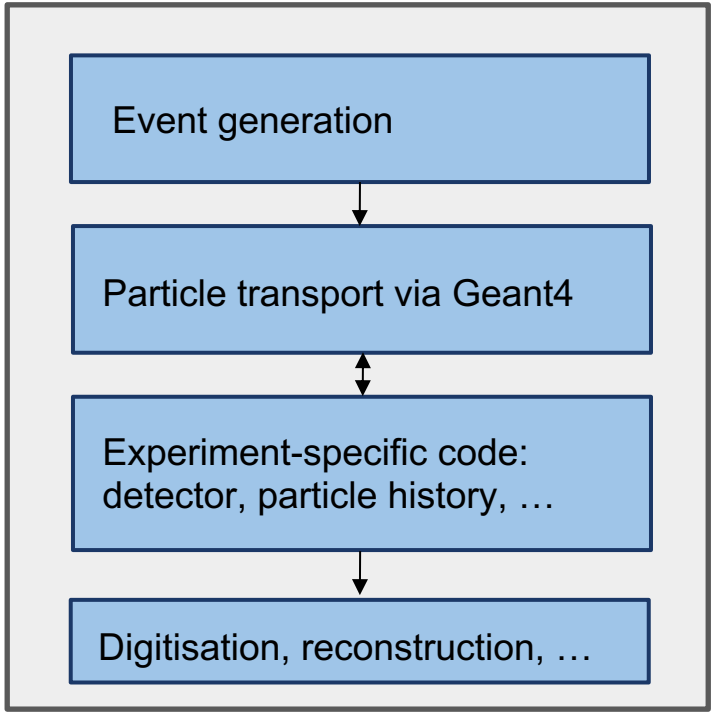
Accelerating Geant4 detector simulations on GPU with AdePT

Severin Diederichs,
CERN, severin.diederichs@cern.ch
on behalf of the AdePT project

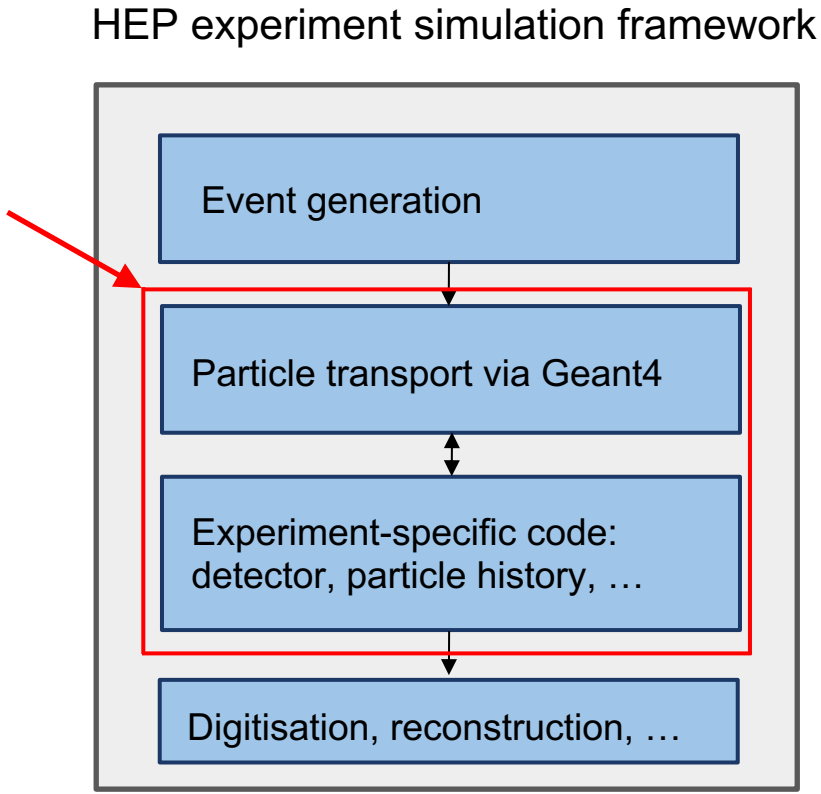
Understanding the target

HEP experiment simulation framework

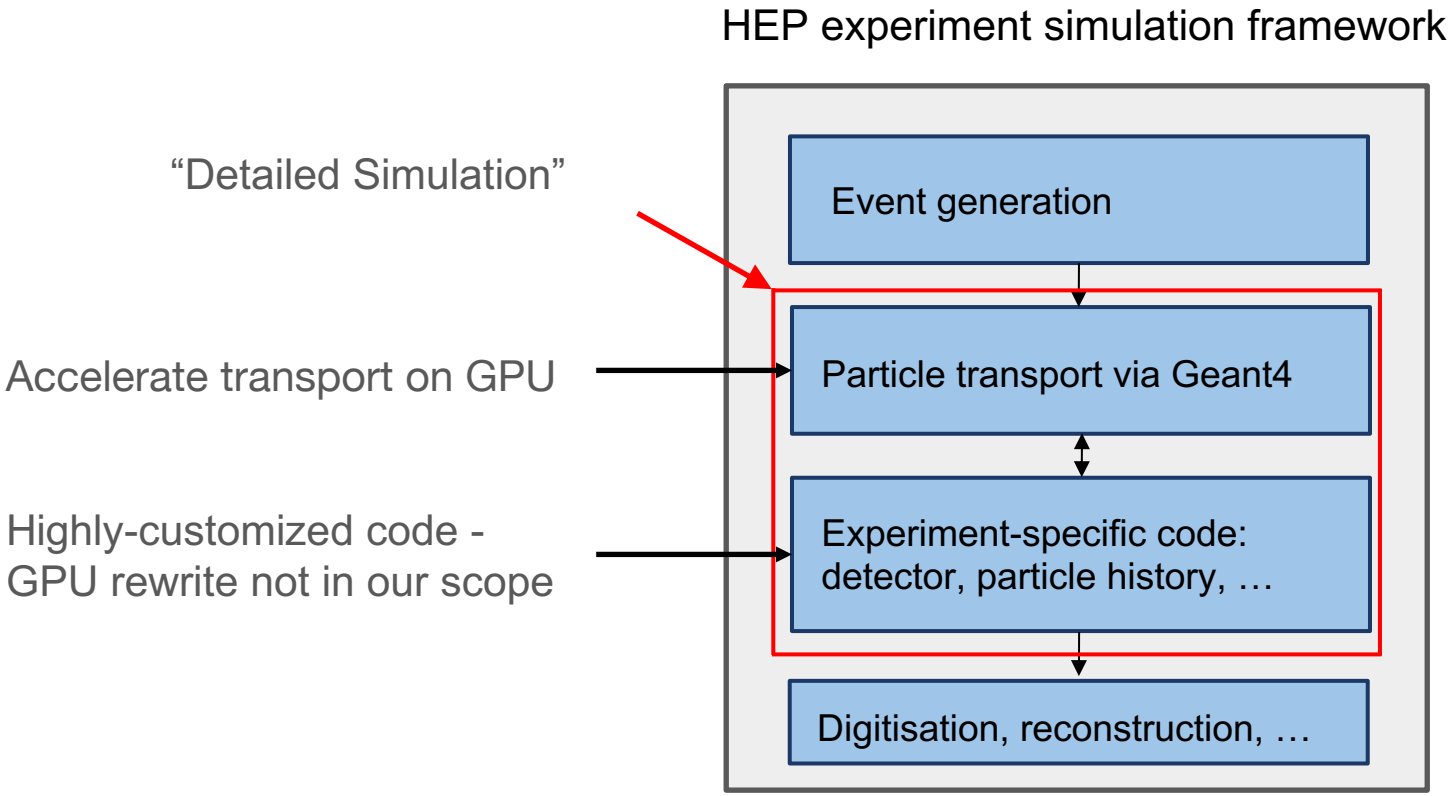


Understanding the target

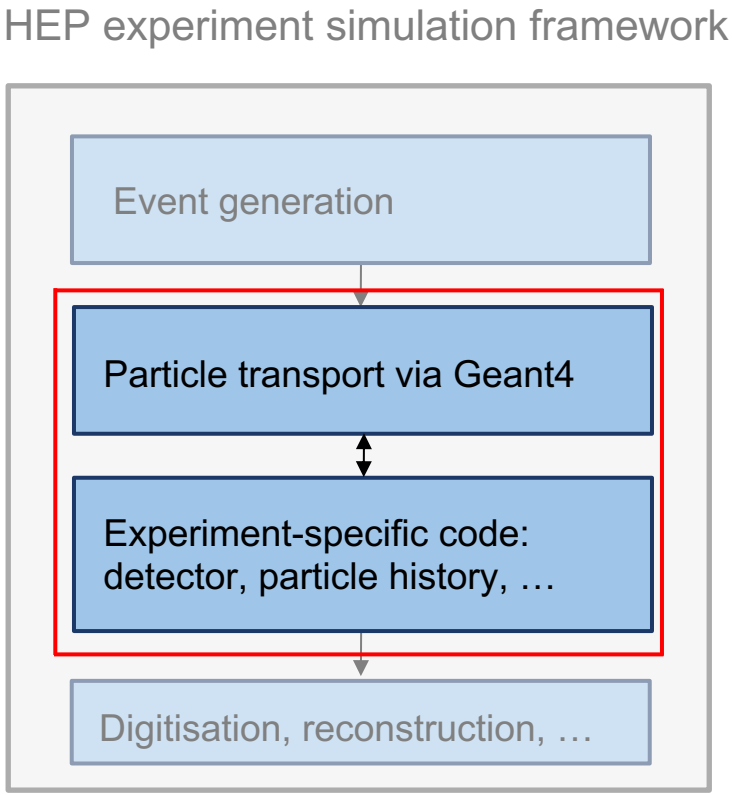
“Detailed Simulation”



