R&D IN GEANT4

Naples HSF meeting
March 2018

Marc Verderi
LLR, Ecole polytechnique
For the Geant4 Collaboration

Layout

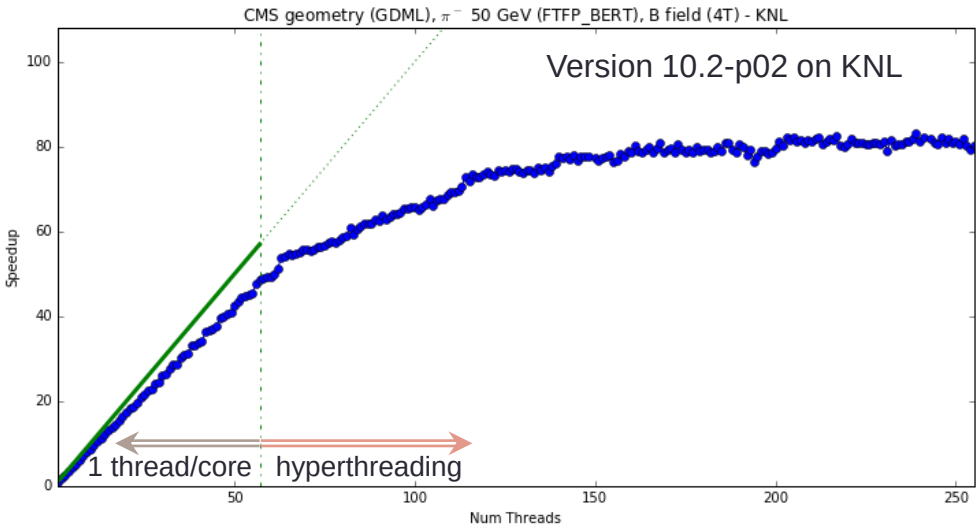
I. Medium Term Evolutions, Some Highlights
II. Longer Term Evolutions
III. R&D external to Geant4

Medium Term Evolutions

Some Highlights

Sub-Event Level Parallelism

- Geant4 MT initially designed to process events in parallel
  - With independent processing per thread of series of events
    - Independence of threads makes next to perfect scaling of throughput with #threads
  - But scheme assumes events are small enough to fit into the memory of one thread



- Sub-event parallelism generalizes the approach:
  - To serve the case of applications requesting large memory per event:
    - e.g. : ALICE, HL-LHC, air shower
  - One event is split into “sub-events”
    - e.g. : each primary track = a sub-event
  - Each sub-event is sent to a thread, and merged back to the original full event later
    - Geant4 will provide tools to easily enable this feature
    - ATLAS already has this mechanism within ATHENA

Ions, muonic atoms, radicals, hyper-nuclear…

- Most common particles –$e^{\pm}, \mu^{\pm}, p, n, \alpha, \pi^{\pm,0}$ ...– are each represented by a dedicated class
- Ions ($> \alpha$) have a common treatment through one “generic ion” class
  - Allows to treat more than 7000 ions species
  - With a single and common instantiation of the related physics
- But ions physics vectors accessed through a special case during tracking
  - i.e. an “if” statement

- Recently, for Intensity Frontier, muonic atoms were introduced:
  - i.e. atoms in which an e^- is replaced by a μ^-
  - Mechanism for ions is duplicated, and related code is protected by an “#ifdef”…
- And further:
  - Radicals, e.g. OH^-, are also handled at low energy in the DNA module
  - Hyper-nuclear physics is planned:
    - Hyper-nucleus = nucleus in which an hyperon stands in place of a proton or neutron
    - Starting with hyperon = lambda, and nucleus transported (today it is decayed immediately)

- Inflation of particle species ($\gg$ 10k) in at least four different families
  - Each being a special case

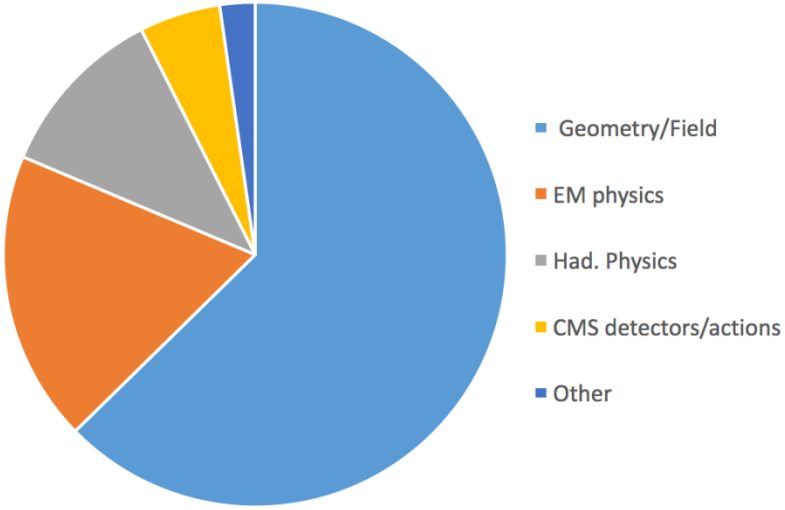
- Plan to design an uniform treatment of these “non-most-common” particle types

