hierarchy and indirection; random access*
  - *Memory allocation is a particular problem*



Credit: [Seth Johnson (ORNL)]

Event-parallel to Track-parallel Stepping Workflow



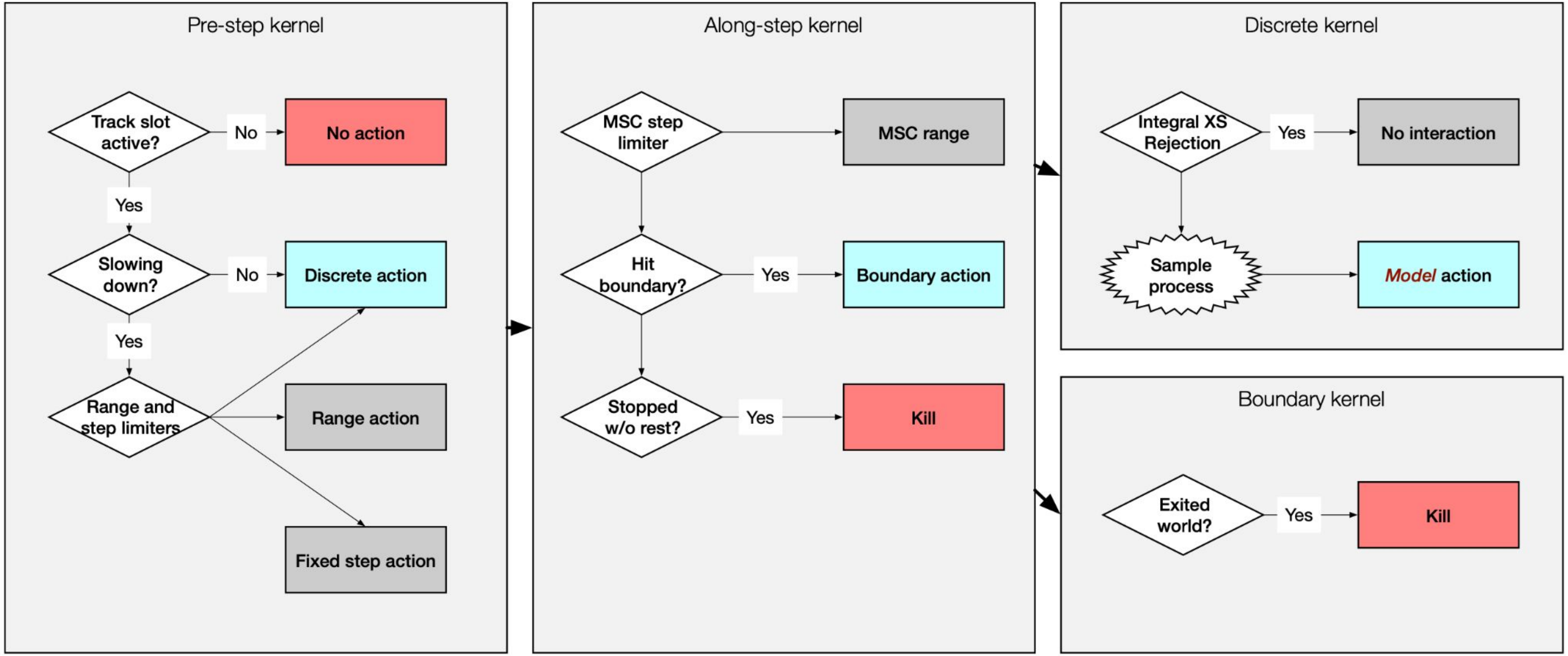

```
extend_from_primaries          ▷ Copy primaries to device, create track initializers
while Tracks are alive do
    initialize_tracks          ▷ Create new tracks in empty slots
    pre_step                   ▷ Sample mean free path, calculate step limits
    along_step                 ▷ Propagation, slowing down
    boundary                   ▷ Cross a geometry boundary
    discrete_select            ▷ Discrete model selection
    launch_models              ▷ Launch interaction kernels for applicable models
    extend_from_secondaries    ▷ Create track initializers from secondaries
end while
```

