Geant 4

Introduction to version 10.2
~ Kernel part ~

Makoto Asai (SLAC SD/EPP)

Geant4 Technical Forum @ LAL Orsay

May 3rd, 2016

Version 10.2

- Released on December 4th, 2015
  - Patch release v10.2-patch01 on March 2nd, 2016
- Further improvements in both computing speed and memory consumption.
  - Preliminary benchmarks show a few % speedup with sizable memory reduction
- New features include:
  - C++11 native code
  - Unified Solids Library is now installed as an external package
  - Updated Goudsmit-Sounderson multiple scattering model for e+/e-
  - Full Auger atomic de-excitation cascade simulation
  - Alternative Geant4-DNA physics models for liquid water
  - Improved simulation of low-energy, evaporated nucleons by Fritiof model coupled with Precompond-de-excitation model
  - A new model for nuclear gamma de-excitation
  - Neutron_HP is extended to Particle_HP, which also deals with proton, deuteron, triton, 3He and alpha
  - Substantial revision of the Radioactive Decay model
  - “Batch plotting" facility in G4tools
  - Dedicated thread of Visualization for “real-time” event display

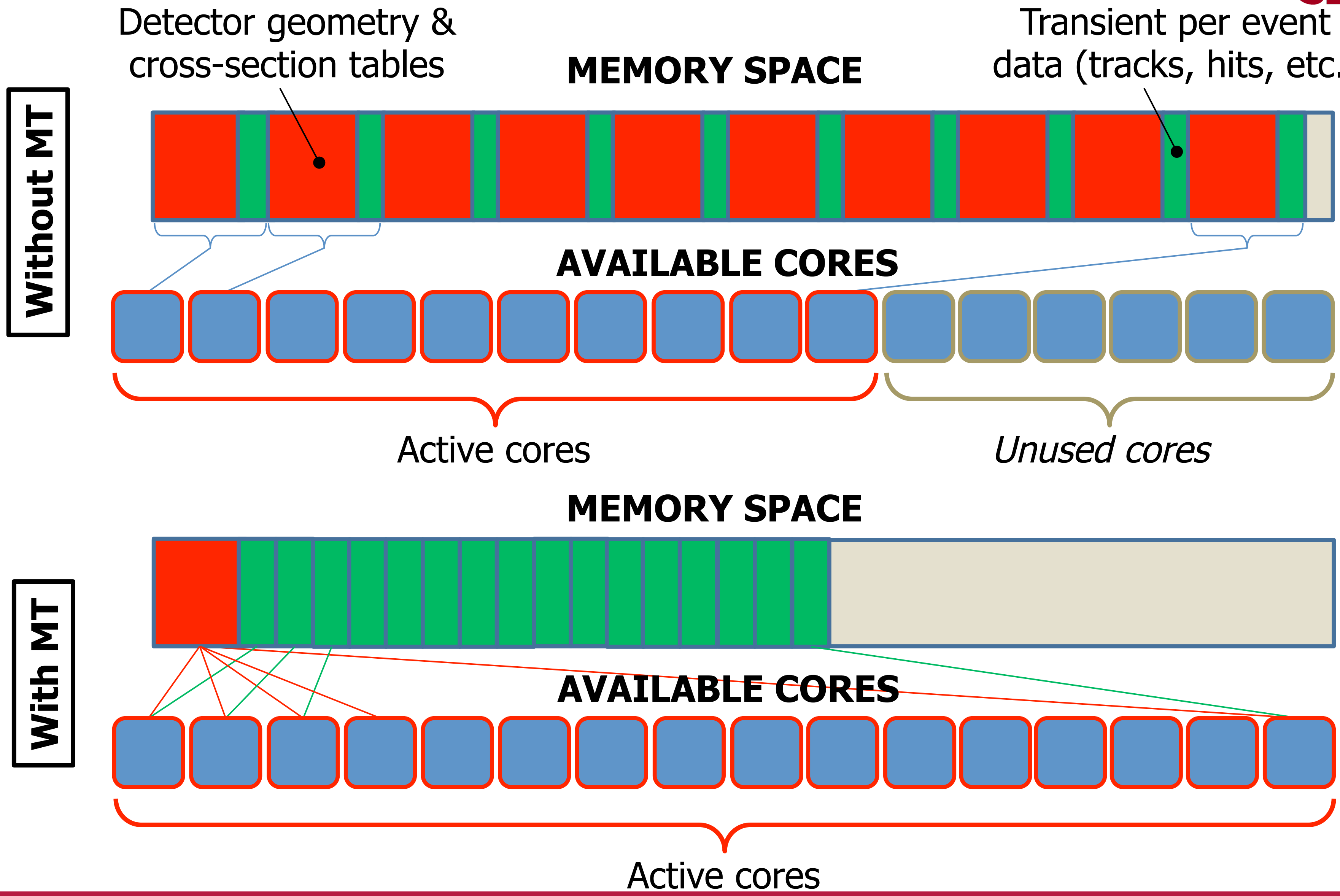
Platforms

- Compilation using the C++11 Standard is now mandatory.
- Cmake 3.3 or higher is required.
- Platforms:
  - Linux, gcc-4.8.3 on 64 bit architectures (Intel or AMD) with CERN CentOS Linux 7 (CC7) (based on CentOS Linux 7).
  - MacOSX 10.11 with clang-3.7 (Apple LLVM/Clang-7.0.0)
  - Windows7 with Visual C++ 14.0 (Visual Studio 2015)
- More verified and tested configurations (64 bits):
  - Linux, gcc-4.9.3, gcc-5.2.0, clang-3.6
  - Linux, Intel-icc 15.0, 16.0
  - MacOSX 10.10 with clang-3.6
  - MacOSX 10.9 with clang-3.5
  - Windows7 with Visual C++ 12.0 (Visual Studio 2013)
  - Linux for Intel Xeon Phi with Intel-icc 15.0, 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.

