



Keynote Lecture:

Recent Geant4 Developments

IV Geant4 International User Conference at the physics-medicine-biology frontier
24th - 26th October 2022,
Napoli,
Marc Verderi (LLR)

Overview



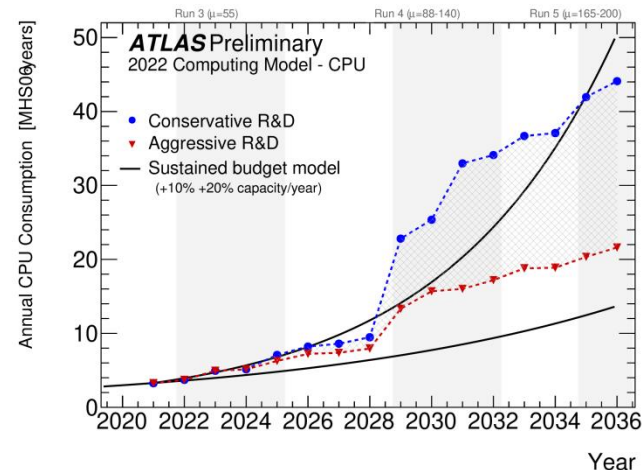
- The LHC Context & Medical
- Tasking : A Fruitful Parallelism Scheme
- Few Selected Topics
- Collaboration Evolution

The LHC Context & Medical

Context wrt HL-LHC



- The Geant4 Collaboration has to face the serious challenge of trying to respond **the strong HL-LHC demand** (High Luminosity phase of LHC, starting 2026)
 - Involves quite a large fraction of human resources
 - Solutions can drive broad evolution of the Geant4 software
- Issue of HL-LHC:
 - Under pressure HL-LHC computing budget
 - Simulation being O(50 %) of it (depending on experiment) !
 - Simulation is asked to be
 - **10 times faster !**
 - Simulation throughput, all types included (full & fast sim.)
 - **3 times better !**
 - That is $\sqrt{10}$; **to not increase contribution to systematic errors**
- Approach is:
 - **Detailed simulation as much as possible**
 - **On CPU:** with **adiabatic evolution (CPU & phys.) & smartness** (eg Woodcock tracking)
 - **On GPU:** **provided current R&Ds succeed & an efficient workflow is established**
 - On other future hardware ?
 - **Fast simulation as much as needed** Few % gain saves M€ !
 - Classical fast simulation or **Machine Learning based**



▪ These efforts, motivated by HEP domain, are of interest **to many medical users !**

Tasking : A Fruitful Parallelism Scheme

Tasking



- **Tasking is the default parallelism scheme since major release 11.0** (December 2021)
 - It is a re-thinking of the previous multi-threading scheme, but more flexible
 - It is based on based on PTL (Parallel Tasking Library) v2.0.0, distributed with Geant4
 - Migration is easy :
 - **auto runManager = G4RunManagerFactory::CreateRunManager();** Introduced in 11.0
 - Will create the proper run manager, depending on your build system, selecting the G4TaskRunManager if present

- **Main idea:**
 - **Makes tasks logically independent of threads** (hardware threads or hyper-threading ones)
 - Program becomes then a set of tasks to be served
 - Tasks can be linked to guarantee a proper ordering of execution

- **Advantages:**
 - Tasks can be served by threads, as before
 - Load balancing is made easier
 - Tasks may also be served by other backend !
 - eg GPU !
 - Offloading the CPU from specialized work that can be efficiently served by a GPU
 - Makes easier the road to hybrid-computing !
 - Functionality not yet exploited, but expected in not distant future !
 - Ease adoption of “sub-event parallelism”
 - That is the splitting of a “big event” into several smaller ones, executed in parallel on several threads
 - And maybe other !

- **Certainly a functionality that would be worth thinking in the context of medical applications !**
 - Eg : simulation of large imaging system, or CPU-GPU combination in treatment planning, etc. ?

Few Selected Topics

Keynotes on Geant4 Software

- Keynote: Update on the G4-Med project and on the Geant4 Advanced Examples for medical applications
 - Susanna Guatelli
- Keynote: Hadronic aspects in Geant4
 - José Manuel Quesada
- Keynote: Overview of the Geant4-DNA project
 - Sébastien Incerti
- Keynote: EM aspects in Geant4
 - Vladimir Ivantchenko

G4[particle]GeneralProcess

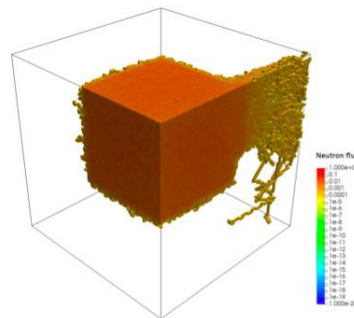
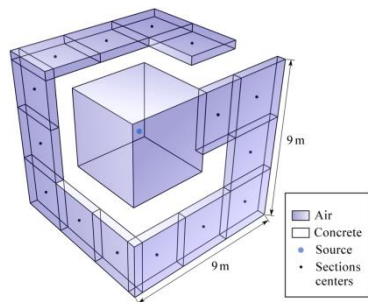


