# GEANT4 10.1 highlights

*kernel modules*

Gabriele Cosmo, CERN PH-SFT

for the Geant4 Collaboration

# Outline

- Major features introduced in release 10.1
  - Geometry
  - Kernel & Interfaces
  - Visualization
  - Physics (see talks by V.Ivantchenko and A.Ribon)

➢ *Detailed release notes:*

- **http://geant4.cern.ch/support/ReleaseNotes4.10.1.html**

➢ *All planned features for 2014:*

- **http://geant4.cern.ch/support/planned_features.shtml**

# Geant4 10.01

*Consolidation release*

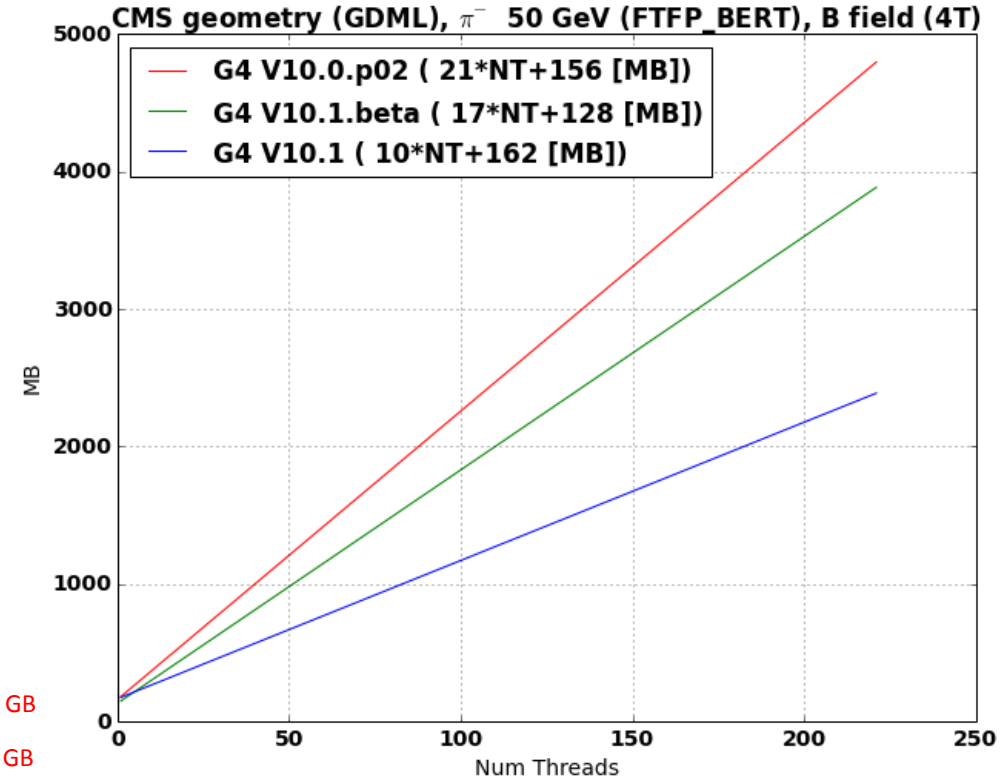
- Consolidation of multi-threading and improved thread-safety
  - Improved memory footprint
  - Improved memory management
- Improved CPU performance for both sequential and MT modes
- Extended set of primitives for USolids
  - USolids module now external
  - Updated GDML schema
- Revised policy for the treatment of ions and isomers
- Bug fixes & some new features
- Updated documentation

# Multi-threading

## *Memory management*

- Many fixes, improvements for multi-threading
  - Fixed issues of thread safety
  - Further sharing by making const more shared classes/data
- Benchmark: LHC detector geometry - CMS 2008 ($10^6$ volumes, 300 materials, no hits or trajectories)
  - Each **additional thread** requires less memory **10 MB** (reduced from 19MB) excluding hits
  - Rough estimate of savings is 100-200 MB/thread depending on physics setup
- Feedback from CMS (100 t-tbar events CMSSW full simulation) on release 10.0 with MT
  - 1.2 GB memory shared
  - 200 MB additional memory per-thread
  - i.e.: multi-processing on 12 threads: 12*1.4 = 16.8 GB

