7B: PARALLEL & VECTOR

J. Apostolakis, CERN EP/SFT

TALKS

Contribution list | **Timetable**

Thu 28/09

Print | PDF | Full screen | Detailed view | Filter

Time	Talk	Speaker / Time
16:00	**Deterministic transport independent of the order of tracking** *Building 6, room 210*	*Dmitry Savin* 16:00 - 16:20
	Update on GeantV *Building 6, room 210*	*Federico Carminati* 16:20 - 16:40
	Enabling vectorisation in Geant4 and single-particle GeantV tracking *Building 6, room 210*	*John Apostolakis* 16:40 - 17:00
17:00	**Discussion** *Building 6, room 210*	17:00 - 17:30

Deterministic transport independent of the order of tracking

D. Savin, J. Apostolakis, S. Wenzel

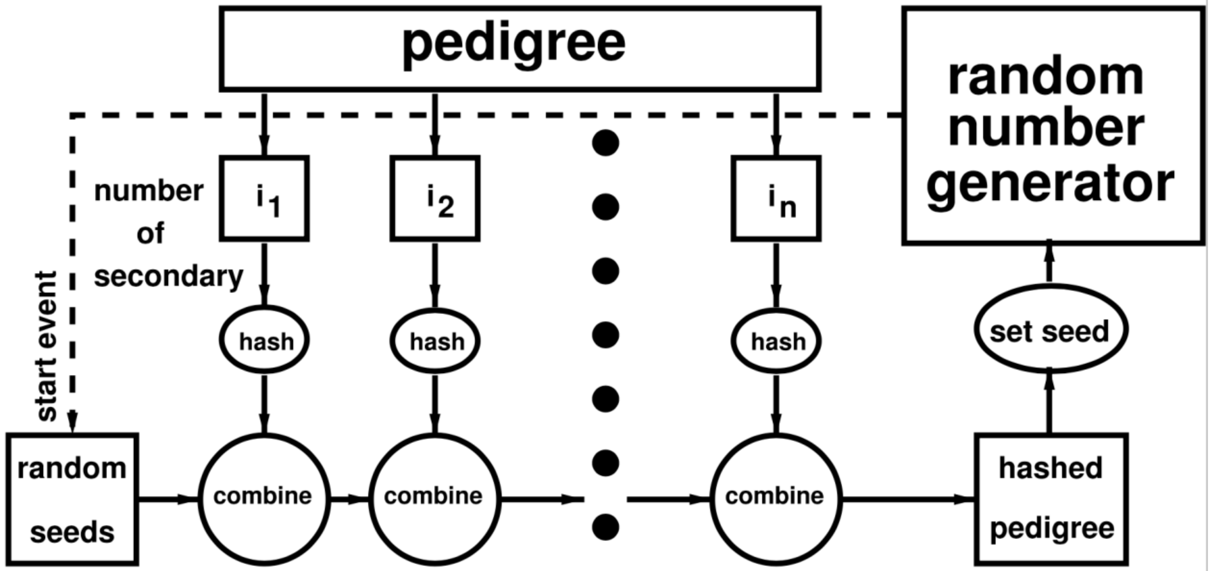
Motivation

- Monte Carlo simulation relies on pseudo-random numbers
- The numbers must seem uncorrelated
- Reproducibility from any point of simulation is needed

Pedigrees

- Define:
  - the rank of a track i_k - the number of sibling tracks created by its parent track before it
  - the pedigree of a track $\{i_1, i_2, \dots, i_n\}$ - the sequence of ranks of its ancestors
- the hashed pedigree can be stored by the track and used to seed the generator

Merkle–Damgård hash calculation



Random number engines

- Ranecu
- Ranlux
- Ranlux64
- HepJamesRandom
- Ranshi
- DualRand
- Mersenne Twister
- MixMax
- CBPRNG

CAVEAT:

- Ranecu, Ranlux, HepJamesRandom, Ranshi, DualRand – legacy engines kept for compatibility and should not be used
  - Not indicated in the manual!

