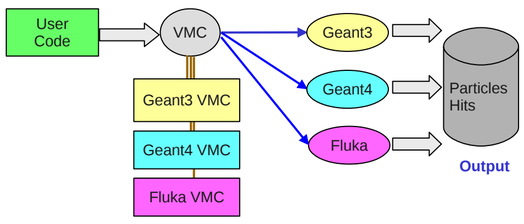
The Geant4 Virtual Monte Carlo

I. Hřivnáčová

Institut de Physique Nucléaire (IPNO),
Université Paris-Sud, CNRS-IN2P3

Virtual Monte Carlo (VMC)

- VMC has been developed by the ALICE Offline project in the close collaboration with the ROOT team
- Since 2002 the VMC is distributed with ROOT
  - http://root.cern.ch/root/vmc/VirtualMC.html
- Now it is in use in more experimental frameworks



Thanks to an abstract VMC layer to the MonteCarlo transport codes, the same user application code can be run with different simulation programs

Geant4 VMC

- Presented already at previous CHEP conferences (2003, 2007, and in the context of ALICE, 2010), here we concentrate on new features, the multi-threading prototype and the performance test
- Documentation:
  - http://root.cern.ch/drupal/content/geant4-vmc

Releases:

- New tag with ROOT releases if needed and with each Geant4 release
- In general, the tagged Geant4 VMC version can be used
  - with the ROOT version which it was tested with and the higher ones
  - with the Geant4 version which it was tested with including its patches (but not with the higher ones)
- Patch versions:
  - Only fixes are applied to the based version, usually maintained for the last two versions (based on the last two versions of Geant4, actually 9.4.x and 9.5.x)

New Features and Developments

Non physics

- New class TG4FieldParameters allowing a user customization of the Geant4 magnetic field integrator and the accuracy parameters
  - See an example of new commands in E02/g4config.in
- A new utility class, TG4ParticlesChecker, for comparing particles properties defined in ROOT and Geant4
- Implemented a possibility to select sensitive volumes by the user; in this case TVirtualMCApplication::Stepping() function is called only when a track is located in a sensitive volume.
  - Can speed up a user application, especially when the accounting of MC informations happens in few volumes
- Implemented new TVirtualMC functions for drawing tracks from ROOT: SetCollectTracks(..), IsCollectTracks()
- New, faster implementation of the VMC functions: VolId(..), VolName(..), GetMediumID()
- Passing the random number seed from TRandom in CLHEP::HepRandom

Physics Selection

- Closer integration of G4PhysListFactory: the available hadron physics lists names are now taken directly from Geant4
- New TG4ExtraPhysicsList allowing a user selection of the following optional Geant4 builders: G4ExtraPhysics, G4OpticalPhysics, G4RadioactiveDecayPhysics
- New TG4EmModelPhysics allowing a user selection of an extra EM energy loss and fluctuations models.
- The messenger class for user customization of optical physics was adopted in Geant4 9.5 (and could have been removed from Geant4 VMC)
- New class TG4CrossSectionManager allowing a user inspecting of hadronic cross sections
- Included light anti-ions (anti-deuteron, anti-triton, anti-alpha, anti-He3) and added a test (in E03)
- A new test for processing all available G4 physics lists

Geant4 Physics With VMC

- Geant4 VMC defines a composed physics list (derived from G4VUserPhysicsList) which is a a composition of three physics lists
- Users can then select the physics list configuration via a string option in their configuration macro

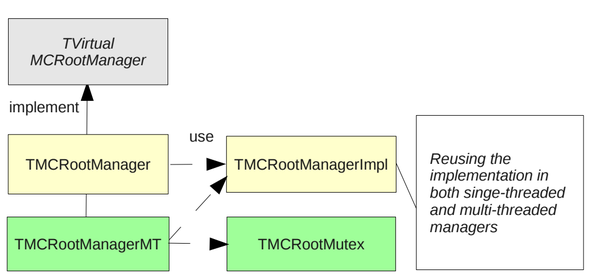
G4 reference physics list	+	Extra Physics	+	Special Physics (VMC features)
Selection of G4 reference list via G4 factory:		Selection of extra builders provided In G4:		Selection of special VMC builders::
FTFP_BERT FTFP_BERT_EMV LHEP QGSP QGSP_BIC QGSP_BIC_EMY		+extra +optical +radDecay Supported builders: G4ExtraPhysics G4OpticalPhysics G4Radioactive DecayPhysics		specialCuts +stepLimiter +stackPopper The VMC builders (TG4*Physics): SpecialCuts StepLimiter StackPopper UserParticles ExtDecayer ProcessMCMap EmModel

