General introduction of version 10.4 and prospects

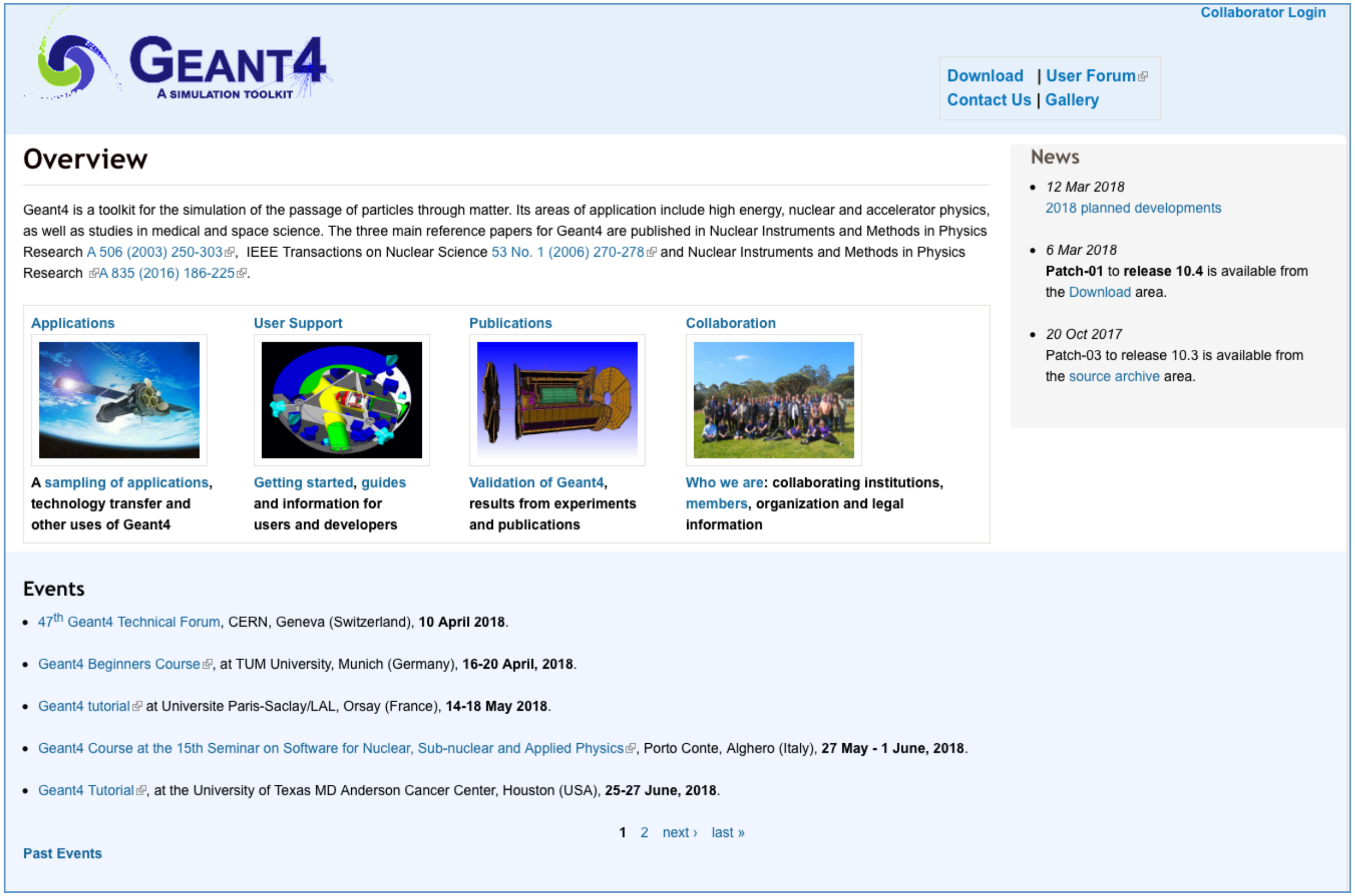
Makoto Asai
SLAC National Accelerator Laboratory
May 17th, 2018
@ EIC Software Consortium Meeting