Understanding the target



Accelerating Geant4 with a GPU plugin

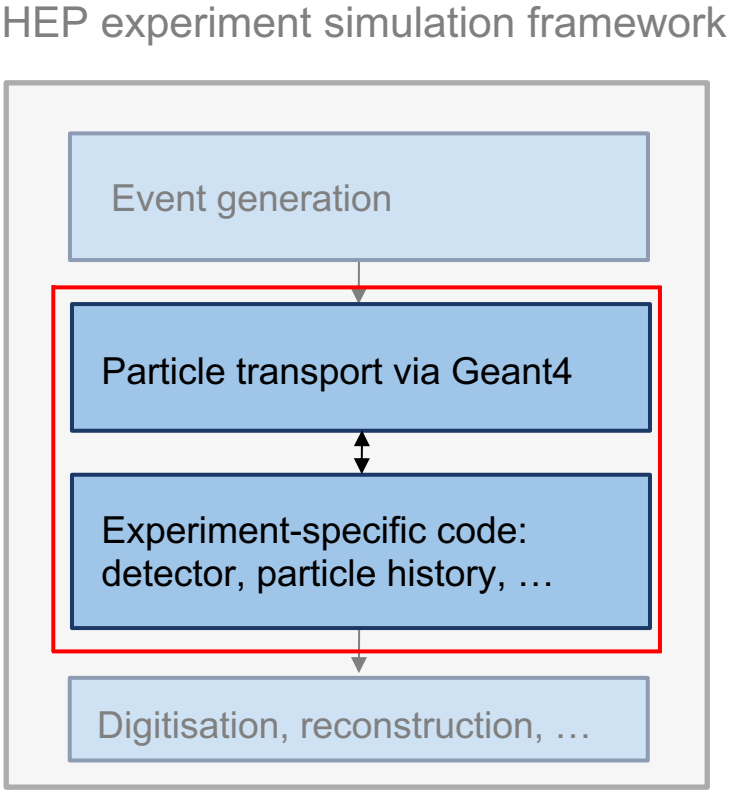


Accelerating Geant4 with a GPU plugin

Current approach:

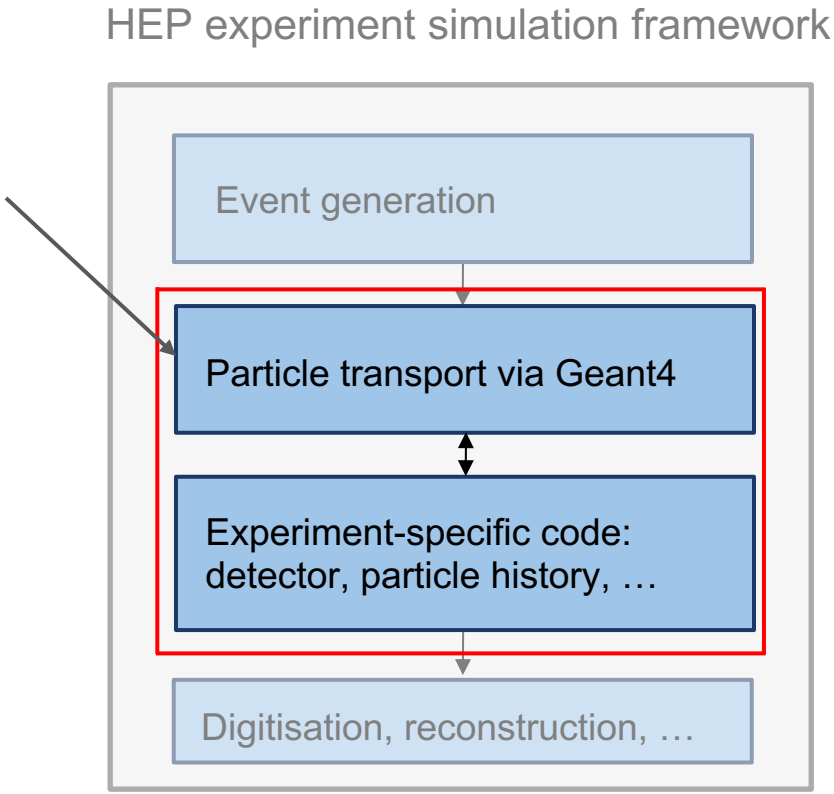
Offloading e^-, e^+, γ to the GPU because they

- have self-consistent physics
- require a large fraction of the compute time



Accelerating Geant4 with a GPU plugin

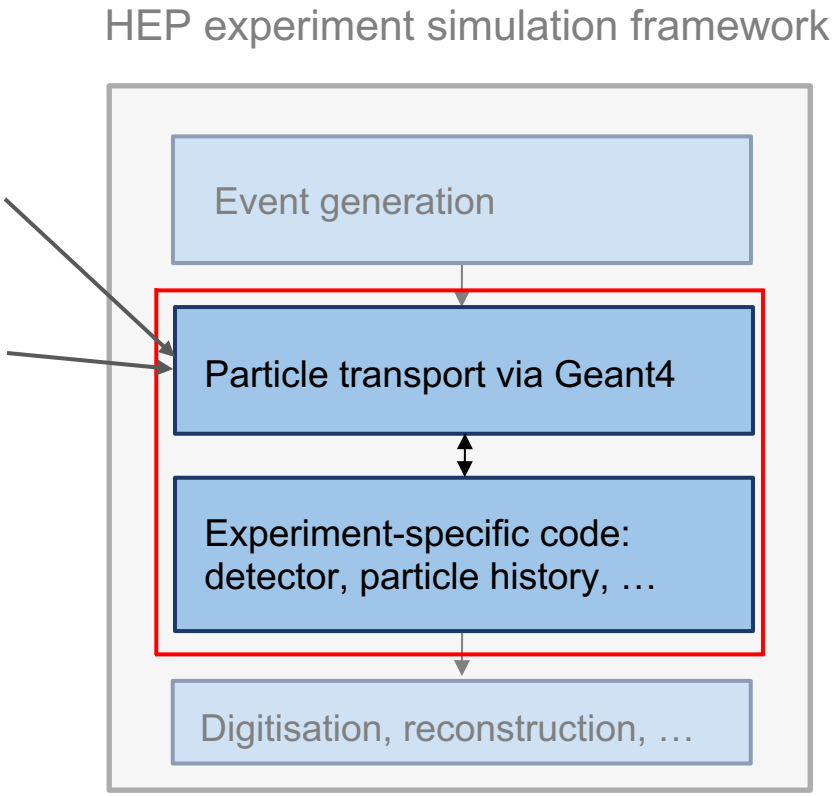
1. Data must be available on GPU
   - Physics: cross sections, magnetic field
   - Geometry: volume hierarchy, materials
   - **Automatically converted from Geant4**



Accelerating Geant4 with a GPU plugin

1. Data must be available on GPU
   - Physics: cross sections, magnetic field
   - Geometry: volume hierarchy, materials
   - **Automatically converted from Geant4**

1. **Moving particles** between CPU and GPU

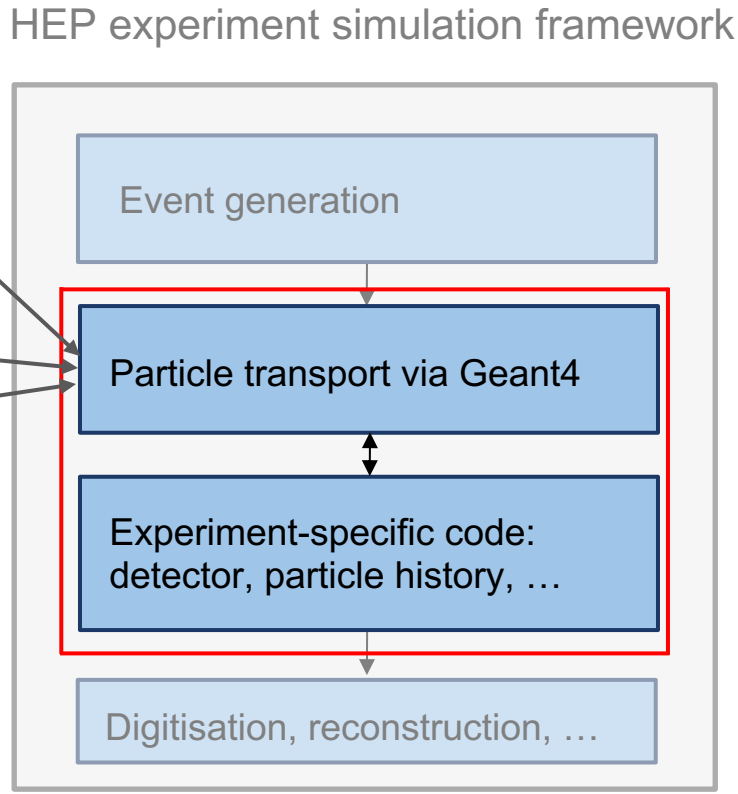


Accelerating Geant4 with a GPU plugin

1. Data must be available on GPU
   - Physics: cross sections, magnetic field
   - Geometry: volume hierarchy, materials
   - **Automatically converted from Geant4**