Geant4 (VMC) MT

- Geant4 MT prototype – first public release on 31 October 2011, based on 9.4.p01
- Geant4 VMC MT - the same approach as in Geant4 MT
- Singleton objects -> singletons per thread
  - Both TGeant4 and UserMCApplication are instantiated per each thread
- New function in TVirtualMC to initialize multi-threading
  - TVirtualMC::InitMT(Int_t threadRank)
  - Available in the ROOT development version, to be included in v5.34/00
- Volume IDs can be taken directly from Geant4 logical volumes ID (which exist only in the MT version)
- Special care was needed in the use of G4Allocator (in TG4TrackInformation), which has to be declared thread local
- New classes for the application multi-threading management: G4ParRunManager, G4ThreadManager
  - Developed according to Geant4 MT examples
  - Not specific to Geant4 VMC – included in Geant4 MT
- Geant4 VMC MT application
  - Main program linked with all other packages libraries (ROOT, Geant4, ...) ,
    - what is different from a usual VMC application which is run from the ROOT main
  - It handles instantiation of threads (via included G4ParTop.icc)
  - Example E02 as an equivalent of ParN02 and ParN02Root
  - Its simulation time on 4 cores CPU scales with the number of threads in a similar way as ParN02 or ParN02Root

Geant4 VMC Multi-Threading (MT)

The classes for ROOT output management in Geant4 VMC:



- The Geant4 VMC MT prototype based on the Geant4 MT branch is available in the SVN development branch: geant4_vmc_mt
- First tagged version is foreseen after the public release of Geant4 MT based on Geant4 9.5.x

Geant4 (VMC) MT + ROOT Output

- Geant4 MT introduces parallelism per event
- The ROOT output is introduced per threads: each thread opens and writes on its own ROOT file
- No need for a final merge: user analysis can chain files
- Geant4 MT + ROOT output:
  - New example ParN02Root in Geant4 MT: adds ROOT output to ParN02
  - Added classes: RootManagerMT, RootMutex
  - Use of TThread, locking ROOT IO only till first TTree::Fill() in each thread
- Geant4 VMC MT + ROOT output:
  - TMCRootManager* classes included in E02 (but not specific to E02)

Test Application

- Geant4 extended example analysis/A01 rewritten in VMC
  - Better reflects the real applications than the existing VMC examples introducing four sensitive detector types (associated with six volumes) and geometry including placements with rotations and replicas
- Modifications in the Geant4 example:
  - Local magnetic field changed to a global field with 0 value outside the volume with the field, as local fields are not supported in VMC
  - PVParameterised volume changed to PVReplica in order to be able to run the example with all supported geometry options as parameterised volumes are not supported in VGM used in Geant4 VMC for geometry conversions between Root and Geant 4
- Example setup:
  - The VMC example is run with the original Geant4 geometry (using the geometry option “geomGeant4”)
  - FTFP_BERT physics list with 1 mm range cuts
  - Primary generator with periodically changing particles between e+, mu+, pi+, K+ and proton with fixed momentum
  - Measured time of a run with 100 events with 100 primary particles each
- Platform:
  - Geant4 9.5.p01, Root 5.32.03; Geant4 VMC trunk (r. 607) @ Fedora Core 14, gcc 4.5.1 , Intel Core i7

Performance Test

The purpose of this test was to evaluate the overhead of the VMC layer comparing to a Geant4 native application

Results

		Time [s]	Time/ Ref. time I	Time/ Ref. time II
G4 local field		97.93	0.81	0.70
G4 global field	(Ref. I)	121.43	**1.00**	0.87
VMC	(Ref. II)	139.04	**1.15**	1.00
VMC storing hits		141.23	1.16	1.02
VMC storing stack		178.37	1.47	1.28
VMC storing all		179.08	1.47	1.29

- The overhead of VMC in the tested example is **around 15 %**
- 2% time added when storing hits
  - The transient data representation is identical with their persistent representation (in the ROOT framework)
- 28% time added when storing particles
  - The transient representation (in the Geant4 framework) is different from their persistent one (ROOT framework)
  - It should be noted that the A01 VMC example is using the Ex03MCStack with no optimizations which stores all secondary particles
- The full VMC application is by 47% slower than the original Geant4 application, but a greater part of this slow down is caused by the added output and not the VMC interface.

International Conference On Computing in High Energy and Nuclear Physics 2012, New York, United States, 21-25 May 2012