

GeantV current status and future roadmap

ANDREI GHEATA FOR THE GEANTV PROJECT

CWP SIMULATION MEETING

MAY 22, 2017

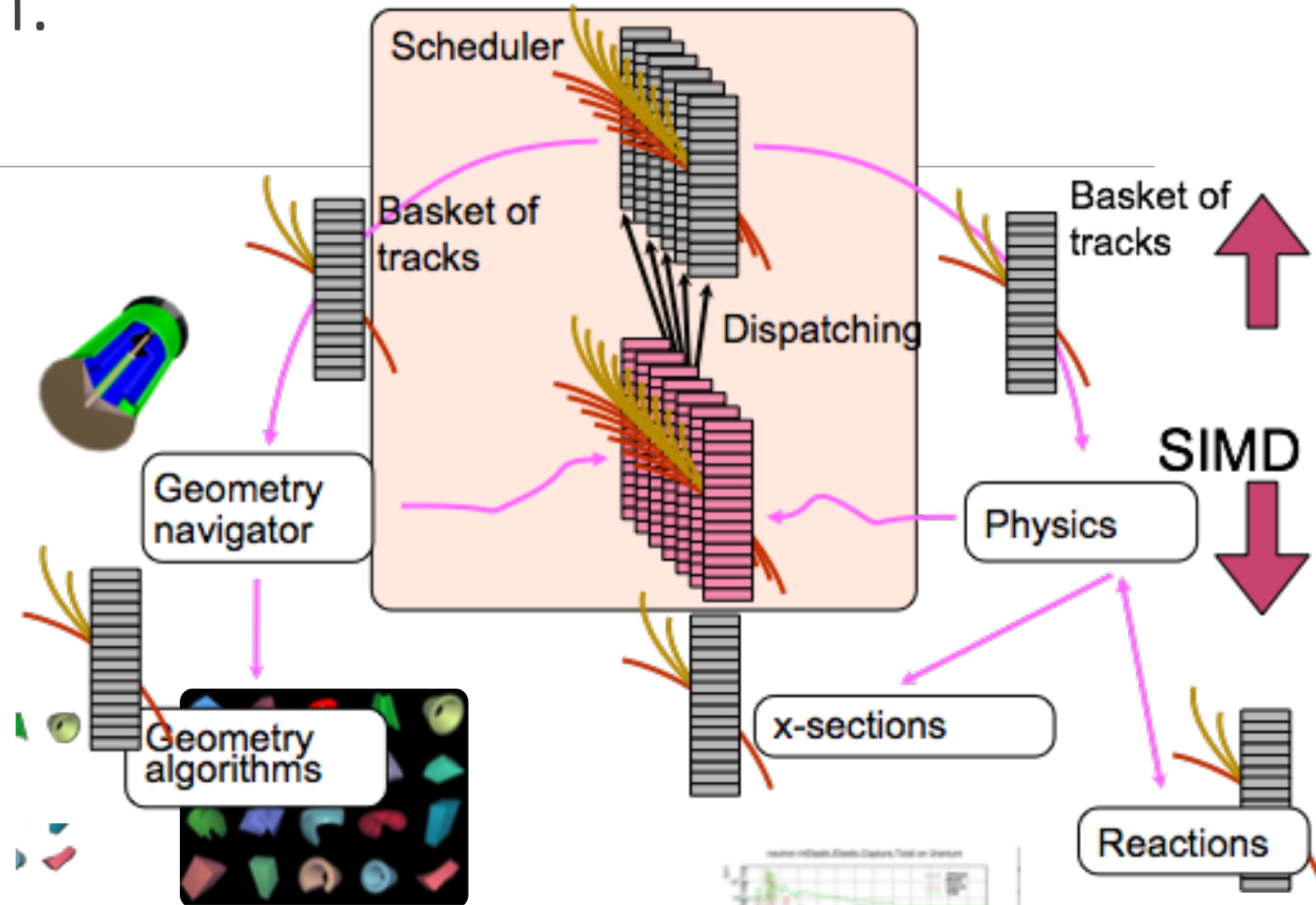
GeantV approach: boosting vectors

Transport particles in vectors
("baskets")