1. **Moving particles** between CPU and GPU

1. **Transport on GPU**



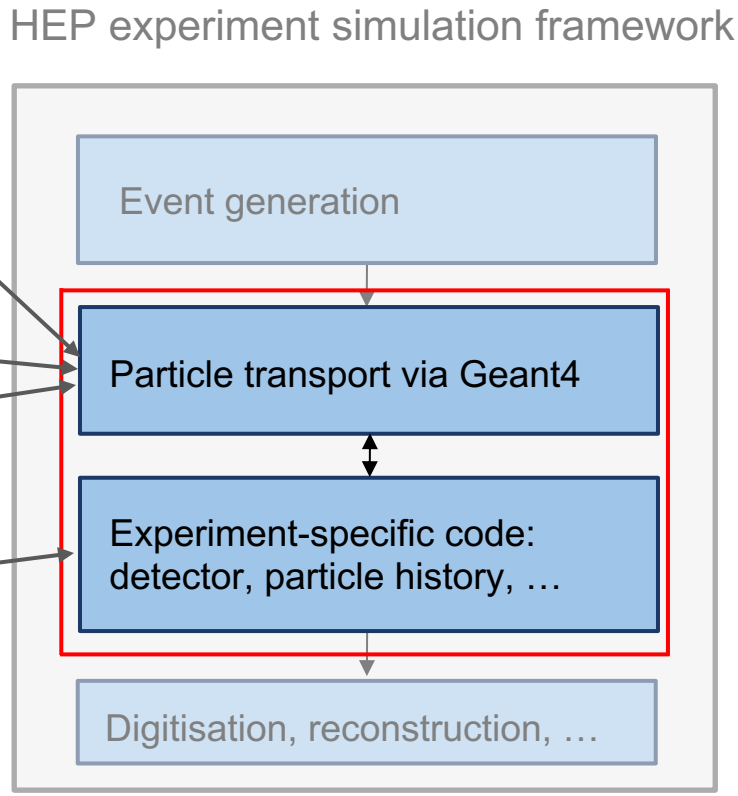
Accelerating Geant4 with a GPU plugin

1. Data must be available on GPU
   - Physics: cross sections, magnetic field
   - Geometry: volume hierarchy, materials
   - **Automatically converted from Geant4**

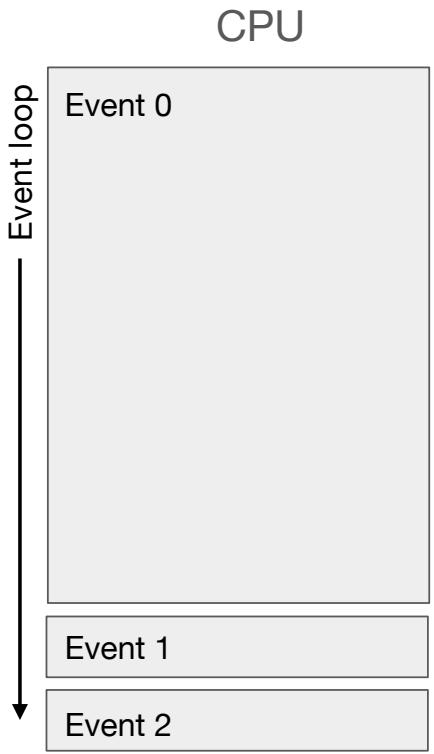
1. **Moving particles** between CPU and GPU

1. **Transport on GPU**

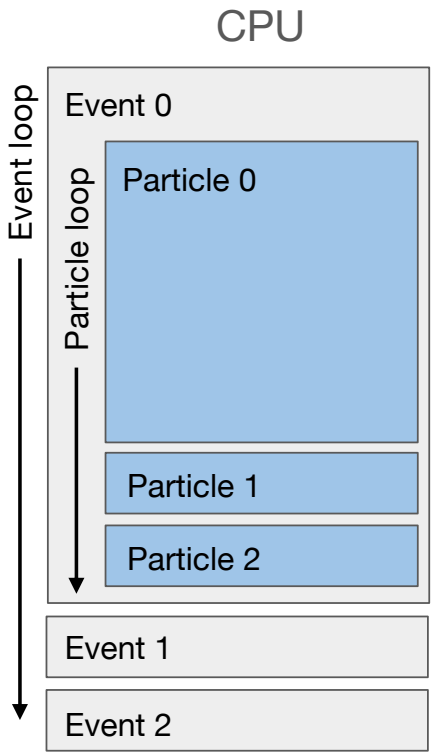
2. **Returning steps** from GPU to CPU



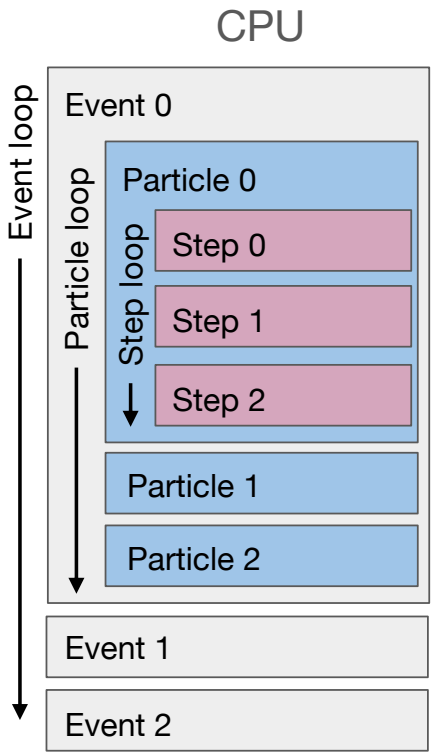
GPUs require massive parallelism to be efficient



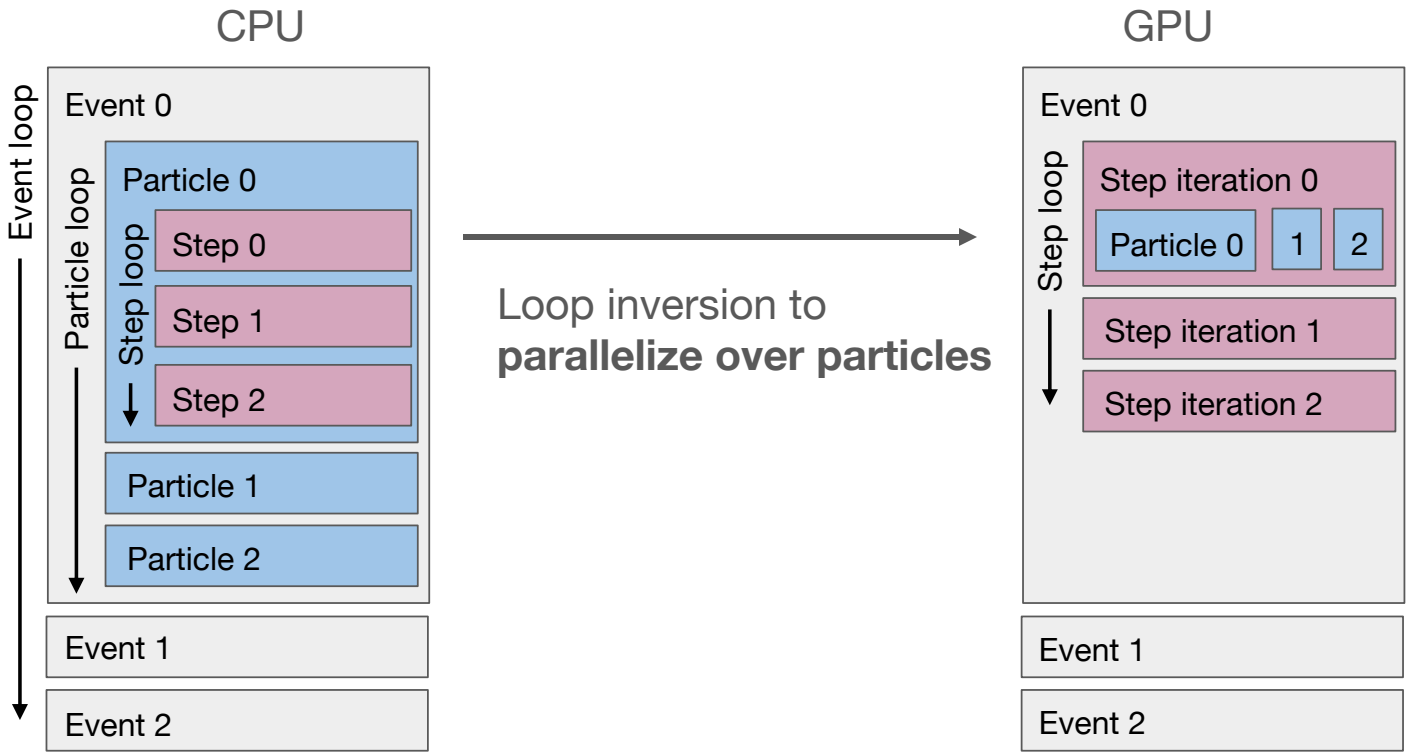
GPUs require massive parallelism to be efficient



