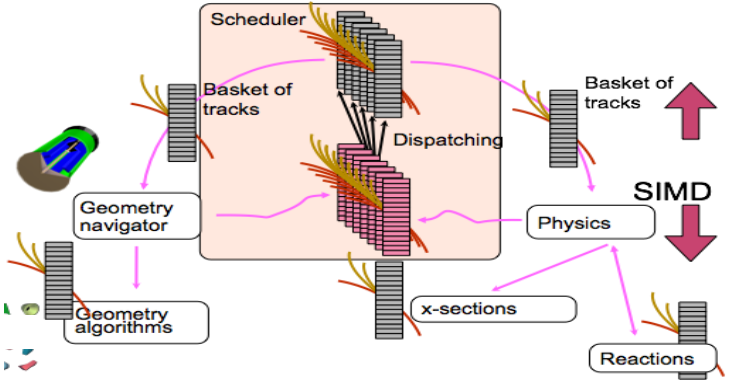


Status and plans for the vectorized particle transport demonstrator

Witek Pokorski for the GeantV team

Joint WLCG & HSF workshop

28.03.2018

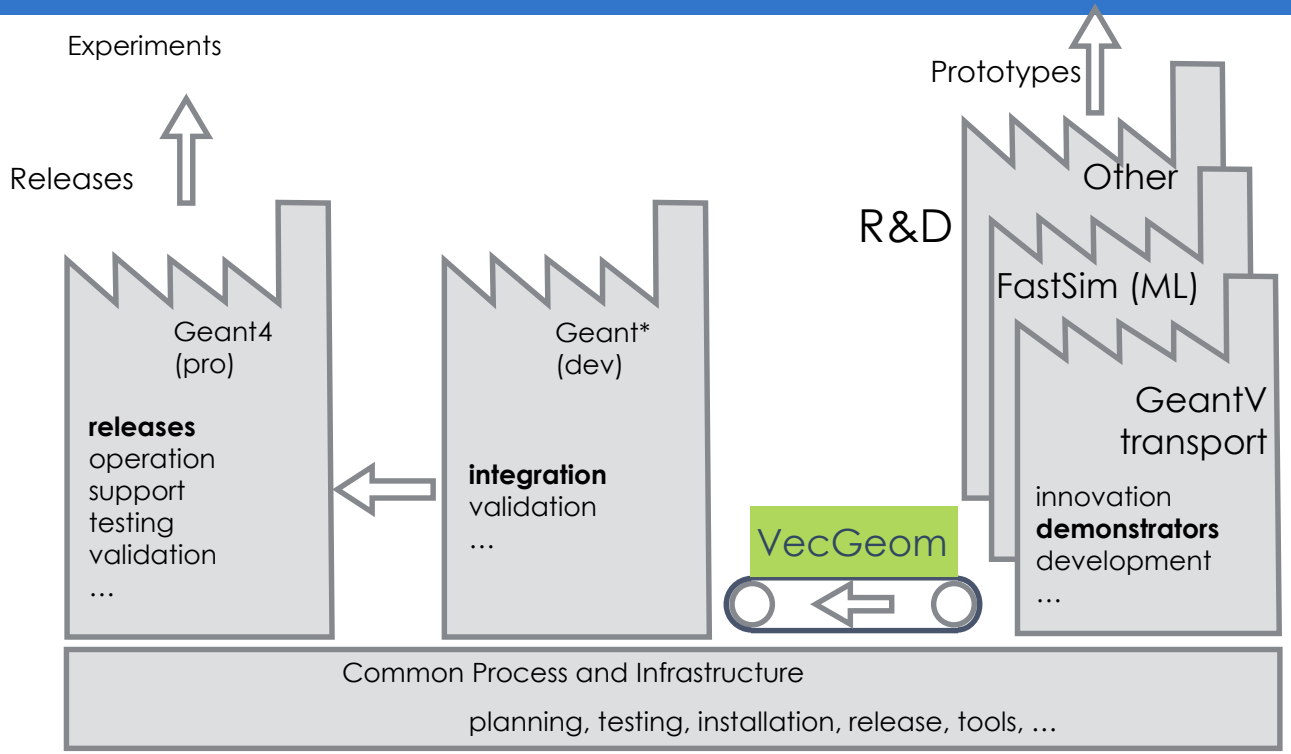
Outline

- motivation and goals
- status
- plans

GeantV motivation and goals

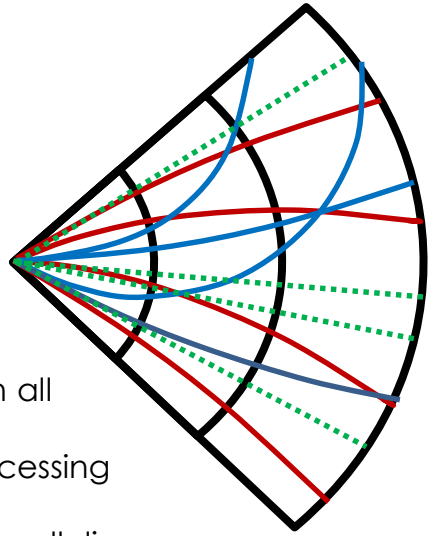
- future accelerators (HL-LHC, FCC) experiments will benefit with large speed-up in detector simulation (one of dominant CPU-time consumers)
  - some say they require 1-2 orders of magnitude speed-up in simulation
- GeantV R&D explores vectorized particle transport for next generation simulation toolkit