Production Thresholds (aka “cuts”)

- Geant4 has a production threshold mechanism at the tracking level:
  - controls the conformance of secondary particles returned by processes wrt to the defined thresholds
  - Generic mechanism applied to all processes
- Mechanism will be re-examined:
  - In practice, only a few processes must have such thresholds defined:
    - Bremsstrahlung and ionization: as subject to diverging or heavy productions
  - For other processes :
    - more a matter of convenience / taste, not fundamental
- Keep mechanism at tracking level or move down responsibility to processes ?
- Expected benefits from moving to processes:
  - simplification of tracking classes
    - As offloaded of the control of secondary particle production
  - better performance
  - clearer responsibilities on cuts appliance
    - today shared between tracking and processes

Refactoring Transportation [1/2]

- The “transportation” is a Geant4 process
- It manages the navigation in the geometry:
  - It cares about volume boundaries
  - It takes into account the fields in the propagation of particles sensitive to such fields
- Currently, only one transportation exists:
  - It deals with all particle types:
    - neutral and charged ones,
    - optical photons,
    - phonons, etc.
  - Results in frequent “if” branches
    - on the charge to decide to apply field computation or not,
    - to use group velocity or not
    - …



Sources of CPU consumption Geant4 CMS simulation
Courtesy of Vladimir Ivanchenko

Refactoring Transportation [2/2]

- Idea is to provide at least two flavors of transportation:
  - One for charged particles
  - One for neutral particles
  - Eventually one also for optical photons
    - As velocity calculations differ from other particles
- Case of other fields -e.g. gravitation- could be treated with more transportation flavors

- Further extensions/specializations to be also considered:
  - VecGeom navigation: optimized implementation with modern C++
  - À la DagMC: direct and efficient navigation in CAD geometries
  - DNA navigation: better serve the case of radicals

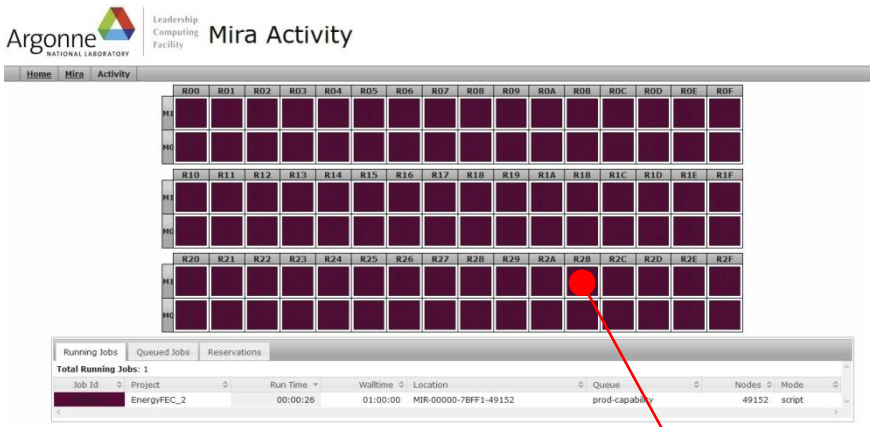
- Revision extended to “Coupled Transportation”:
  - Transportation process dealing with several parallel geometries simultaneously
  - Has many use-cases: e.g. layered-mass geometry: allows to switch between several representation of a same detector, depending on particle

Use of HPC

- **Please, see Andrea's presentation this afternoon.**

Geant4 ran on Mira Bluegene/Q Supercomputer @ ANL
with all of its 3 million threads, in a single application

- Setup:
  - Combination of MPI and Multi-threading
  - Full-CMS geometry & field
- Good linearity observed
  - Even if some issues on data reduction (collection) exist, to be tackled in 2018
- Why targeting detector simulation here
  - Processing can start quickly (no data access, few seeds to set,…)
  - Allows for an opportunistic computing strategy:
    - Exploit remaining time left by heavy consumers.
    - Which still represents a lot of computing power.



Single color, full squares =
one application occupies Mira all

This strategy would allow to cover about
70% of the ATLAS simulation needs !