Version 10.4

- Released on December 8th, 2017
  - Mar 6th, 2018 – Geant4 10.4-patch01 release
  - Geant4 10.4-patch02 is in preparation. To be released in a few days.
- Cmake 3.3 or higher is required.
- Platforms:
  - Linux, gcc-4.8.5.
    - Tested on 64 bit architectures (Intel or AMD) with CERN CentOS Linux 7 (CC7) (based on CentOS Linux 7).
  - MacOSX 10.13 High-Sierra with clang-4.0 (Apple LLVM/Clang-9.0.0)
  - Windows-10 with Visual C++ 14.11 (Visual Studio 2017)
- More verified and tested configurations (64 bits):
  - Linux, gcc-4.9.3, gcc-5.3.0, gcc-6.3.0, gcc-7.2.0, clang-3.9
  - Linux, Intel-icc 18.0
  - MacOSX 10.10, 11, 12 with clang-3.6, 3.7, 3.8
  - Windows-7, 10 with Visual C++ 14.0 (Visual Studio 2015)
  - Linux for Intel Xeon Phi with Intel-icc 16.0 (gcc-4.9 compatibility layer)
- Note: Windows platforms are supported/verified only for the sequential mode. Multithreading capability is not yet supported on Windows.

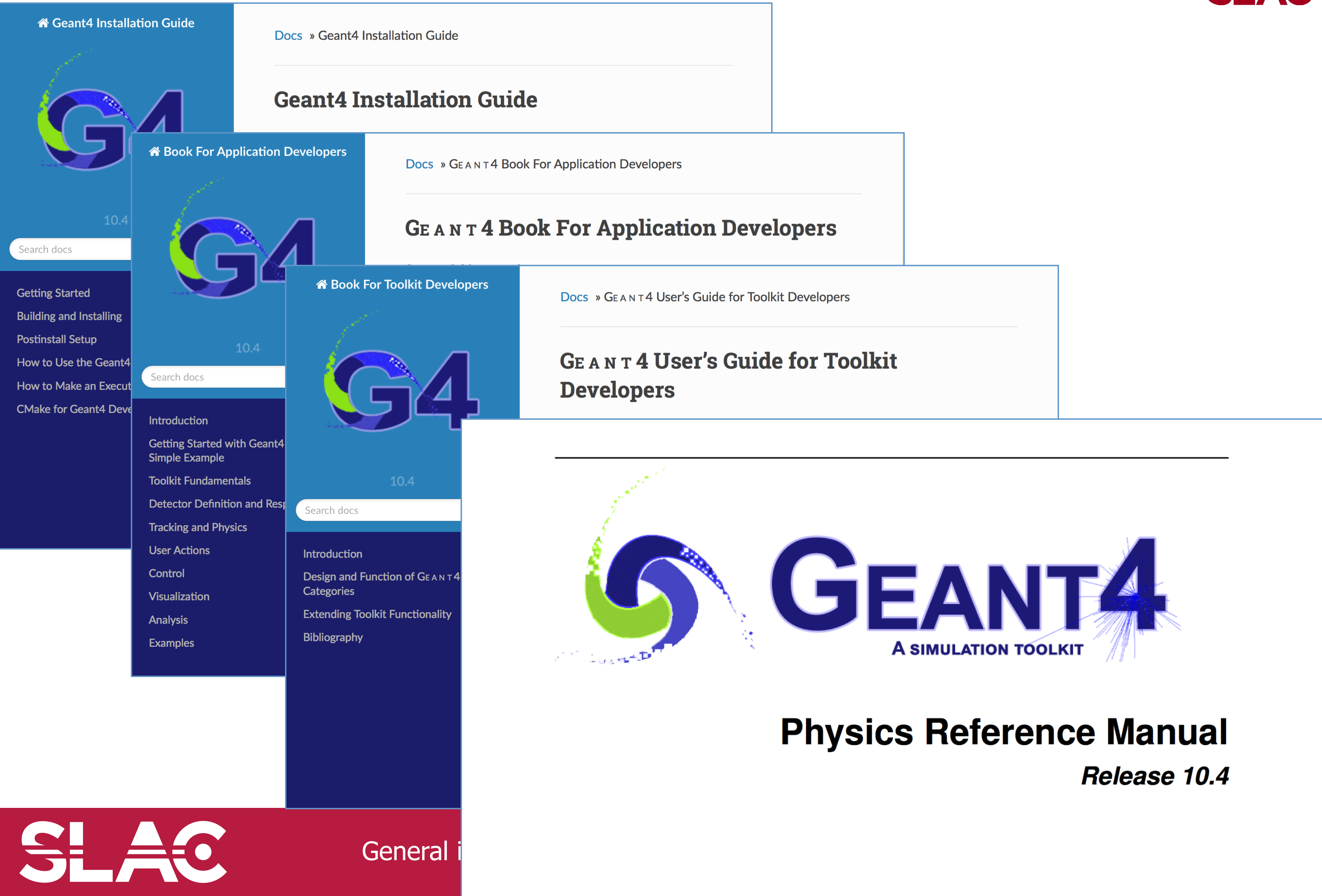
We moved to a new Drupal site

- http://geant4.org/ (http://cern.ch/geant4/)



- We will continue updating/enriching pages.

10.4 came with new user’s guides and new logo



Releases and events

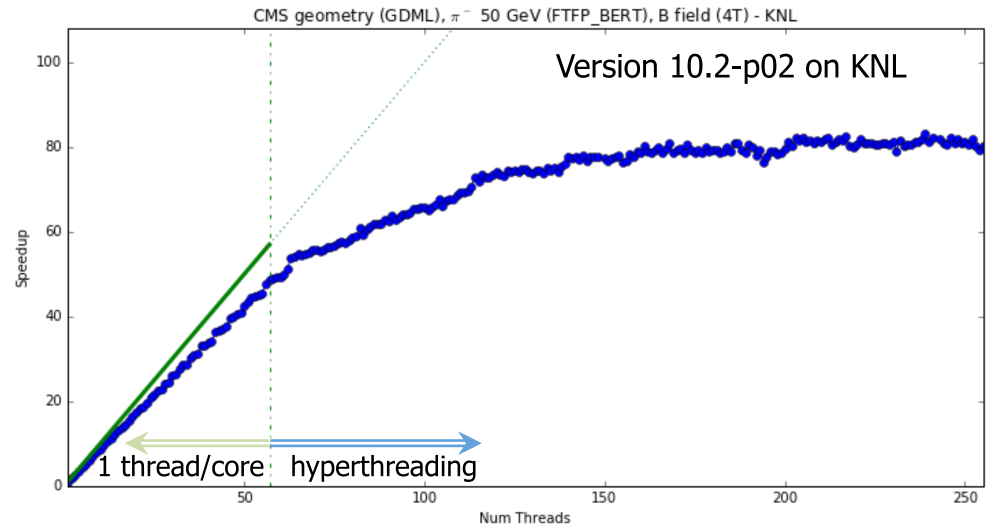
- As announced, 2018 work plan is listed here.
  - http://geant4.web.cern.ch/support/planned_features
- Coming releases
  - Next public release : December 7th, 2018
    - Next beta release : June 29th, 2018
  - Patch releases for version 10.4 will be made as needed.
  - Patches for older versions may be discussed.
- Other events
  - Several tutorial courses are listed on the event section of the collaboration home page.
  - Future tutorials will be announced.
- 2019 Geant4 Collaboration Meeting will be held at JLAB (or Hampton U.)
  - Nuclear Physics will be highlighted.
  - Associated users' meeting will be arranged.
- SLAC continues acting as a liaison to EIC (and nuclear physics in general).