- Attempt to reduce the “generic communication” between the tracking and individual processes by **grouping some processes into “general processes”**
 - Applied to **G4GammaGeneralProcess** and recently to **G4NeutronGeneralProcess**
 - G4GammaGeneralProcess = {Compton, Conversion, Gamma Nuclear, Photo-electric, Rayleigh}
 - Already present and exercised since several releases
- **Effects:**
 - Grouping processes indeed leads to O(5%) improvement
 - not negligible !
 - The consequence is also that **individual processes do not appear anymore !**
 - So, **have to be aware of possible side effect if talking directly to processes !**
- **Woodcock tracking for gamma:**
 - Grouping processes has a second advantage : the **easy access to the total cross-section**
 - For G4GammaGeneralProcess this is used to include the Woodcock tracking option.
 - Method useful (and invented) in the medical domain !
 - To track faster inside phantoms, skipping many volume boundaries at once
 - With an implementation by GATE
 - Method demonstrated recently to be also useful in HEP !
 - Eg : ATLAS calorimeter, where a **O(10 %) performance gain is obtained !**
- **The G4GammaGeneralProcess is planned to become the default in 11.1 !**

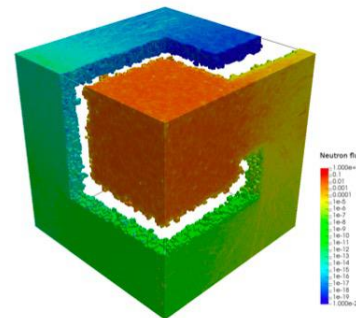
AMS : a Future Functionality for Medical Radioprotection ?



- A topic presented at the last collaboration meeting, end of September (maybe be released in 2023)
- The Adaptive Multilevel Splitting algorithm is a technique allowing to determine the flux reaching a distant region along a complicated path
 - New CEA members, experts of TRIPOLI-4, are proposing this technique to Geant4
- In a few words:
 - Perform successive attempts as:
 - Shoot tracks (neutrons) and split the track which got closer to the target during its path
 - Kill the track which got farther



(a) Analog neutron flux



(b) AMS neutron flux

- Used in radioprotection problems !

▪ **Could this be of interest for treatment planning rooms ?**

An R&D : EM HEP on GPU



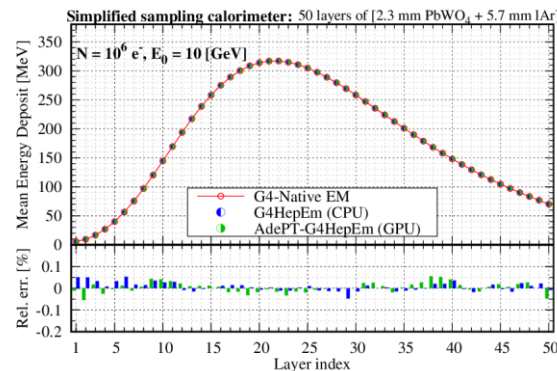
- High Energy EM showers in calorimeter :
 - A « **go/nogo** » test for what porting to GPU is concerned
 - « Simple » physics
 - Limited to e^+ , e^- , γ
 - **Expensive simulation on CPU**
 - Hence interesting for acceleration
 - VecGeom package used for geometry porting

- Rewriting HEP EM physics : G4HepEm
 - Work by Mihaly Novak
 - Compact C-like code, compatible CPU & GPU
 - Utilized on GPU by Jonas Hahnfeld in the context of the CERN AdePT project
 - Also used in a similar US project, Celeritas

• **An encouraging milestone !** But more realism needed (setup, and workflow) before measuring **performances** Expected in O(1 year).

• An R&D of interest to medical users too !

- Perfect physics G4/G4HepEM comparison



- Similar performances found on a simplified setup between
 - NVIDIA card GeForce RTX 2070 SUPER
 - And 24 threads on AMD Ryzen 9 3900

Collaboration Evolution

Development & Distribution Model



Development



GitLab

- **Development repository (geant4-dev)**
 - Development and experimental area
- Restricted to Geant4 members
- Developer proposes a Merge Request (MR)
 - MR evaluated by WG coordinators
 - Undergoes system testing, continuous & nightly
- Order of ~1000 MRs / year
- Campaigns of validation for public releases

Distribution

Geant4 web page

GitHub

- **Public repository (geant4)**
 - Mirror of geant4-dev for public releases & patches
- Open to anyone Pull Request (PR)
 - Evaluated by responsible Geant4 developers, and then eventually integrated in geant4-dev
- Order of ~10 PRs / year



