

# GeantV scheduler – status and plans

A.Gheata

21 Jan 2014 Fermilab

# Outlook

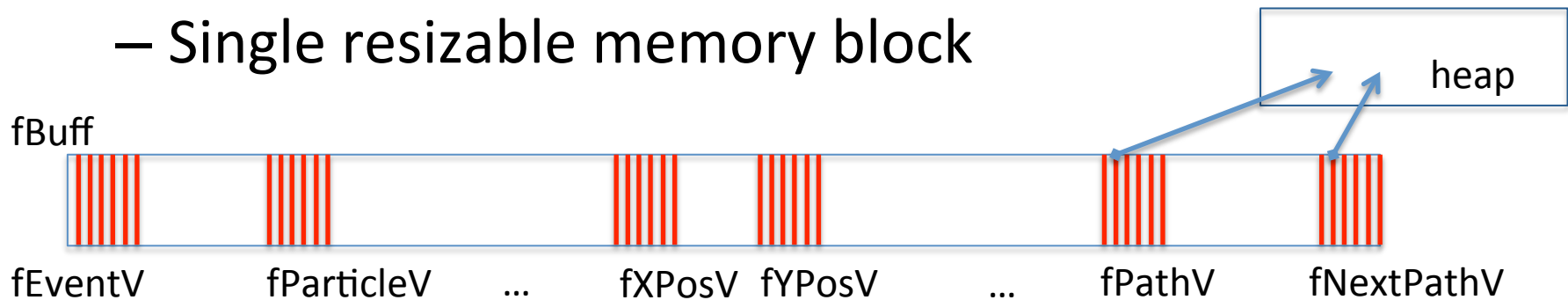
- Event structure and containers
- Baskets and data queues
- Basket managers (per LV)
- Transport (physics and geometry) and track phases
- Scheduler class and scheduling thread
- Scheduling policies and multithreading
  - Connection to vector geometry
  - Connection to physics
  - Connection to GPU prototype
- Monitoring

# GeantTrack

- Track identifiers: event, slot (memory management), track ID, PDG, code
- Particle identifiers: PDG, GeantV code, charge, mass, **species**
- Kinematics: position, direction, momentum, energy
- Status: status, N steps, **N null steps**, boundary flag, **pending flag**
- Geometry/physics context: process, proposed step, current step, distance to boundary, safety, current path, next path
- $\text{sizeof}(\text{GeantTrack}) = 192 \text{ bytes} + 2 * \text{sizeof}(\text{TGeoBranchArray}) = 192 + 2 * 48 + \text{depth} * 4 + 16 = \mathbf{344 \text{ bytes}}$  in average
- **Can this be reduced? Size influences memory requirements AND CPU overhead for reshuffling operations in vector mode.**

# Track vectorizable container

- Track data format not used directly by the transport – only used to import tracks from generators/processes
  - Track data imported into GeantTrack\_v
- SOA matching GeantTrack using internal memory management for vector performance
  - Single resizable memory block



# GeantTrack\_v

- Buffer management: allocate, copy, resize
- Import tracks from GeantTrack or GeantTrack\_v
  - And track removal
- Management of holes (i.e. tracks that finished transport in the current propagation cycle)
  - Vector not efficient-> compact tracks
    - Hole finding algorithm based on TBits + memory copy overhead
- Sorting by track status, needed to vectorize different propagation stages
  - E.g. transport of neutral tracks

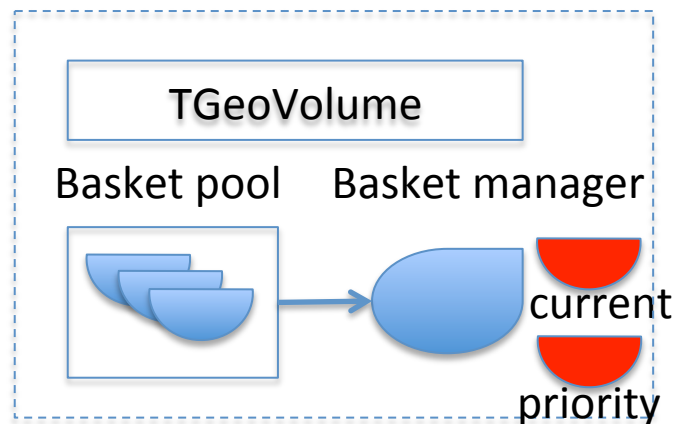
# Baskets of tracks

- Unit of work for the transport thread
  - Containing only tracks in a single logical volume
- Current implementation: basket = input and output GeantTrack\_v
  - Provision for hits and digits to be added
  - Could be handled in a single container, but overhead bigger

