Open requirements: LHC and other HEP energy frontier experiments

Mihaly Novak (CERN, EP-SFT)

23rd Geant4 Collaboration Meeting

27-31 August 2018, Lund, Sweden

OUTLINE

- **Status and open requirements:**
  - ATLAS: John Chapman, Heather Gray, John Apostolakis
  - CMS : Vladimir Ivantchenko, Daniel Elvira
  - ALICE : Ivana Hrivnacova, Andreas Morsch, Sandro Christian Wenzel
  - LHCb : Gloria Corti, Dominik Muller, Michal Kreps, Philip James Ilten, Matteo Rama

ATLAS

ATLAS: current production

- **13 TeV MC production (MC16) well underway now**
  - default production release uses G4 10.1 patch03, CLHEP 2.2, 64-bit, gcc 4.9, SLC6, C++14 (Some samples produced with later releases built using gcc 6.2.)
  - compiling G4 as part of their nightly builds
  - significant number of updates to ATLAS user code (geometry and detector response), including several speed ups
  - HPCs, Amazon cloud, BOINC in use; Testing underway for icc, clang, and ARM builds. Could be used for production if they prove useful
- **Over 11.1B events simulated so far with this configuration!!!**
- **MC16 will be the main campaign for 2018**
- **Will soon move MC16 Simulation to use Geant4 10.1.patch03.atlas05 which contains G4Solid improvements from G4 10.4. This gives a 4% speed-up in their simulation.**
- **Still running tails of (much) older production campaigns, MC15:**
  - Geant4 9.6 patch03, CLHEP 2.1, 64-bit, gcc 4.7, SLC6, C++11
  - No updates requested from analysers here for a number of months now though.
- **The next MC campaign (preparing for LHC Run 3) will most likely use Geant4 10.3 (outside possibility of using Geant4 10.4 - depending when the campaign starts).**

ATLAS: requirements I.

- **Identical output between serial and MT runs of Geant4:**
  - difference seen in safety between serial and MT runs (ATLASSIM-3630)
  - observed different results of safety calculation by serial and MT jobs for the same event
  - reproducible with latest G4 release (10.4-p02)
- **Enabling use of hadronic process(es) in MT inside external application (Fast Simulation) SIM-607**
- **Continued support for Geant4MT**
- **Enable ‘true’ zero decay time for pre-defined decays, for use in B-Bbar oscillation:**
  - currently if t=0, it is assumed that it is not set
- **Issues found in how G4Commands are processed by the G4UIManager in G4 10.1.:**
  - would be good to get the patches for this into whichever release they next use for production
  - would like confirmation that their **suggested workarounds** on the Athena side are reasonable too

ATLAS: requirements II.

- **Continued development of G4Solid library:**
  - have observed 3% (5%) to full (fast) simulation already from this work
- **Continued improvement of hadronic shower simulation:**
  - FTFP_BERT_ATL in Geant4 10.1.patch03 is an improvement over FTFP_BERT in Geant4 9.6, but the agreement with data is still not as good as they had with QGSP_BERT in Geant4 9.4.
- **“Looping” tracks killed by G4Transportation:**
  - need to re-appraise the rate of such tracks after the recent (ATLAS)G4Geometry & G4Solid improvements
  - may then have some requests about the circumstances under which such tracks are killed (or not)
- **Small step issue is still hanging around in G4 10.1.:**
  - they want to re-check the situation after improvements (and with newer G4 versions), and
  - try out some of the new options for the navigator in the G4 10.5 beta
- **Continued support of AtlasRK4 stepper (SIM-697)**
  - aim to trial G4Nystrom stepper in Athena too

CMS

CMS: status of CMS full simulation

- **CMS used Geant4 10.0p02 for 2015-2016 Run-2 simulation:**
  - > 16 billion events are produced
  - sequential Geant4 in production
  - QGSP_FTFP_BERT_EML Physics List
- **Geant4 10.2p02 is used for 2017 Run-2 simulation production:**
  - > 10 billion events are produced
  - minor fixes from Geant4 10.2p03 are added
    - FTFP configuration from Geant4 10.1 is used
  - FTFP_BERT_EMM PhysicsList
  - multithreaded Geant4
  - recently two extra fixes were backported
- **Geant4 10.4 is adopted as a production version for 2018 MC production for Run-2**
  - relevant known fixes are added on top of 10.4 as CMS private patches
  - the MT mode, production platforms:
    - slc6-amd64-gcc700
    - slc7-amd64-gcc700

CMS: status of CMS full simulation

- **New features implemented in CMS full simulation for 2018 production**
  - MixMax random number generator for both hits and for digitization
  - VecGeom external geometry library
  - new Geant4 stepper for computing of trajectories in magnetic field G4DormandPrince475
  - smart tracking in magnetic field option allowing faster computations for low-energy changed particles
  - **overall simulation speedup is ~30%**

CMS: plans for LS2

- **Target Geant4 version for Run-3:**
  - the plan is to use the version of December 2019
- **CMS has interest to all variants of performance improvements in G4:**
  - improvements to EM and Had physics models and physics lists
  - improved geometry and tracking using R&D work on Geant4 and GeantV
    - faster transportation(s)
    - faster navigator
  - FastSim may benefit from interfaces to ML tools
    - at least, on level of running examples
- **Test-beam activities toward new forward calorimeters will be increased in order to have final design**
  - close cooperation, to understand test-beam results, will be required
- **CMS full simulation may be performed at HPS**
  - optimizations of Geant4 for this use-case is required
- **Status of migration to DD4Hep:**
  - CMS has uncovered serious issues related to dependences, thread safety, and duplication (DD4Hep interoperability with the CMSSW framework, VecGeom, TGeo)
  - CMS will produce a requirements document to be shared with developers
  - CMS will meet with tool developers (DD4Hep, VecGeom, Geant4, Root) to discuss potential solutions to address the requirements

