

ATLAS Requirements

Zach Marshall for the ATLAS Sim Team

Geant4 Tech Forum

10 Dec 2013



Current Production (Familiar?)

- ATLAS is still in “MC12” production mode
 - Approaching 10B simulated events during this production
- Production using G4 9.4 + (official and private) patches
 - Supporting both 32 and 64-bit, gcc 4.3, SLC5
 - From March: “expect to stay in similar configuration through early 2014
- Many migrations underway
 - Shortly will launch simulation for “Data Challenge 14”, the first major simulation campaign in preparation for the 14 TeV data
 - Will launch with G4 9.6 and FTFP_BERT
 - Will move soon to gcc 4.7, C++11, SLC6, 64-bit only (probably), CLHEP 2.1 (no internal G4 CLHEP to be consistent with reco)
 - Testing most of these platforms already, and some are quite advanced
 - Also testing icc, clang, Mac OS X builds (still no production plans)
- G4MT Trial / prototype in testing for a while
 - Development within the context of our new ISF
 - Looking forward to upgrading this to G4 10 – thanks for your responsiveness and helpfulness with the G4MT prototypes!!

Outcomes from Benchmarking

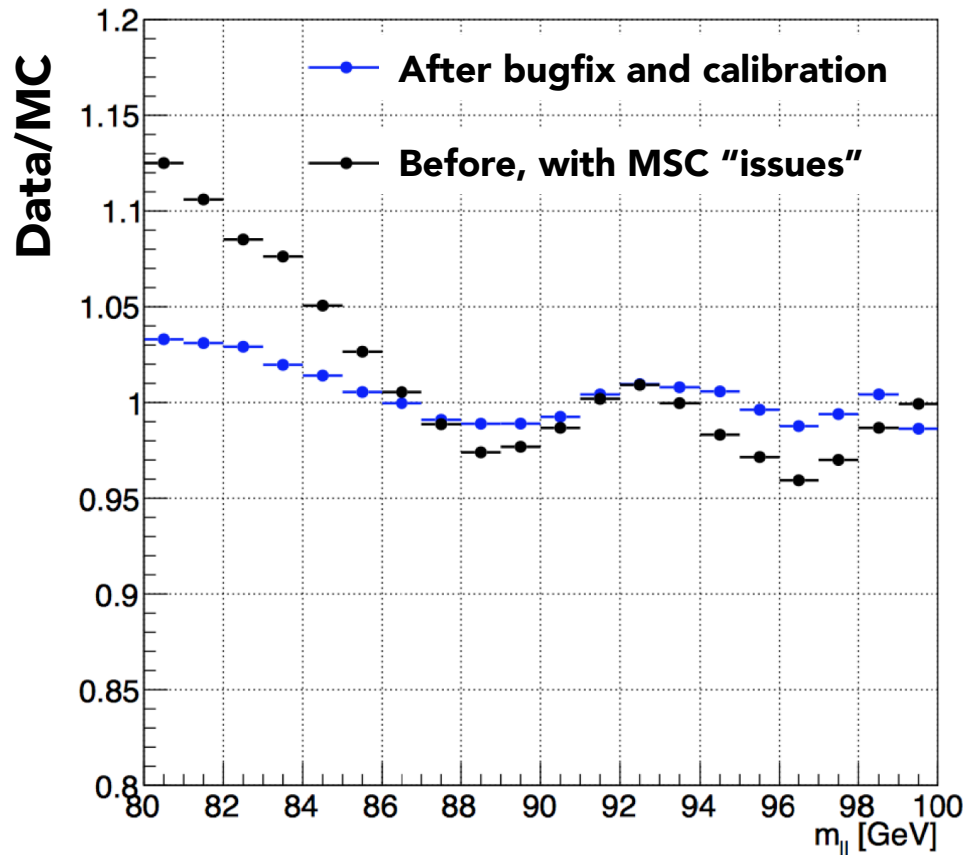
- We showed in September (also at your collaboration meeting by JA) some benchmarking for the first time in a while
- Now we're going back to correct and revisit some old features
 - Includes range cut revision in the calorimeters, for example
 - Includes digging in again to the EMEC code... our old friend
- We are in the process of validating ApplyCuts for EM physics
 - Clearly we need smaller range cuts to use apply cuts, so the optimistic gain is a bit optimistic. Still, it looks like we could be talking about a 5-10% gain, which is not nothing.
- Also finding some features in our code, but a lot of this archeology takes time
 - Why were these things there?

```
//When I wrote this, only God and I understood what I was doing  
//Now, God only knows
```

```
//  
// Dear maintainer:  
//  
// Once you are done trying to 'optimize' this routine,  
// and have realized what a terrible mistake that was,  
// please increment the following counter as a warning  
// to the next guy:  
//  
// total_hours_wasted_here = 42  
//
```