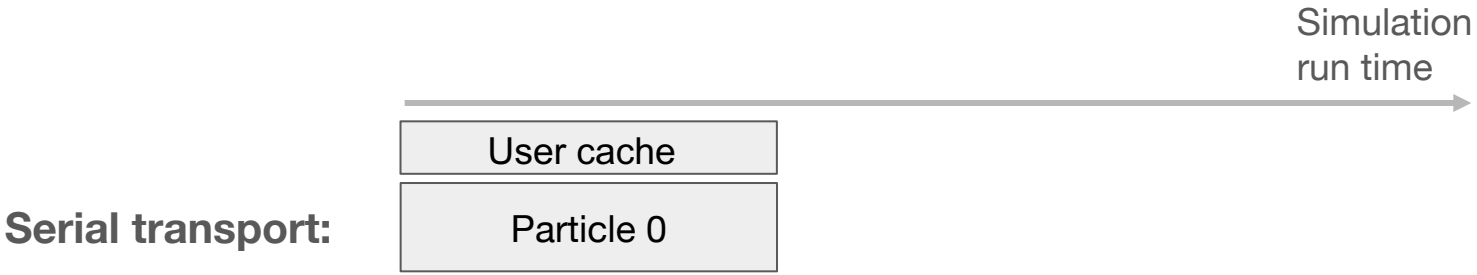
GPUs require massive parallelism to be efficient



GPUs require massive parallelism to be efficient

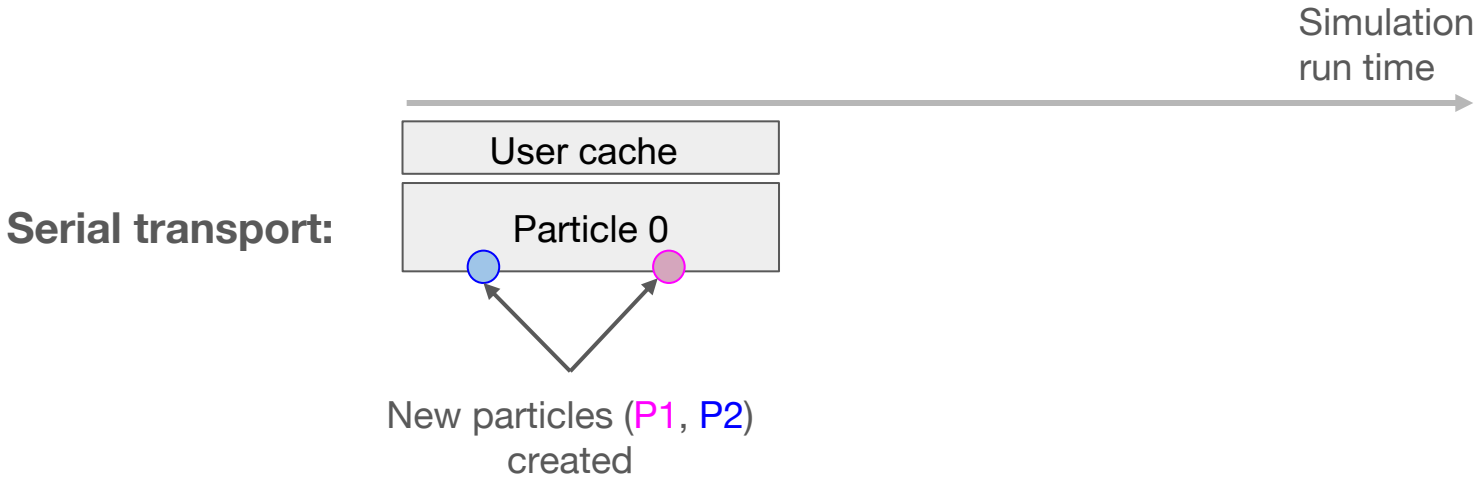


Consequences of parallel transport



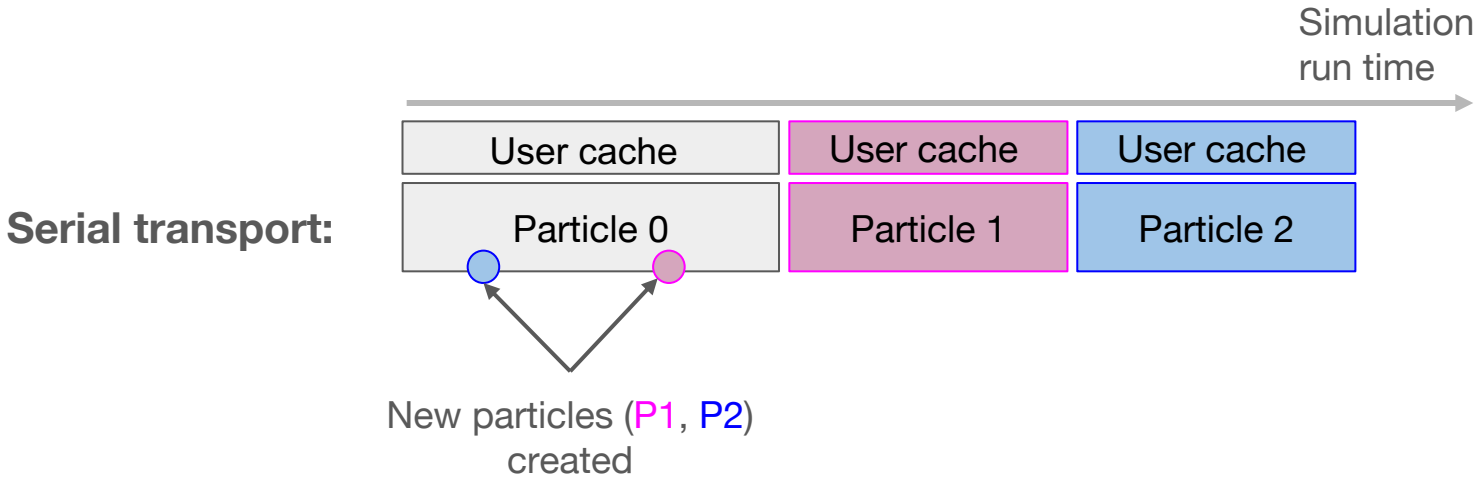
Caching is used to **optimize performance**

Consequences of parallel transport



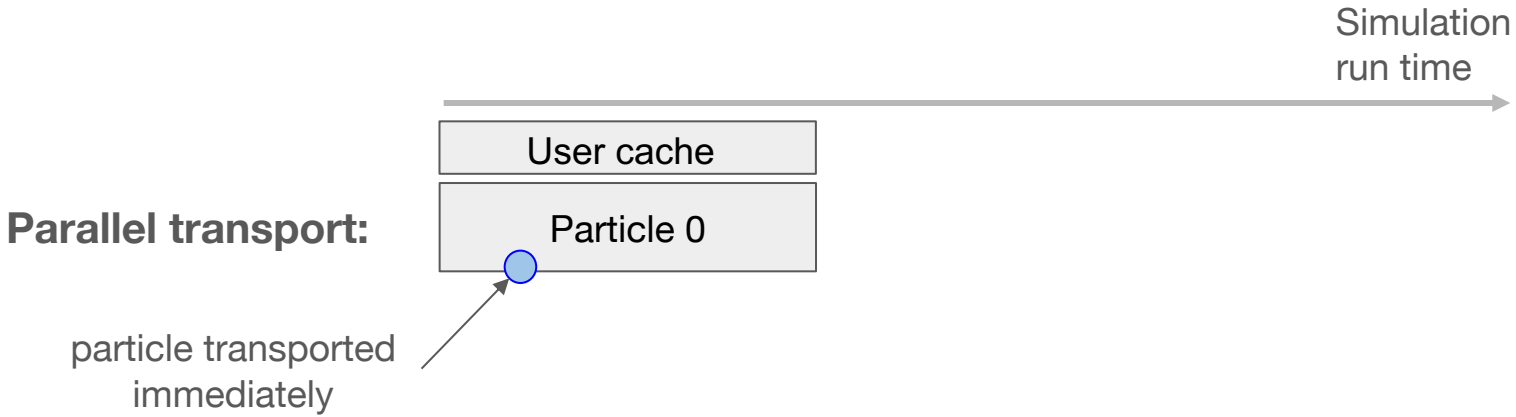
Caching is used to **optimize performance**