LHCb

LHCb: future simulation needs

- **LHCb Upgrade for Run3[1]:**
  - full software trigger with high signal purity
  - analysis directly on trigger output
  - simulation will be dominating the CPU needs, even more than today
  - MUST speed up the simulation:
    - by implementing faster or parametrized simulations
    - by reducing the CPU consumption of the detailed Geant-based simulation

    while maintaining high quality physics modeling
- **Run3 is the occasion to modernize the whole LHCb software**
  - new Multi-threaded Gaudi framework
    - better use of multi-processor CPUs
    - reduce memory usage
    - optimize cache performance
    - remove dead code
    - modern data structures
    - enable code vectorization
    - enable algorithmic optimization
- **In the simulation software also exploit more modern features in HEP-wide libraries used**
  - 'newish' Multi-threaded Geant4 (same benefits as above)
  - thread-safe generators, e.g. HepMC 3

[1] C. Bozzi, "Challenges for LHCb trigger", WLCG & HSF Workshop, Naples, March 2018
R. Matev, "Real-time analyses - the LHCb case" , WLCG & HSF Workshop, Naples, March 2018

LHCb: from current to future Gauss

- **LHCb is developing Gaussino as an experiment-independent framework built on top of modern, multithreaded and functional Gaudi[1]:**
  - a working prototype for an interface to multithreaded Geant4 within Gaussino is currently being extended to run the LHCb geometry
  - the G4 worker is running in separate threads to allow maximal flexibility. Implemented a modified run manager and Gaudi-thread <--> G4 thread exchange
- **The future Gauss will be build on top of Gaussino with the LHCb specific configurations**

[1] D. Muller, "Adopting new technologies in the LHCb Gauss simulation framework", CHEP 2018, Sofia, July 2018

LHCb: Gaussino and requirements for Geant4 MT

- **LHCb wants to be able to manage the event loop and flexibly control the handover of particles to G4. Ideally they would like to easily configure this using Gaudi configurables at run time.**
- **Flexibility is also very important. LHCb would like to be able to seamlessly change between '1 generated event in Gaudi handed to one worker thread' and 'break up one generated event and assign to many G4 threads':**
  - the latter is especially interesting when looking at opportunistic resources such as very short jobs during lower load on our trigger farm

- **Geant4 and timing:**
  - most of the time to produce the majority of the simulated samples in full simulation is spent in Geant4
  - LHCb has been validating new productions with Geant4 10.4 and exploring faster ways to use it:
    - clear indication of a significant speed-up moving from Geant4 9.6 to 10.4 combined with the move from gcc 4.9 and 6.2.
    - no indication of speed-up using new VecGeom geometry package. Trying to understand why that is the case: mostly simple shapes?
    - **Can any improvement for simple shapes be provided?**

LHCb: fast simulation of the Calorimeter system

- **A generic infrastructure has been developed in the Geant4 interface of the current Gauss framework to allow to interrupt the G4 transport for a given particle and inject a given fast simulation counterpart:**
  - makes use of Geant4 utilities (e.g. G4Region)
  - create the same calorimeter cell hits with identical format as those produced from Geant4
  - works with any fast implementation – also for detectors other then calorimeter
- **Two fast parametrization solutions currently under development[1]:**
  - classic Frozen Shower Libraries
  - hits generation based on Generative Adversial Networks (GAN)
  - … not necessarily mutually exclusive
  - **Require continuing support for fast simulation integration**

[1] M. Rama, "Fast Calorimeter Simulation in the LHCb framework" , CHEP 2018, Sofia, July 2018
V. Chekalina, "Generative models for fast Calorimeter simulation: LHCb case " , CHEP 2018, Sofia, July 2018

ALICE

ALICE: requirements

- **ALICE switched to FTFP_INCLXX physics lists:**
  - the FTFP_BERT list, which is the recommended physics list for hadronic calorimeter simulations, shows much less light nuclei (d, t, 3He) which come from secondary particles
  - for this reason, ALICE switched to FTFP_INCLXX physics lists for MC prductions, what however brings a performance penalty:
    - ALICE would like to get the light nucleon production in FTFP_BERT improved
    - and keep the support for FTFP_INCLXX physics list, until they can switch back to FTFP_BERT
- **Support for "sub-event" parallelism across G4 threads:**
  - since they have very big events it would be very valuable to get one event done quicker
- **ALICE likes to benefit from VecGeom developments:**
  - it would be nice having G4 interface with VecGeom navigation, the same way possible for TGeo
- **ALICE is interested in tests with monopole physics:**
  - As the monopole physics is provided only in a Geant4 example, they have to duplicate the classes (G4*) from the Geant4 example in Geant VMC. This duplication could be avoided if these classes would be provided within the Geant4 source.

QUESTION?

BACKUP

- Most of the time to produce the majority of the simulated samples in full simulation is spent in Geant4
- LHCb has been validating new productions with Geant4 10.4 and exploring faster ways to use it
  - Clear indication of a significant speed-up moving from Geant4 9.6 to 10.4 combined with the move from gcc 4.9 and 6.2.
  - No indication of speed-up using new VecGeom geometry package. Trying to understand why that is the case: mostly simple shapes?

Geant4 v10.3.3 and VecGeom v0.3.rc.
Tested with:
- Scalar backend + SSE4.2
- Scalar backend + AVX2
- Vc backend + AVX2

Can any improvement for simple shapes be provided?

Shapes used in LHCb for current detector

Shapes	Quantities
G4Box	2219
G4Cons	122
G4Polycone	20
G4Sphere	133
G4Trap	399
G4Trd	18
G4Tubs	1490