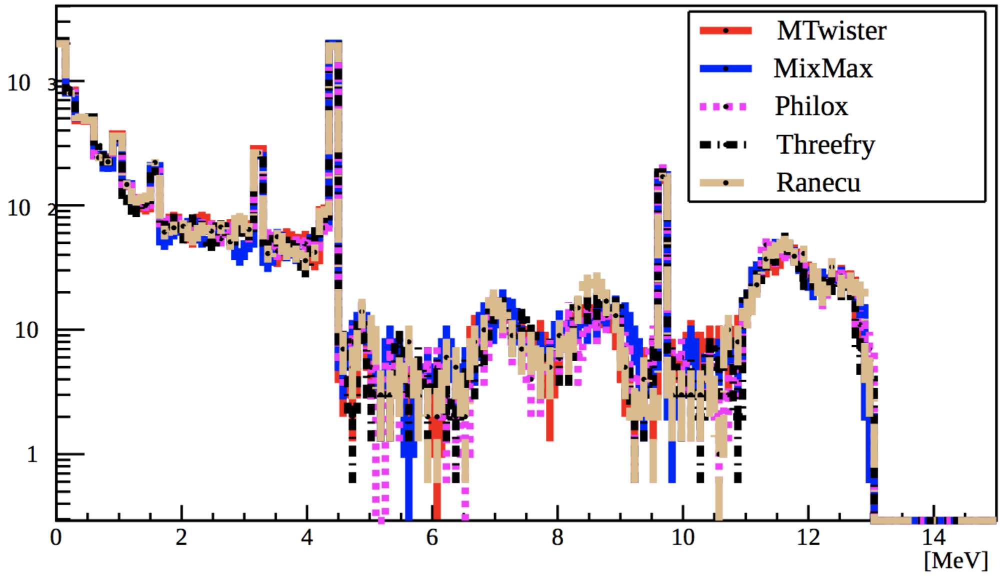
Counter-Based Pseudo-Random Number Generators

- State transition is a simple increment
- Randomness produced by the output function
- Seeding is a single assignment
- Cryptographic function for robustness, simpler functions for performance
- Vectorization may increase performance
- Philox1x64, Threefry1x64 from Random123 library were wrapped in a CLHEP interface

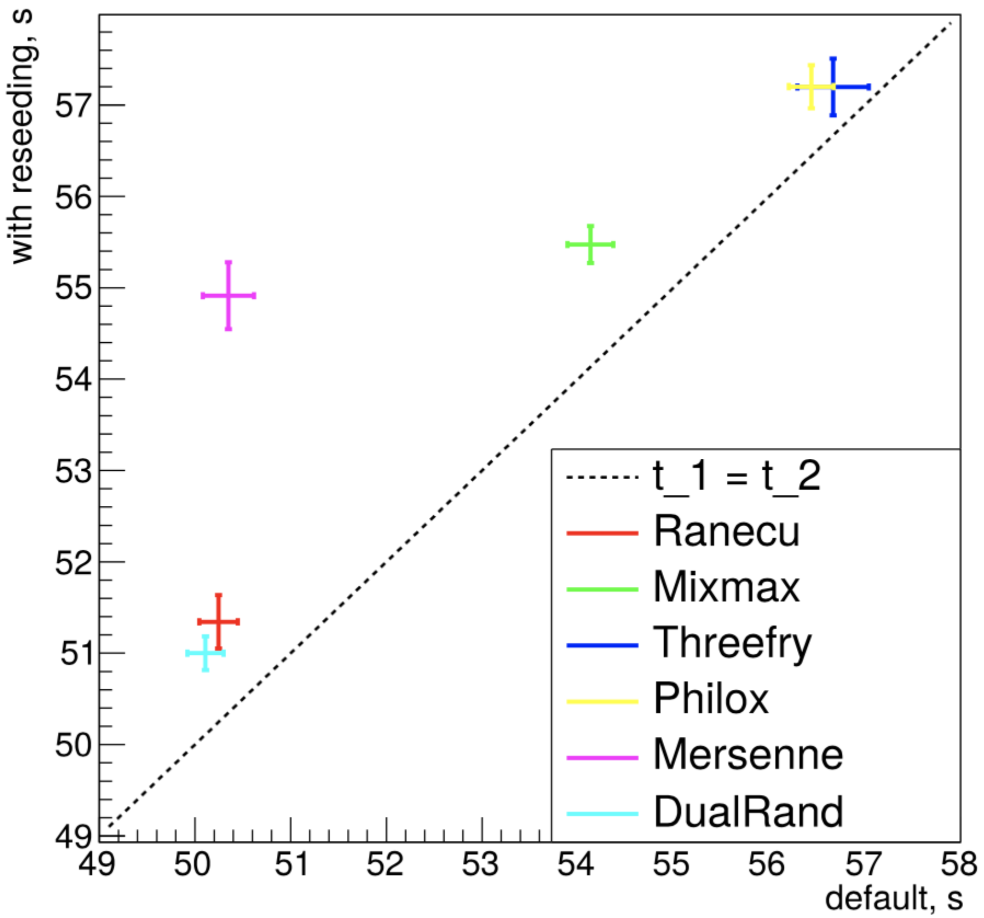
Engines testing
Performance results

Engines testing
Physics results

energy spectrum of emerging gamma



Run time



Speeding up single-particle tracking in Geant and GeantV using vectorisation

John Apostolakis, CERN

In GeantV non-trivial fraction, O(10%) is currently tracked in single particle mode