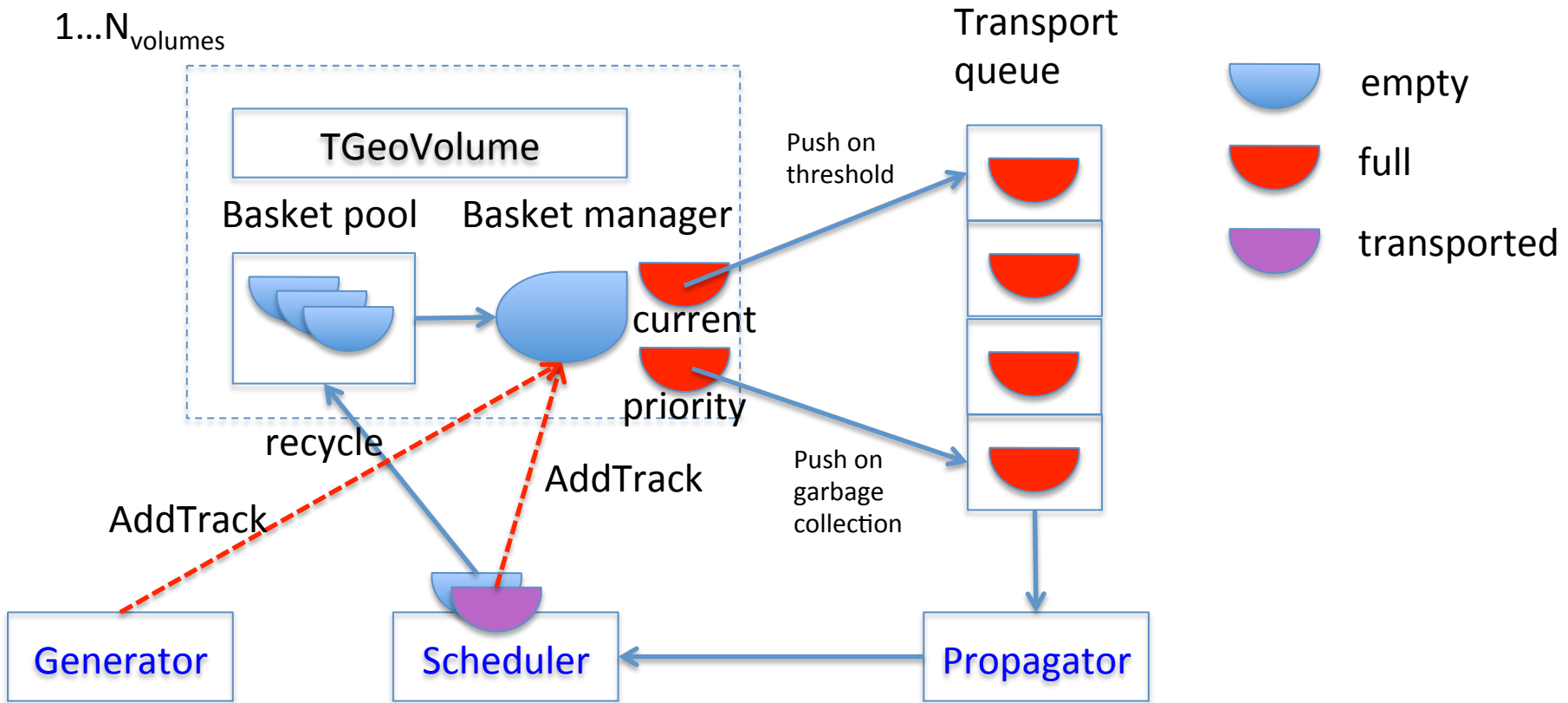
# Basket managers

- One basket manager per volume
  - Receiving tracks entering the volume from generator or scheduler
  - Accessed by scheduler only
- Pool of empty baskets, one current basket + one basket for prioritized tracks
- Lock-free access for unique scheduler (only one thread can add tracks)
- Transportability threshold per manager
  - If threshold reached when adding tracks, the current basket is pushed in the work queue and replaced from the pool. Tracks added with the priority flag go to the priority basket which gets pushed to the priority side of the queue
  - $\text{Threshold}(\text{vol}) = \text{Ntracks\_in\_flight}(\text{vol}) / 2N\_threads$  rounded to %4 (min 4, max 256)