Consequences of parallel transport

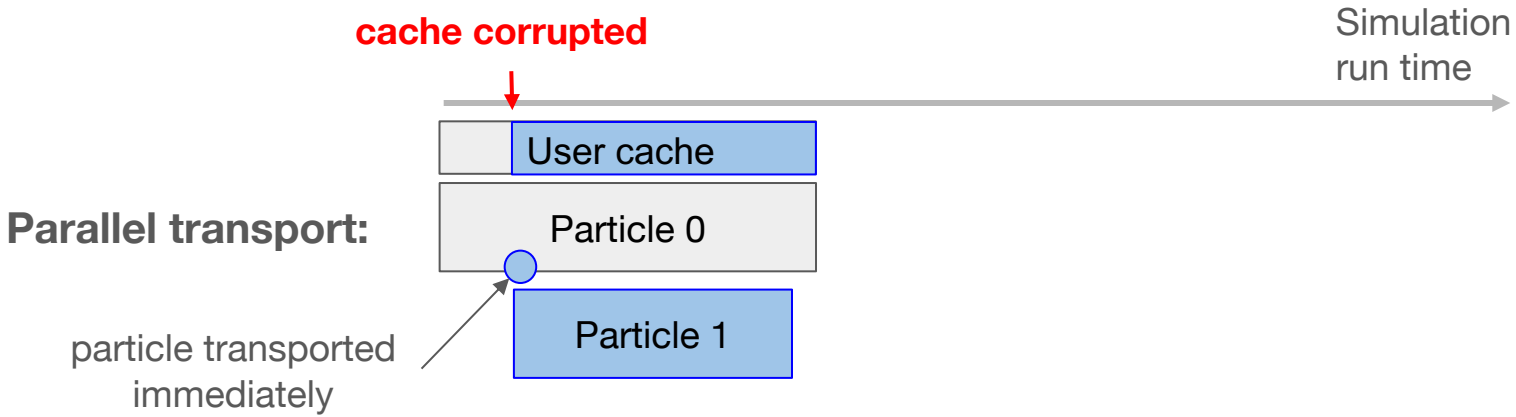


Caching is used to **optimize performance**

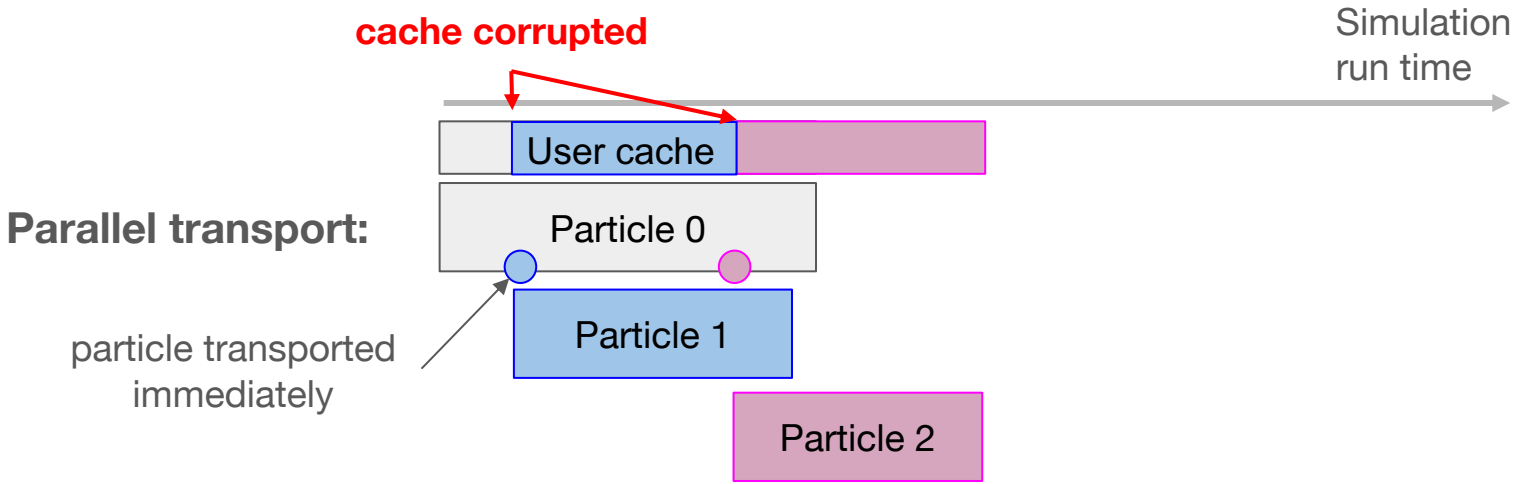
Consequences of parallel transport



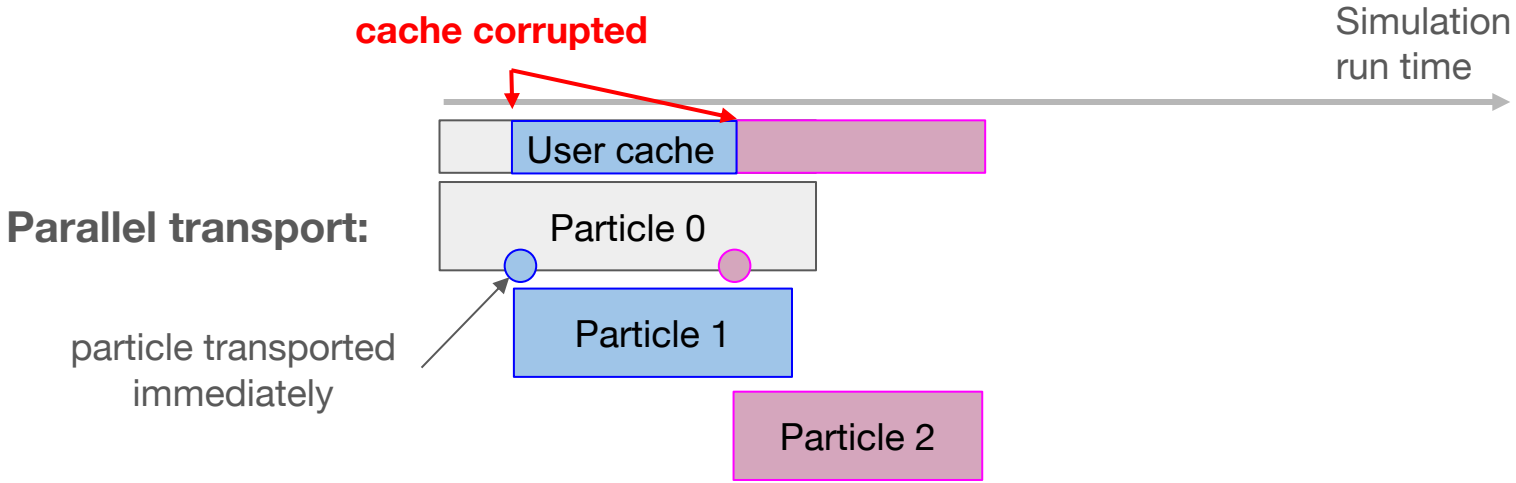
Consequences of parallel transport



Consequences of parallel transport



Consequences of parallel transport



Perfectly-valid Geant4 applications may not be compliant with parallel transport and may require adaptation. ***These are fundamental, algorithmic limitations!***
More information in the documentation

Current capabilities and limitations of AdePT

Particle types	e^-, e^+, γ
Physics	G4HepEm (gives ~20% speedup over native G4) provides specialized transport on CPU for region-based offloading G4EmStandardPhysics Option 0,1,2 + nuclear reactions
Geometry	GDML/Geant4 (with few exceptions)
Geometry backends	VecGeom
B-field (via covfie)	3D
Compute backends	NVIDIA

For more information, see the AdePT documentation

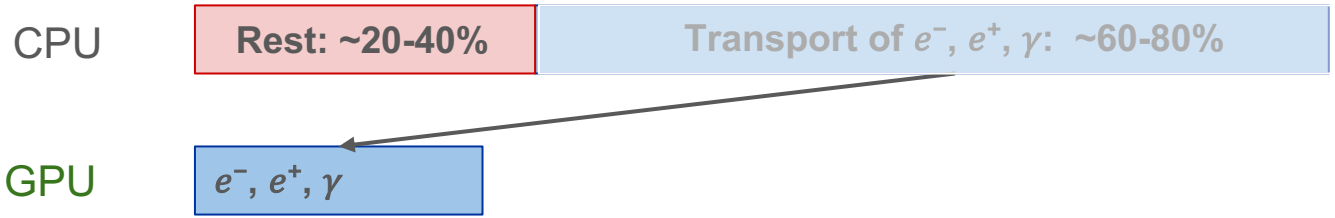
Results: LHC

Setting the expectations

Typical LHC production run:

Setting the expectations

Typical LHC production run:



Offloading the transport of e^-, e^+, γ to a GPU could give up to **~2.5x to 5x** speedup, depending on the setup

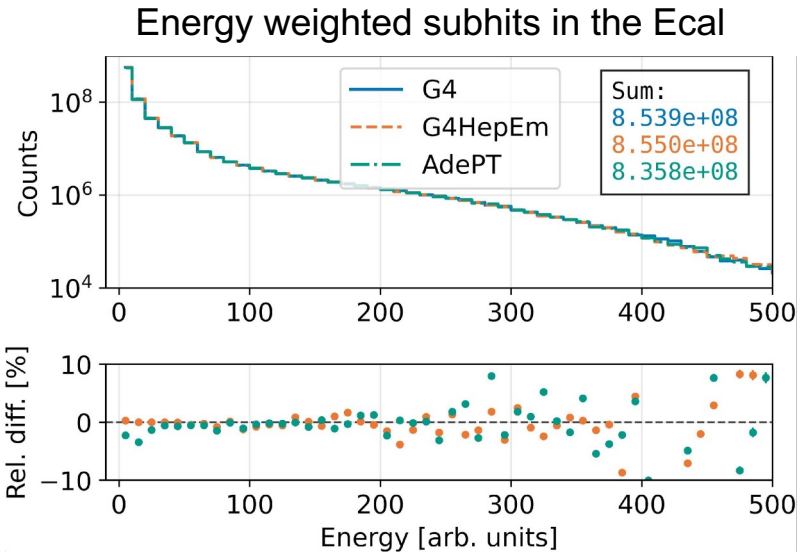
Results: LHCb

AdePT in Gauss (LHCb)