- Filter by geometry volume or physics process

Redesign library and workflow to target fine grain parallelism

Use an abstraction for vector types and their operations to achieve portable vectorization



Aim for a 3x-5x faster code, understand hard limits for 10x

Alpha release of GeantV (2017)

Providing stable interfaces and allowing experiments to “give it a try” with GeantV software

GeantV scheduler version 3

Finalized user interfaces, experiment integration with parallel flow (CMSSW)

Vectorized Runge-Kutta propagator

Vectorized geometry

EM physics most/all processes for e+/e-/gamma in scalar mode, first assessment on vectorization potential

Hadronic physics: Glauber-Gribov cross sections + low energy parameterisations, elastic scattering

Fast simulation hooks in GeantV, scope definition, integration and proof of concept based on examples

Full hit/MC truth cycle demonstrator

GPU demonstrator

Beta release of GeantV (2018)

Providing most of GeantV features/optimisations in terms of geometry, EM physics (partially hadronics), I/O and fast simulation. Allowing to actually integrate experimental simulations with GeantV as toolkit.

Production-quality scheduling, including error handling at the level of track/event, HPC demonstrator

Production-quality geometry supporting full features (construction and navigation) of Geant4 and ROOT

MC usage demonstrator based on realistic use cases. Integration with experiment SW.

EM physics – full shower physics (e+, e-, gamma), most CPU consuming models vectorized

Hadronic physics: Bertini cascade(?) + (partial)QGS, realistic model level and application level benchmarks

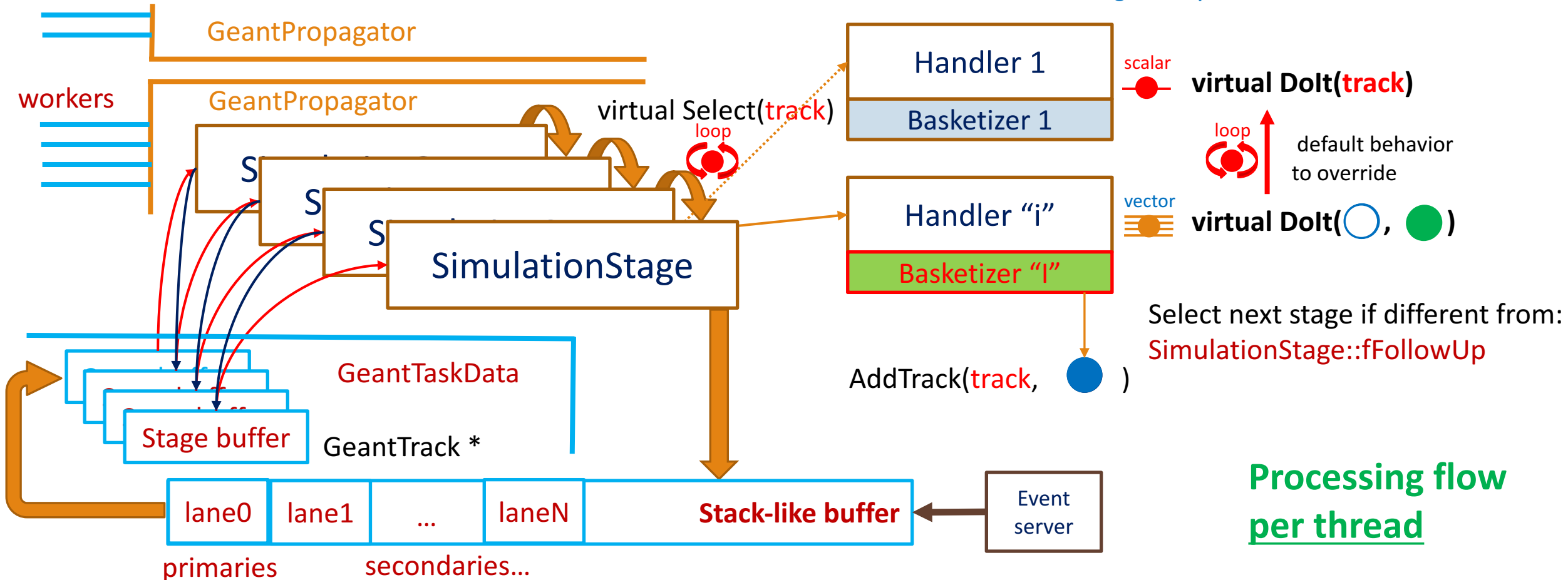
Integration of fast simulation with experimental frameworks, ML-based standalone tool + demonstrators for concrete cases