  - aiming at demonstrating the speed up of simulation using a novel approach to concurrent processing and data handling
    - allows to exploit vector operations on modern CPU
- goal for GeantV of speed-up x2-5
  - achieving higher speed-ups by embedding fast simulation

Simulation software R&D



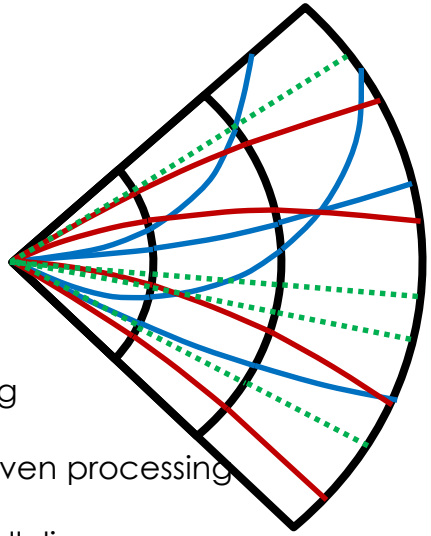
The GeantV idea

Classical simulation



- one track at a time through all stepping stages
- sequential stack-driven processing
- single event transport
- event-level embarrassing parallelism
- cache coherency – low
- vectorization potential – low (scalar auto-vectorization)