Recent patch releases

- Nov 30th, ’12 – Geant4 version 9.6 release
  - Feb 4th, ’15 - Geant4 9.6-patch04 release ← Version frozen
- Dec 6th, ’13 – Geant4 version 10.0 release
  - Mar 6th, ’15 - Geant4 10.0-patch04 release ← Retroactive patch release
- Dec 5th, ’14 – Geant4 version 10.1 release
  - Feb 5th, ’16 - Geant4 10.1-patch03 release ← Retroactive patch release
- Dec 4th, ’15 – Geant4 version 10.2 release
  - Feb 26th, ’16 - Geant4 10.2-patch01 release ← Current version

- Versions 10.0 and 10.1 will be supported with necessary patches at least for 2016.
  - They are kept compliant with C++98.
- We stop full support of version 9.6.
  - Individual cases may be discussed.



Without MT

Detector geometry & cross-section tables

MEMORY SPACE

Transient per event data (tracks, hits, etc.)

AVAILABLE CORES

Active cores

Unused cores

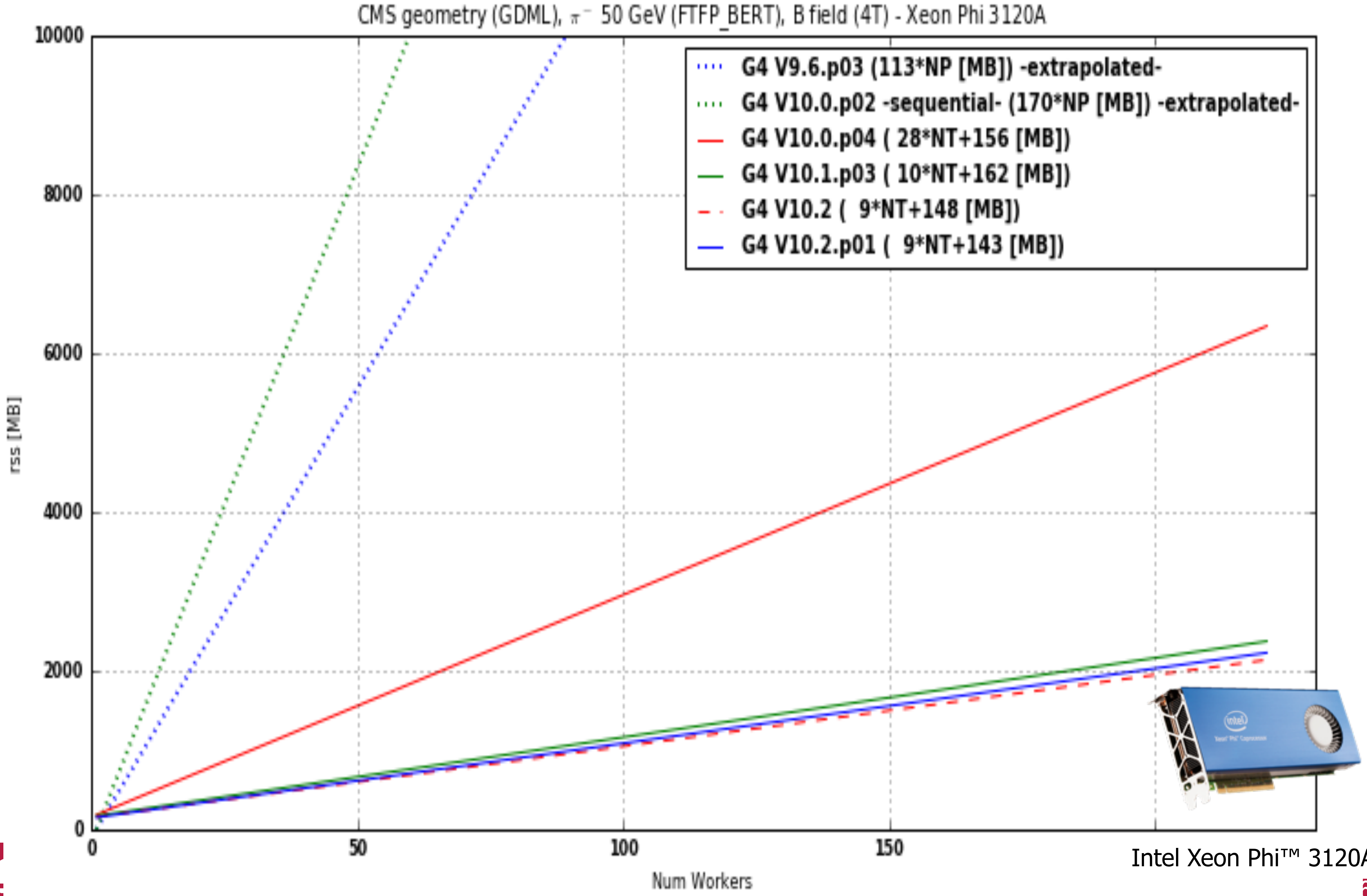
With MT

MEMORY SPACE

AVAILABLE CORES

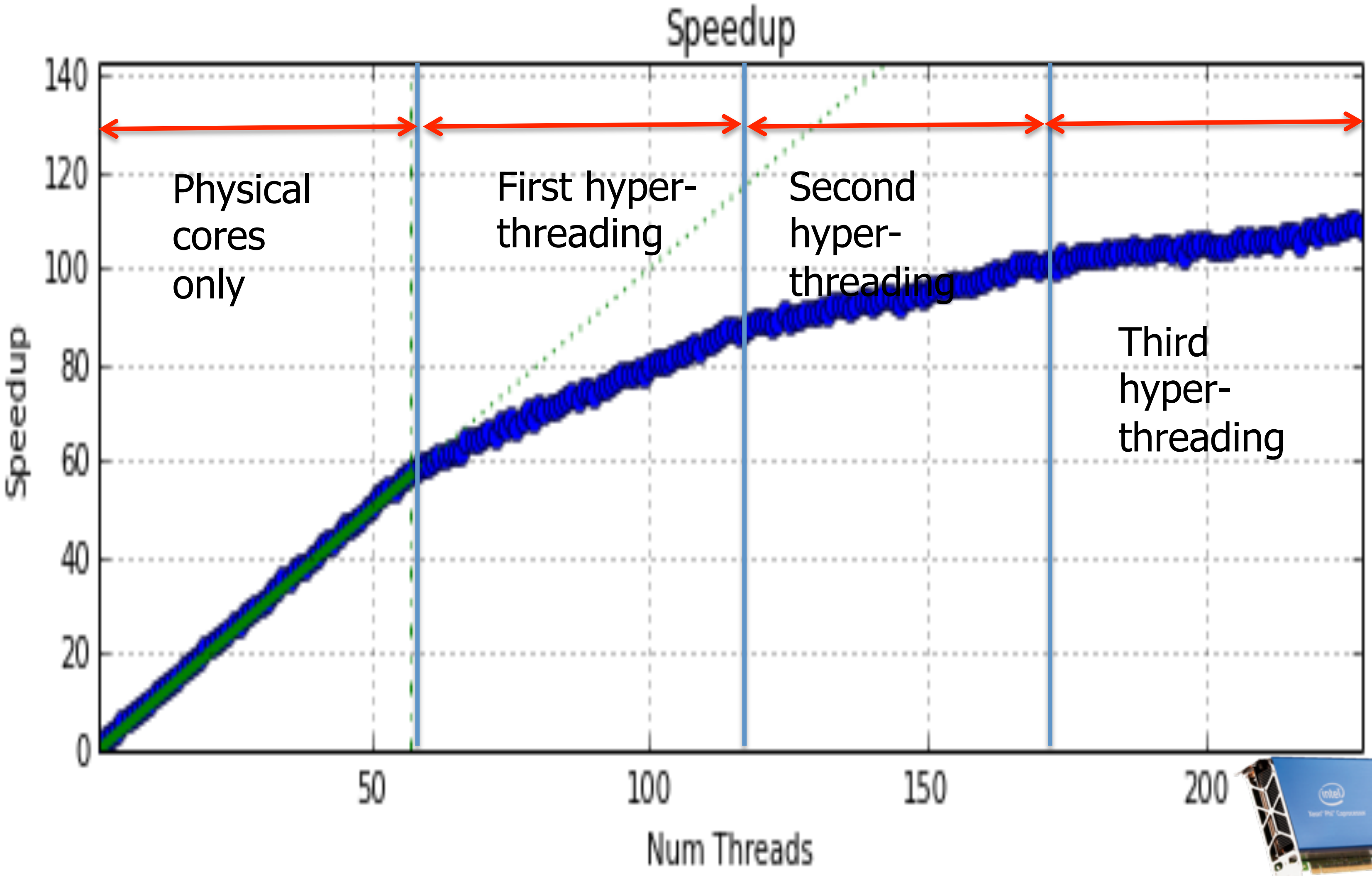
Active cores

Memory consumption on Intel Xeon Phi



Intel Xeon Phi™ 3120A

Scalability on Intel Xeon Phi



Physical cores only

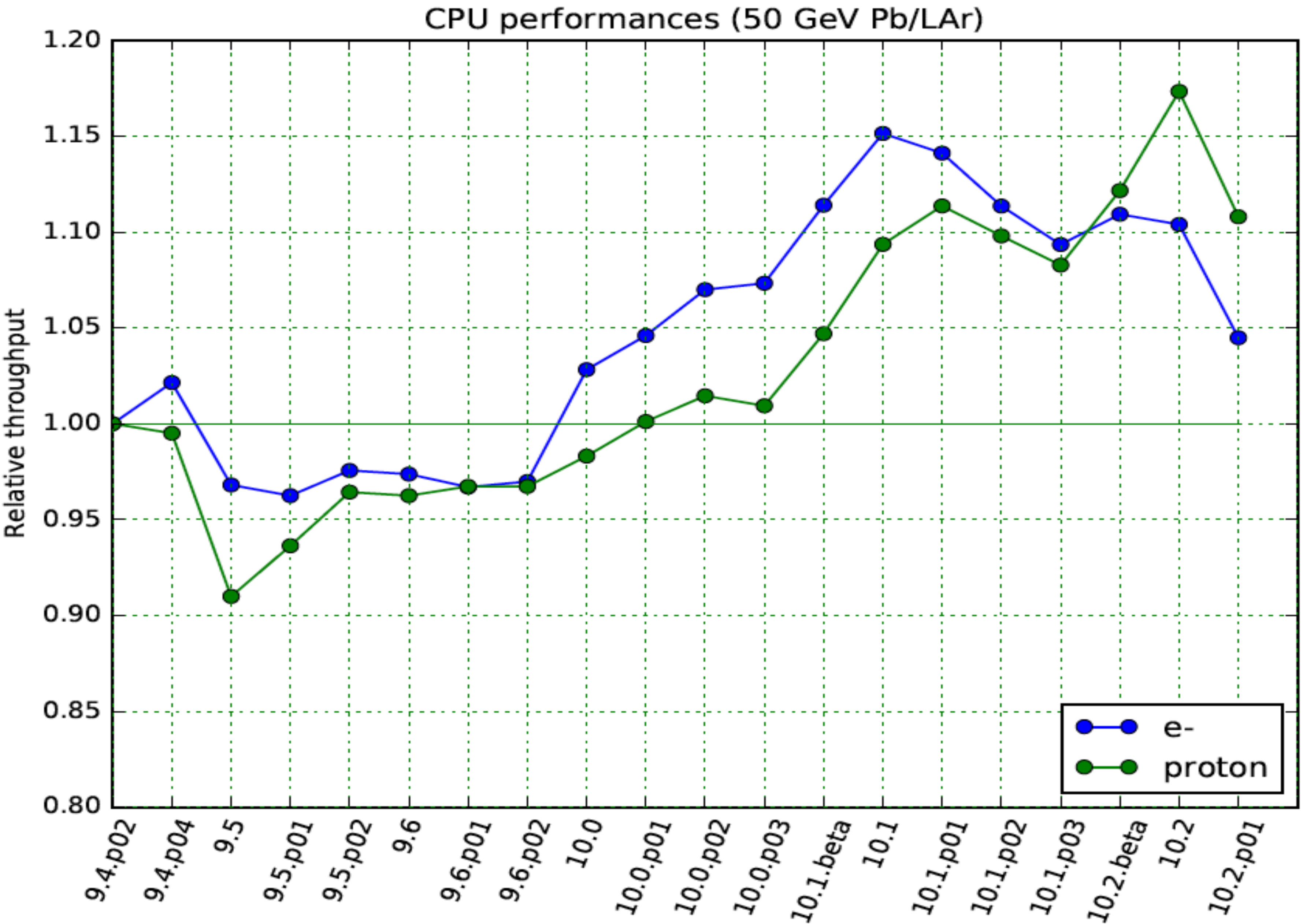
First hyper-threading

Second hyper-threading

Third hyper-threading

Intel Xeon Phi™ 3120A

Throughput in sequential mode

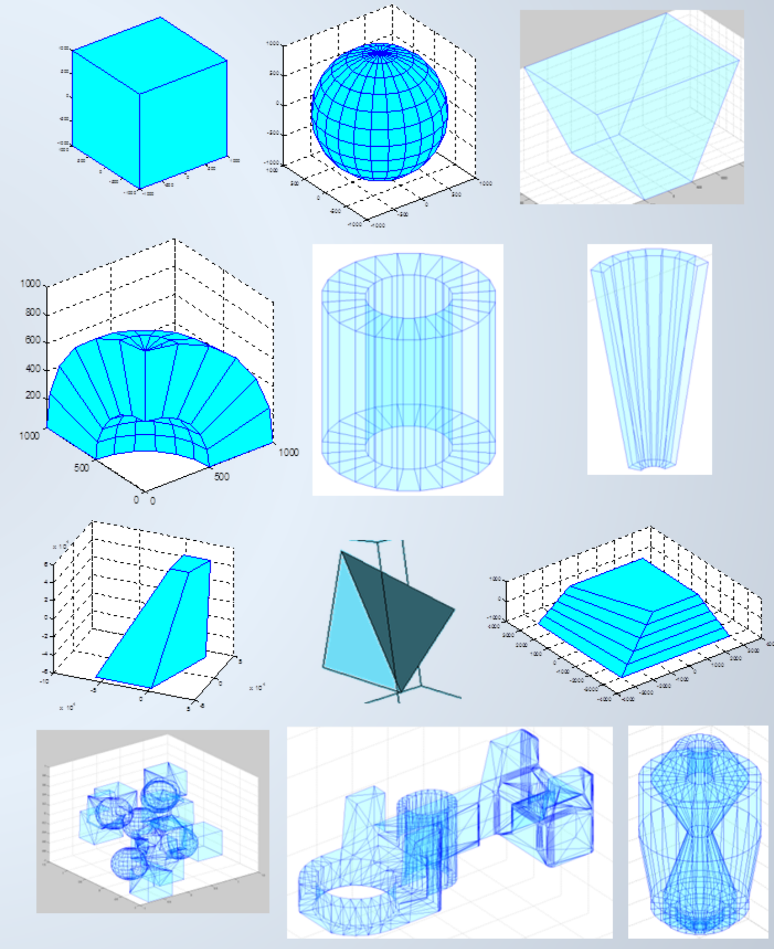


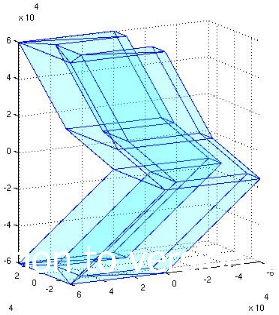
User's code migration

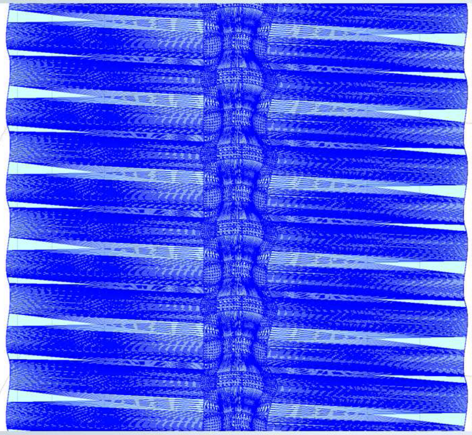
- If you have a running code with version 9.6 and you want to stick to sequential mode, you do not need to migrate. It should run with version 10 series.
  - Except for a few obsolete interfaces that you had already seen warning messages in v9.6.
- Migration of user's code to multi-threading mode of Geant4 version 10 series should be fairly easy and straightforward.
  - Migration guide is available.
  - Geant4 users guides are updated with multi-threading features.
  - Most of the examples have been migrated to multi-threading.
  - Geant4 tutorials based on version 10 series have already started.
- G4MTRunManager collects run objects from worker threads and "reduces".
- Toughest part of the migration is making user's code thread-safe.
  - It is always a good idea to clearly identify which class objects are thread-local.
- Every file I/O for local thread is a challenge
  - Input : primary events : examples are offered in the migration guide.
  - Output : event-by-event hits, trajectories, histograms