1...N<sub>volumes</sub>

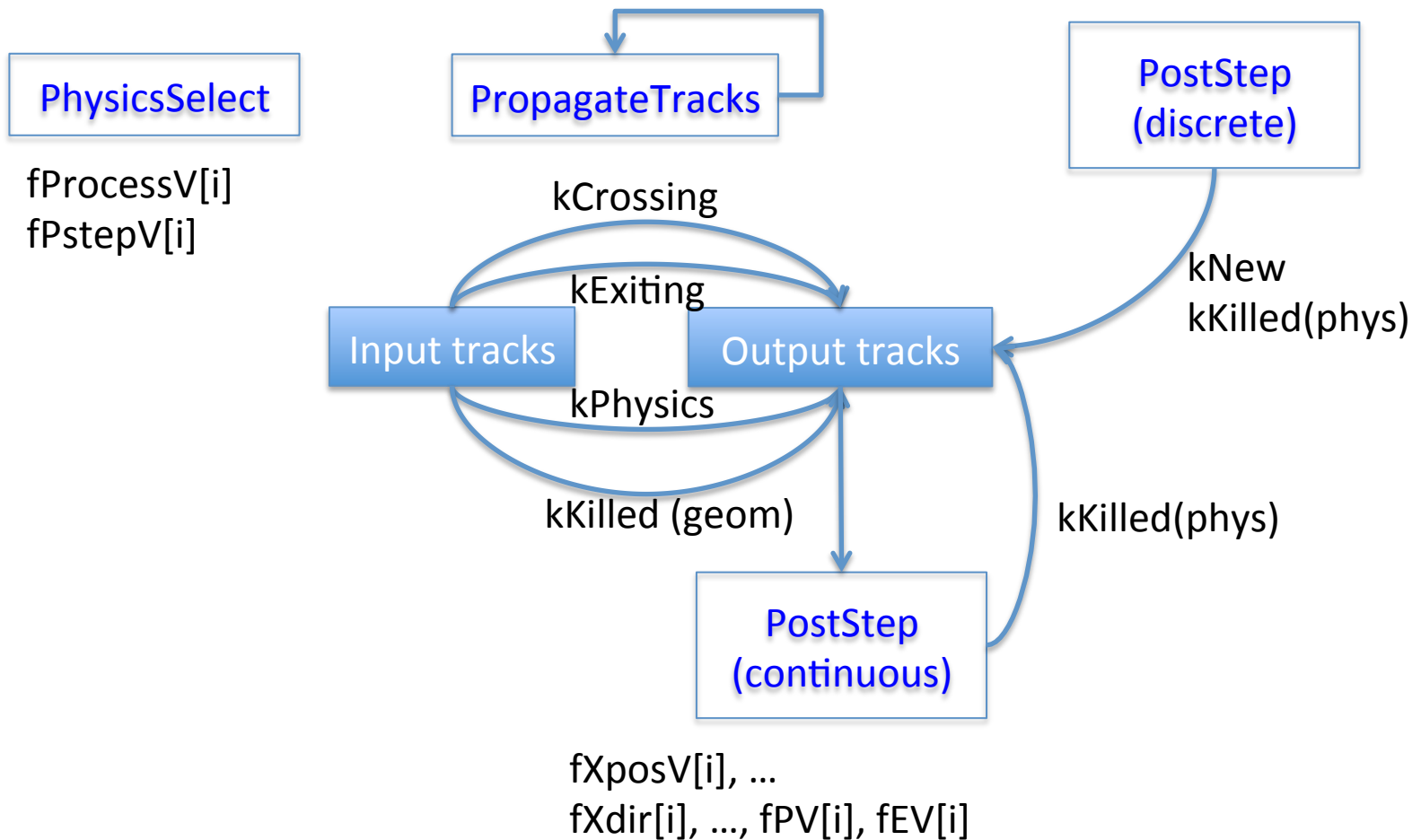


# Basket lifecycle

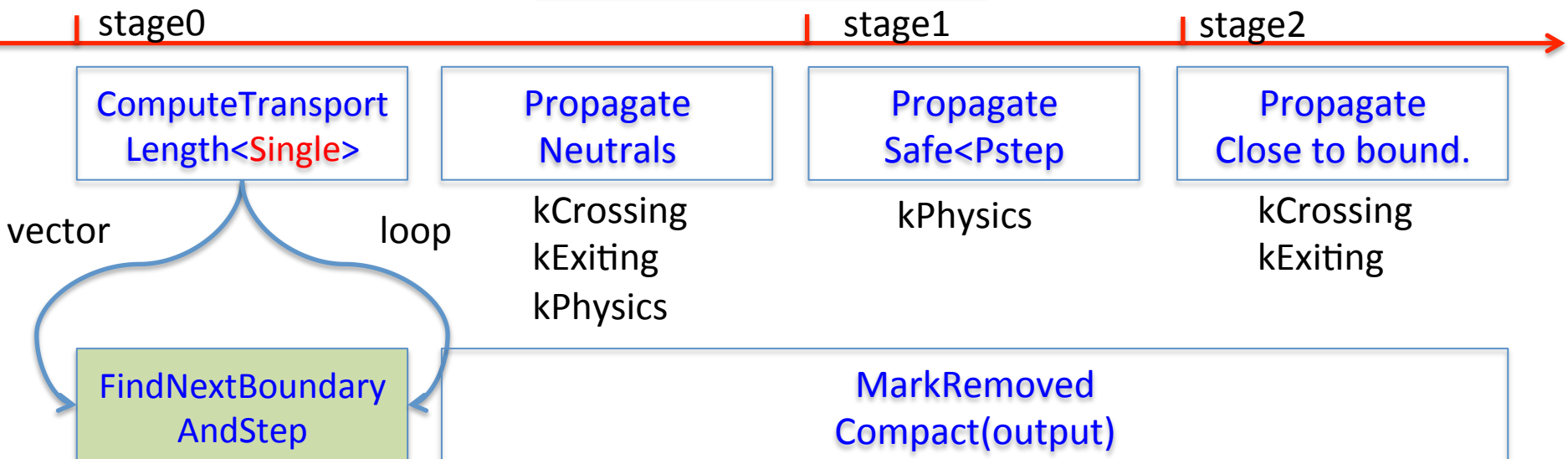




# Basket transport



# PropagateTracks

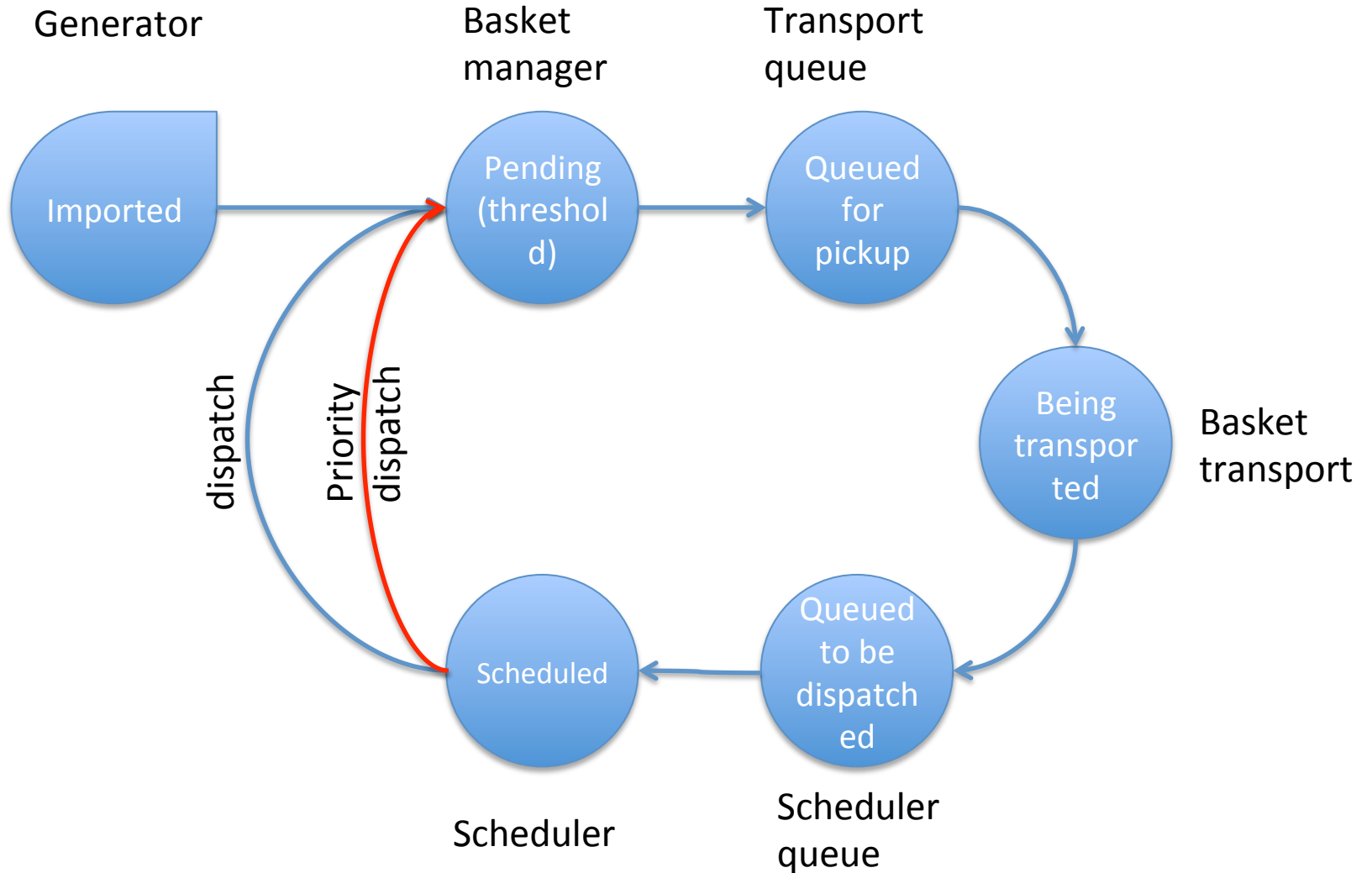


fSnextV[i], fSafetyV[i]

- kVector – continue in vector mode
- kSingle – call PropagateTracksSingle at the given stage
- kPostpone – copy remaining tracks to output
- MarkRemoved + Compact – compact holes and copy these tracks to the output

Propagate with safety

# Track stages



# Scheduler

- Pulls transported baskets, dispatches tracks to basket managers per volume
- Single thread, one scheduler/ multiple thread, multiple schedulers
  - TBB task approach, to be investigated after understanding and tuning the scheduler with real physics
- Applying policies to:
  - Provide work balancing
  - Keep memory under control
