ATLAS Detector Simulation Performance

G4CPT - March, 8 2010

J. Apostolakis, A. Buckley, A. Dotti, Z. Marshall

CERN

Geant 4

Outlook

- Introduction
- Tracking of particles optimization
- Code optimization
- Memory churn optimization

Tests done with G4 9.2.p2

Introduction

- A mixed ATLAS / PH-SFT group has been set up with experts from the ATLAS sw and G4 to assess the performance of the ATLAS detector simulation
- Mandate of the group was to produce a report indicating hot-spots in the simulation and possible solutions
- A report has been produced: ATL-COM-SOFT-2010-008 ; CERN-LCGAPP-2010-01
- Different areas for improvement has been identified classified in: review of production thresholds and code improvements

G4Step Distribution

- The simulation of ttbar and MB events have been studied: typical topologies
- The basic assumption is that the simulation time is proportional to the number of G4Steps performed
- Any attempt to improve performances should concentrate on e.m. showers in EM-calos

Number of G4Steps, units in 10^6	e-	e+	gamma	charged pion	proton	neutron	other	Total (%)
Px	2.32	0.24	2.37	0.68	0.14	0.30	0.21	***0.3%***
Sct	2.42	0.37	6.52	0.82	0.20	0.89	0.25	***0.6%***
Trt	5.37	1.05	40.20	1.82	0.50	5.46	0.53	***2.9%***
Sev	49.88	7.35	40.63	0.42	0.35	3.82	0.46	***5.4%***
Cry	48.92	6.48	49.01	0.34	0.40	7.38	0.41	***6.0%***
Pre	3.99	0.49	10.41	0.09	0.06	1.73	0.03	***0.9%***
EMB	79.94	11.76	66.26	0.74	0.56	32.59	0.38	***10.2%***
EMEC	137.13	22.73	114.97	1.34	1.03	63.34	0.65	***18.1%***
HCB	10.10	0.96	14.88	0.14	0.18	11.46	0.25	***2.0%***
HEC	17.49	1.67	23.01	0.23	0.30	20.79	0.58	***3.4%***
EM FCal	303.56	31.82	204.14	1.79	1.61	64.26	0.80	***32.2%***
Had FCal	90.15	13.18	42.67	0.91	0.83	87.99	0.72	***12.5%***
FCO	10.23	1.46	3.80	0.10	0.12	1.01	0.03	***0.9%***
LAr	2.50	0.25	2.97	0.06	0.11	0.54	0.05	***0.3%***
Mu	31.85	4.34	22.09	0.11	0.23	4.25	0.50	***3.4%***
Other	4.89	0.86	10.13	0.13	0.11	1.05	0.04	***0.9%***
Total (%)	***42.4%***	***5.6%***	***34.6%***	***0.5%***	***0.4%***	***16.2%***	***0.3%***	***100%***

