Geant4 General Status

15th Geant4 Space Users' Workshop,
5th – 7th December 2023,
Hyatt Place Pasadena,
Marc Verderi (LLR)

Overview

I. Software Aspects

- Geant4 releases & highlights since Last Users' Space Workshop
- Some highlights of 11.2
- Migration to 11.2 & supported platforms

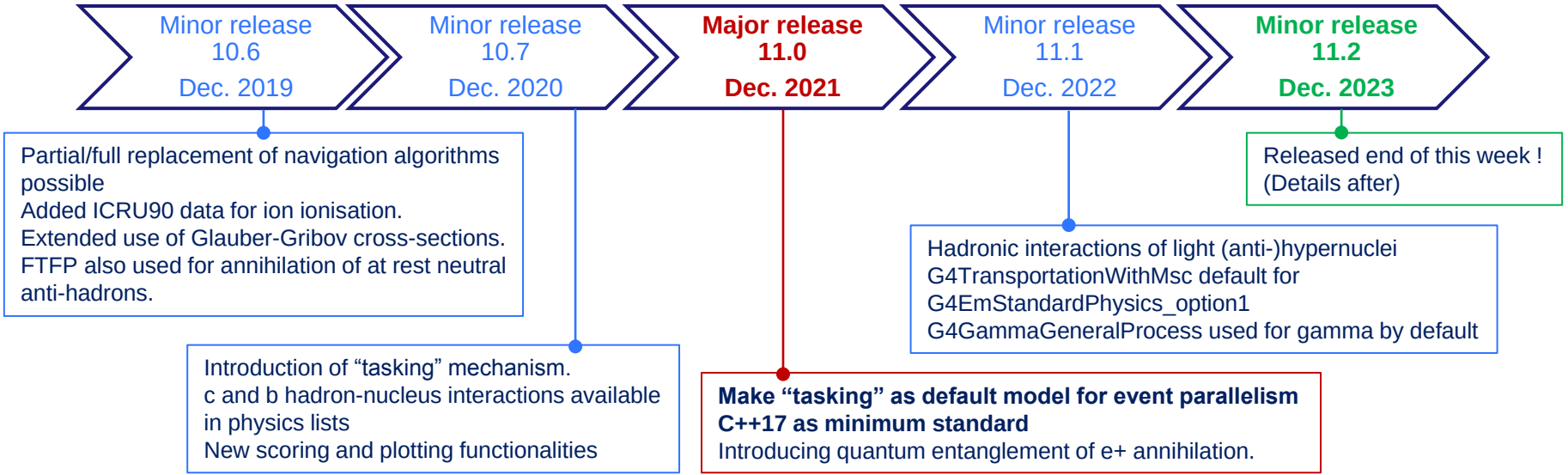
II. Collaboration Aspects

- New website
- The new "Contributor" status
- R&D for HEP EM physics on GPU
- Building a Community for Python (and Others) Binding ?

I. Software Aspects

Geant4 Releases & Highlights Since Last Users' Space Workshop

- Geant4 release at the time of the last workshop (Oct. 2019, Korinthia) was **10.5**
- Since then:



Some Highlights of 11.2

- New process **G4XrayReflection** to describe X-ray surface scattering based on data provided in G4EMLOW-8.5.
- New class **G4MicroElecCapture** for better estimation of the non ionizing energy deposition.
- In **G4RadioactiveDecay**, changed default threshold for the time beyond which radioactive decays are ignored, **from twice the age of the Universe to one year**.
  - If ions radioactive decay is important to you, put a very large threshold.
- Added **new physics lists to the physics factory, for HPT variants** (i.e. with special treatment of elastic scattering of thermal neutrons) : FTFP_BERT_HPT, … , Shielding_HPT, …
- Added new **physics constructor for neutrino physics**, G4NeutrinoPhysics, and its messenger. New constructors **G4ChargeExchangePhysics and G4NeutrinoPhysics may be added on top of any modular Physics List.**
- **First implementation of G4SubEvent** and related mechanism for registration and merging in a run.
- **New Quantum State Simulation (QSS) integration method**
  - an alternative method of integration offering built-in interpolation capability and enabling faster finding of the intersection of the trajectory with surfaces.
- New G4AnalysisManager functions for deleting selected histograms/profiles/n-tuples, with related UI commands.
- **First version of the TSG_[QT,X11,XT,WINDOWS]_ZB sub drivers**, allowing for interactive 3D rendering bypassing native graphics systems.
- **New Geant4-FLUKA interface and two hadronic examples**, providing access to FLUKA-Cern hadron-nucleus inelastic physics.