Physics & Code Evolution

- Extension of physics coverage both to higher and lower energies:
  - High EM and hadronic physics, for HL-LHC and FCC
    - See Farah's presentation this afternoon
  - Very low energies with electron-hole and phonon transport
- Extension to crystalline structure on the way
  - Use-case of beam extraction with bent crystals, based on channeling
- Further and easier-to-use event biasing schemes

- Revision and simplification of the physics processes framework
  - Generalize "process = { set of models }" at base classes level
  - Simplification of the tracking logic wrt the various ways processes can act
  - Refactoring of INCL, reengineering and easier use of LEND, etc.

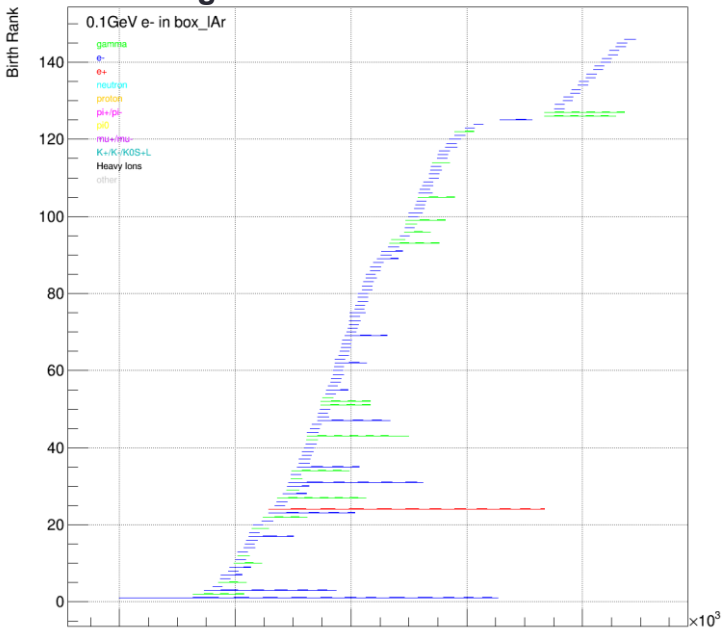
- Following C++ language evolutions:
  - C++11 standard adopted & evolution towards (C++14) C++17
    - with continuous transformation of past code
  - Full adoption of C++11 threading models and workspace/tasks-based parallelism
    - will allow Geant4 to transparently support MT on Windows platforms
    - better compatibility with external tasks-based frameworks

Longer Term Evolutions

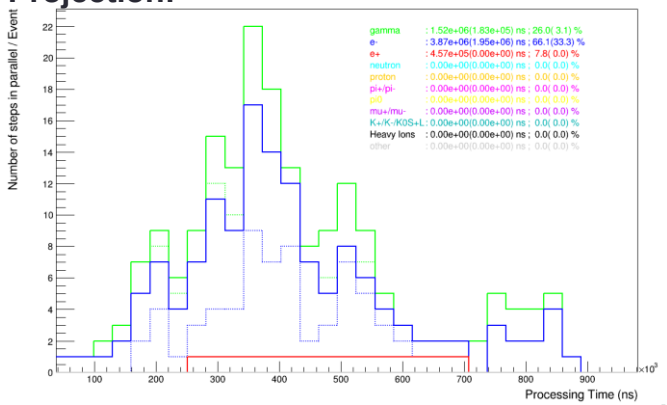
Track-level Parallelism

- Geant-V prototyping a basketized transport
  - 2 – 5 / 3 – 5+ speed-up advertised
- VecGeom as option since Geant4 10.2
- Interest in generic problem of parallel track transport
  - Tracks physically independent → track parallelism introduces a technical coupling between them
  - At what cost ?
    - In term of waiting time in memory, number of tracks in memory, variance on this number, etc.
  - What scalability with number of cores, vector register length, memory space, memory traffic speed,… ?
  - What articulation with user's code ?
  - What asymptotic performances expected ?
  - What intrinsic limitations ?
  - Unicity or not of the algorithm ? What FOM if several ?
- Emulation of problem with regular Geant4 may help to answer some of these questions

"Chronogram" on an "∞-core" machine:

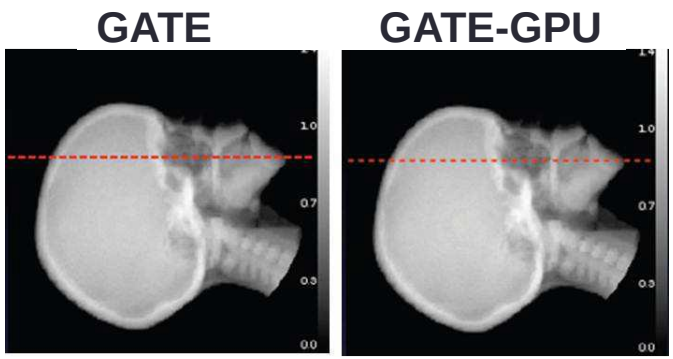


Projection:



GPGPU & Hybrid Computing, and more…

- GPGPU:
  - Some Geant4 members participate to simulation on GPU
  - GPU can be excellent in addressing specific use-cases:
    - Radiotherapy, imaging
    - Optical photon transport
    - Thermal neutron transport
    - Radical transport in microdosimetry
  - Impressive boosts O(100) even O(1000) obtained
  - But only used with limited physics & limited geometry
    - Eg: radiotherapy : E < 1 GeV & only boxes
    - Optical photon : few surface and absorption processes
  - General case of HEP does not fit here.