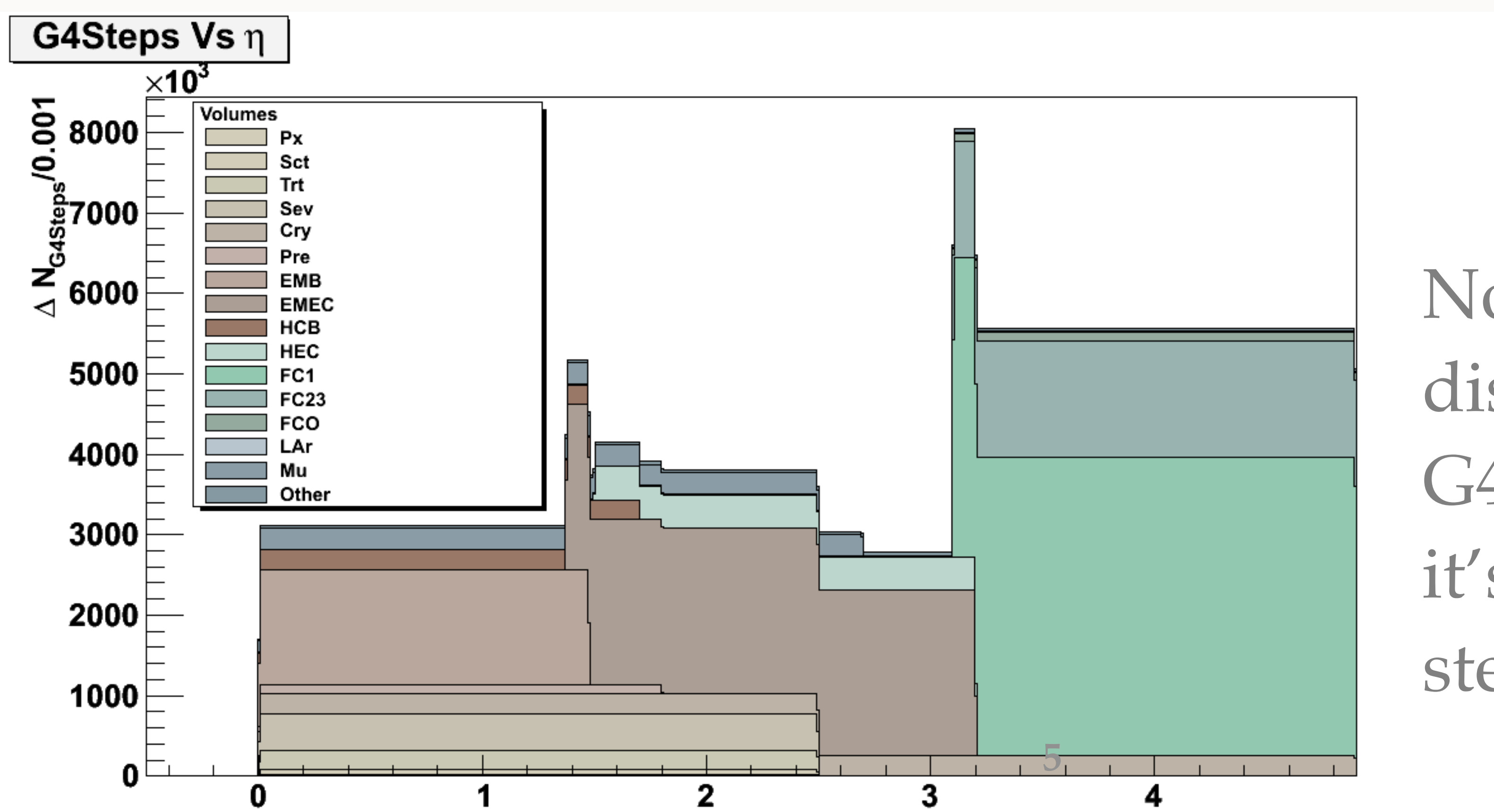
ID

Calos

Eta Dependence

Subdetector	Range Cut	Absorber Thickness	Liquid Argon Thickness
EM Barrel	100 µm	1.13-1.53 mm	200 µm
EM Endcap	100 µm	1.7-2.2 mm	200 µm
Hadronic Endcap	1 mm	25-50 mm	8.5 mm
Forward Calorimeter	30 µm	Varies	269-508 µm