Migration to 11.2 & Supported Platforms

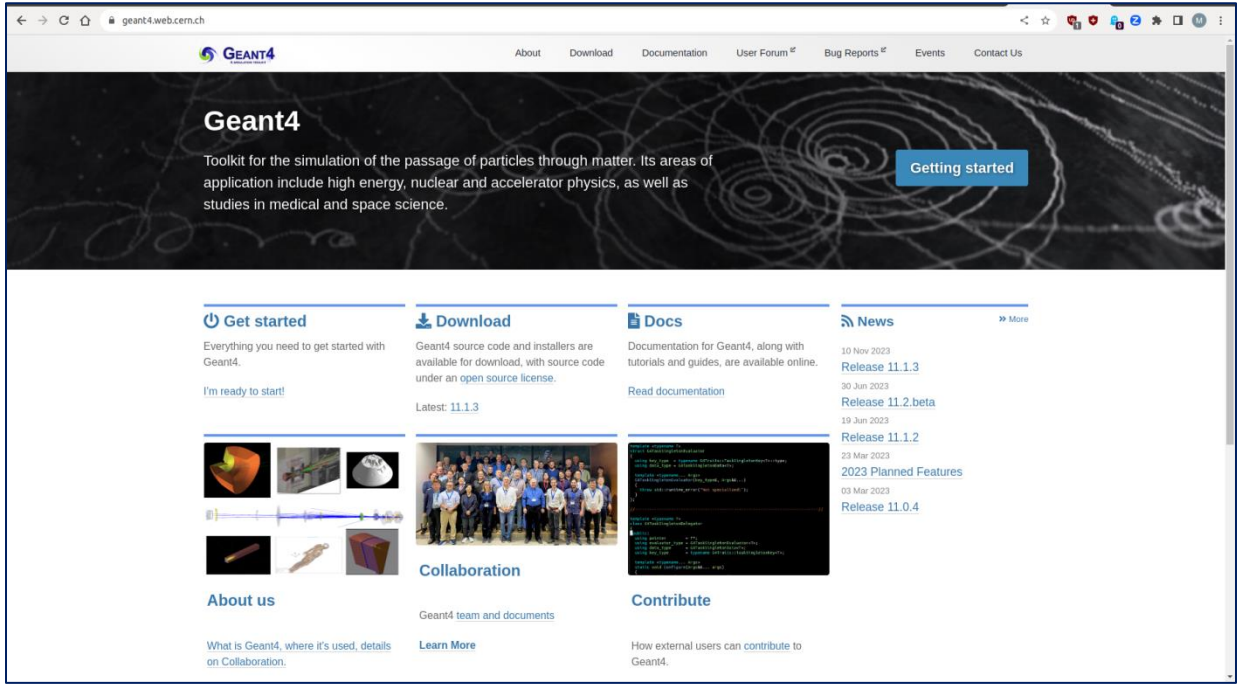
- **Some items for migration of the user code:**
  - **Simplification of touchable handling** : inheritance level for G4TouchableHistory removed (G4VTouchable now typedef of G4TouchableHistory)
  - **G4NeutrinoPhysics** constructor added.
  - **Time threshold for radioactive decays of ions** (used to ignore decay) : ~twice the age of universe → 1 year.
  - **G4Persistency library split** as G4mctruth, G4geomtext, G4gdml.
  - **Data sets:**
    - **New low-energy data set version, G4EMLOW-8.5:**
      - Updated microelec data for e^- ; Updated DNA data.
      - New data subdirectory for XRayReflection.
      - Updated MicroElec data inside Inelastic and Structure subdirectories.
    - **New nuclear shell effects data set version, G4ABLA-3.3:**
      - New file mass2016.dat with experimental atomic mass evaluation from 2016.
      - New file mass2020.dat with binding energies from AME2020.
    - **New data set version for p & n density profiles, G4INCL-1.2:**
      - New files for antiproton annihilation at rest of INCL.
      - New data files for in-flight antiprotons (Channel probabilities).