New features in geometry

- New command-based geometry overlap checks
  - Built-in check with random points on surfaces
  - Resolution and tolerance are tunable
- Introduction of gravity field and magnetic field gradient
- Optional new AIDA Unified Solid library
  - Optimized implementation of a large number of primitives and constructs
  - For example, the time required to compute intersections with the tessellated solid was dramatically reduced.
- Several new shapes including extruded solid
- New multi-union structure
  - Replace multiple use of binary Boolean unions
  - Far better performance for combining more than O(10) unions





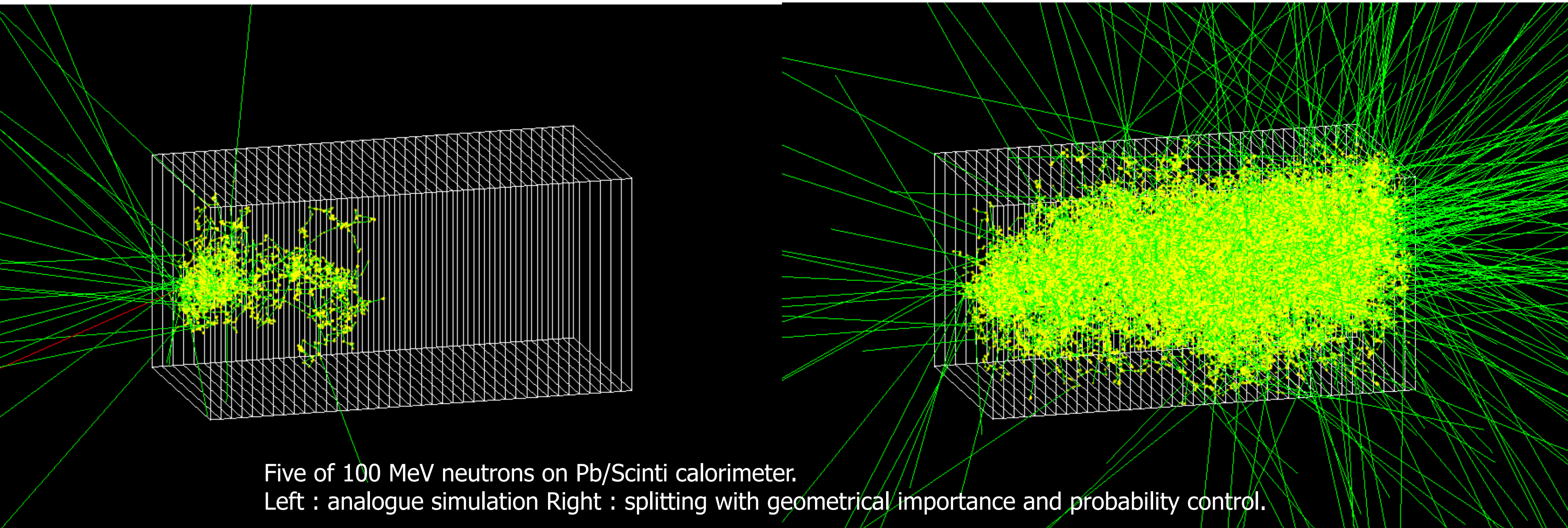


Method	Speedup
Inside	2423x
DistanceToIn	1334x
DistanceToOut	1976x

Information	Value
Number of facets	164.149
Number of voxels	100.000
Memory saved compared with original Geant4	22% (51MB)

New biasing scheme

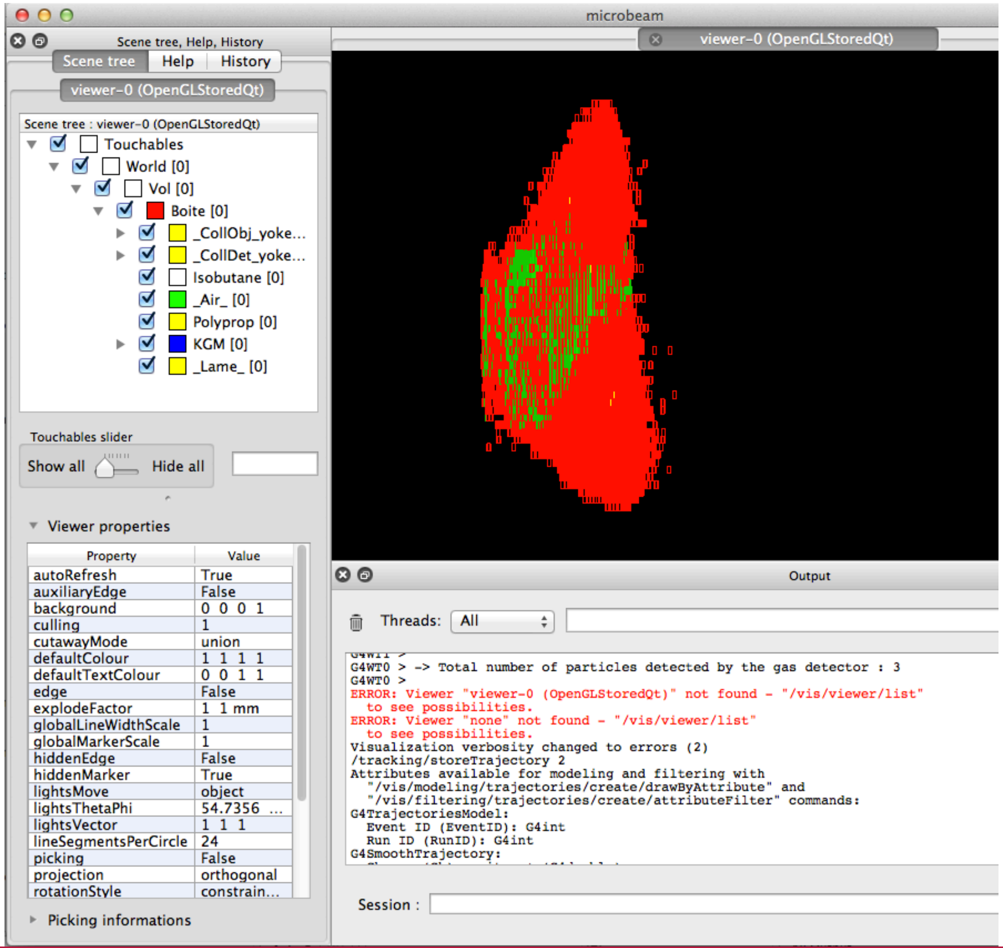
- Event biasing (a.k.a. variance reduction) scheme has been fully revised at version 10.
- It allows treating many biasing options in coherent manner.
- Such options include:
  - Physics process biasing : alters physics process
    - Cross-section biasing, forced interaction, forced passage, etc.
    - Biasing final products of an interaction, e.g. distribution
  - Non-physics biasing : alters the transportation of particle
    - Geometrical importance, splitting / Russian roulette, weight window, etc.
- Easily extensible to new (or user-defined) options
- Well-integrated with built-in scoring functionalities.
- New examples are available.