- volumes with small flux or volume
- ‘straggler’ tracks to end event(s)

Components with vectorisation potential

VecGeom is an existing common component. Two parts suit vectors:

- Solids with several similar faces
- Intersection of bounding boxes
- Intersection of same solids*

Runge Kutta **integration** of charged particle trajectories

- integrates 6-8 variables at once

Others?

Runge Kutta integration

Intertwines

- Vector operations over 6 (x,p) or 8 (x, p, E, t) variables
- calls to field evaluation and force calculation

Using templated ‘calls’ to inline field methods, all the code can be put into one compilation unit

- Prototype in G4 by Josh Xie (GSoC ‘14), reused in GeantV

Plan to investigate vectorisation potential for scalar tracks - potential GSoC project in 2018

$f_1 = f(y_0 + c_1 f_0 s)$

$b_2 = a_{20} f_0 + a_{21} f_1$

$f_2 = f(y_0 + c_2 f_0 s)$

$b_3 = a_{30} f_0 + a_{31} f_1 + a_{32} f_2$

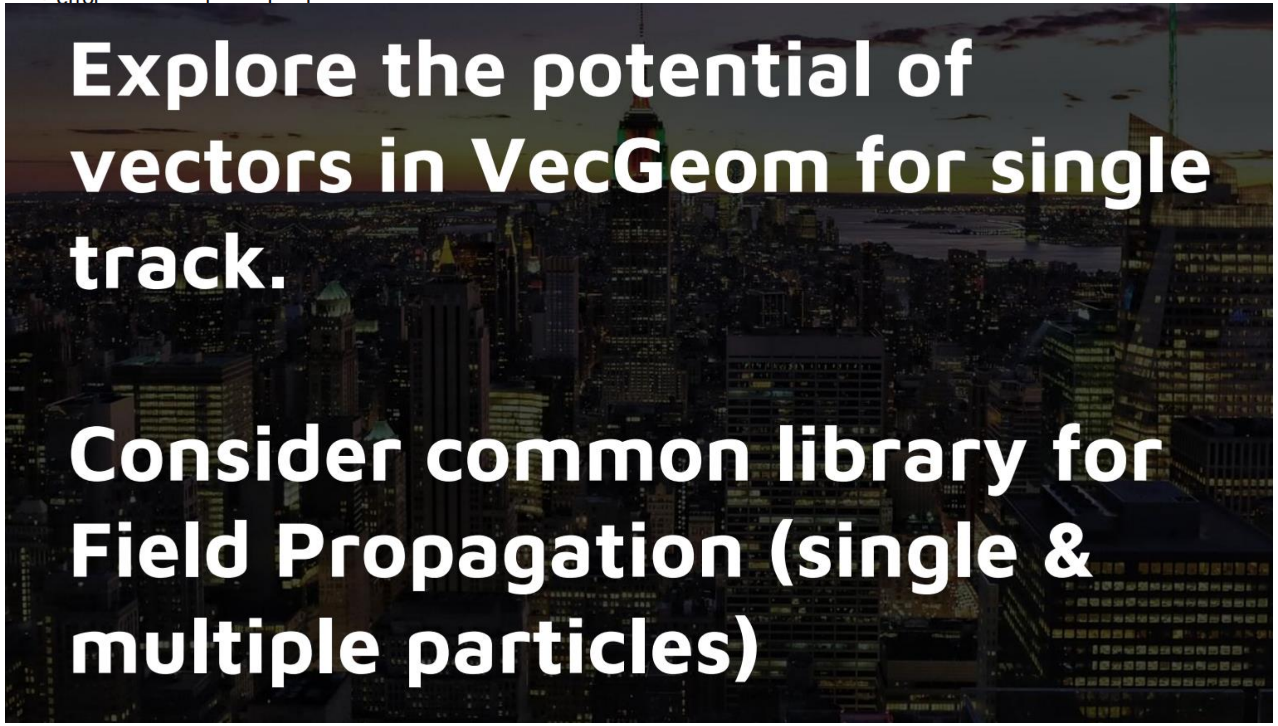
$f_3 = f(y_0 + c_3 f_0 s)$

$b_4 = a_{40} f_0 + a_{41} f_1 + a_{42} f_2 + a_{43} f_3$

....