    multi-threading on 12 threads: 12*0.2 + 1.2 = 3.6 GB

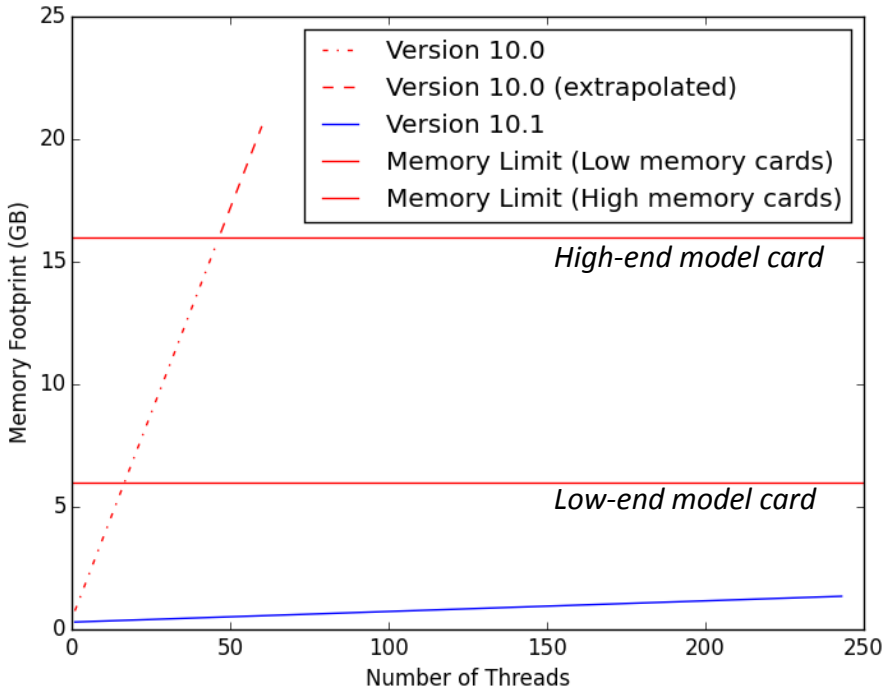


# Multi-threading

*Memory management - GPS*

- Considerably improved memory management for General Particle Source primary particles generator

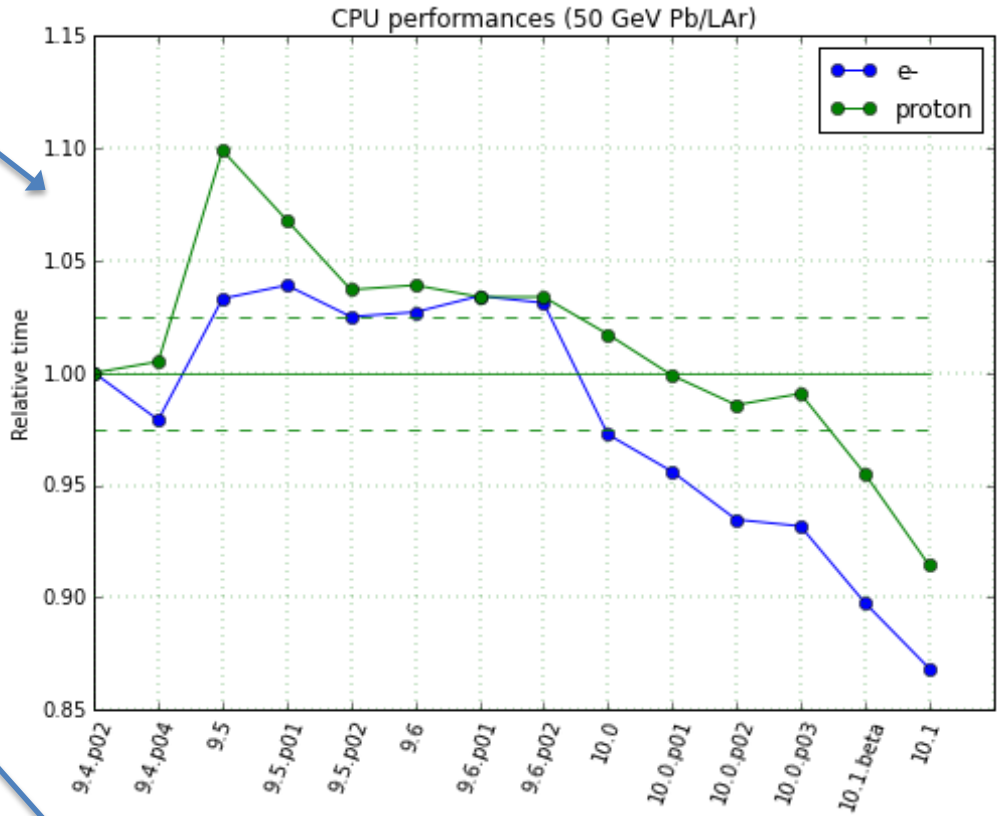
- Benchmark: proton-therapy application using GPS with hundred of point-sources each with its own distribution
  - Adopting sharing of distributions and other “read-only” data
  - Replaced C-arrays with more efficient STL containers