- **1 GPU thread per Alive Track**
- Action based control flow
- Step broken down into Kernels determining next **Action**, or performing an **Interaction**
- *Example from Celeritas, AdePT's is similar but with larger kernels at present*

Oak Ridge National Laboratory

Credit: Seth Johnson (ORNL)

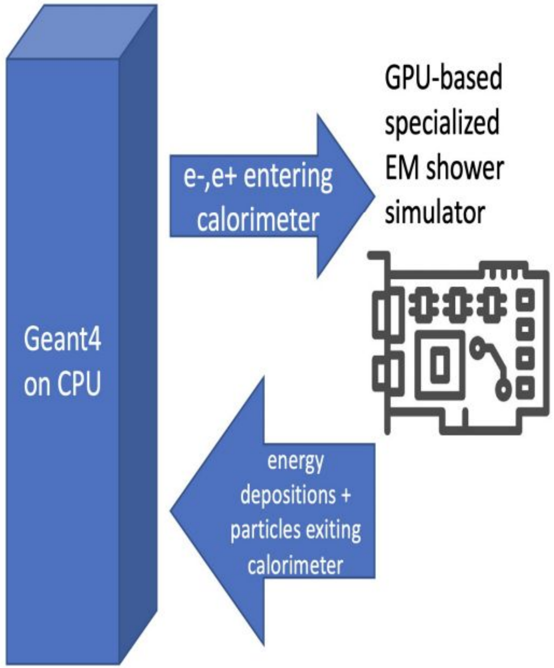
Celeritas: Inside Kernels



Credit: [Seth Johnson (ORNL)]

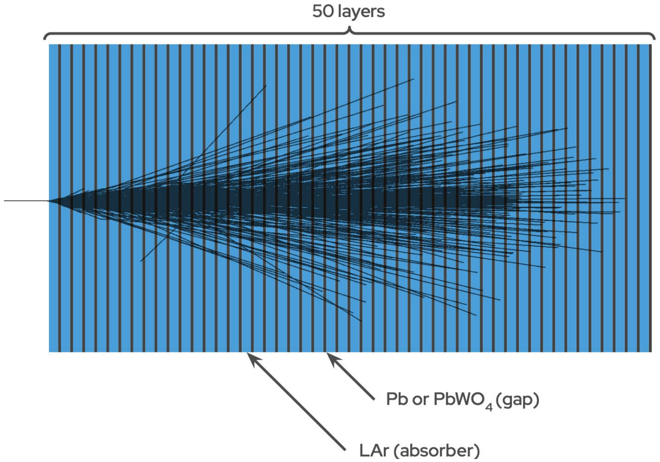
Strategies for integration with Geant4 applications

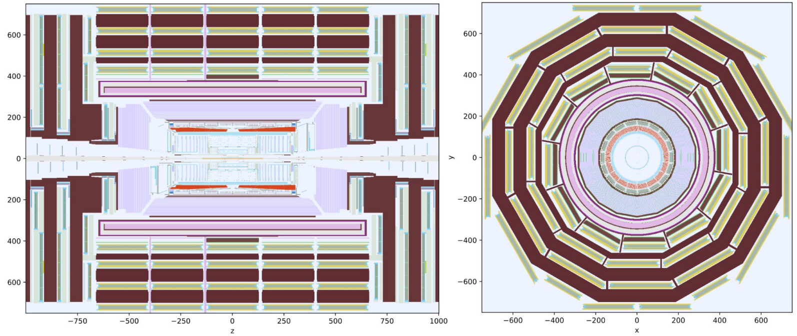
- AdePT and Celeritas only model e-/e+/g physics at present, so cannot be used standalone for simulating a **full** EM+hadronic experiment
- Instead, use them as a “service” to offload tracks to the GPU according to preconditions such as particle type, or geometric region.
- Use Geant4 **Fast Simulation** or **Tracking Action hooks**, both with same basic challenges:
  - *Minimizing number/size of on/offload actions*
  - ***Allowing user-defined actions on GPU, such as scoring/hits***
  - *Handing back particles (e.g. exiting particles, hadrons from photonuclear processes) from GPU to CPU*