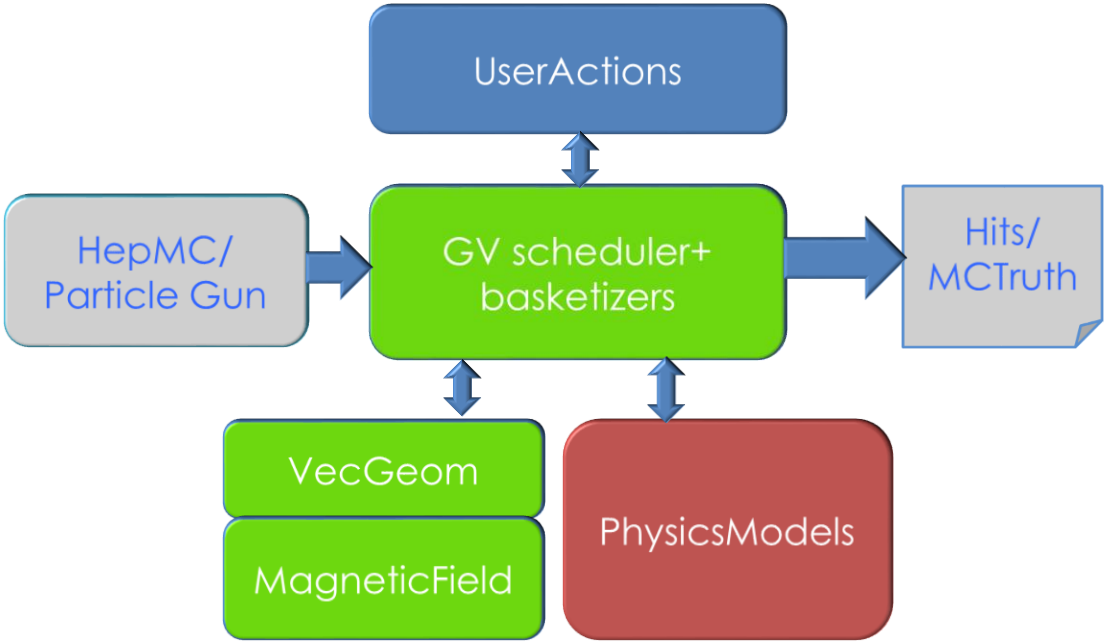
GeantV simulation



- groups of tracks executing together each stage
- non-sequential basket-driven processing
- multi event transport
- track-level fine-grain parallelism
- cache coherency – high
- vectorization potential – high (explicit multi-particle interfaces)

Current overall GeantV status

- alpha tag
  - complete version of scheduling, basketization
  - vectorized geometry
  - first vectorized magnetic field
  - full EM physics (scalar)
  - ‘user actions’
  - examples
    - realistic simulations
    - validation against Geant4 and data



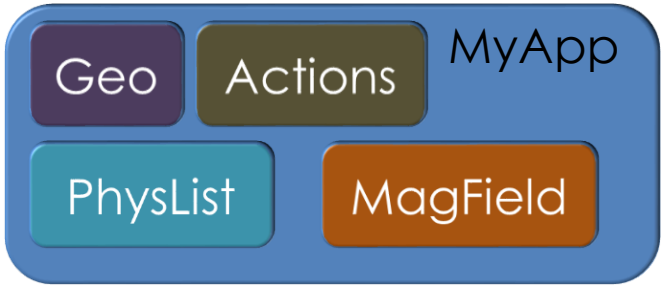
http://geant.cern.ch

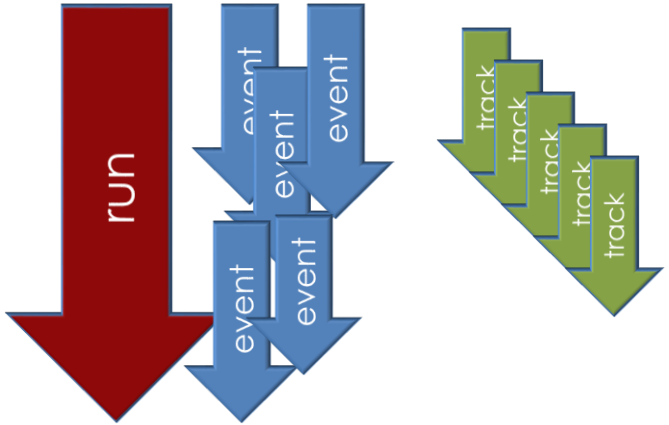
Scheduler status

- scheduler providing basket flow to all processing parts
  - physics models, magnetic field propagation, user actions
- user interfaces and helpers adapted for multi-threaded, multi-event and multi-particle handling
  - demonstrated by few examples
- possibility of interfacing with external frameworks and their event loops

User interaction status

- user application, detector construction, physics list, magnetic field description very similar to Geant4
- notifications during transport like for the other simulation packages
  - Geant4 style: Begin/End run, event, primary
- ‘user actions’ are more complex, due to:
  - concurrency: scoring data has to be handled per thread -> thread safety
  - event mixing: data has to be allocated separately per event slot!
    - GeantV provides services to deal with the extra complexity, with several examples





