Report from C++11 Task-Force

A. Dotti for the Geant4 C++11 Task Force ; SLAC SD/EPP/Computing

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Overview

SLAC

Geant4 Steering Board agreed to create a task-force, lead by Gabriele, to guide the migration to c++11/14 in view of version 10.2

Immediate goals:

- identify strategy for c++11 migration
- provide documentation and guidelines
- migrate system testing
- study possible issues

C++11

SLAC

Platforms we plan to support for version 10.2 are:

- OS:
  - SLC6 with latest compiler
  - Linux CentOS-7 (coming with gcc-4.8.2 vanilla)
  - MacOS Yosemite
  - Window 7 or 8 (or 10)
- Compilers:
  - **gcc-4.8.1 or greater**
  - **clang-3.5 or greater**
  - **icc-15 or greater**
  - **Visual-C++ 14 (Visual Studio 2015)**

Status of migration: cmake and system testing

SLAC

cmake is now able to recognize if the configured compiler supports c++11 and it check for needed features

- -std=c++11 flag is added automatically when needed

System testing has been migrated to c++11 compilers and OS:

- Linux SLC6 w/ gcc 4.8, 4.9, 5.1, 5.2, clang3.6
- Linux CentOs 6 w/ gcc 5.1 and icc 15
- Linux Ubuntu 14.04 w/ gcc 4.8
- Linux CC7 (CentOS 7, to replace SLC6): w/ gcc 4.8 and 5.1
- Mac OS 10.9 w/ clang 3.5 ; 10.10 w/ clang 3.6
- Xeon Phi: icc 15 (more work needed)

Documentation

SLAC

A minimal guide with suggestion has been prepared (Ivana):

http://geant4.cern.ch/collaboration/c++11_guide.shtml

- initial feedback from SB received, ready for public release

A twiki has been created to list issues and notes:

https://twiki.cern.ch/twiki/bin/view/Geant4/Cxx11MigrationTaskForce

Note: some features (parallelization and rng) should not be used directly by developers, instead continue use G4 wrappers

System support

SLAC

Compilers support is satisfactory and everything works as expected on Linux (gcc, clang, icc) and Mac (clang)

- Issues found with icc for Xeon Phi, input from Intel received, need update to mpss stack but workaround in G4 code found
- On Mac OS X stick to clang: alternative gcc is not working

Windows: Visual Studio 2015 should have full c++11 support (possibly allowing MT to finally work on WIN)

- initial tests with WIN 10 technical preview show good support for needed features (std::thread, thread_local)
- new collaborator joined (Brian Smith, Kromek) with some experience w/ WIN programming

On Linux w/ ICC observed a substantial slow-down (factor 2-3) due to CLHEP use of std::shared_ptr and thread_local: possibly linked to limitation observed on Xeon Phi, workaround exists

On Linux w/ ICC compilation did not succeed (internal error) due to a massive static std::vector with 4k non-POD elements in LEND lmodel, workaround was to split data into smaller chunks

Random Number Generation

SLAC

Initial report that new STL RNG engines are faster than CLHEP ones:

- e.g. MarsenneTwister 50% faster than CLHEP
- however need to better understand some compatibility and correctness of results, mt19937_64 failing standard tests, will not migrate until all issues are solved

Migration to std::thread and thread_local

SLAC

For multi-threading builds:

- migration from __thread to thread_local done
- for compilers that support it, in other cases use __thread
- notable exception is ICC for Xeon Phi in some cases, to be studied
- migration from pthread to std::thread to be evaluated if realy needed for 10.2
  - real advantage is the use for WIN, since this is not critical, we may postpone it to next year

Several tags with pure c++11 code have been already accepted.

For example visualization for MT uses std::thread

Conclusions

SLAC

Geant4 code inclusion of C++11 features is well under-way

Ready for some features to be widely adopted for new developments

We'll evaluate, case by case, migration of legacy code to c++11 based on effectiveness: e.g. possible speedup with RNG, std::thread

- expect activities to continue in 2016 and possibly beyond 10.3