Five of 100 MeV neutrons on Pb/Scinti calorimeter.
Left : analogue simulation Right : splitting with geometrical importance and probability control.

New features in analysis, GUI and visualization

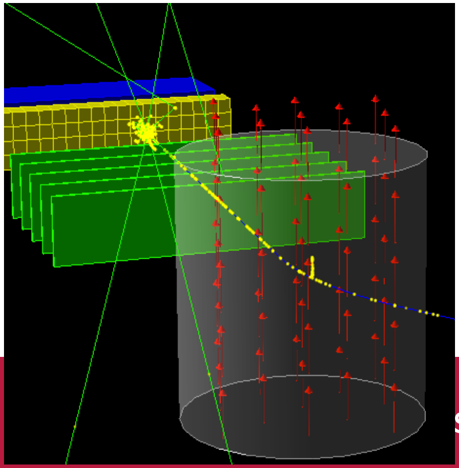
- New built-in fully-multithreaded histogramming tool
  - 1-D and 2-D histograms and scatter plots, n-tuples
  - Data format compatible with ROOT, XML, AIDA, CSV
    - Extensible to other format
- GUI and visualization
  - New Qt driver with OpenGL
    - Viewer properties and picking panel, dock-able widgets
    - Multithread output filtering
  - More than 30% faster drawing on OpenGL
  - Magnetic field lines



Property	Value
autoRefresh	True
auxiliaryEdge	False
background	0 0 0 1
culling	1
cutawayMode	union
defaultColour	1 1 1 1
defaultTextColour	0 0 1 1
edge	False
explodeFactor	1 1 mm
globalLineWidthScale	1
globalMarkerScale	1
hiddenEdge	False
hiddenMarker	True
lightsMove	object
lightsThetaPhi	54.7356 ...
lightsVector	1 1 1
lineSegmentsPerCircle	24
picking	False
projection	orthogonal
rotationStyle	constrain...

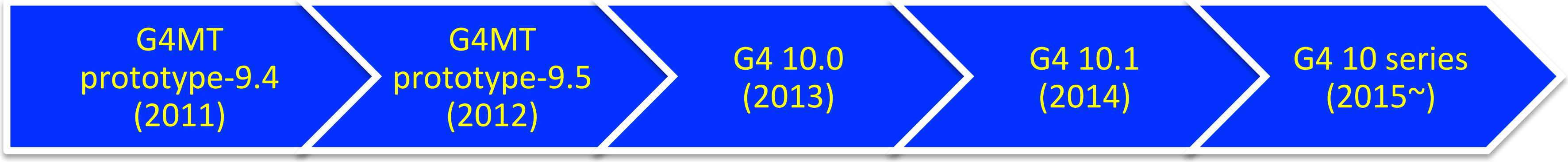
```
G4WT0 > -> Total number of particles detected by the gas detector : 3
G4WT0 >
ERROR: Viewer "viewer-0 (OpenGLStoredQt)" not found - "/vis/viewer/list"
  to see possibilities.
ERROR: Viewer "none" not found - "/vis/viewer/list"
  to see possibilities.
Visualization verbosity changed to errors (2)
/tracking/storeTrajectory 2
Attributes available for modeling and filtering with
  "/vis/modeling/trajectories/create/drawByAttribute" and
  "/vis/filtering/trajectories/create/attributeFilter" commands:
G4TrajectoriesModel:
  Event ID (EventID): G4int
  Run ID (RunID): G4int
G4SmoothTrajectory:
```

Session :



Geant4 version 10 series

- Geant4 version 10 series.
  - Performance improvements (both in physics and computing)
    - E.g. Memory size required per thread for full LHC/CMS will be reduced to half 28MB (v10.0) → < 9MB (v10.2)
  - Missing functionalities yet to be migrated to multithread are arriving
    - E.g. Real-time visualization in multithreaded mode with v10.2
  - Additional APIs
    - E.g. Better "integratability" with TBB (Intel® Threading Building Blocks), and also
    - Smoother integration of MPI (Message Passing Interface) and MT, in particular for merging results
  - Additional functionalities
    - E.g. More biasing options
  - New physics
    - E.g. Multi-TeV hadronics, neutrino physics, channeling effects in crystal