Progression Problems for Benchmarking/Validation

- Both AdePT and Celeritas have adopted two *primary* test cases for benchmarking and validation
  - *Run on CPU, GPU, CPU+GPU hybrid modes*
- “**TestEM3**” taken from Geant4 examples as a core test case
  - *50 layer Pb (or PbWO4) / LAr sampling calorimeter*
  - *1-10GeV e- primaries in beam*
  - *Validation, basic scoring and performance measurements*
- **CMS 2018** GDML geometry
  - *Same primaries, also HepMC3 input*
  - *Use of more complex workflows, scoring*
- ATLAS, LHCb, medical use cases to be added

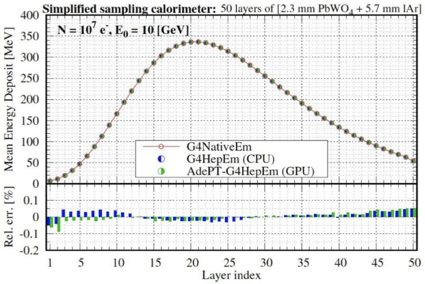




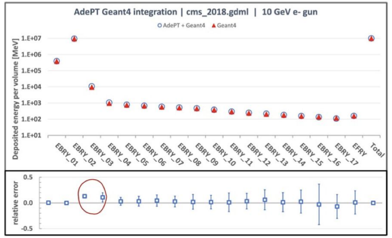
Physics Validation

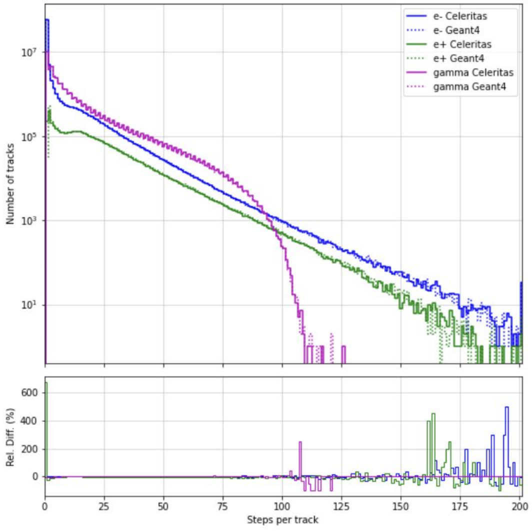
- [G4HepEM](#) in AdePT, CPU/GPU implementation/use of Geant4 models/cross-sections in Celeritas.

Sampling calorimeter example



AdePT integration with Geant4





TestEM3 MSC step count verification (Amanda Lund)

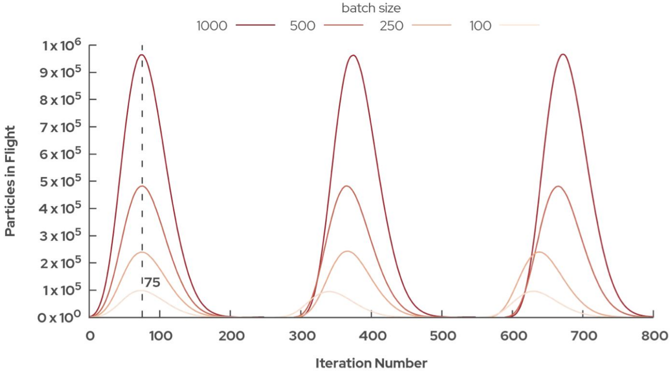
Particle	Process	Model
γ	photon conversion	Bethe–Heitler
	Compton scattering	Klein–Nishina
	photoelectric effect	Livermore
	Rayleigh scattering	Livermore
e±	ionization	Møller/Bhabha
	bremsstrahlung	Seltzer–Berger
		relativistic
	Coulomb scattering	Urban MSC
e+	annihilation	→(γ, γ)
μ	bremsstrahlung	μ brems

