G4 Computing Performance Summary

V. Daniel Elvira

Fermi National Accelerator Laboratory

Computing Performance Topics

- Time profiling of G4 applications
  - Using FAST
- Profiling memory in hadronic models
  - Using IgProf
- A look at Geant4MT performance
- G4 performance improvement opportunities
- G4 performance at the LHC experiments

Profiling Geant4 Applications

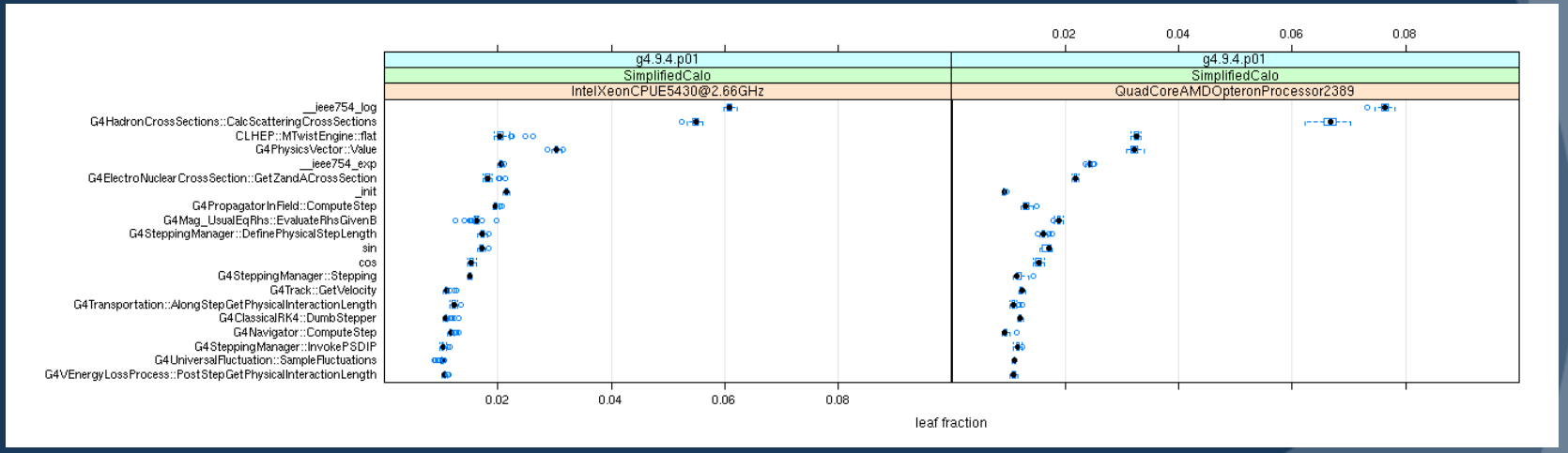
K. Genser (FNAL) using FAST tool from J. Kowalkowski, M. Paterno (FNAL)

Goal: Routinely evaluate new releases of G4 using “representative” G4 applications, make results publicly available.

Tool: FAST (Flexible Analysis and Storage Toolkit) has components for the collection, analysis, and display of the performance data.

FAST is available at https://cdcvs.fnal.gov/redmine/projects/fast

Applications: simpleCalorimeter, CMS, Mu2e using QGSP_BERT



Leaf count fractions for top functions

Profiling Geant4 Applications

K. Genser (FNAL) using FAST tool from J. Kowalkowski, M. Paterno (FNAL)

Geant4 version	SimplifiedCalo	cmssw426	Mu2e109
9.4	√	√	
9.4.p01	√	√	√
9.4.p02	√	√	
9.4.r06	√		
9.5.b.01	√		

- Preliminary profiling results are available at:
  http://oink.fnal.gov/perfanalysis/g4p (the location is likely to change)

Eventually define "protocol" to be run by shifters
(applications, input datasets, standard plots, storage, flag anomalies for experts to look at)
Extend to profile memory (IgProf, Valgrind)

Profiling Mem in Hadronic Models

M. Kelsey (SLAC) using IgProf

Recent Analysis

Reduced allocation footprint of Bertini from 120 MB (9.3) to 3 MB (9.4) in benchmark job (1,000 proton-nucleus interactions)