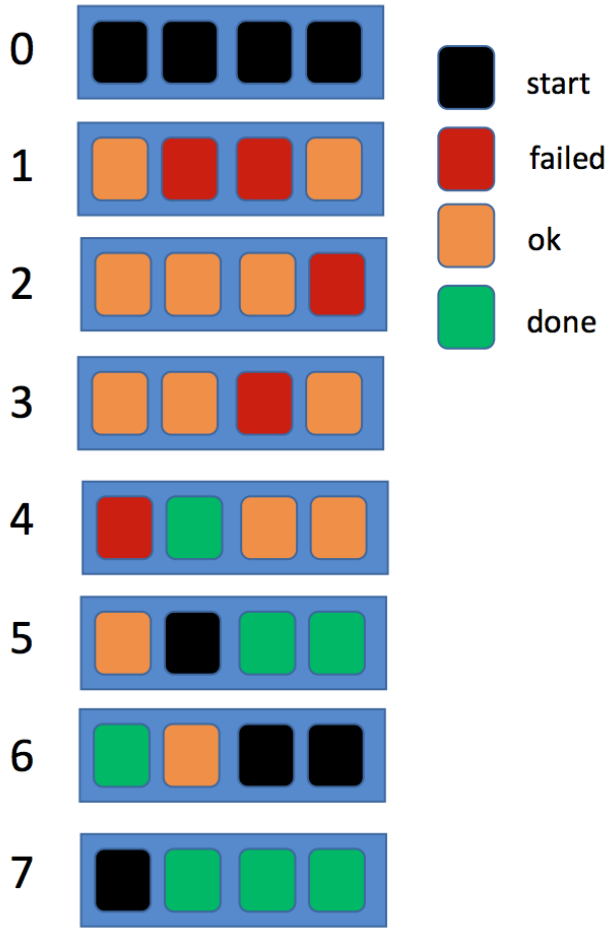
GeantV concurrency

- many events transported concurrently, tracks mixed among them
  - contract: limited number of event “slots”
- two modes supported:
  - standalone – configurable number of static threads managed by GeantV
  - external concurrency – transport behaving like a re-entrant service task transporting a number of events with concurrency orchestrated by external framework
- GeantV provides services to deal with data per event/per thread
  - task local data whiteboard, merging service, I/O service

Field propagation status

- field propagation involves solution of Ordinary Differential Equation
  - typically Runge-Kutta methods are used (as in Geant4)
- in GeantV created **vectorized Runge-Kutta** propagation
  - charged tracks in a basket are sent to the FieldPropagation classes
  - vectorized over tracks
- challenge to ensure that all vector lanes are working & use mostly vector operations

- Runge-Kutta step: estimate **x**, **p**, **Δx**, **Δp**
- successful if
  - $|\mathbf{\Delta x}| < \varepsilon * s$
  - $|\mathbf{\Delta p}| < \varepsilon * |\mathbf{p}|$
- different tracks (vector lanes) can take different number of iterations to finish integration



GeantV physics status

- EM showers can be fully simulated in scalar mode (models not vectorized) in single and multithreaded mode
  - general physics framework can handle continuous, discrete and at-rest processes are handled
    - all the EM models implement final state sampling with both alias sampling tables and rejection sampling
      - controlled by a flag at initialization, possible to set by region
  - every model tested and verified against the corresponding Geant4 model (cross section per atom, cross section per volume, and kinematic of primary and secondary particles)