GeantV scheduler upgrade

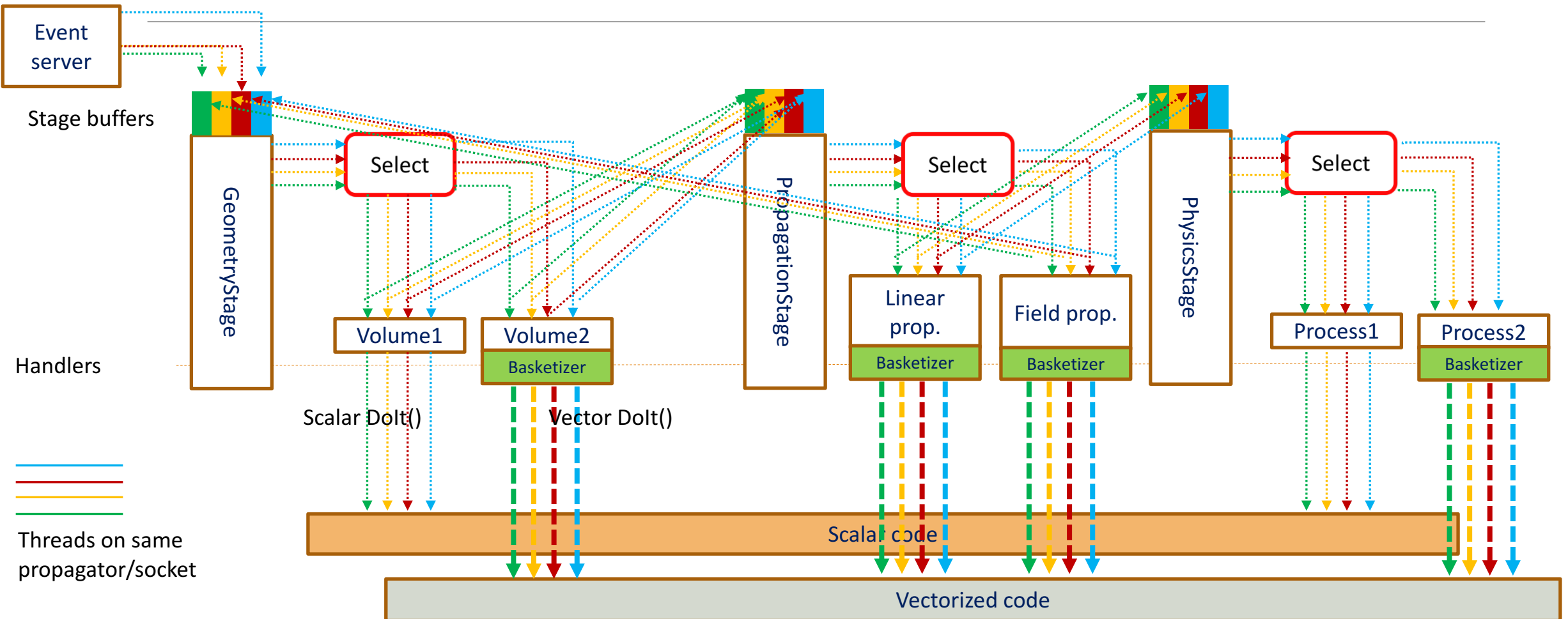
Scheduler version 2	Scheduler version 3
Geometry-centric basketizing approach	“Democratizing” the concept of basketizing to allow for physics multi-particle vectorization
SOA container handling: overheads for scatter/gather, reshuffling, concurrency	AOS handling in basketization, light SOA on demand for dispatching to vector code
“Avalanche” memory behavior: tracks are never released but only created, the full shower has to be kept in memory	More stack-like behavior, favoring transporting secondaries/low energy tracks with priority
Basket-driven concurrency based on non-local queues, adding contention points	Thread-local data and containers, relying less on common concurrency services (use my own data and containers as much as possible)
System-driven allocation of resources (threads, memory)	NUMA-aware allocation of resources, relying on topology discovery