Application: Gauss + Gaussino
Geometry: LHCb detector
Field: 3D field map (v6r1 down)
Input: 20’000 min bias events

Production settings except user-caching particle history and optical photons

Offloading e^-, e^+, γ to GPU in: entire detector



Good physics agreement with small differences due to B field integrators

Performance results in Gauss

Hardware: Desktop PC
GPU: Nvidia **RTX4090**
CPU: AMD Ryzen 9 **16 cores**

No $e^{\pm}$, γ: **2.7x**

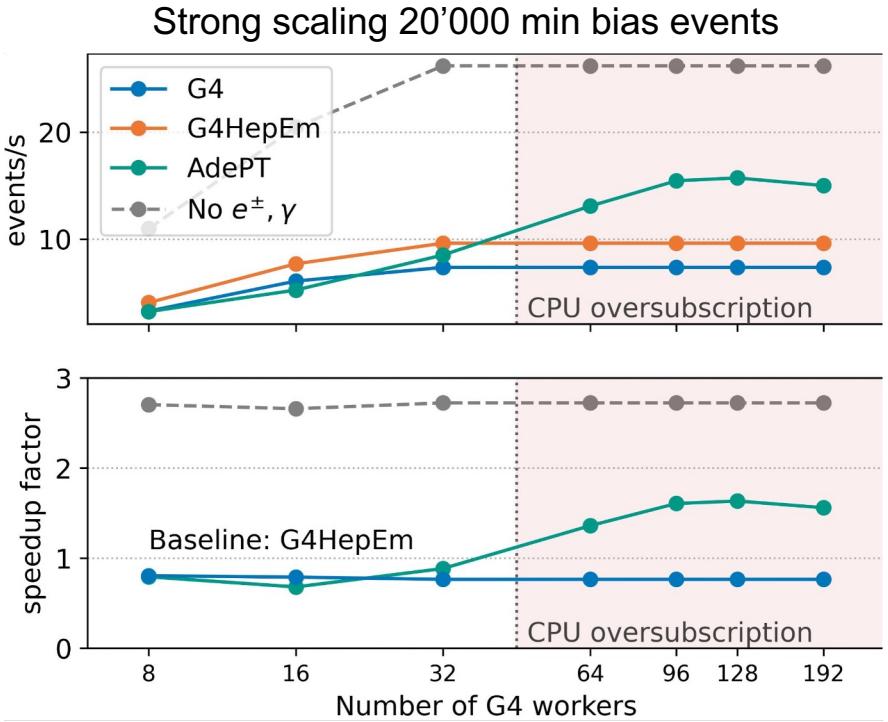
Performance results in Gauss

Hardware: Desktop PC
GPU: Nvidia **RTX4090**
CPU: AMD Ryzen 9 **16 cores**

No $e^{\pm}$, γ: **2.7x**

Current speedup:
1.6x at 128 threads

Oversubscription (4x)
needed to fill the GPU

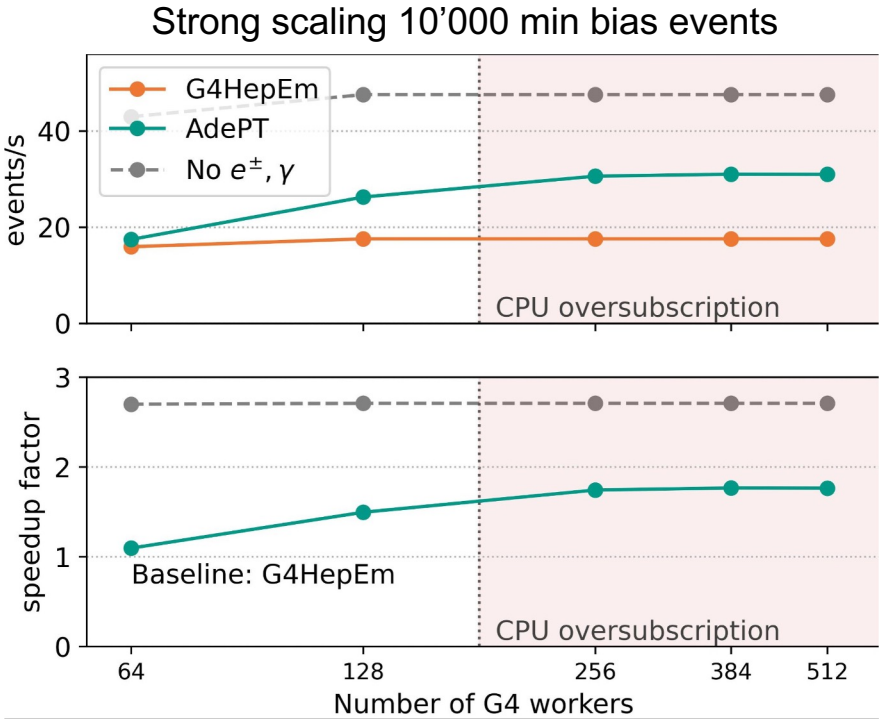


Performance results in Gauss

Hardware: 1x Perlmutter node
GPU: 4x **A100**
CPU: AMD EPYC 7763 (64 cores)

No $e^{\pm}$, γ: **2.7x**

Current speedup:
1.8x at 384 threads



Energy usage:
CPU + GPU vs CPU
0.258 kWh < 0.306 kWh
See talk on energy-efficiency

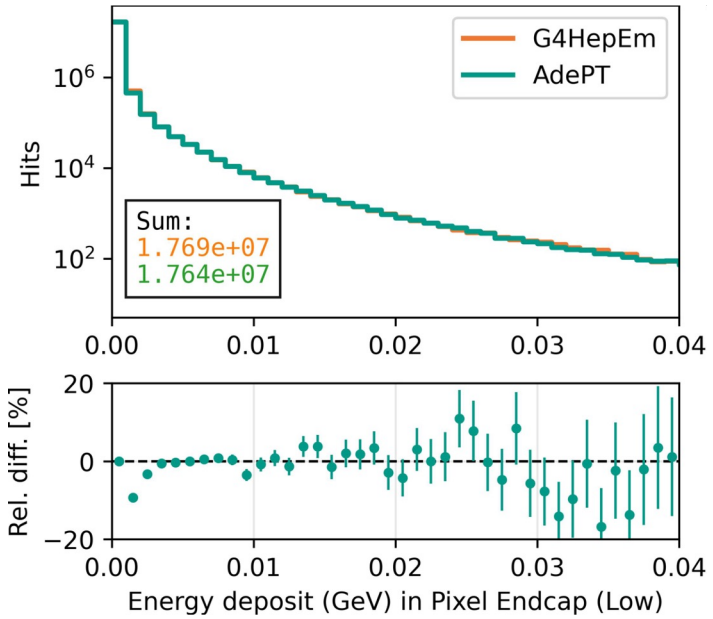
Results: CMS

AdePT in CMSSW (CMS)

Application: CaloSimHitStudy
Geometry: CMS detector,
Run 4 geometry (D110)
Field: 3D field map interpolated
Input: 20’000 ttbar events

Production settings except user-caching particle history and sensitive detectors

Offloading e^-, e^+, γ to GPU in:
Everything except HCAL



- Can reproduce some histograms, some differences need to be understood
- Others are off, due to caching