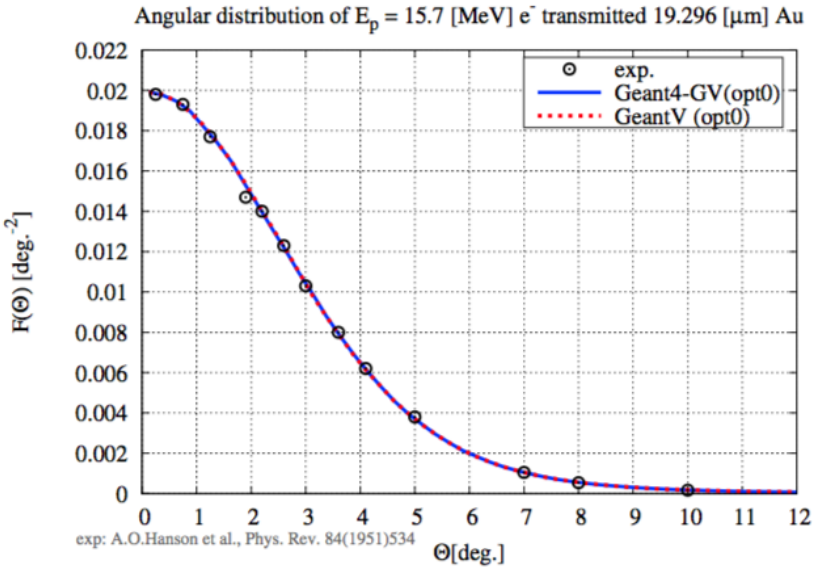
GeantV EM Physics

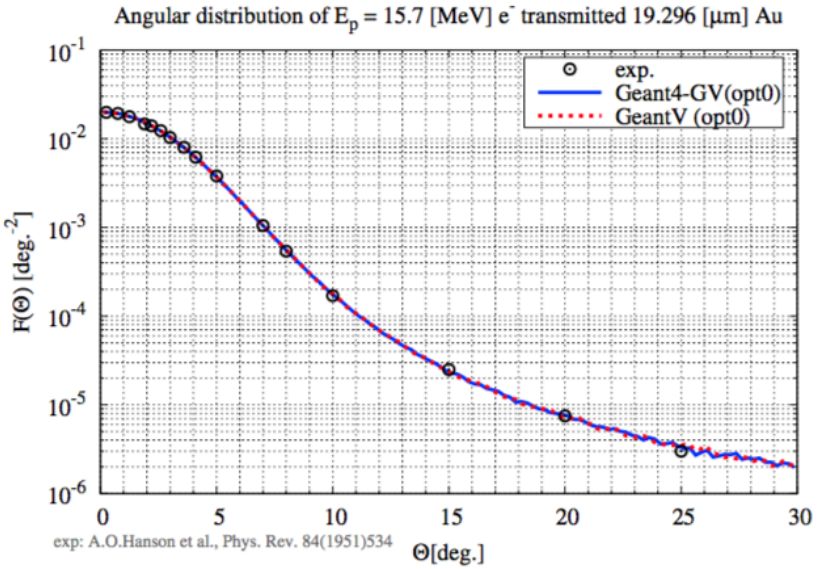
Current State

particle	processes	model(s)	
		GeantV	Geant4
e^-	ionisation	Møller [100eV-100TeV]	Møller [100eV-100TeV]
	bremsstrahlung	Seltzer-Berger [1keV-1GeV]	Seltzer-Berger [1keV-1GeV]
		Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]	Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]
	Coulomb sc.	GS MSC model [100eV-100TeV]	Urban MSC model [100eV-100MeV]
			Mixed model [100MeV-100TeV]
e^+	ionisation	Bhabha [100eV-100TeV]	Bhabha [100eV-100TeV]
	bremsstrahlung	Seltzer-Berger [1keV-1GeV]	Seltzer-Berger [1keV-1GeV]
		Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]	Tsai (Bethe-Heitler) w. LPM. [1GeV-100TeV]
	Coulomb sc.	GS MSC model [100eV-100TeV]	Urban MSC model [100eV-100MeV]
			Mixed model [100MeV-100TeV]
	annihilation	Heitler (2γ) [0-100TeV]	Heitler (2γ) [0-100TeV]
γ	photoelectric	Sauter-Gavrila + EPICS2014 [1eV-100TeV]	Sauter-Gavrila + EPICS2014 [1eV-100TeV]
	incoherent sc.	Klein-Nishina$^+$ [100eV-100TeV]	Klein-Nishina$^+$ [100eV-100TeV]
	e^-e^+pair production	Bethe-Heitler$^+$ [100eV-80GeV]	Bethe-Heitler$^+$ [100eV-80GeV]
		Bethe-Heitler$^+$ w. LPM [80GeV-100TeV]	Bethe-Heitler$^+$ w. LPM [80GeV-100TeV]
	coherent sc.	-	Livermore
+	energy loss fluct.	-	Urban