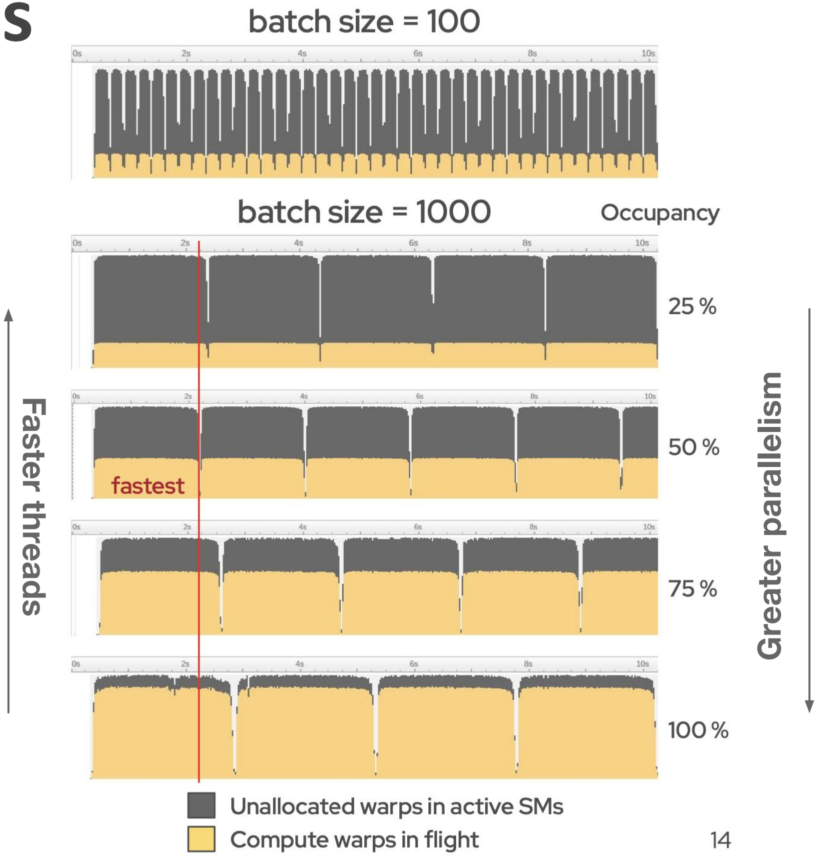
Overall excellent agreement with Geant4, but ongoing validation studies across problem space

Credits: [Witek Pokorski (CERN)](#), [Seth Johnson (ORNL), Amanda Lund (Argonne)](#)

AdePT: Runtime Characteristics

- Configuration space for GPU runs
  - *Number of input particles per batch*
  - *Number of registers per thread*
  - *Number of threads per block*
- Higher batch size => more work per N steps
  - *Limited by available memory and tracks*
- **Hints of strategies to fill "work gaps"**
  - *More CPU threads, i.e. concurrent events*



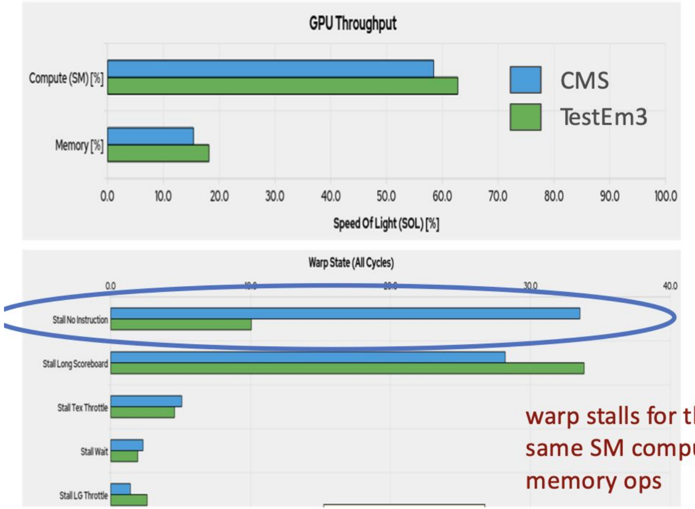


Credit: AdePT Developers

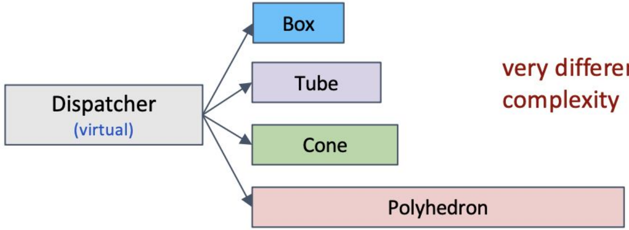
Key Performance Limiter: Geometry on GPUs