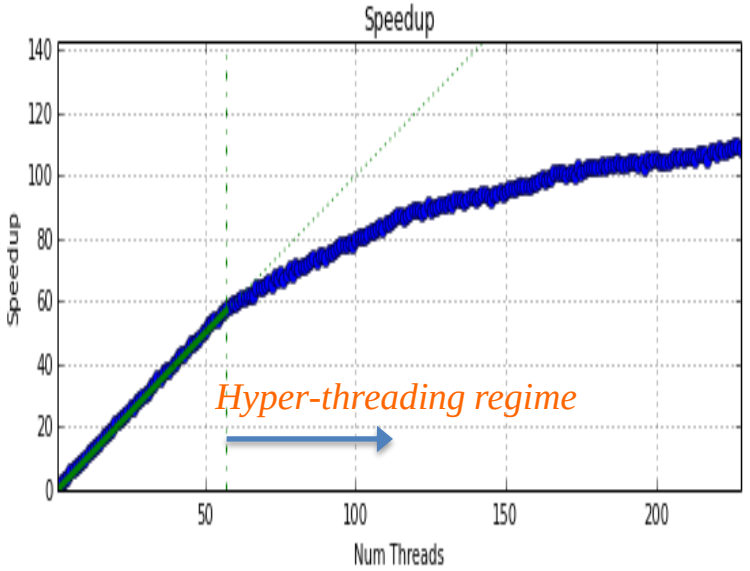


# Performance

*Improved speed for sequential/MT modes*

- Considerably improved CPU speedup and excellent MT linearity
  - Benchmark: Simplified Calorimeter Pb/LAr
    - Up to 10% speedup versus release 10.0