- all the models are developed in a **"vectorization-friendly"** way
- still plenty of **conditional branches** (requires use of Mask operation)

EM physics validation





Geant4/V TestEM5

Hits I/O and MCTruth status

- infrastructure for hits-handling implemented
  - example (FullLHCb) implemented simulating full LHCb detector and saving (concurrently) hits
- implemented necessary hooks for users to store particles history
  - simple example implemented based on HepMC3 event record

GeantV examples

- main GeantV applications
  - TestEm5: simulation of particle transmission through a simple slab
  - TestEm3: general simplified sampling calorimeter simulation to study EM shower simulation
    - optional MCtruth handling
  - fullCMS: general simulation
  - fullLHCb: general simulation + hits handling
  - cmsToyGV: example of interfacing of GeantV with a simplified multi-threading version of CMSSW
    - see next talk by Kevin on interfacing GeantV with real CMSSW

GeantV plans for 2018

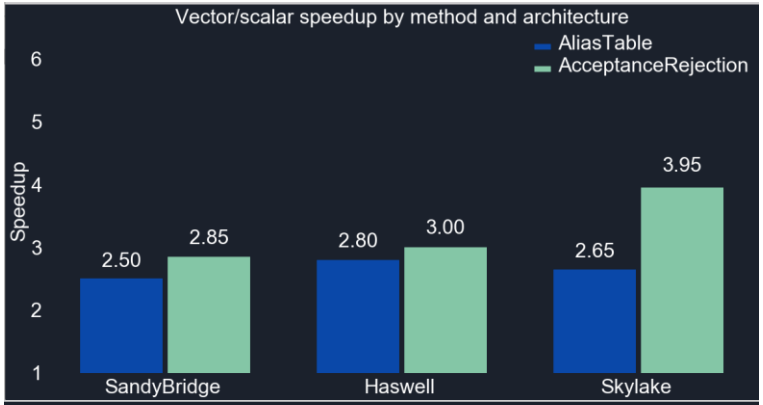
- beta tag to demonstrate achievable speed-up in realistic conditions
- target: full EM shower transport vectorized, with close to design performance
  - vectorization of EM physics models, magnetic field propagation and geometry – main activity
  - performance tuning for the core scheduler
  - performance tuning for magnetic field
- more realistic examples also including MC truth processing, I/O and fast simulation integration
- remaining geometry development
  - more solids using internal vectorization

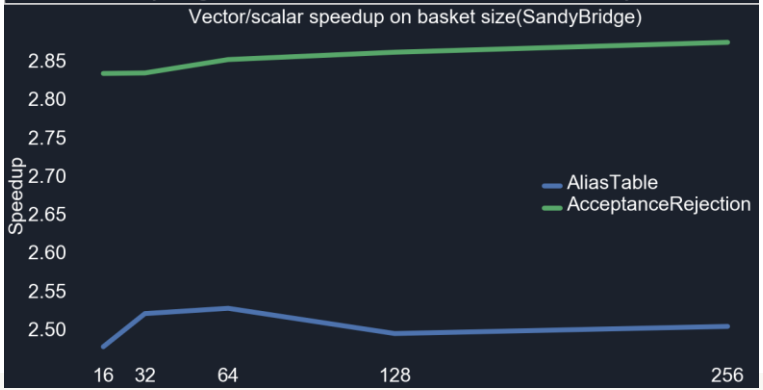
Scheduler plans

- further optimizing GeantV scheduler for concurrent vectorized flow
  - workload balancing, tuning parameters related to basketization, data flow for getting VecCore types to physics models
- error handling mechanisms for handling track/event/worker loss
- specialized geometry navigators to remove part of the scalar operations preventing efficient geometry vectorization in multi-particle mode
- event reproducibility studies
  - RNG development