- **Several problems in CSG based approach of VecGeom library on device:**
  - *Virtual dispatch*
  - *Recursion in relocation algorithms*
  - *Divergence from differences in algorithmic complexity for solids*
- Consequences on GPU:
  - *Large stacks & register-hungry code limits number of concurrent warps*
  - *Divergence limits concurrency per warp*
- Moving from simple to complex geometry => *longer stalls within a warp for same SM compute*

TestEM3 = 100 simple layered boxes
CMS = full CMS_2018 geometry

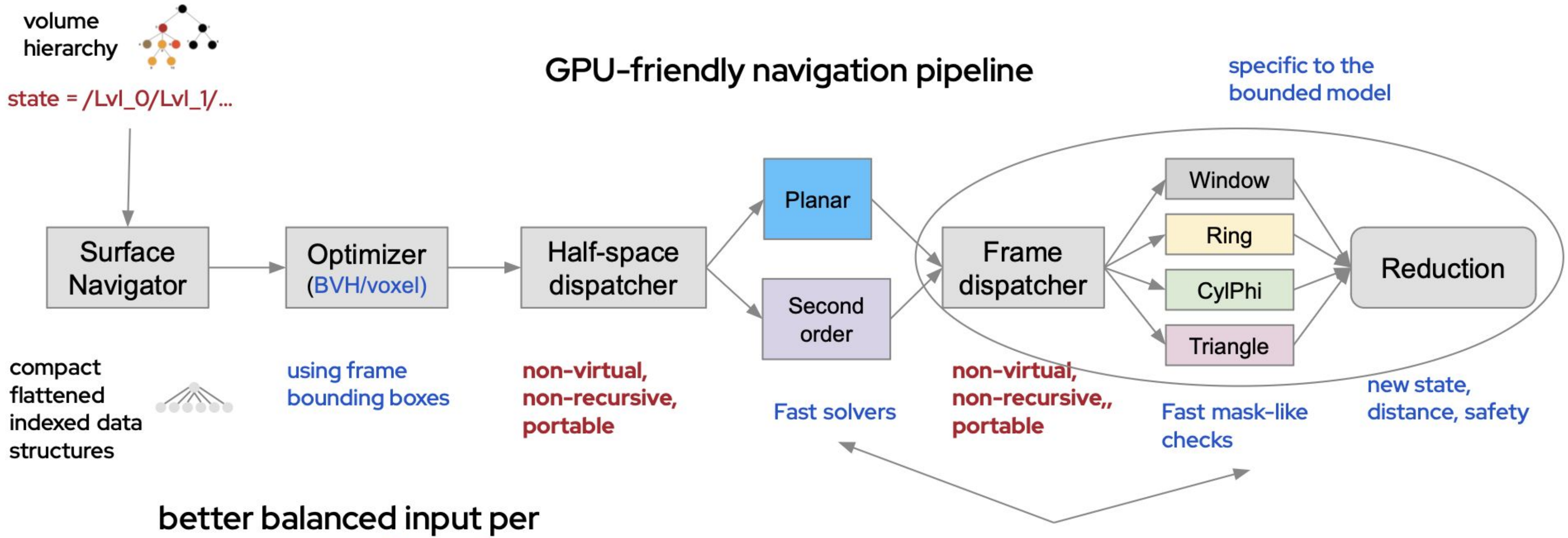


warp stalls for the same SM compute & memory ops



very different code complexity

Credit: Witek Pokorski (CERN)