GeantV version 3: A generic vector flow approach

e.g. ComptonFilter::Dolt



Processing flow per propagator/NUMA node



Performance preliminary V3 vs. V2

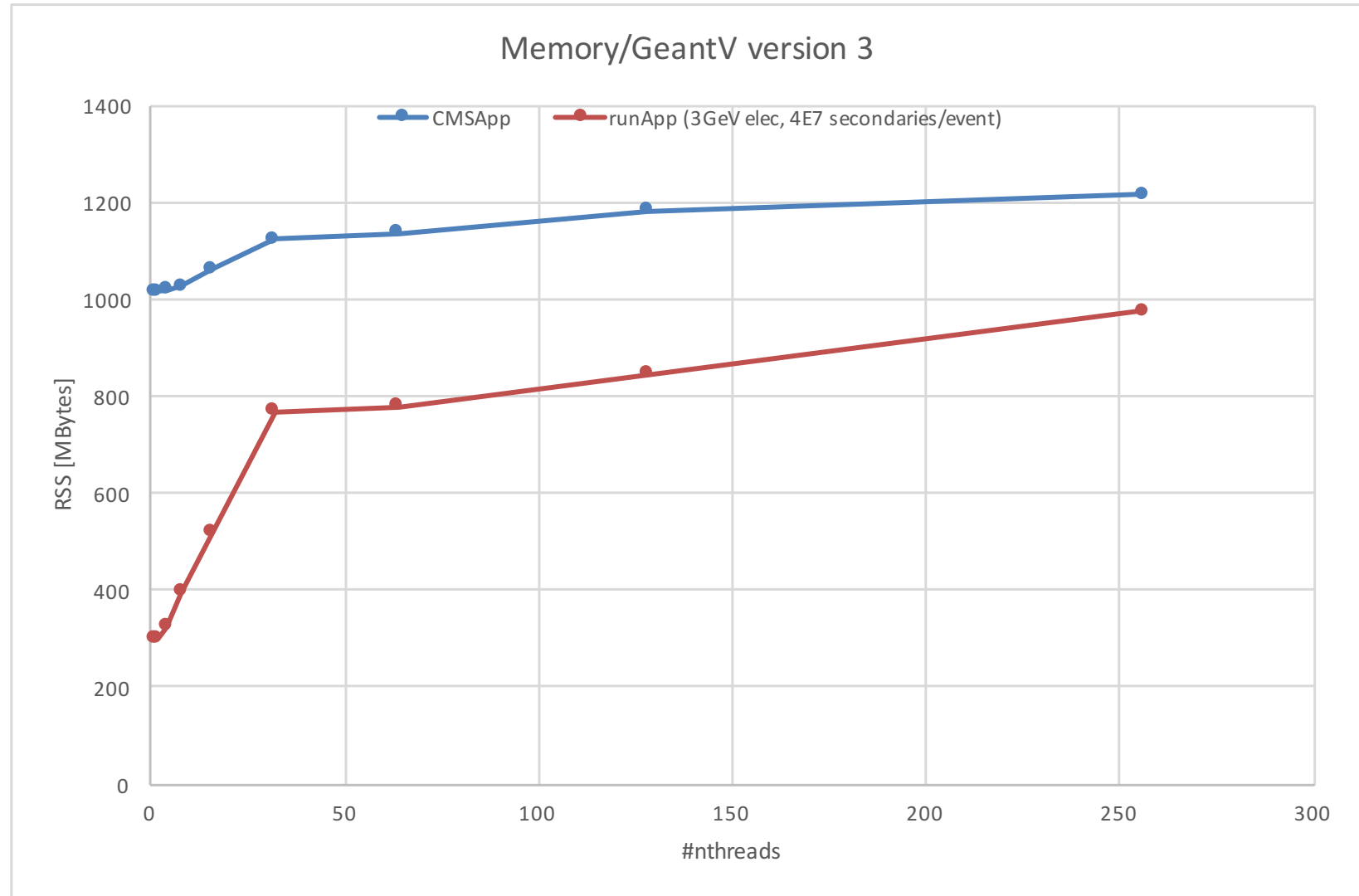
V3 VERSUS V2, MEMORY, SCALABILITY, NUMA, TUNING KNOBS
(FOR NOW JUST TABULATED PHYSICS, CMS SETUP & SIMPLE
CALORIMETER)