Head CT scan simulation
Courtesy of OpenGate Collaboration

- Hybrid Computing:
  - Can more be offered ?
  - By offloading the CPU from some heavy computation ?
    With fast CPU ↔ GPU communications ?
  - Issue of future HPC: hybrid computing might be the choice of HPC
- More:
  - Machine learning:
    - No Geant4 members involved for now, but interest in what could be obtained
  - Quantum Computing:
    - Sometimes mentioned….

R&D External to Geant4

Donation, Collaboration, Third-party…

- Several R&D's external to Geant4 are of interest to the simulation community:
  - Developments of physics models, or data libraries of cross-sections, etc.
  - Interfaces to CAD systems
  - Geant-V prototype / parallel track transport
  - Machine Learning
  - Etc.

- Up to now, we handled external R&D's either by:
  - Code donation, in simple cases:
    - This appeared for several classes, e.g. some solid shapes, GUI functionalities
    - Sustainability concerns though, to validate/maintain/improve
  - Joining the Geant4 Collaboration, for more complex cases:
    - Particle_HP, INCL++, … most recent case of accurate polarized pair-production model (5D sampling)
    - Joining the Collaboration ensures the needed level of communication wrt complexity
    - And import the needed competence, ensures long term maintenance and support

- We also promote third-party extensions:
  - Sizeable developments or tools the authors distribute by themselves, for example:
    - CADMesh : A CAD file interface for GEANT4
      - https://github.com/topics/cadmesh
    - DAGMC: Direct Accelerated Geometry Monte Carlo
      - https://svalinn.github.io/DAGMC/#
  - Geant4 Collaboration will enhance promotion of these through its web pages and other means

Going further ? [1/2]

- Geant4 will be moving to Git for its code management this year
  - Note that Github is already used to distribute public releases (“mirror mode”)
  - With pull requests possible on this site
    - Even though not integrated in the development tree
- Could we using Git for external pull requests on the development tree ?
- Some issues, not exhaustive:
  - Geant4 Collaboration differs from HEP experiment ones:
    - {EXPERIMENT software users} $\subset$ {EXPERIMENT collaborators}
    - {Geant4 software users} $\gg$ {Geant4 collaborators}
      - Higher risk for contradictory/diverging requests
        - Eg : a meaningful pull request from a medical user may not fit HEP case
      - Higher costs for evaluating requests and resolve potential conflicts
  - Ongoing developments can not be made public:
    - They are work in progress, not to be used
    - They must benefit of the regular validation done by the Collaboration
    - Last but not least : developments are subject to publications by developers themselves
  - Once reaching some level of development, we go for physics validation
    - An heavy validation, to be examined carefully
    - As part of the mandate of the Collaboration

Going further ? [2/2]

- Testing & validation evolve with new use-cases
  - A new development must come in general with an additional testing
  - Can not define an *a priori* “one fits all” testing & validation suite
- Effectiveness of the tool is determined by the people running it:
  - They may not have the appropriate skill to analyze a request or setup the testing
  - Hence the need for at least close communication with requesters and likely their involvement in Geant4 to addressing complex requests
    - no “drop-and-go” possible in complex cases

- ***Human interaction is by far the most important aspect***
  - Have to see and understand how such tools can help in the case of Geant4
  - How they can improve the “let’s work together”

- We propose to further discuss these at Geant4 Technical Fora
  - Discuss the pull requests that may come
    - As an other way we respond to requests
  - Further discuss the process
  - **Remember : these Technical Fora are joint users – Geant4 public Fora**

Conclusion

- Geant4 is a mission critical tool for many applications, including HEP ones
- We must:
  - Improve and extend it
  - While keeping user's interfaces as stable as possible
- Many activities going on to improve the physics and performances, rationalize design aspects, extend the capabilities, …
  - Introducing sub-event level parallelism, handling of ions, muonic atoms, radicals, hyper-nuclear, revising the production threshold mechanism, refactoring of transportation, …
- Longer terms activities are investigated
  - Parallel track transport,
  - GPGPU and hybrid platforms,
  - …
- Geant4 welcome external contributions
  - Either "easy ones", handled by code donation,
  - More complex ones, handled by joining the Collaboration, or
  - Third-party contributions