  - Keep the vectors up (most of the time)

# Scheduling policies

- Workload balancing
  - Divide the work evenly to scale with number of workers
  - Queue control: garbage collection on work queue depletion
  - **Improvement: schedule physics as separate task (process selection and discrete processes post-step)**
- Memory management
  - Not active currently, the idea is to trigger hit/digits collection and memory cleanup on thresholds
- Keep large vectors
  - Raise transportability thresholds per volume
  - Postpone sparse tracks when not in garbage collection mode
- Trigger single track mode when vectorization gives just overhead

# Connection to physics & geometry

- Currently trivial approach to physics, have to interface to the new physics code
  - Process selection based on total xsec
  - Redo process interface for actions (along and post-step)
- Connect to vectorized navigator
  - Even limited to simple setups, we need to understand gains and overheads + tuning
- Connect scheduler to GPU transport
  - Using a manager thread to take and transport baskets from the main CPU queue
  - We have to understand if there are extra requirements
  - Can be done for both geometry and physics baskets

# Monitoring

- Internals of track dynamics
  - Track counters in different phases, efficiency to prioritize events
- Basket dynamics
  - Number of baskets, size per volume, transportability threshold
  - Vector size
- Memory monitoring
- Multithreading efficiency
  - Locks and waits analysis, concurrency
- New class GeantTrackStat
  - Used if GEANT\_DEBUG=1
  - Track counters for number of tracks/steps per event, read in the different track phases
- Separate monitoring thread with graphics to be done