Memory control

Stack-like control using a special buffer inserted in the stepping loop

- Higher generation secondaries flushed with priority

Very good behavior even for high number of threads/secondaries



NUMA awareness

Implemented using hwloc > 1.8

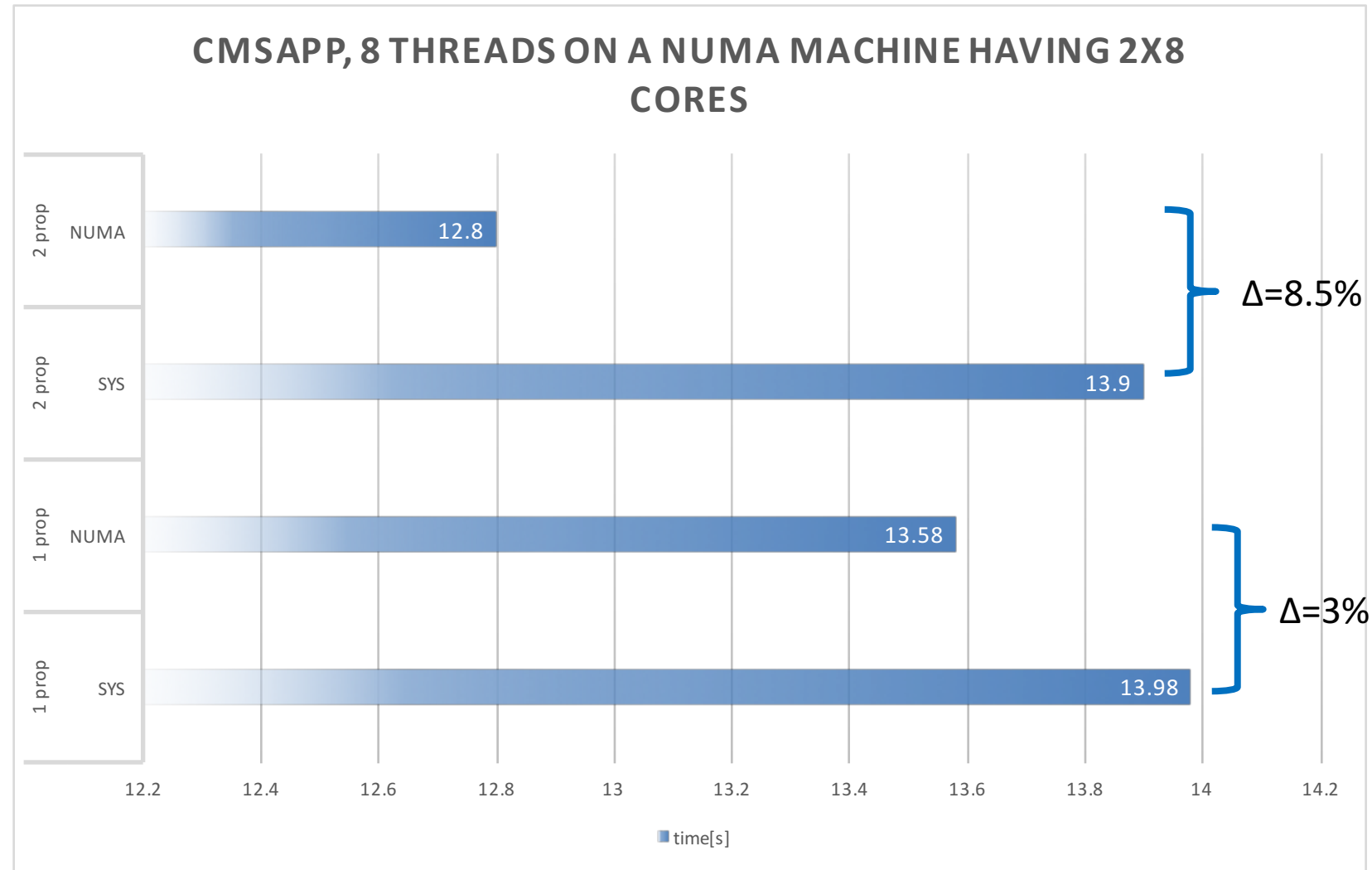
- Enumerating NUMA nodes, cores, CPU's
- Threads are bound to CPU's