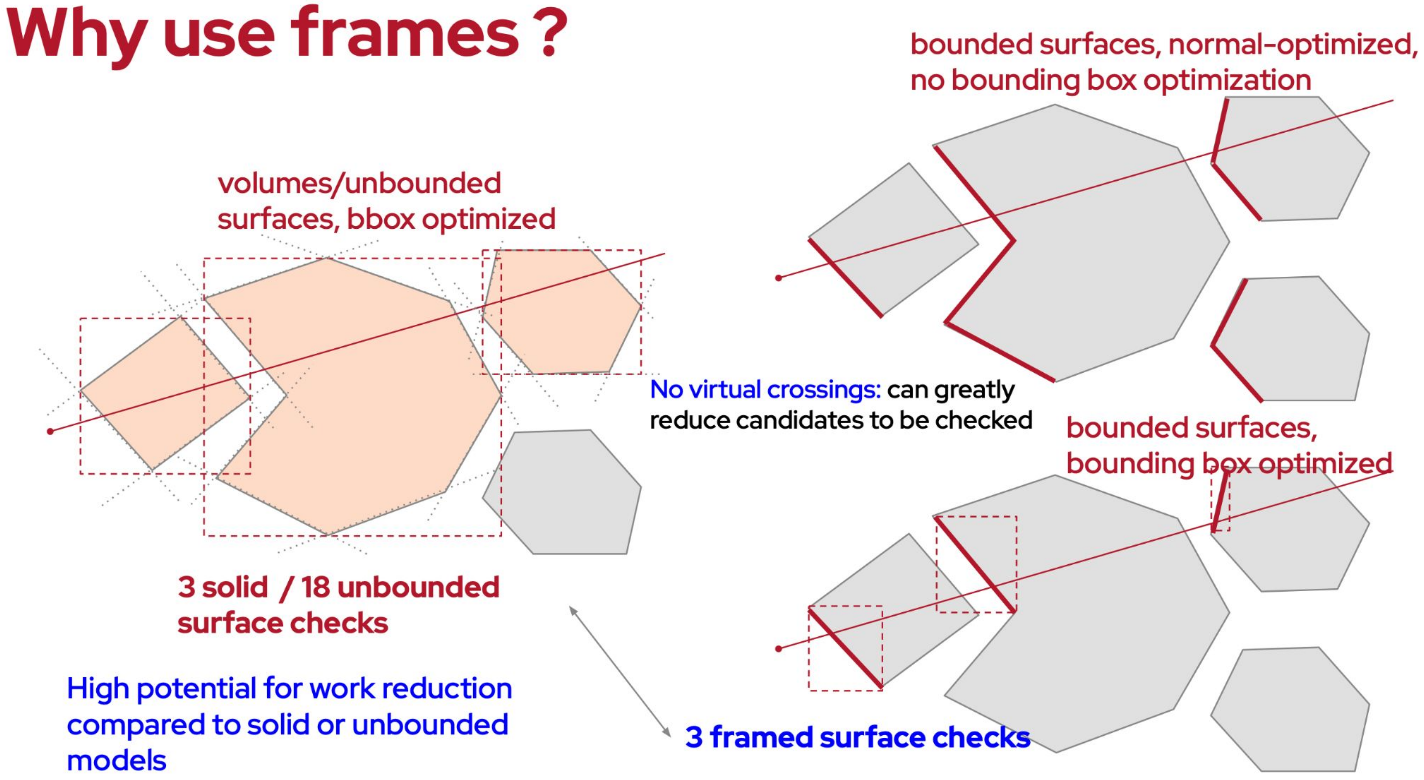


better balanced input per particle due to flattening and mixing surfaces from different volumes

faster divergent sections with fewer branches

From Solid to Surface Based Geometry Models

Initial work since Summer 2022 between ORNL/CERN



ORANGE: *Unbounded Surfaces*
VecGeom: *Bounded Surfaces*

Complex, fine grained HEP geometries may benefit from the use of frames to reduce search

Also ongoing contacts with track reconstruction projects such as ACTS and detray.

Bounded vs Unbounded Surfaces

See talks at SWIFT-HEP Workshop for more details

Summary

- **Simulating HEP detectors a key task in experiment design, operation, and production of physics results**
    - ***Geant4*** *provides the primary particle transport code used to implement simulation in HEP and other domains*
- Significant increase in simulation throughput needed for next 10-20 years of experiments to realise physics goals within available compute/resource budgets
    - ***R&D in AdePT/Celeritas projects on general EM particle transport on GPUs: ExCALIBUR-HEP and now SWIFT-HEP contributing to this work***
    - ***R&D on GPU friendly geometry modeling/navigation is critical: ExaTEPP is contributing to this work***
    - ***Complementary to other work on "Fast" and Optical Photon Simulation***