CVMFS, mainly for LHC experiments

- Monthly development snapshots
- For early feed-back

Geant4 & Open Development Model



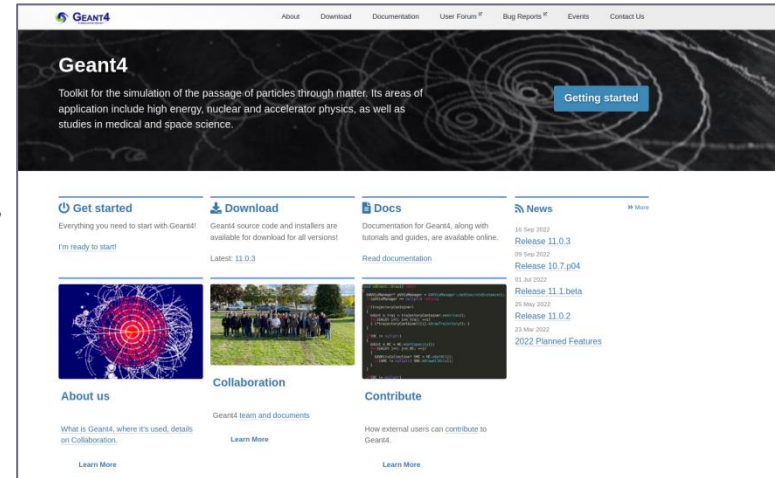
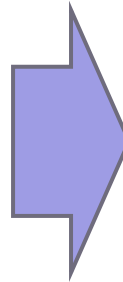
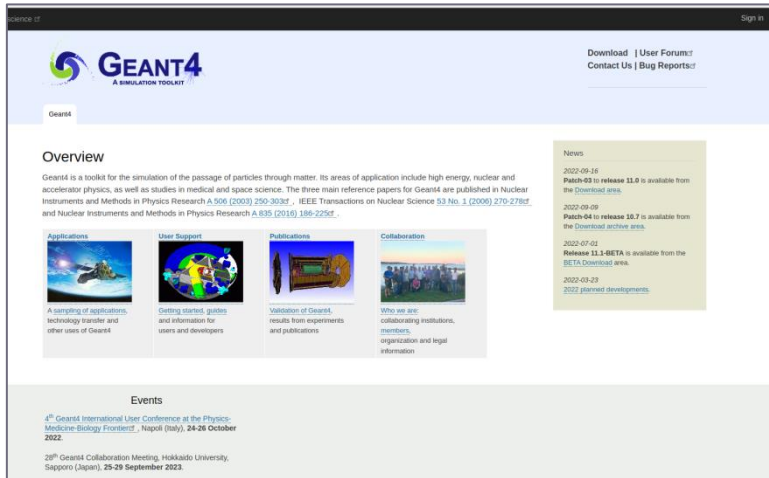
The new “contributor” status

- Open Development Models are popular
 - And Geant4 has been encouraged to do so
 - Namely : opening GitLab to public
- But we think this fully open model is **not adapted** to Geant4:
 - Snapshot of repository code is **not usable** at any arbitrary time t:
 - **Physics development time scale**
 - **Physics validation time scale**
 - Intellectual property of work in progress
- But we believe that a **partial** opening maybe be beneficial
 - **Knowledgeable** users could provide bug fixes or additional light functionalities
 - **Could help to attract new developers**
- Geant4 Collaboration has agreed for a new “**contributor**” category
 - Intended for users who want to take an active role and follow closely their contribution in Geant4
 - Not a replacement, but an addition to the existing ways to contribute
 - Enrollment web form based, renewed every year, with declaration on intended contribution and **commitment to not distribute or publish results from non-public releases**
- Could this be also a way to develop extra/non-core functionalities : eg **python binding** ?

New Website !



- Geant4 has revised its website
 - “Background” task which spanned over ~2 years (so many details to tackle !)
- Moves from Drupal technology (not supported anymore at CERN) to Jekyll one + workflow managed under GitLab; and strong input from young generation !



- **New site to be released within O(few days), so don't be surprised ! ;)**
 - And of course, in case of any problem, please contact us !

Summary



- Plenty of activities on-going
 - Sub-minimal sample shown here
- Geant4 faces the HL-LHC challenge
 - But this is not to the detriment of the medical domain !
 - Solutions developed/investigated are often of general interest
- Tasking has demonstrated its ability to take over the previous Multi-threading scheme
 - And looks a promising road toward hybrid computing
- Many developments going on in physics
 - With strong focus in EM on performance gains (general process, Woodcock tracking...)
 - Which show that interest is bi-directional : ie Woodcock tracking benefits to HEP !
- Geant4 takes a step towards the Open Development Model
 - And we will see how this goes
- **Many things** not discussed in this presentation !
 - Some will be covered during **dedicated presentations** !
 - And if others are not planned, **coffee breaks can be a fruitful way to recover** !