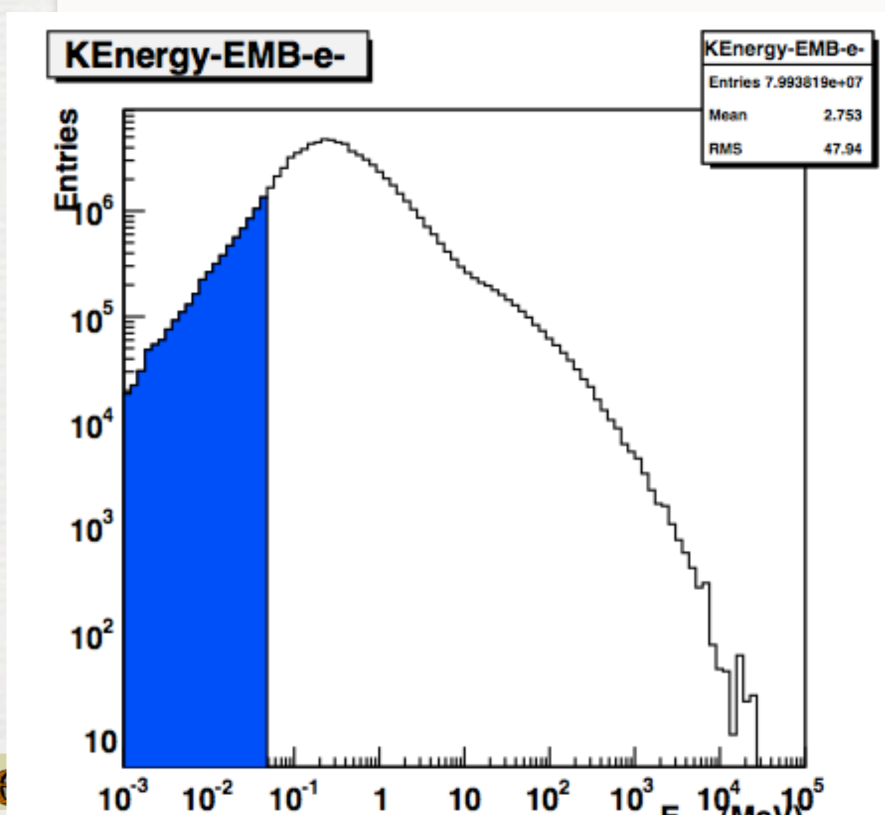
- It is recomended to review the threshold cuts for different G4Regions:
  - At the moment a large fraction of track is simulated in the very forward region (less stringent physics performances required)
  - This is at the moment the most promising strategy to reduce simulation time
  - potential improvement: 30%



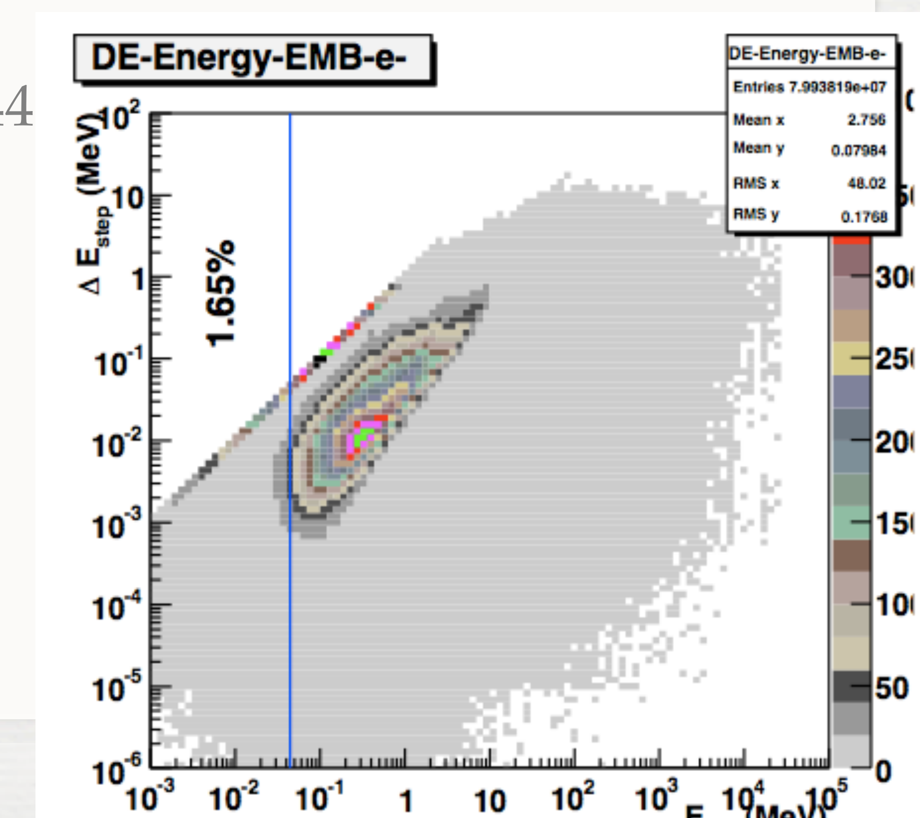
Note: real η distribution for G4Tracks
it's likely even more steep to high values