Recently applied **IgProf** to **FTF** (using Bertini for cascade propagation) and to **G4PreCompoundModel** (for de-excitation after Bertini)

- **FTF** adds 90 MB allocation footprint in benchmark job
- **PreCompound** adds 50 MB allocation footprint in benchmark job

Profiling Mem in Hadronic Models

M. Kelsey (SLAC) using IgProf

Significant Inefficiencies

Memory Checks

Analyzing **9.4-ref-07** release

G4VParticipants 64 kB per interaction *Eliminated*

G4Fancy3DNucleus 47 kB per interaction *Eliminated*

G4ReactionProductVector 10 kB per interaction *Reduced 90%*

G4VFermiFragments, G4FermiFragmentsPool 3.5 kB per interaction

G4DiffractiveSplitableHadron 2.5 kB per interaction

G4HadFinalState, G4HadSecondary 2 kB per interaction *Eliminated*

G4KineticTrack(Vector) 0.5 kB per interaction

All due to vectors of pointers, and vector objects being created and deleted on every interaction

Profiling Mem in Hadronic Models

M. Kelsey (SLAC) using IgProf

Recommendations

Memory Checks

Stay away from the devil!

- ⋆ Systematic use of **IgProf** or other memory profiling tool to identify inefficiencies
- Class data members for vectors, lists, etc. to allow reuse
- Vectors of objects, not pointers, to reuse allocations
- **G4Allocator** model for pointers in "exposed" containers
- Improve constness of interfaces
- Clarify ownership of returned pointer objects/containers

A look at Geant4MT

Ph. Canal (FNAL) using code from G. Cooperman & Xin Dong (NEU)

- Study memory changes in N02 and CMS
  - No significant gain in N02, 140 Mb of resident memory savings per thread/ process out of 170 Mb used per single thread
- Instability when number of threads increases
  - On a 4 cores CPU, at 12 threads, segfault half the times

  FIXED – Need to be tested
- Lower CPU performance
  - 25%-35% slower than regular G4
- Requires non trivial changes to user code

Geant4MT saves memory, slower at the moment, for full debugging needs a set of examples providing full code coverage

Geant4 Performance Opportunities

Ph. Canal (FNAL) using simplifiedCalo example by A. Dotti (CERN)

The simplifiedCalo example tests shower shapes using selected calorimeter setups – 7 GeV neutrons

- Largest fraction of the time spent in log/exp during init
- G4HadronCrossSection::CalcScatteringCrossSection next largest contributor (18% of DoEventLoop)
- 1% of time spent in 'IsApplicable' routines
- 1% of time spent in: G4HadronCrossSection::GetParticleCode
- G4hPairProductionModel:: ComputeDMicroscopicCrossSection and G4HadronCrossSections::CalcScatteringCrossSections take RunInit and DoEventLoop respectively, called many times with few distinct inputs

Single threaded Geant4 presents no clear hot spot, some improvement opportunities left, structural changes needed for significant gains

Geant4 Performance in ATLAS

J. Apostolakis (CERN)

Large production campaigns, computing performance limiting factor, more demand as luminosity increases

- Move to G4.4.9 in production
- Revision of geometry, added dead material
- Chose to remain with QGSP_BERT
- CPU reduced by 15% by moving to G4NystromRK4 stepper
- Shower parameterization validated for production (3-5 speed up factor)
- Did not profile simulation recently

Geant4 Performance in CMS

J. Apostolakis (CERN) for S. Banerjee (Kolkatta)

Large production campaigns, computing performance limiting factor, more demand as luminosity increases

- Use same version of CMSSW 440pre7 and build it with G4.9.4 with CMS specific patches, G4.9.4p02, G4.9.5.b01
- Use (slc5_amd64_gcc434) on a 8-core Intel® Xeon® E5410 2.33GHz CPU
- Many sets of events tried: QCD, Z(ee), ttbar, minbias
- Chose to migrate to with QGSP_FTFP_BERT_EML
- CPU performance stable at ~15s/evt (minbias), ~92s/evt (ttbar), ~55s/evt (Z), ~310s/evt (QCD multi-Tev jets)