ATLAS Requirements

The ATLAS Sim Team

Geant4 Tech Forum

13 Jan 2015

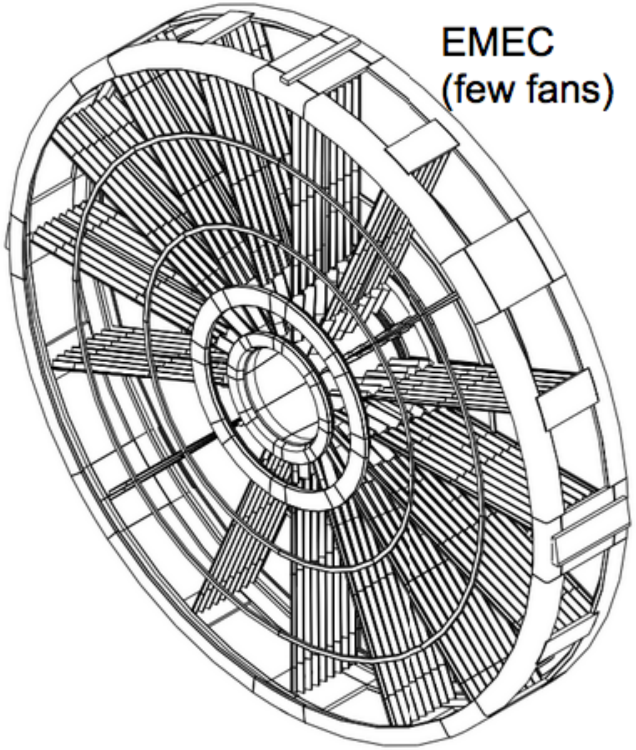
Current Production

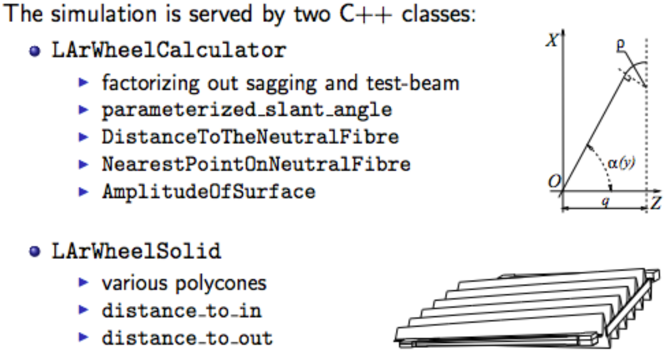
- ATLAS is still in "MC12" production mode
  - Still producing about 5B events per year
  - Using G4 9.4 + (official and private) patches, 32/64-bit, gcc 4.3, SLC5
- Also running test production ("data challenge 14")
  - Using G4 9.6 – but really this was a warm up for this year's MC
- Now beginning MC15 production (13 TeV)
  - Launching with G4 9.6 and FTFP_BERT
  - Enormous number of geometry updates both to improve our description of the detector (and its response) and to include the changes to the detector that were made during LS1
  - Using gcc 4.7, C++11, SLC6, 64-bit only, CLHEP 2.1 (no internal G4 CLHEP to be consistent with reco)
  - Still testing icc, clang, Mac OS X builds (still no production plans)
  - Expecting this will be in production through early 2016

Production Plans

- Working on a reasonable physics list variations
  - FTFP_BERT_EMZ/EMY for EM physics variations in testing
  - Received hadronic physics variation options for systematic studies (thanks Alberto!); now preparing to test these in the production system
  - The next run will include particles hitting a calorimeter that have higher energy than ever before. Are we sure the Geant4 hadronics and EM models are going to be up to it? We will feed back as soon as possible what we have, and we hope you all can help respond and improve the high-energy models as we get new information!
- Working on ApplyCuts for calorimeter simulation speed up
  - Is it clear whether this will just cancel out the gain from _EMZ?
- Starting to plan MC16 for the end of 2015
  - Hoping that this will be our move to G4 10.1 in production
  - Expecting we would then be in 10.1 through end of 2017 – quite possibly into LS2
- G4MT Trial / prototype in testing for some time now
  - Looking forward to upgrading this to G4 10 – thanks for your responsiveness and helpfulness with the G4MT prototypes!!

Our Developments




The simulation is served by two C++ classes:

- LArWheelCalculator
  - factorizing out sagging and test-beam
  - parameterized_slant_angle
  - DistanceToTheNeutralFibre
  - NearestPointOnNeutralFibre
  - AmplitudeOfSurface
- LArWheelSolid
  - various polycones
  - distance_to_in
  - distance_to_out

- Major improvements to EMEC geometry methods (custom solid)
  - Improves total CPU by ~10-15%
- Many, many incremental improvements
  - 3% here, 5% there, many 1%-level improvements
- Improved error reporting on the grid
  - Soon should be able to cull G4 error messages properly from production jobs and give better reports and rates to you guys
- Implemented simulation of quasi stable particles (e.g. B hadrons, taus) that the generator decays
  - Using SetDaughter(…) assignment for G4PrimaryParticles using generator daughters
  - Recovers silicon hits for highly boosted particles
  - Might be nice to have this in Geant4. Most HEP experiments read HepMC files (well-established event generation standard) and convert to G4PrimaryParticles. Save the FCC folks the trouble?

Patch Saga

- It looks to us like it's time for another public patch of G4 9.6
  - Patch for muon capture infinite loop (thanks!)
  - Patches for G4ProtonField and G4NeutronField
  - Patches to clean up multinavigator use
  - Patch to binary cascade for deuteron crashes
  - Patch to G4NavigationLevelRep to avoid temporary
  - Patch to G4HadronicProcess to avoid Tachyons
- Meanwhile, began use of particle "white list" provided by G4 developers
  - The first draft included quarks, which probably G4 doesn't know how to move
- ISF identified a major CPU performance penalty coming from memory pool use (G4EnhancedVecAllocator) in G4TouchableHistory
  - Memory pool use was removed, issue was also worked-around
- G4TouchableHistory was responsible for significant new and delete
  - Expected after that previous change (75GB/50 event job – ¾ of total churn)
  - A. Dotti adapted the G4 10.1-β memory pool (should be more CPU efficient)
  - Still opportunities to improve the pool CPU usage, discussing with G4 devs

More Plans for the Year

- Hardening of G4 9.6 production setup
  - We still have some level of crashes, but we don't have firm numbers
  - Still trying to get full event-level reproducability
- More serious testing with G4 10.1
  - First tests show nice CPU performance (just trying to disentangle effects now)
- More optimization work to try to shave some more CPU off
- Rewrite of lots of infrastructure in a more Gaudi-friendly way
  - Should not deeply affect simulation speed, but will dramatically affect ease of initialization
  - Will allow an easier move to Gaudi Hive, eventually
  - May lead to some useful discussions about future Geant4 developments
- Testing on HPC systems
  - Simulation is low I/O, so a good use-case for HPC systems. In testing and validation for some systems. Major production system overheads; need to significantly improve performance before we use this for major production
  - Patch to use databases instead of many small files for data could be helpful here
- Lots of work on fast simulation planned
  - Not so interesting for this audience, we think…
- Should we have an LPCC simulation workshop at the end of this year?
  - Might include the first 13 TeV data…