Other Strategis

- Alternatives options to be studied with more details have been identified:
  - em option 2 (EMX phys lists) (<20% effect)
  - Tracking cut (0-20% effect)
  - Use of different cuts for e and gammas (few % effect)
  - More granular G4Regions (few % effect)



Effect of a tracking cut at 44 keV in LAr:
8% of G4Steps are below threshold,
these deposit 1.65% of E



Callgrind Studies: High Priority

- The simulation program has been studied with the help of the valgrind suite
- Callgrind call trees have been identified some hot-spots areas for improvement have been identified:
  - Significant time is devoted to evaluation of the magnetic field (26%). In particular the time is spent in accessing the value of the field. These results have been obtained using the ATLAS patch number 3 that already includes improvements in this area (new 9.3 stepper should be tested in the future)
  - Significant time is spent in geometry related methods (distance to in, distance to out) in the EMEC wheel (15% of the CPU time in ttbar events). The EMEC is implemented as a single, large custom solid. Potentially considerable additional time is spent in energy collection methods because of this design.
  - Given the importance of the Bertini cascade for the ATLAS simulation, it is important to improve its code. An area of improvement to pay particular attention to is the use of standard library containers. The ongoing re-design of Bertini should bring benefits in this area.

Callgrind Studies: Low Priority

- Multiple Scattering (Urban Model 2) takes about 4% of the CPU time, a deeper analysis of the code should be foreseen to identify if it is possible to improve performances.
- Retrieving cross-sections for the calculation of steps length takes about 5% of the total CPU time. This time is distributed equally among several models and different implementations
- Two utility methods are called several times and are responsible for 2% of the CPU time: G4Track::GetVelocity and G4PhysicsVector::GetValue. Another 2% comes from the use of log and exp functions.
- The, relatively rare, gamma-nuclear process takes 0.4% of CPU time. Possible optimization?

Memory Churn

- Previous studies carried out by ATLAS have shown that reducing the number of malloc (new/delete) plays an important role in sw performances. The ATLAS Hephaestus tool has been used to study this aspect:
  - G4TouchableHistory is responsible for 25% of the memory churn
  - QGSP_BERT (in particular the Bertini cascade) and its related methods are responsible for 40% of the memory churn
  - G4NavigationHistory's constructor is responsible for 25% of the total churn
  - LArG4Identifiers and related methods are responsible for 10% of the memory churn
- atlas3 sw patch solves the first issue, this has been ported to G4 9.3
- Bertini code is undergoing a re-write, it is suggested to reduce memory churn and re-design the use of stl containers

Conclusions

- The study has identified some points of improvement
- ATLAS specific:
  - Most promising to reduce CPU time is the use of more aggressive production threshold and pseudo-rapidity cuts
  - Re-design of EMEC custom solid
  - LArG4Identifiers redesign to reduce memory churn
- Common -B Field-:
  - New stepper: G4Nystron (available G4 9.3) and B-Field caching
- G4 specific:
  - Re-design of BERT code: reduce of memory churn
  - G4TouchableHistory , G4NavigationHistory re-design to reduce memory churn
  - UrbanModel2 is responsible of 4% of CPU
  - Cross sections retrieval: 5% of CPU
  - G4Track::GetVelocity and G4PhysicsVector::GetValue: 2% CPU
  - Use of exp and log: 2% CPU (part of this is also ATLAS specific)