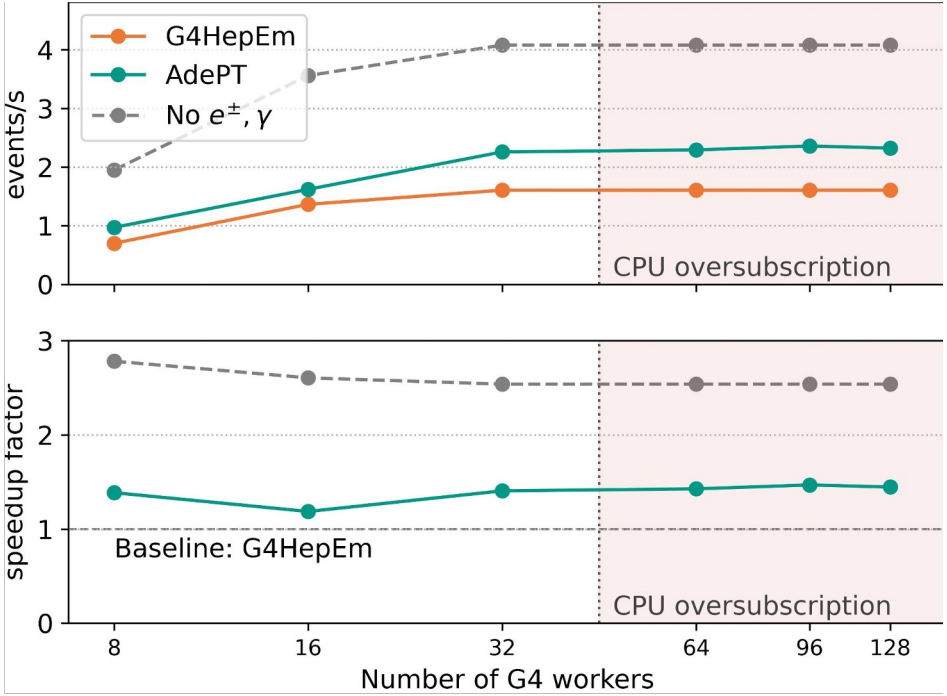
Performance assessment in CMSSW

Hardware: Desktop PC
GPU: **Nvidia RTX4090**
CPU: AMD Ryzen 9 **16 cores**

No $e^{\pm}$, γ: **2.6x**

Current speedup:
1.4x at 96 threads

Strong scaling of 2'000 ttbar events



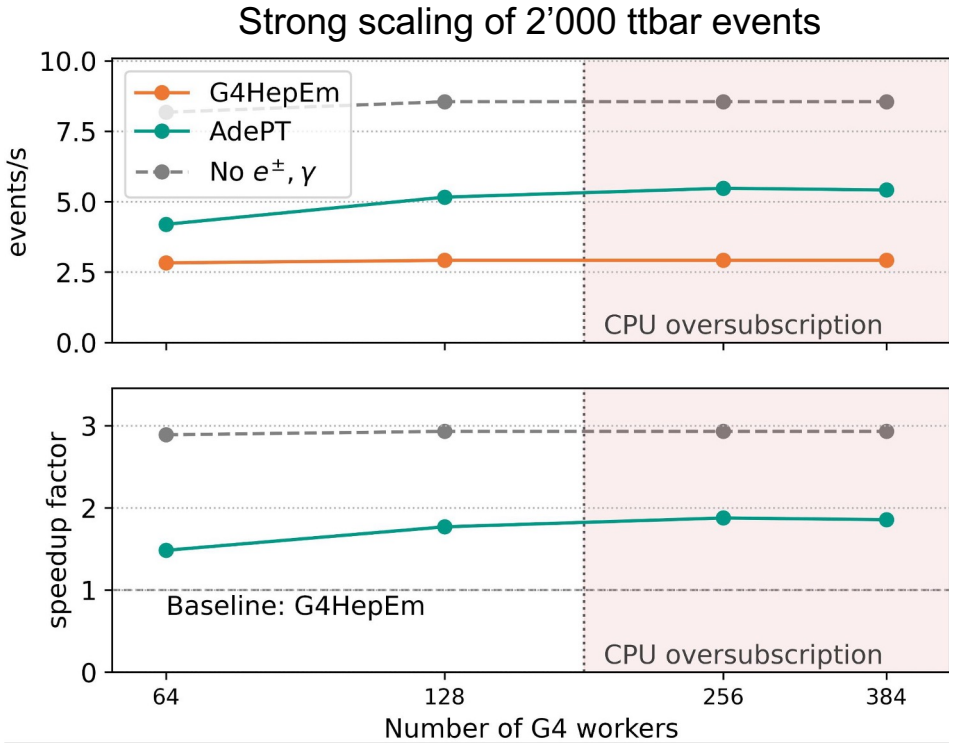
Performance assessment in CMSSW

Hardware: 1x Perlmutter node
GPU: 4x **A100**
CPU: AMD EPYC 7763 (64 cores)

No $e^{\pm}$, γ: **2.9x**

Current speedup:
1.9x at 256 threads

Energy usage:
CPU + GPU vs CPU
0.330 kWh < 0.370 kWh
See talk on energy-efficiency



Results: ATLAS

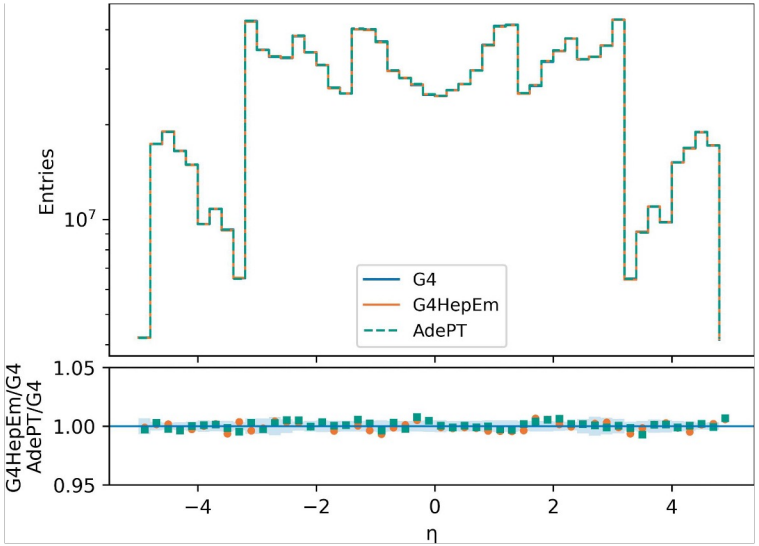
AdePT in Athena (ATLAS)

Application: AtlasG4_tf
Geometry: ATLAS detector, Run 4 geometry
Field: 3D field map interpolated
Input: 10’000 ttbar events

Production settings except user-caching particle history

Offloading e^-, e^+, γ to GPU in:
EMEC, EMB, HEC, PreSampLAr

Energy-weighted hit histogram along eta



AdePT on GPU shows **excellent agreement** with G4 on CPU in Athena

Performance results in Athena

Hardware: Desktop PC
GPU: **Nvidia RTX4090**
CPU: AMD Ryzen 9 **16 cores**

Entire detector, no $e^{\pm}$, γ:
~4.4x speedup

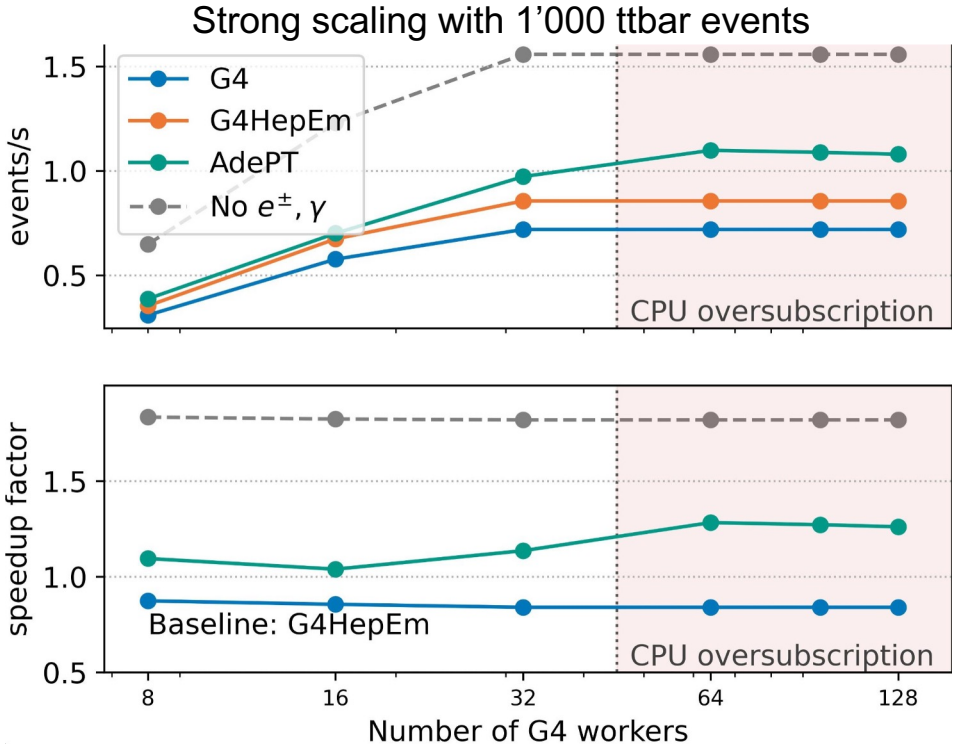
Offloading to GPU:
EMEC, EMB, HEC, PreSampLAr
No $e^{\pm}$, γ: **1.8x**

Performance results in Athena

Hardware: Desktop PC
GPU: **Nvidia RTX4090**
CPU: AMD Ryzen 9 **16 cores**

Entire detector, no $e^{\pm}$, γ:
~4.4x speedup

Offloading to GPU:
EMEC, EMB, HEC, PreSampLAr
No $e^{\pm}$, γ: **1.8x**
Achieved: 1.3x