- **Supported and Tested Platforms:**
  - **Platforms:**
    - Linux, gcc-11.3.1.
    - Tested on 64 bit architectures (Intel or AMD) with Alma Linux 9 (based on RedHat Linux Enterprise 9).
    - macOS 14.1.1 Sonoma with Apple LLVM/Clang-15.
    - Tested on 64 bit architectures (Intel or Apple Silicon).
    - Windows-10 with Visual C++ 14.36 (Visual Studio 2022).
  - **More verified and tested configurations (64 bits):**
    - Linux, gcc-9.4/10.3/11.3/12.1/13.2, clang-14/16/17
    - Linux, Intel-icx 2022.2
    - macOS 13.6 Ventura with Apple LLVM/clang-15
    - macOS 12.7 Monterey with Apple LLVM/clang-14
    - Windows/10 with Visual C++ 14.29 (Visual Studio 2019)

II. Collaboration Aspects

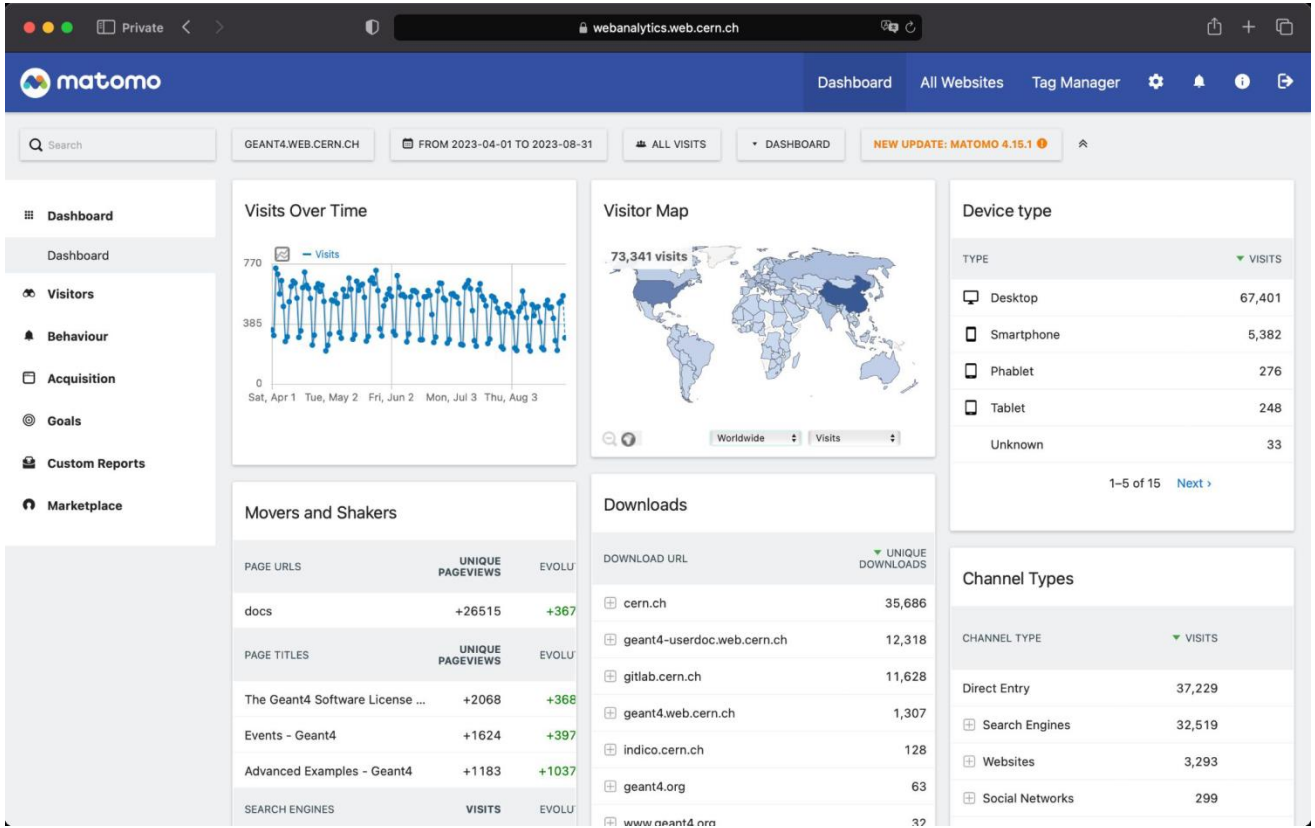
New Website

- **Website renovated**, using **Jekyll**, files maintained in a GitLab instance
  - **Git permissions** used to allow all G4 members to make MR, with validation by working group coordinators etc.



- **Needed to re-write all Drupal-based site !** Scripts for some automation (eg : release page) created too.