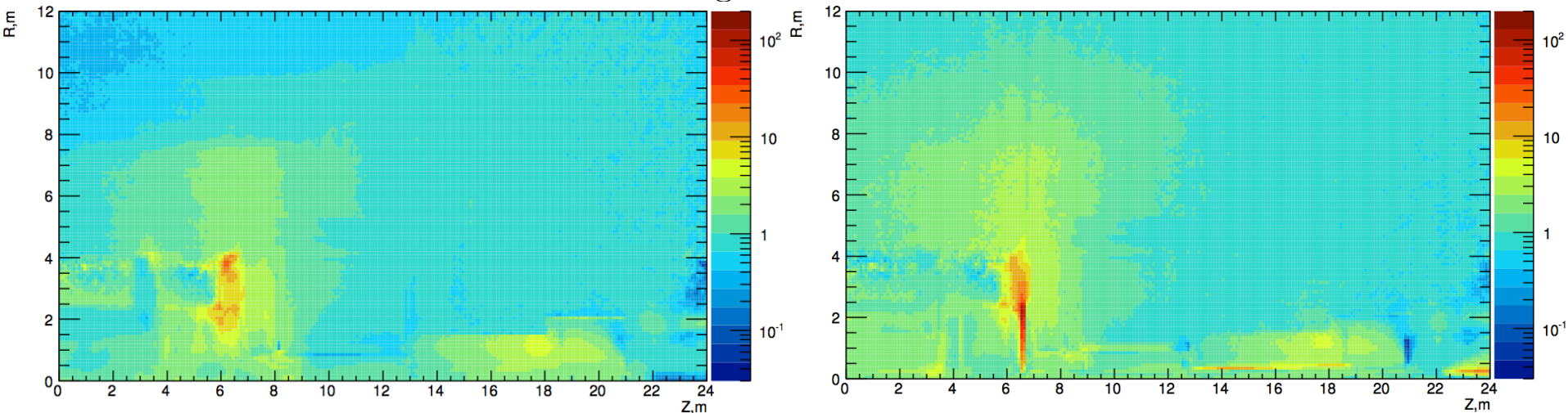
Z Line-shape Saga

- Long investigation of $Z \rightarrow e\bar{e}$ led to “discoveries” of issues in multiple scattering
- Now solved, and data/MC agreement vastly improved
 - Thanks for all your help with this!!



Cavern Background

- After a *very* long saga, the cavern background fluxes in G4 compare quite favorably
 - Left for neutrons, right for photons
- Remaining features seem to be lower level than they used to be, and it's no longer obvious who is “right”
 - The stream on the right comes because the two have an actual difference in description that hasn't been resolved
 - In the bulk, the differences are quite small – almost to the point where we want to see data to distinguish the two!



Bug and Crash Reminders

- Still quite a few open issues
- For a *crash*, pay a factor of ~**FIVE** penalty in CPU time
 - Because of grid retries
 - Can be reduced only in the case that we can recognize the failure and that it will repeat – working on that now
- For a *loop*, pay a factor of ~**TWENTY** penalty in CPU time
 - Grid retries + waiting for the job to time out
 - These are almost impossible to distinguish from grid nodes getting borked – very hard to auto-detect and shut down early
- New production system should improve our situation, but these still make us very nervous
- This is the primary reason that we are not rushing into G4 10
 - Already with G4 9.6 the loop rate is borderline for production

Latest Bugs and some Requests

- Apparent infinite loop in muon capture processes
 - http://bugzilla-geant4.kek.jp/show_bug.cgi?id=1536
 - We seem to find about one of these a year. Would it be possible or reasonable to try to troll through the code and find the remaining do-while statements that are not protected against infinite loops?
- “Solved” the geometry bug reported last TF
 - Came down to negative length sides, where the G4Box constructor protection was incorrect (and our protection was non-existent)
- Still struggling with an (apparent) bug in G4PolyCone
- Still struggling with difficulty to re-start a run at event N
 - This seems to be getting close, but isn’t quite there yet (even with G4 9.6 p2)
- We have had a number of problems with generators handing out particles that G4 knows nothing about
 - Could the G4 developers please provide a “white list” (PDG IDs are great here) of particles that **could** reasonably be treated by G4??
- Still trying to understand a few benchmarking “features”
 - Binary cascade makes its own stepper? Does it really need the full G4ClassicalRK4?

Plans for the Year

- Testing and playing with G4 10
 - We are about to launch a major simulation campaign. Switching takes about a year, all in, for validation. Now is not the time.
- Further validation and debugging of G4 9.6
 - I've been bad about digging into bug reports; should improve soon
- Serious optimization work to try to shave some more CPU off
 - Plenty of ideas, but not a lot of people for this one
- Lots of work on fast simulation planned that we won't bore you with here
- Preparation for the March LPCC workshop – we look forward to showing you some interesting results
 - If there are studies you've seen from us in the past that you particularly liked, there is still time to put in a special request
 - Did we get anywhere with “systematic uncertainty variation” lists?