A propagator will use threads bound to the same NUMA node

- More propagators can be bound to the same NUMA node

Compact policy used for threads on same propagator, scatter for distributing propagators on different nodes

Task data stage buffers, stack-like buffer, baskets and tracks bound to memory on the same node as the propagator owning the thread



Scalability

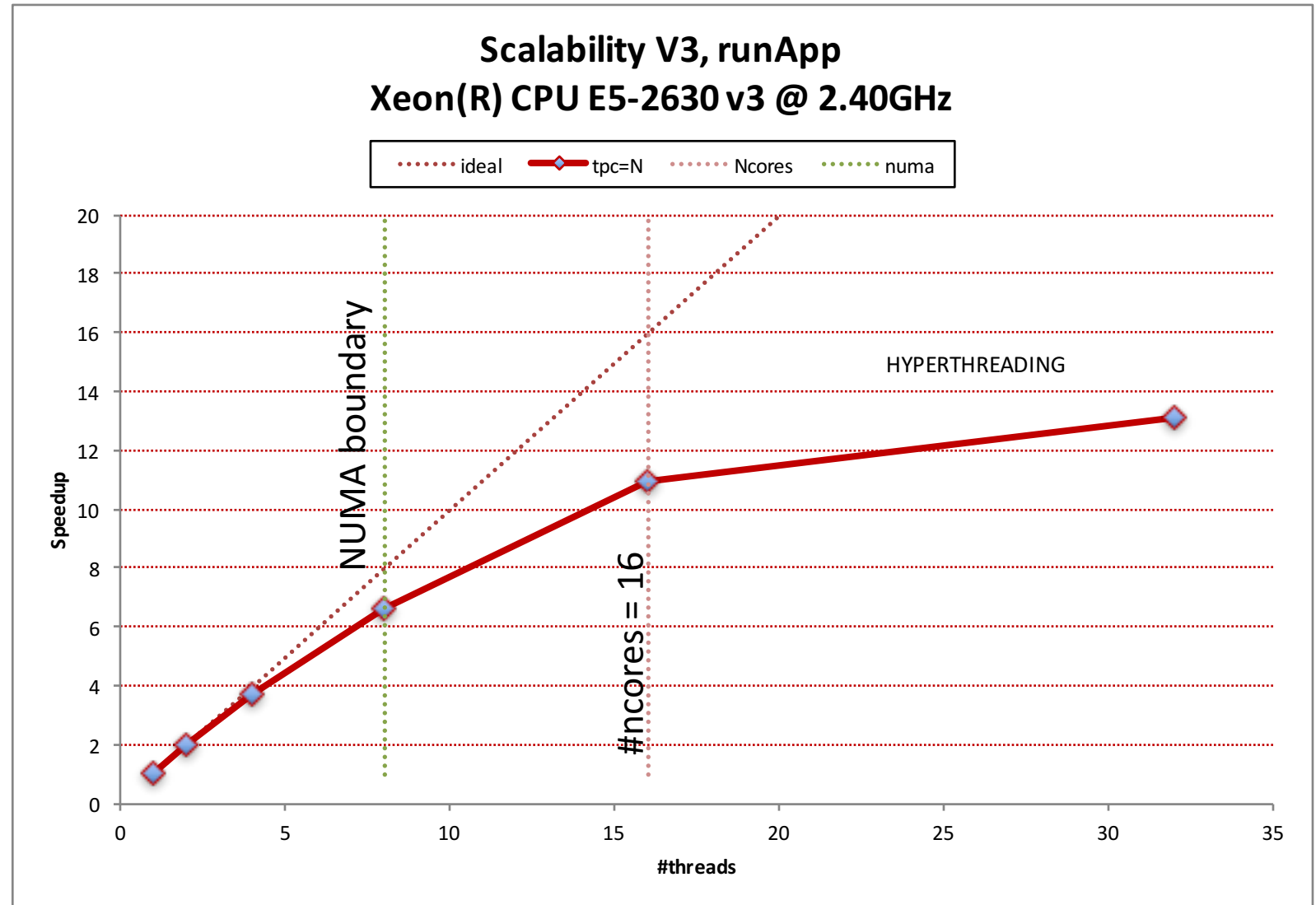
Not as good as expected

- Interaction between threads lesser, removed contingency points, SOA basketizing, no more basket queue

Profiling comparison N/2N threads does not reveal obvious hotspots

- To be further pursued

Memory operations are high in the profile, we expect picture to improve when having a more balanced scenario with more (vector) work on physics side.