Physics vectorization work in progress

- implementation and testing of the **Filtering/Basketizing** mechanism for physics
  - handlers per physics processes/models
  - **'LightTrack'** in **SOA** format **-** customised per "model needs"
- **Sauter-Gavrila Photoelectric** and **Klein-Nishina Compton** are the first models tackled
- vectorization of the main functions/blocks of code used in the physics library:
  - alias/rejection sampling
  - RotationToLabFrame
  - generation of secondaries



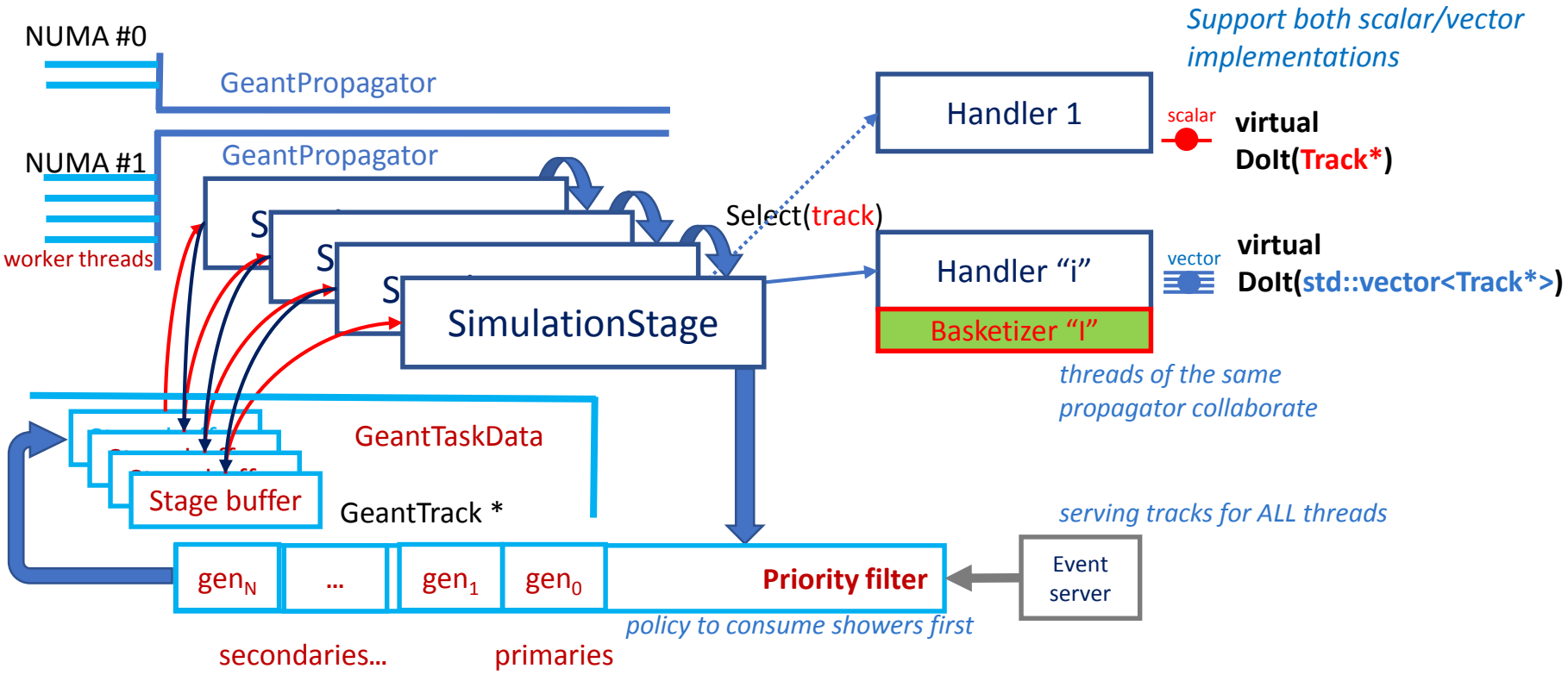


Summary

- GeantV R&D aims at demonstrating benefits of vectorized particle transport
  - valuable components already delivered to the community via Geant4/ROOT: VecGeom/VecCore, improved photoelectric model
  - aiming at complete EM shower simulation in a vector flow
- status
  - Alpha tag: full EM transport, vectorized geometry/magnetic field, scalar physics, user interfaces, examples of different complexity
- plan for 2018:
  - demonstrate the achievable speed-up in realistic application
  - Beta tag: EM physics models fully vectorized, more examples, more feedback from community