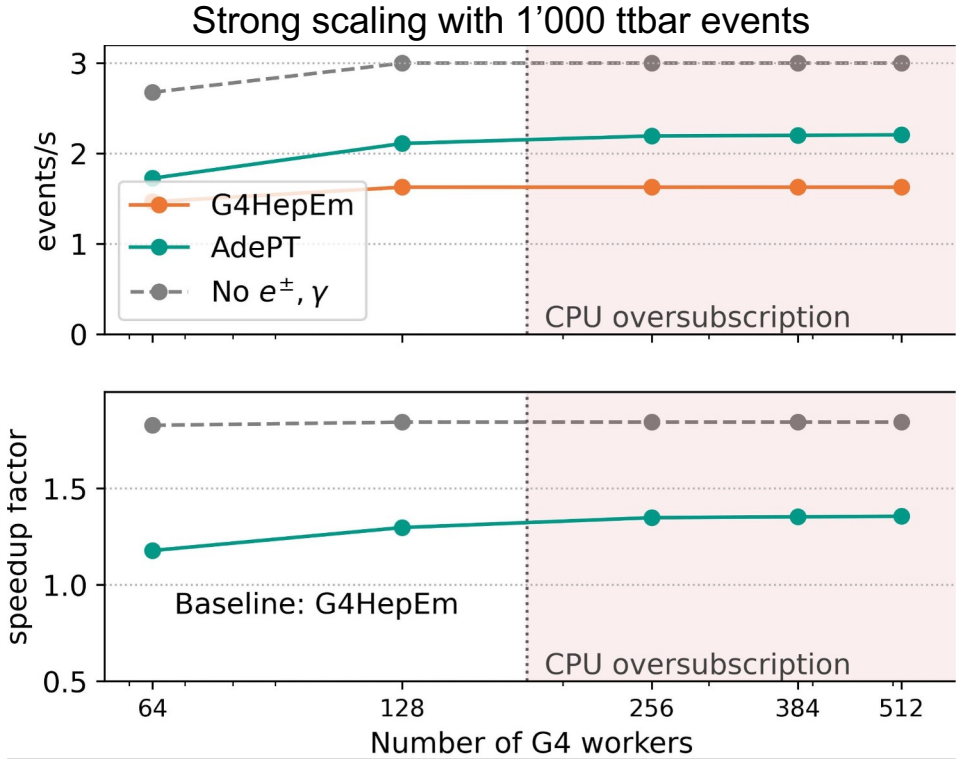


Performance results in Athena

Hardware: 1x Perlmutter node
GPU: 4x **A100**
CPU: AMD EPYC 7763 (64 cores)

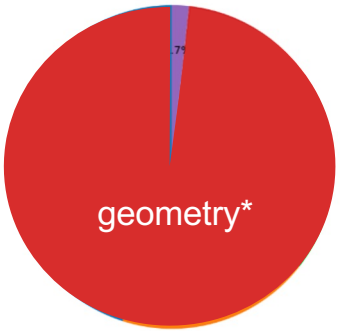
Offloading to GPU:
EMEC, EMB, HEC, PreSampLAr
No $e^{\pm}$, γ: **1.8x**
Achieved: 1.4x

Energy usage:
CPU + GPU vs CPU
0.321 kWh ≈ 0.316 kWh
See talk on energy-efficiency

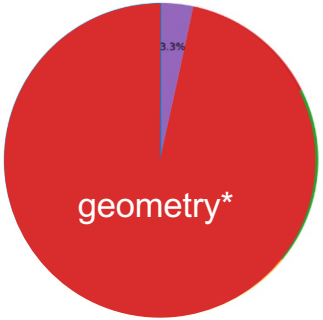


Geometry is the current bottleneck

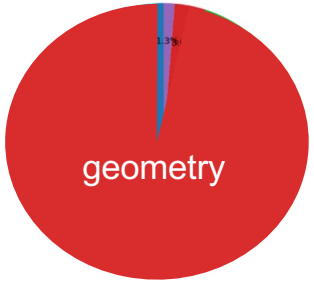
Electrons ~ 36 %



Positrons ~ 31 %



Gammas ~ 24 %



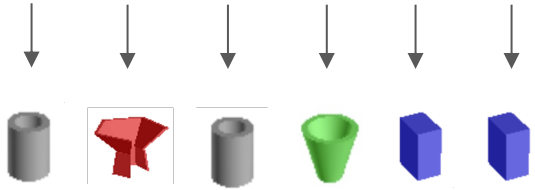
Compute time in ATLAS production runs on GPU:
>90% geometry kernel*

* some kernels also do other work, but are dominated by geometry

Geometry is the current bottleneck

Divergence and **random memory access** are the biggest performance penalties on GPU:

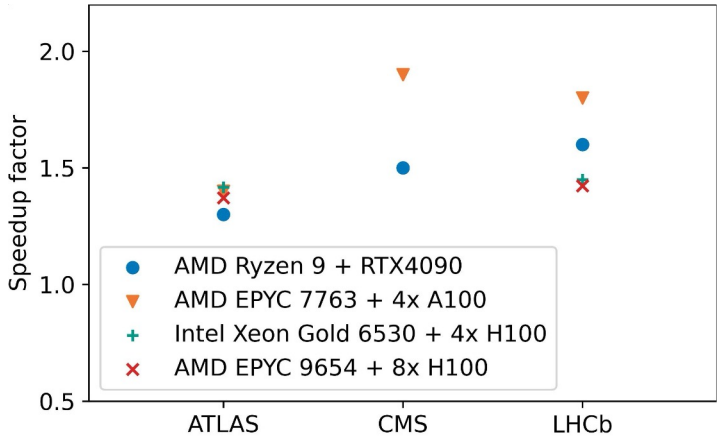
Different particles hit different volumes



Inherent problem in large detector setups!
Joint plan to improve VecGeom by AdePT & Celeritas teams

Current performance results with AdePT

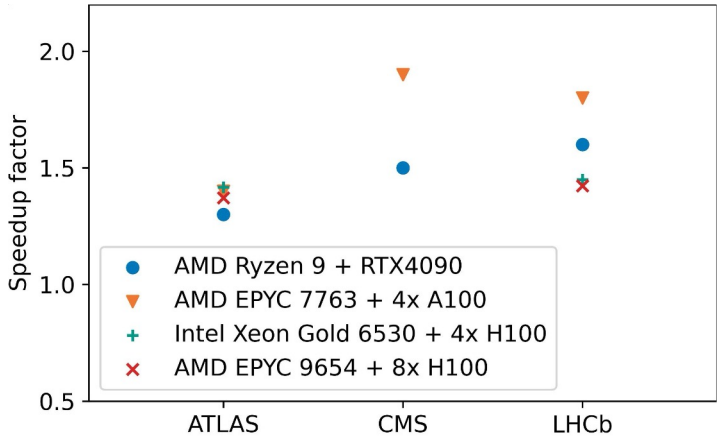
Performance results are strongly hardware-dependent - cheaper gaming GPUs might be an option



Current performance results with AdePT

Performance results are strongly hardware-dependent - cheaper gaming GPUs might be an option

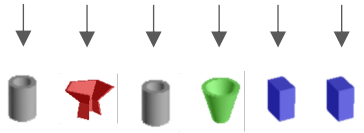
Proper benchmarking is needed -
first HepScore on GPU with AdePT + Athena is available (see Robin Hofsäss' CHEP talk)



Next steps

For AdePT:

- Improve raw performance (focus on geometry)
- Enable transport of more particle types on GPU
- Support optical physics:



First initial effort to use AdePT for EM physics and Simphony (EIC-Opticks) for optical physics, see presentation by Gábor

Next steps

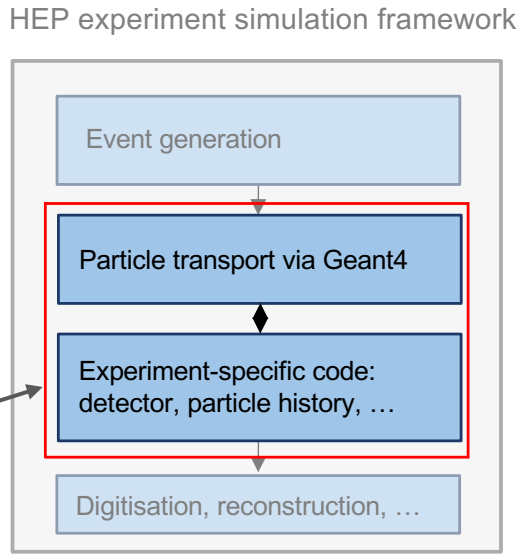
For AdePT:

- Improve raw performance (focus on geometry)
- Enable transport of more particle types on GPU
- Support optical physics:

  First initial effort to use AdePT for EM physics and Simphony (EIC-Opticks) for optical physics, see [presentation by Gábor]

For the HEP experiments:

- Comply with parallel transport of the GPU
- Experiment with direct scoring on the GPU



Summary

1. **Parallel transport on GPUs is a paradigm shift for detailed simulation -** software usually needs to be adjusted to use them

1. **Production-quality detector simulations on GPU are available** for LHCb, CMS, and ATLAS

1. **Energy-efficient simulations using GPUs have been achieved** although buying them for detailed simulation might not be cost effective

Conclusion:

We believe that GPU-accelerated detector simulations will soon be regularly used for production simulations

Acknowledgments

AdePT:

Current contributors: John Apostolakis, Wouter Deconinck, Severin Diederichs, Andrei Gheata, Juan Gonzalez Caminero, Stephan Hageboeck, Jonas Hahnfeld, Ben Morgan, Mihaly Novak, Witek Pokorski

Past contributors: Guilherme Amadio, Nazar Misyats, Bernhard Gruber

Funding sources:
This work has been [partially] funded by the Eric & Wendy Schmidt Fund for Strategic Innovation through the CERN Next Generation Triggers project under grant agreement number SIF-2023-004. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy User Facility using NERSC award HEP-ERCAP 0033546.

External thanks:

ATLAS: Marilena Bandieramonte, John Chapman, Michael Dührssen-Debling, Johannes Elmsheuser, Julien Esseiva
CMS: Vladimir Ivantchenko, Shahzad Muzaffar, Kevin Pedro
LHCb: Gloria Corti, Marco Clemencic
NERSC: Zhengji Zhao , Kevin Gott
CERN IT: Amine Lahouel, Robin Hofsaess, Domenico Giordano