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

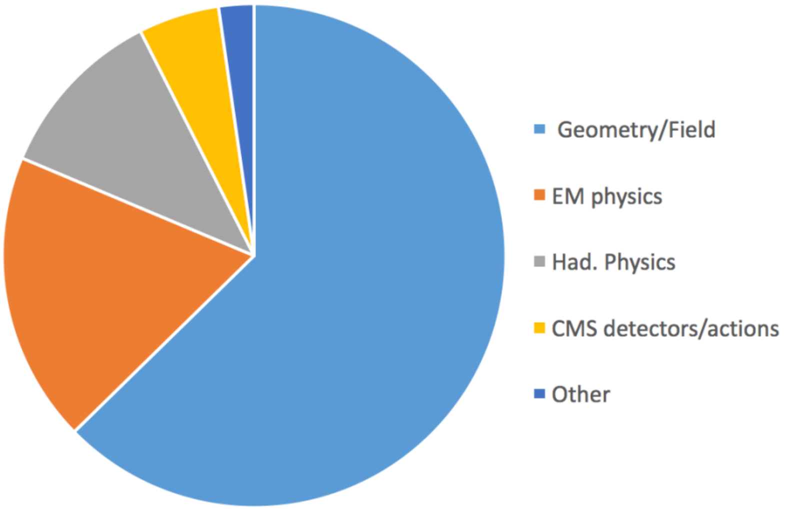
- Recently, for Intensity Frontier, muonic atoms were introduced:
  - i.e. atoms in which an e^{-} is replaced by a μ^{-}
  - Mechanism for ions is duplicated, and related code is protected by an "#ifdef"…
- And further:
  - Radicals, e.g. OH^{-}, are also handled at low energy in the DNA module
  - 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

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
    - …

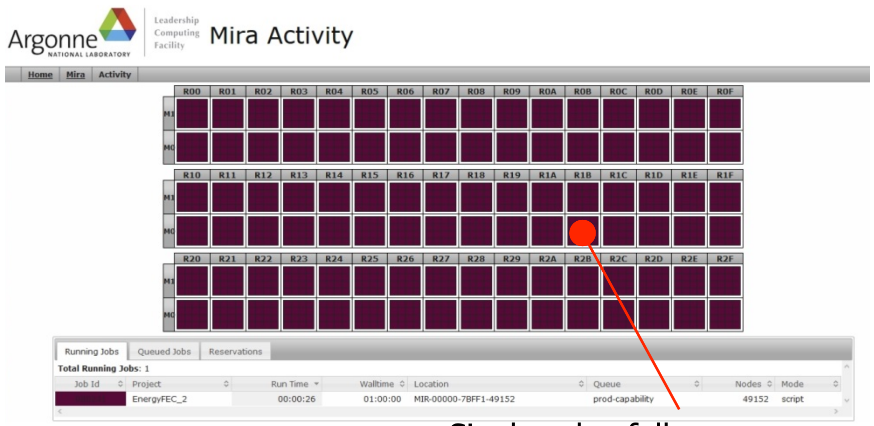


Sources of CPU consumption Geant4 CMS simulation
Courtesy of Vladimir Ivanchenko

Use of HPC

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



Single color, full squares = one application occupies Mira all

- 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.

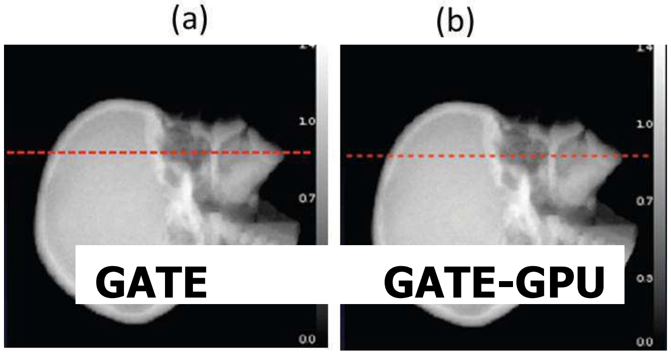
Physics & Code Evolution

- 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

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.
- 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….



Head CT scan simulation
Courtesy of OpenGate Collaboration
(Note GPU project note anymore developed by GATE)