$y_{end} = y_0 + s * \square d_i f_i$

$y_{error} = \square (d_i - d^*_i) f_i$

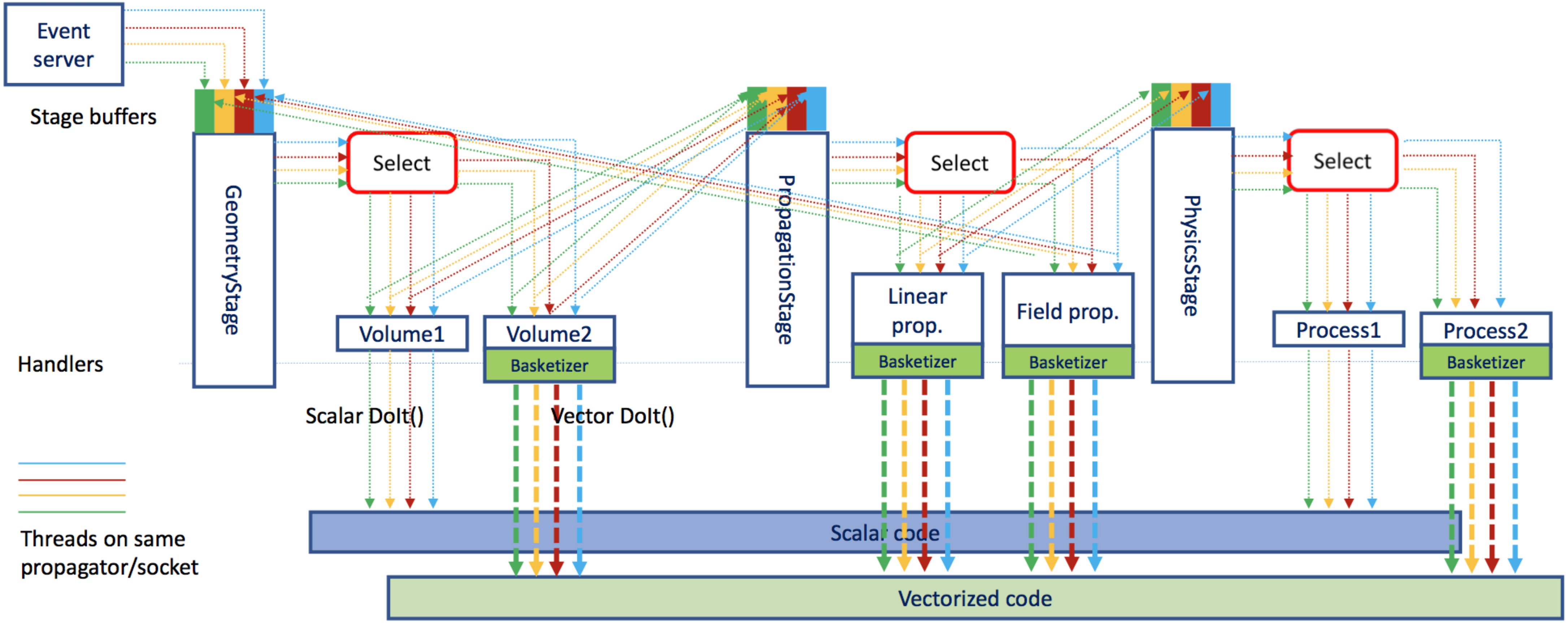


Update on GeantV

F.Carminati
for the GeantV collaboration

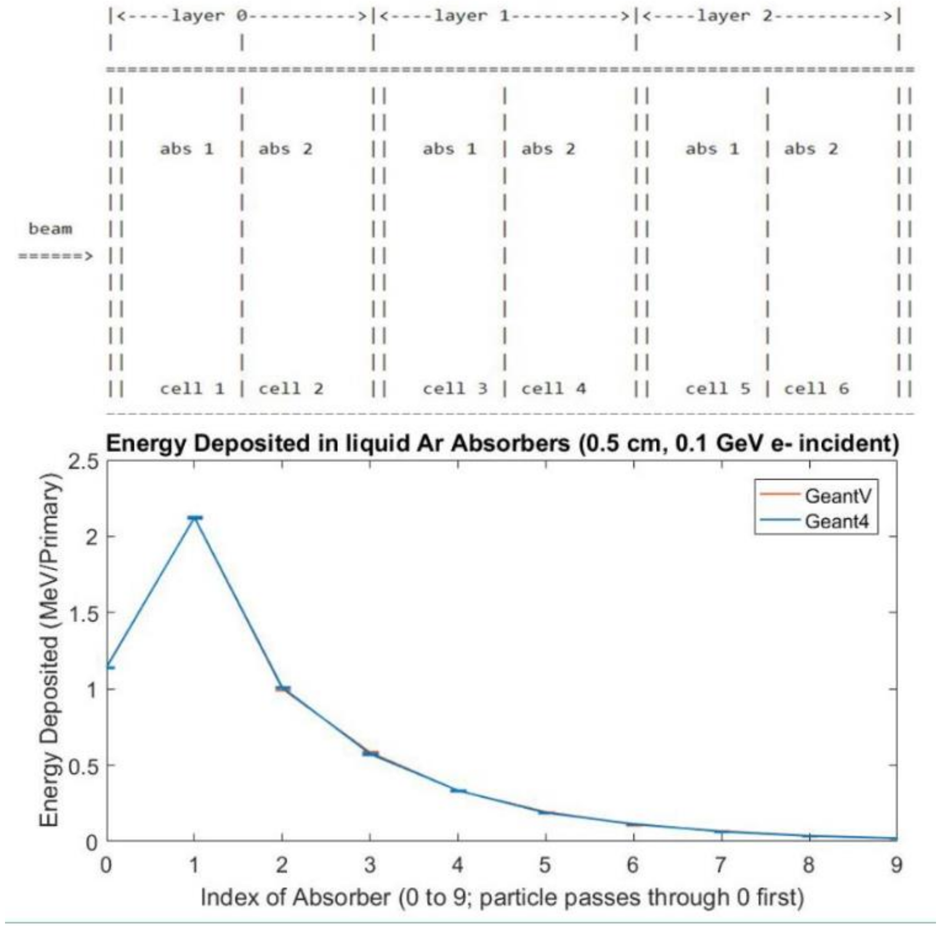
22nd Geant4 Collaboration Meeting – 25th - 29th September 2017

New transporter with scalar and vector processing combined



User application: examples

- Fully configurable GeantV example applications
  - Including configurable user defined detector construction, geometry, physics list with custom physics process, primary generator
  - Showing how to do scoring
  - Geant4 equivalent also demonstrated and compared
- Current examples
  - Simple calorimeter
  - More examples being prepared



GeantV roadmap

RELEASE	CORE	USER INTRFACES	GEOMETRY	PROPAGATOR	EM PHYSICS	HADRONIC PHYSICS	FAST SIMULATION	OTHER
alpha (2017) stable interfaces allowing experiments to create own examples	v3, vectors (all) NUMA, TBB	functional	vectorized, most solids	RK vectorized	shower (e^+, e^-, γ), most processes, scalar/validated	elastic scattering validated, scalar	Simulation stage hooked in core v3	examples, demonstrators vs. Geant4
beta (2018) most of GeantV features/optimisations. Allowing to actually integrate experimental simulations with GeantV as toolkit	vectors (all) NUMA, TBB HPC, device[1]	finalized	vectorized, full set, fully validated	RK vectorized	shower (e^+, e^-, γ), all processes, most vectorized[2]	elastic scattering, vectorized[2] Bertini, EPOS, QGS[3] (?)	Inference from ML-based module	Reproducibility, error handling at event/track level, integration with exp. frameworks

[1] non-standard processor & GPU
[2] full assessment of vectorization potential not available for all models
[3] partial implementation

MODULARITY

- VecGeom is a first library in common between Geant4 and GeantV
- "It has pried the door open for a modularisation of Geant4"
- The advantages are potentially large
  - Can concentrate on the "core business" (physics)
  - Be more open to external contributions
  - Allow a simpler evolutionary path
- It could serve as an example – or be once off
  - Other potential case(s): integration of ODE for field propagation,
- Geant4 could embraces this approach, to mutual benefit

GEANTV OUTLOOK

- GeantV developments on a tight schedule towards the end-of-year release
  - Program of work became much more focused on releases
  - Most components almost ready (core, EM physics, interfaces & examples), others still under development (field propagation, hadronic physics, more complex examples with new EM physics, complete flow including GPU)
- The alpha release will feature mainly shower EM physics
  - Delivering part of the design performance (vectorized geometry but scalar physics)
  - Interfaces mature enough to make the product usable by experiments for testing
  - Many more features and design performance expected in the beta version
- Important R&D still ongoing
  - Event loop steered by user application + CMSSW demonstrator
  - Generic fast simulation approach + integration as simulation stage
  - GeantV optimization for HPC