BTW, download statistics



Courtesy of Ben Morgan (U. Warwick)

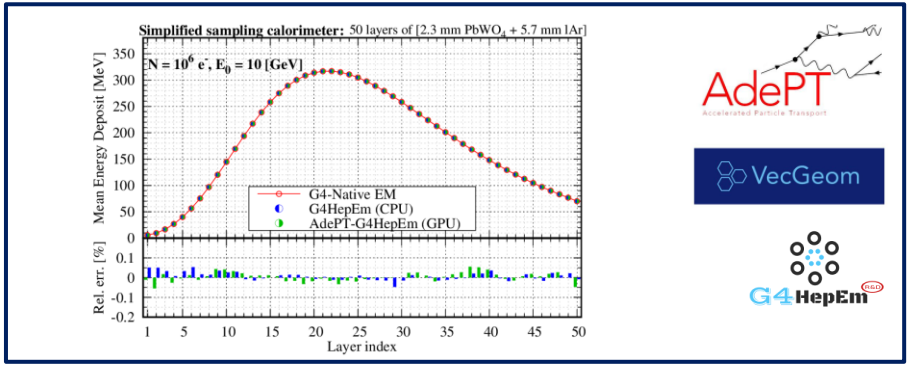
The new “Contributor” status

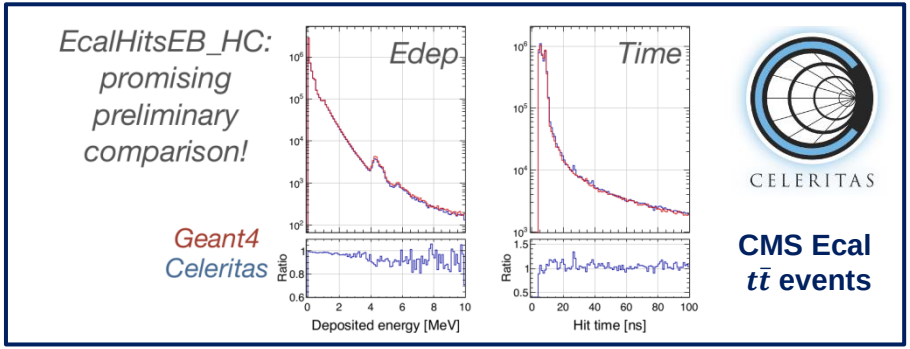
- Status introduced beginning of 2022
  - Inspired from “Open Development Model”
  - But with restrictions, as an open development model is not applicable to Geant4
    - Physics development and validation time scales prevent this
- **Intended for people having a good understanding of Geant4 and who want to provide light functionalities/bug fixes, with an active role about these in Geant4**
- **Full access to the GitLab development repository** (MR and CI)
  - Candidates proposed by WG coordinators, and approved by the management
  - Specify what is the intended contribution with **commitment to not distribute or publish results from non-public releases**
- After almost two years, and about ~10 contributors, status allows to the barrier to contribute to Geant4 and should ease recruiting new full members !
- **So, you’re welcome to contribute Geant4 this way !**

R&D for HEP EM Physics on GPU

- HEP EM physics in calorimeters is by far the **most expensive part to simulate** in term of CPU usage
  - But the physics itself is “relatively simple”:
  - Ie : few particle types, few processes
- **This makes EM physics a good candidate to try to port on GPU**
  - As a **“go / no-go”** for what HEP simulation on GPU is concerned
  - Note : low EM physics already ported to GPU in medical applications, for **simple** geometries

- Two projects investigating this:
  - **AdePT : CERN**
    - VecGeom for geom., and G4HepEM for physics
  - **Celeritas : US**
    - Developing ORANGE for geometry (surface based representation), and using G4HepEM + own physics implementation





- Speed-up GPU/CPU ranges ~O(1 – 6)
- **Assessment of GPU projects next week !**

Building a Community for Python (and Others) Binding ?

- Python, Julia, (others ?) interfaces are popular
  - Motivations : “C++ is too hard”, “This improves usability”, “Young developers expect such interfaces”, etc.
- Several private developments exist, often tailored to some application domains
  - Sometimes puzzling features: “I simplified the interfaces”, “SetProtonMass(…)”
- **Strong demand, various needs & use-cases, but Geant4 has no specific manpower on this !**

- How to respond ?
  - **Initiative by Ben Morgan** (U. Warwick)
  - **Idea : embark experienced developers (outside Geant4) along an “open development model”**

- Proposes **formation of an “Interest Group”** (see HSF), **under Geant4 aegis**, and proceed as:
  - **Step 1: Gathering Interest and Requirements**
    - List of requirements, existing solutions, counting level of interest/FTE for development
  - **Step 2: Determining common solution potential**
    - Under Geant4 aegis, but Geant4 would not be there to a force particular solution.
  - **Step 3: Development and Sustainability**
    - Proposes to develop bindings as open projects on GitHub, against latest public releases of Geant4
    - Project members would build a sustainable support and user community
    - Feed-back on Geant4 interfaces will be welcome

- **Plan to start this process early 2024 !**

Thank you for your attention !