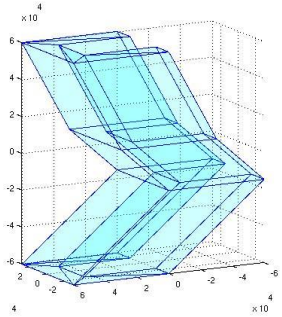


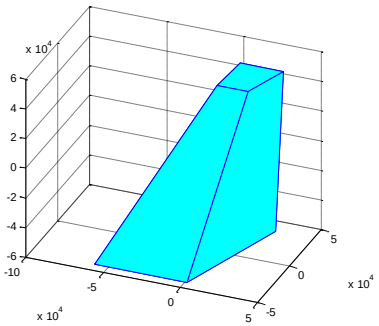



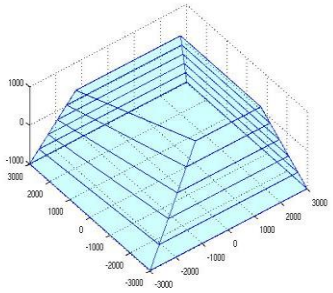

- Benchmark on Xeon-Phi: CMS 2008
  - π- @ 50 Gev with 4T field

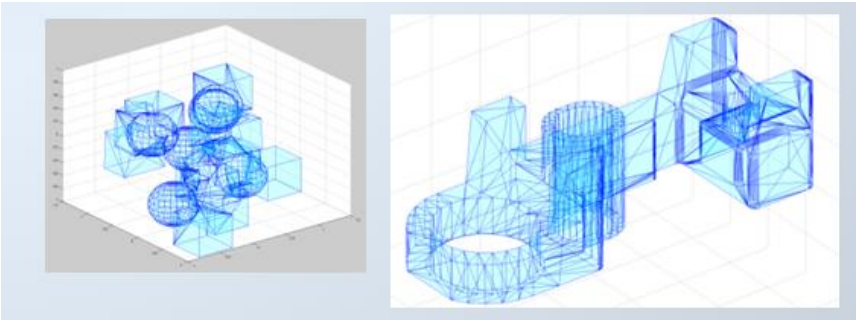
# Geometry

## *Geometrical primitives - USolids*









- AIDA Unified Solids library update
  - As optional component, for replacing the original solids (G4GEOM_USE_USOLIDS flag)
  - Was embedded in 10.0 inside geometry module; can be adopted as external separate library in 10.1
  - Going to evolve with new and even more optimised implementations from VecGeom
- Optimised implementation and bug fixes for several shapes
  - UPolycone in particular
- Included new shapes
  - UExtrudedSolid, UGenericTrap, Utrap
- New UMultiUnion structure
  - Replaces multiple use of 'binary' Boolean unions for volumes with same material
  - Benefits above 3-4 components
  - Supported in GDML, version 3.1.1

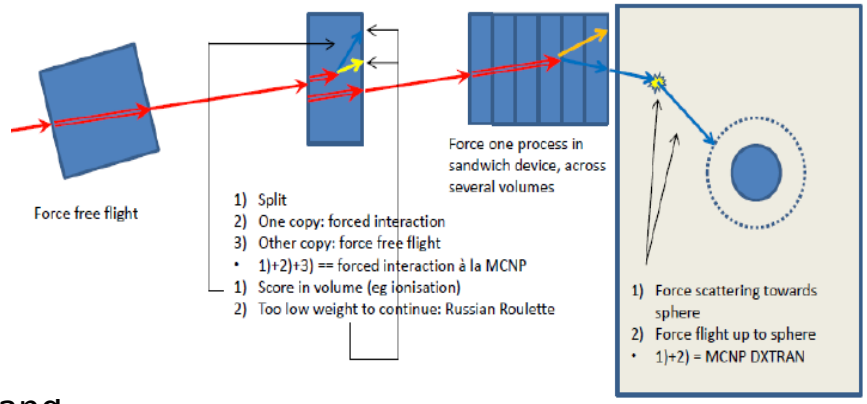
Please, try it out and give us your feedback!

# Physics Lists

- Updated physics lists combinations and options
  - Switched on muon-nuclear by default in all lists
  - Cut on proton set to zero, to simulate all recoils from elastic scattering, in physics lists using neutron-HP
  - Trying out new hadron elastic scattering, still in development, in the experimental physics-list FTFP_BERT_TRV
  - Restructured G4OpticalPhysics constructor and changed UI commands to allow for different particles to have specific settings of optical parameters
  - New experimental lists for Shielding and Neutrino physics

