G4 Computing Performance Task

V. Daniel Elvira (Fermilab)

G4CPT Charge

Draft0 version of the charge emerged from conversations with Geant4 developers expected to evolve with user input

(a) Profiling to identify bottlenecks in Geant4 based on main stream HEP applications. We need to discuss profiling tools, what we want to measure, metrics. EM, Geometry and hadronics are the areas more involved in CPU usage.
(b) Code reviews geared towards improving computing performance and coding practices.
(c) Establish computing performance activities with the medical and space G4 communities.
(d) Identify issues in multi-core-multithread G4.

Today: discuss potential activities and priorities in general, and focus on HEP.
Next meeting: focus on space and medical applications.

Questions to Answer Today

- Any general comments on the charge draft?
- What type of activities should G4CPT undertake?
- Would we benefit from more regular profiling? Public, candidate, reference releases?
- What information? How do we publish/share it?
- What tools? Valgrind (ATLAS, G4), IgProf (CMS, ATLAS?), SimpleProfiler (G4 FNAL team), Perfmon (ATLAS), others?
- What G4 packages would benefit from code reviews?
- How frequently should the G4CPT meet? What format?

Tips for improving CPU Performance of programs using Geant4
https://twiki.cern.ch/twiki/bin/view/Geant4/Geant4PerformanceTips

Today Focus on HEP Applications

Report from ATLAS (A. Dotti).
ATLAS recently went through a Detector Simulation Performance Assessment exercise which produced a document.

Report from CMS (P. Elmer).
Peter leads a CMS computing performance effort which has delivered significant CPU and memory gains for both the CMS simulation application and Geant4.

Issues reported by ATLAS/CMS during the G4 users workshop (October09, Catania)
Migration to the different flavors of the QGSP Bertini physics lists resulted in significant time/event size growth, as well as memory usage at runtime.

I leave the details to the experts… (talks by Peter, Andrea)

http://indico.cern.ch/contributionDisplay.pysessionId=97&contribId=105&confId=44566
(Session on G4 Computing Performance – 2009 G4 User's workshop)

Some ATLAS/CMS Differences

ATLAS and CMS simulations have both achieved high levels of physics accuracy and technical robustness. CMS application is significantly faster because…

- ATLAS's geometry is more complex than CMS's.

- CMS uses by default a shower library in the Forward Hadron Calorimeter (HF) and will probably move to a GFlash shower parameterization both in the HF and ZDC detectors. ATLAS uses G4 showers everywhere.

- CMS uses the faster QGSP_BERT_EMV physics list. ATLAS EM sampling calorimeter is more sensitive than CMS crystal EM calorimeter to multiple scattering. ATLAS uses QGSP_BERT.

- CMS uses field caching to access magnetic field values: re-evaluated field only if G4Step size > 1mm. ATLAS does not do field caching.

FNAL G4 Performance Team

- Past activities (2008): See D. Elvira's presentation in Kobe - 2008

  - In *G4Qhadron & G4QNucleous* (CHIPS model), std::vector<G4Double>* T was replaced by std::vector<G4Double> T with a ~1.5% timing improvement in the CMS offline environment
  - Reorganization of *G4ElementaryParticleCollider* (Bertini), removing 20 of 21 data members resulted in ~4% timing improvement.

- Past activities (2009):

  (I) Profiling

  Event irreproducibility correlated with bifurcation of event processing times across many of the same jobs depending on computer architecture traced to different "firmware" implementations of sin (or, in general, transcendental) functions for different CPU brands.

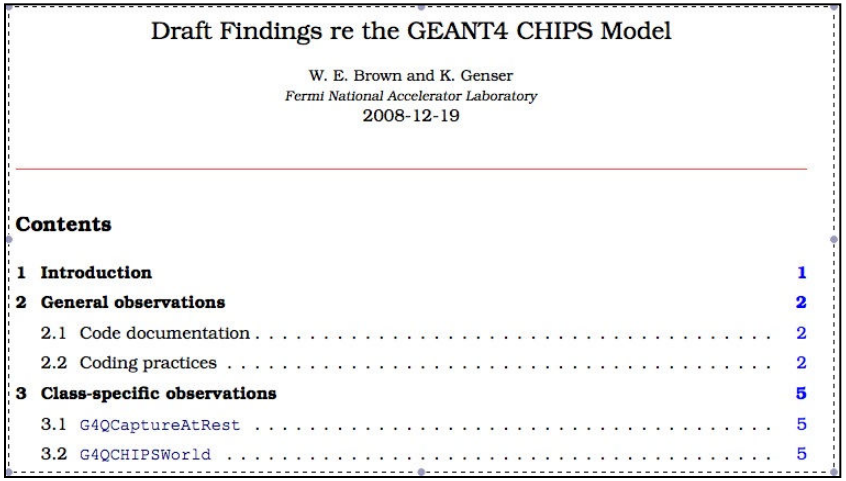
  (II) Code Reviews:

  Review of the CHIPS hadronic model and particle in Field Propagation Modules

FNAL G4 Performance Team

Conclusions from CHIPS and propagation in field code reviews

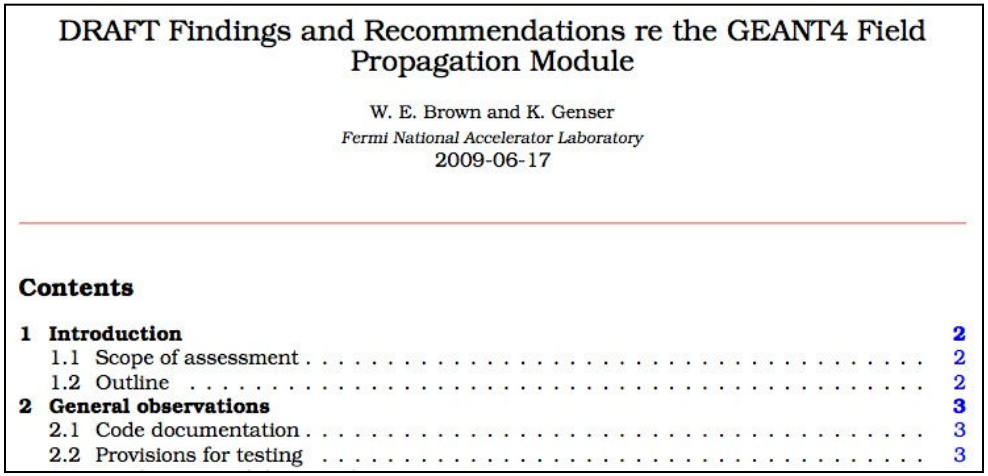
- Most of the comments were related to the C++ coding techniques having impact on code robustness and maintenance
- Among other findings: a potential ~0.5% timing improvement in CHIPS by replacing a collection of pointers to objects with collections of objects.
- No significant opportunities for timing improvement noted in Field Propagation Module.



Draft Findings re the GEANT4 CHIPS Model

W. E. Brown and K. Genser
Fermi National Accelerator Laboratory
2008-12-19

Contents



DRAFT Findings and Recommendations re the GEANT4 Field Propagation Module

W. E. Brown and K. Genser
Fermi National Accelerator Laboratory
2009-06-17

Contents

Krzysztof Genser will report today on current activities

Questions to Answer Today

- Any general comments on the charge draft?
- What type of activities should G4CPT undertake?
- Would we benefit from more regular profiling? Public, candidate, reference releases?
- What information? How do we publish/share it?
- What tools? Valgrind (ATLAS, G4), IgProf (CMS, ATLAS?), SimpleProfiler (G4 FNAL team), Perfmon (ATLAS), others?
- What G4 packages would benefit from code reviews?
- How frequently should the G4CPT meet? What format?

Tips for improving CPU Performance of programs using Geant4
https://twiki.cern.ch/twiki/bin/view/Geant4/Geant4PerformanceTips

Back Up Slides

FNAL G4 Performance Team

Application: G4 standalone tests and CMS simulation application.
Software Versions: G481p01 (2008 studies), G491p03 (CHIPS), G492p01 (field propagation), QGSP_EMV & QGSP_BERT_EMV.
Package/Process: CHIPS, Bertini, field propagation

Profilers: "Simple Profiler" and "Performance Data Base" (developed at FNAL).

About SimpleProfiler:

- It collects data (unbiased measurements) for the program of interest 100 times per second, captures the address of the current function and the address of each function in the call stack. (It thus collects full call path information, not provided by other tools.)

In post-processing, the function names and library location for each function and is determined and all the information is loaded into an SQLite3 database.