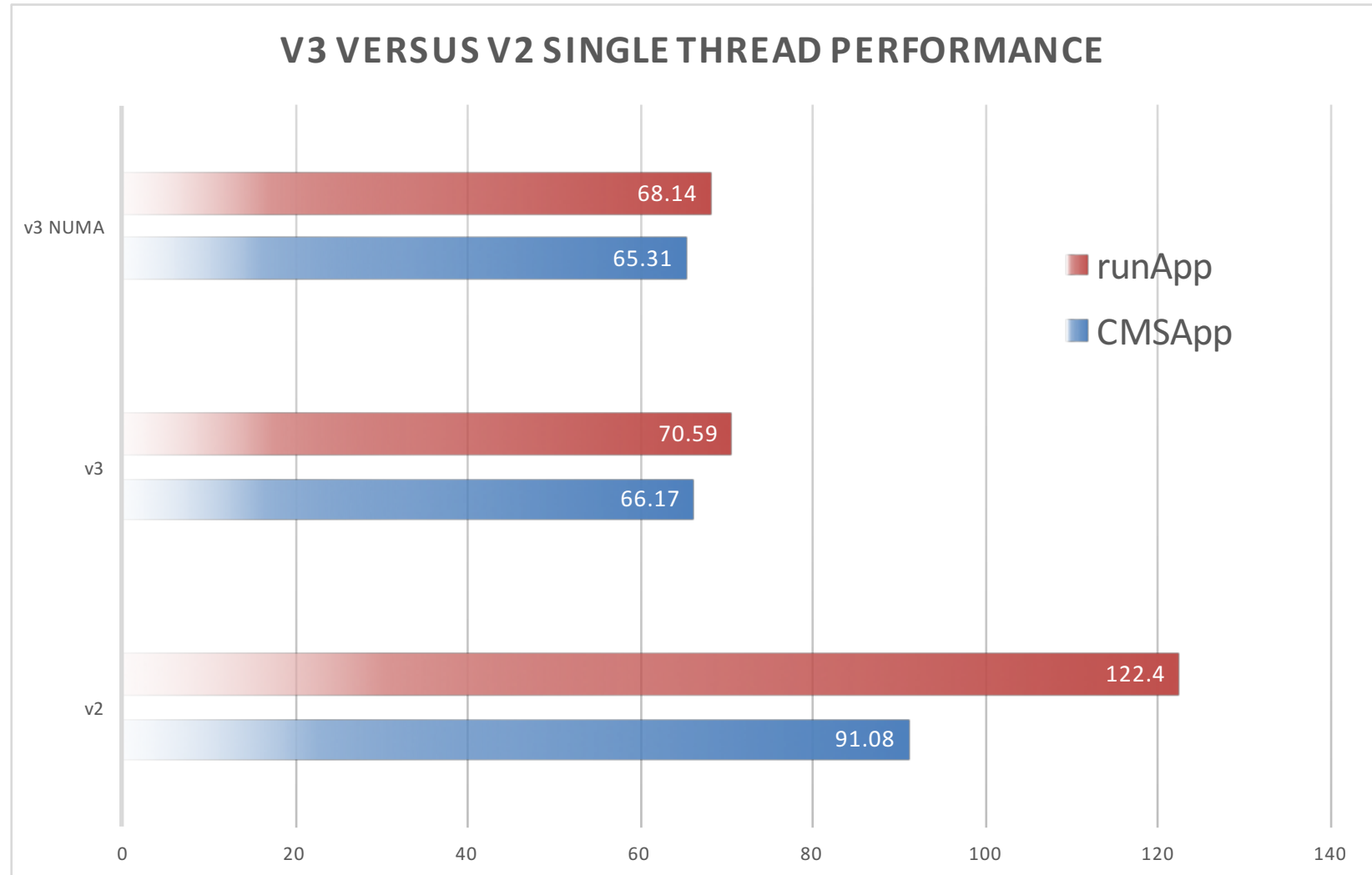


Performance v3 versus v2

Relevant improvements in both single and multi-threaded mode

- Coming mostly from the increase of locality (simulation stages)
- Removal of SOA gather/scatter overheads
- NUMA awareness

Yardstick measurements to be redone



Physics goals

Full EM shower simulation in beta release

- Most important performance-related component for simulation
- Demonstrate important gains due to vectorization + locality (treatment of baskets)

Hadronics covering up to 10 GeV

- Glauber-Gribov cross sections (elastic,inelastic,total) + low energy neutron/pion parameterization (Barashenkov)
- Elastic scattering: rewrite from scratch, converging to a common version for GeantV and Geant4
- Bertini(?) cascade extraction from Geant4
- QGS, probably not the full blown model
- R&D on evolution towards other promising models (e.g. EPOS)
- The EM component for hadronic showers (?)
- Fast simulation allowing functionality as in current Geant4, plus independent generic module based on ML with concrete examples

Today status of RP ("real physics") with V3

10⁴ 100 [GeV] e⁻ in ATLAS bar. simpl. cal. : 50 layers of [2.3 mm Pb + 5.7 mm IAr]

e ⁻ /e ⁺ : ionisation, bremsstrahlung; γ : Compton, conversion						
GeantV			Geant4			
material	E _{dep} [GeV]	length [m]	E _{dep} [GeV]	rms [GeV]	length [m]	rms [m]
Pb	65.425	47.536	65.397	1.139	47.517	0.769
IAr	24.993	116.801	24.987	0.419	116.777	1.771

Mean number of :

gamma	38524	38515
electron	960374	960484
positron	5254	5253
charged steps	1069467	1069700
neutral steps	5172143	5175038

Plans for vectorizing physics

Goal: vectorizing the final state sampling methods of the physics models

Handlers will be automatically created for each different model

The post-step action stage will select tracks for a given model

The vector version of final state sampling method of the models will be possible to invoke

Performance assessment: switching on/off basketizing per model handler

Next priority work (end of summer)

Integration with RP: full shower simulation with scheduler V3

- Most models already integrated with master
- MSC with v3
- Photoelectric
- Integration/testing new RK propagator (scalar/vector)
- Yardstick measurements GeantV vs. Geant4 extended to RP

Geometry vectorization activation for v3, fixes to fully match Geant4 raytracing in complex geometry

Finalize user interfaces

Hadronics

- Include Glauber-Gribov cross sections, finish initial development of elastic scattering process, evaluate feasibility of model “extraction” from Geant4 (e.g. Bertini)