G4MT prototype-9.4 (2011)

- Proof of principle
- Identify objects to be shared
- First testing

G4MT prototype-9.5 (2012)

- MT code integrated into G4

G4 10.0 (2013)

- Production ready
- Public release

G4 10.1 (2014)

- Memory reduction
- First optimizations

G4 10 series (2015~)

- Further refinements

Coming events

- Releases
  - Next release : early December 2016 (date t.b.c.)
    - Next beta release : end of June 2016
  - Patch releases for versions 10.0, 10.1 and 10.2 will be made as needed
- Tutorials
  - Southern Methodist University, Dallas, TX, May 16-19
  - CNRS Geant4 tutorial, LAL Orsay, May 23-27
  - XIII Seminar on Software for Nuclear, Subnuclear and Applied Physics, Alghero, Italy, June 5-10, 2016
    - [includes a Geant4 beginner course]
  - Massachusetts Institute of Technology, Cambridge, MA, June 20-23

Geometry

Geometrical primitives

- Updated USolids library now part of VecGeom
  - Optional replacement of original Geant4 solids
  - Selection made at configuration
  - External library VecGeom v.00-01-00
    - https://gitlab.cern.ch/VecGeom/VecGeom/tree/v00-01-00
  - Possibility to choose adoption of USolids (default) or VecGeom (under development) revised implementation
  - Added shapes (from VecGeom) since last release: Paraboloid, Torus

Geometry

Volumes, Navigation & Transportation

- Enabled parameterisation by solids type in MT mode
  - Feature not supported in previous 10 release series
  - Assumes solids being parameterised are declared thread-local in the user's parameterisation class and allocated just once
- Enabled volume divisions in MT mode
  - Addressing problem reports #1743 and #1750
- Revised implementation of *EstimateIntersectionPoint()* in *G4MultiLevelLocator* for keeping consistent candidate intersection
  - Better treatment of looping tracks in field and improved diagnostics
  - Addressing cases of negative steps and convergence problems with field observed in ALICE

Analysis & Persistency

- Analysis:
  - Added "batch plotting" facility
  - Added classes for management of users parameters
  - Added ability to send/receive histograms through MPI
- Persistency
  - New GDML schema version 3.1.3
    - Added 'userinfo’ field to allow 'global' auxiliary fields; extended auxiliary field by auxunit and pointer to sub-auxiliary fields with no limit on the number of levels
    - Added optional field ‘copynumber’ to physical volumes for specifying copy-numbers associated to normal volume placements
    - Implemented import/export of geometrical regions associated to volumes for importing and storing production cuts and user-limits
    - Added support for writing and reading copy-numbers associated to placements
    - Fixes issues of misplaced volumes in exported geometries of complex detectors
  - Re-enabled detector object persistency
    - Binary persistency using Root I/O with Root-6

Materials, Particles, Track

- Materials:
  - Corrected density effect parameterisation when density of a simple (one component) material differs from the nominal value
    - Addressing problem report #1765
- Particles:
  - Taking into account mass width in Phase Space Decay when daughter particles are resonant particles
    - Dynamic mass given to daughter particles according to the Breit-Wigner formula
  - Removed hard-coded state data for nuclides, now retrieved from the mandatory G4ENSDFSTATE data set
- Track:
  - New G4VAuxiliaryTrackInformation class to be associated to a G4Track
    - Migrated generic biasing code to use G4VAuxiliaryTrackInformation for improved bookkeeping of tracks to be biased

Global, Run, Digits&Hits

- Global:
  - Added treatment of units for fluids (from CLHEP) in G4UnitsTable
    - liter, L, dL, cL, mL
- Run:
  - Increased granularity of MT methods to allow easier sub-classing of run-manager classes and ease integration with external frameworks
  - Added new UI commands for treatment of random numbers
- Digits&Hits:
  - Added G4MultiSensitiveDetector functionality, allowing to assign multiple sensitive-detectors to a single logical-volume
    - Calls to the sensitive-detectors methods are forwarded to all user-defined sensitive-detectors that are added

Visualization & Data sets

- Visualization:
  - Revised visualization system for multi-threading: now adopting dedicated thread for visualization, allowing also for continuous visualization of tracks during event generation
  - New UI commands specific for MT applications
- Data sets:
  - New versions: **`G4EMLOW-6.48`**, **`G4ENSDFSTATE-1.2`**, **`G4RadioactiveDecay-4.3`**, **`G4PhotonEvaporation-3.2`**
  - **`G4ENSDFSTATE`** data for nuclides is a mandatory data-set
  - New optional data set **`G4TENDL-1.0`** for high-precision incident particles