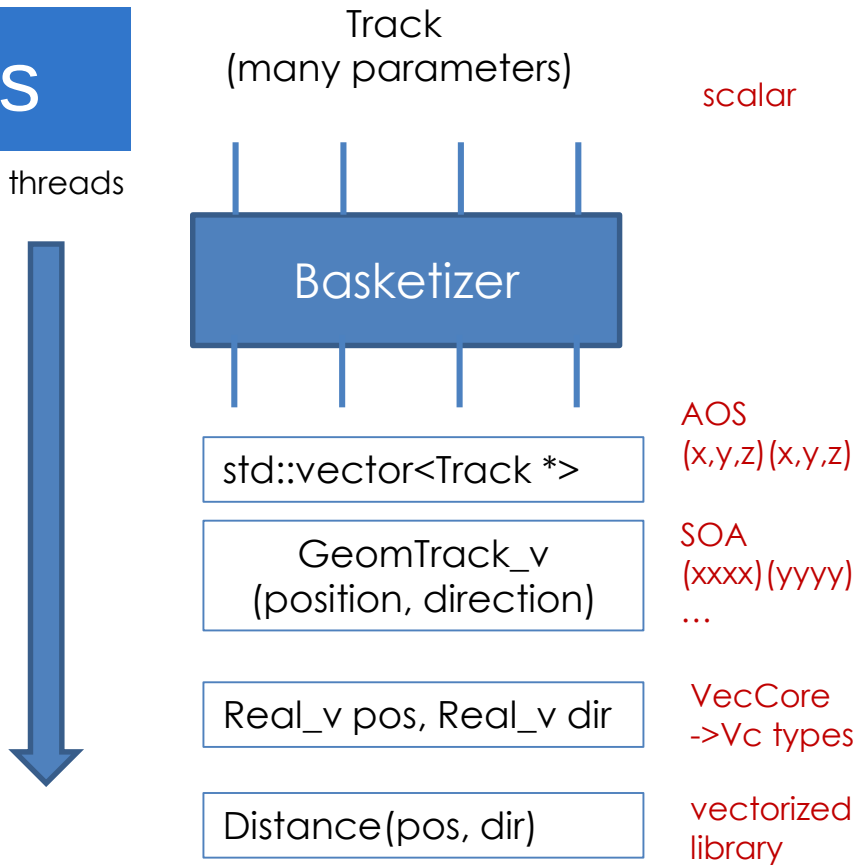
Backup

A generic vector flow approach



Working with vectors

- filter tracks in a scalar flow for a given stage
- basketize concurrently -> vector of tracks
- gather relevant track parameters for the algorithm to execute in SOA form
- cast into native VecCore types
- feed to allgorithms vectorized with VecCore
- added complexity needed to take advantage of SIMD vectorization



Physics vectorization plans

- perform **deep profiling** of the methods to highlight major (sometimes unexpected) **hotspots**
  - indexing of cross-sections/lambda table**: log2 indexing** with *frexp* to extract at “zero-cost" the mantissa and the exponent
  - sampling of the target element index (pe effect) -> **machine learning solution**

- **re-think** and **re-design** some data structures and algorithms
  - to improve **cache locality** and **minimise** the need for **Gather** and **Scatter** operations (mainly scalar)
    - **Alias Table stored as AOS** instead of SOA with some precomputed values

- to reduce the **conditional branches** and **unroll the loops** that kill vectorization
  - **ShellSampling** in photoelectric effect:
    - one single denser dataset based on epics2014 with **indexed** linear Interpolation **(no binary search) -> applicable to other models to avoid log-log interpolations**
      - AliasSampling to sample the sub-shell from a discrete function