# More features …
## *Highlights*

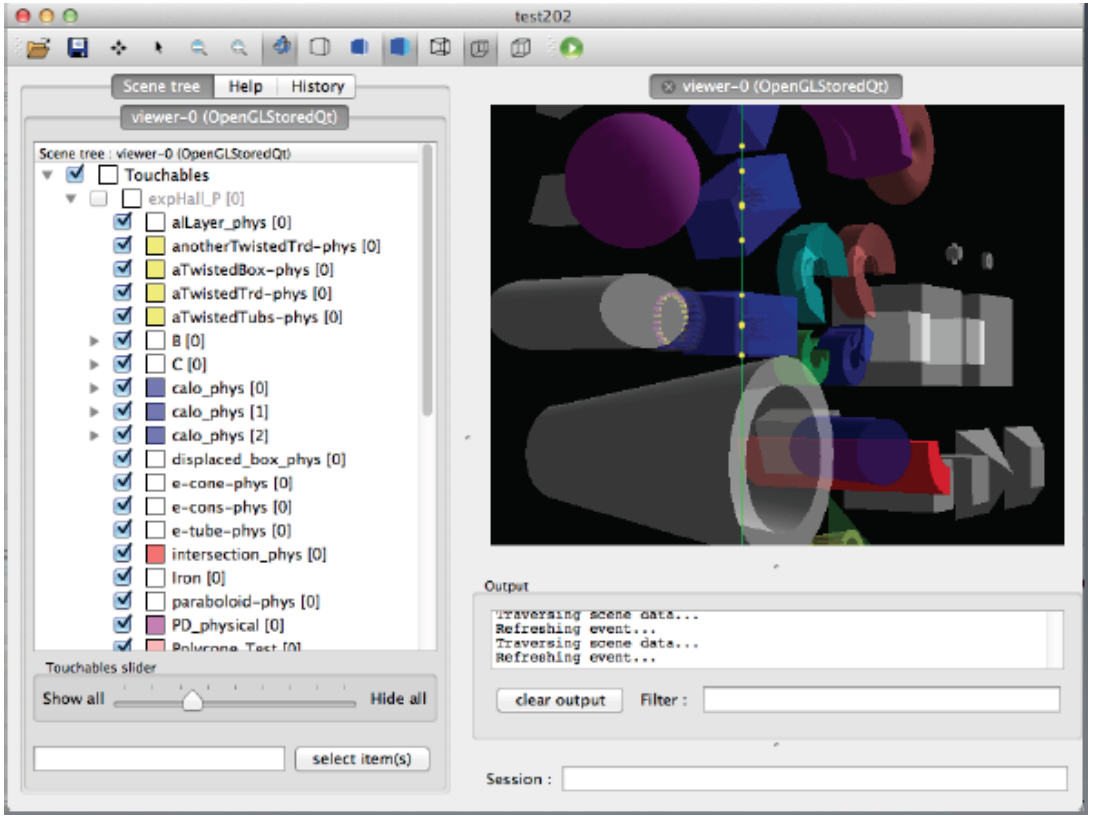
- Adopted fast implementations of exp() and log() from VDT - first in 10.0
  - Use expanded in 10.1 - with benefit in CPU performance
- Introduced memory pool for reuse of G4NavigationHistory objects
  - Replacing use of G4EnhancedVectorAllocator to improve memory churn
- Introduced adoption of “workspaces” for particles and physics-lists MT data
  - Completely transparent to the user
  - Step forward towards better memory handling model also with TBB
- New framework for “generic” biasing for physics-based biasing – first in 10.0
  - Now providing easier access to cross-sections/mean-free-paths of wrapped physics processes
- Ions and isomers now instantiated during the event loop as they become necessary
  - Pre-instantiation still possible through UI command
- First support for “phonon” physics propagation in kernel



# More features …

*Visualization & Analysis*

- Evolved Qt UI/GUI
  - Interactive settings of viewer parameters
  - Ability to detach panels
  - Improved picking features
  - Addition of “Useful Tips”
- Improved support for multi-threading
  - Ability to select output by thread
- New/improved UI commands




- Enhancements to analysis module
  - Support for CSV and XML Ntuples

# External libraries and data sets

- Adopting new CLHEP version 2.2.0.4
  - Converted statics and globals to const, thread_local or atomic to improve thread-safety
  - Required for multi-threading and C++11 enabled setups
  - Still compatible with older CLHEP releases
- Updated data sets:
  - G4EMLOW-6.41, G4NDL-4.5 (compressed), RadioactiveDecay-4.2, G4PhotonEvaporation-3.1

# 10.1: more …

- Configuration (Cmake)
  - Enabled support for “offloading” on Xeon-Phi co-processors
- Particles
  - Updated particle properties to PDG-2014
- G4Py
  - Updated interfaces and completed migration to CMake for building libraries
- Examples
  - Migrated most examples to allow use of multi-threading

# Platforms for 10.1

- Linux SLC6, gcc-4.4.7, 4.7.X, 4.8.X, 64 bits
- MacOSX 10.9, gcc-4.2.1/clang-3.5, 64 bits
- Windows 7, Visual C++ 12.0 (Visual Studio 2013)
  - *No support for multi-threading on Windows yet*

- Also tested:
  - Linux, gcc-4.9, icc-14, icc-15
  - Linux Ubuntu 12, gcc-4.6
  - MacOSX 10.10, gcc-4.2.1/clang-3.5, 64 bits
  - Windows 7, VC++-9.0/10.0/11.0
  - Intel Xeon-Phi co-processor, icc-15

# Thanks!