Configuration & Externals

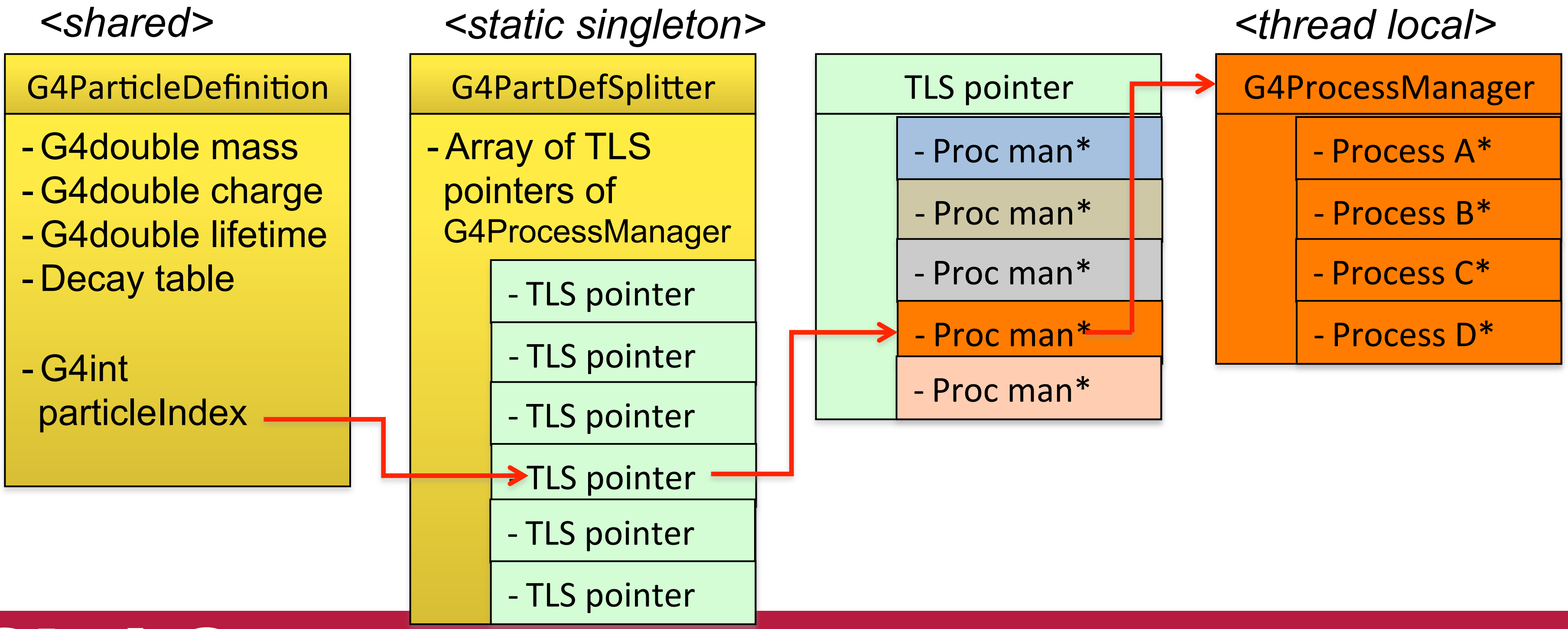
- Cmake:
  - Requiring CMake v3.3 or higher
  - Making use of CMake “Compile Features” to determine and setup the correct C++ standard to use
  - C++11 features now enabled by default
- CLHEP:
  - New version 2.3.1.0 or 2.3.1.1 required
  - Forcing use of C++11
  - New MixMax random engine implementing the "Matrix Generator of Pseudorandom Numbers”
- Updated zlib (1.2.8) and expat (2.1.0) external modules

Geant4 multi-threading : event-level parallelism

- This choice minimizes the changes in user-code
    - Maintain API changes at minimum
- All Geant4 code has been made thread-safe.
    - Thread-safety implemented via Thread Local Storage
- Most memory-consuming parts of the code (geometry, physics tables) are shared over threads.
    - "Split-class" mechanism: reduce memory consumption
        - Read-only part of most memory consuming classes are shared
        - Enabling threads to write to thread-local part
- Particular attention to create "lock-free" code: linearity (w.r.t. #threads) is the metrics we are concentrating on for the v10.0 release.

Split class – case of particle definition

- In Geant4, each particle type has its own dedicated object of G4ParticleDefinition class.
  - Static quantities : mass, charge, life time, decay channels, etc.,
    - To be shared by all threads.
  - Dedicated object of G4ProcessManager : list of physics processes this particular kind of particle undertakes.
    - Physics process object must be thread-local.



Geant4 multi-threading : event-level parallelism

- This choice minimizes the changes in user-code
  - Maintain API changes at minimum
- All Geant4 code has been made thread-safe.
  - Thread-safety implemented via Thread Local Storage
- Most memory-consuming parts of the code (geometry, physics tables) are shared over threads.
  - "Split-class" mechanism: reduce memory consumption
    - Read-only part of most memory consuming classes are shared
    - Enabling threads to write to thread-local part
- Particular attention to create "lock-free" code: linearity (w.r.t. #threads) is the metrics we are concentrating on for the v10.0 release.

Geant4 version 10 series

- Major release of version 10.0 was made on December 6th, 2013.
- One of the highlights of 10 series is its Multithreading capability.
  - Event-level parallelism with memory sharing over threads
  - Geant4 became the first HEP general-purpose software to be fully multithreaded
- 10.0 also included other developments, in particular:
  - Introduction of isomers
  - Alternative low energy neutron model G4LEND
    - Direct interface to GIDI (General Interaction Data Interface)
  - Consolidation of MSC in Electromagnetic physics
  - Decommission of obsolete physics models such as LHEP
  - Extension of event biasing capability to allow process occurrence and final state production biasing
  - Introduction of phonon transport with a new concept of crystal
  - New GUI/Vis built on Qt
    - Note: All these features are available with or without multithreading.
- Next release
  - 10.3 : December 2016 (planned)

Physics Lists

- Neutron elastic cross *G4NeutronElasticXS* used in all non-HP physics lists
  - Instead of Chips neutron elastic cross section
- Glauber-Gribov kaon inelastic cross sections used in all physics lists
  - instead of either Chips or Gheisha kaon inelastic cross sections
- In QBBC, transition between FTFP and BERT : [3, 4] GeV
  - Instead of [3, 12] GeV
- In FTFP_BERT_TRV, use new GS msc model; and transition between FTFP and BERT : [2, 4] GeV
  - Instead of [3, 12] GeV
- QGSP_BIC_AllHP uses ParticleHP for p , d , t , He3 , α below 200 MeV