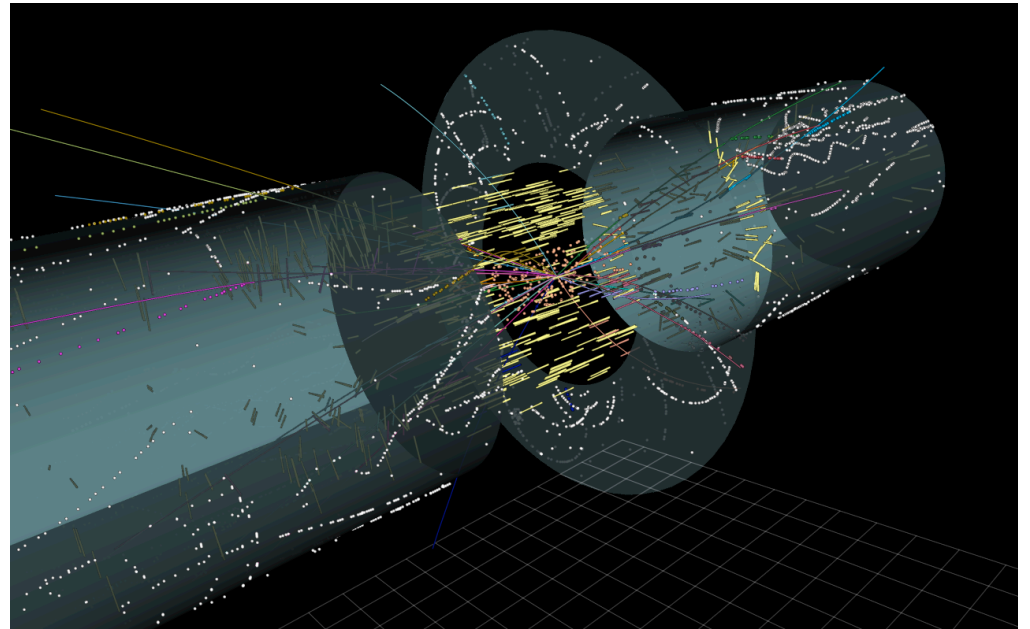




Atlas Geant4 Simulation

Zach Marshall

For the ATLAS Core Sim Team
G4 Tech Forum, 29 Sept 2009



Basics

- ATLAS is currently using:
 - Geant4 9.2 patch02
 - Plan to continue with 9.2 through first data (~end 2010)
 - slc4 gcc34 32-bit (plan to move to slc5/gcc43 soon)
 - CLHEP 1.9.4.2
- Focus in the last few months has been on performance optimization and stability
 - Running valgrind, memory profilers, CPU profilers, memory allocation profilers...
- Good example of how it should go: G4String
 - Found 600MB/event alloc/dealloc from G4String ==
 - Found the problematic method in ATLAS code
 - G4String comparisons were also sub-optimal
 - In a week we had an ATLAS-side and G4-side patch
 - And in a another week this was included in official p2

Stability

- Two open bugs from our point of view
 - All things considered, that's not too bad...
- Crash in hadronics from a corrupt(?) nucleus
 - Seems new to me

```
*** Geant4 Hadronic Reaction Information ***
Process:  , Model:
Nucleus A, Z = 0 0
Projectile was a
```
- 'Stuck' tracks taking tiny steps
 - Probably an old story that we didn't pay close enough attention to
 - Tracks taking millions of very small steps, bouncing between two volumes / materials
 - But taking big enough steps so as to not trigger any of the 'stuck track' mechanisms built into G4Transportation
 - As a result, no warning from G4 - job just continues as long as it is allowed to (may recover after a very long time)
 - Still under investigation by both G4 and our groups
 - But important enough that I'll tell you more about it...

Stability (II)

- First attempted solution (increasing tolerance for zero size steps) resulted in G4 abandoning those tracks - so we found out that G4 can abandon tracks!

```
WARNING - G4PropagatorInField::ComputeStep():
```

```
Zero progress for 51 attempted steps.
```

```
Proposed Step is 1.802421668501153e-10 but Step Taken is 1.802421668501153e-10
```

```
For Particle with Charge =-1 Momentum=0.08665330623096662 Mass=0.51099891
```

```
in the volume Muon::ToroidShieldingInnerPlugs
```

- Some guidance on setting tolerances would be helpful
 - Request a higher default 'warning' energy - 250 MeV is a lot!
 - ATLAS will test with a lower cutoff as well
- This is a rather serious problem in our prod. system!!
 - A track stuck like this can run a job to the CPU limit on the Grid, causing the job to fail
 - The jobs are automatically retried many times in order to allow for transient (Grid teething) failures
 - Our typical jobs take ~1/4 of the grid limit in time, so a stuck track in 1% of jobs (=1/5000 events, or 1/~2x10¹¹ steps) means a **100% increase in CPU for the task!**
 - Current rate between 0.1% and 1% (depending on job type)

Charged Particle Transport

- Several familiar issues with transportation
 - G4Transportation alloc/dealloc lots of memory
 - In ATLAS, >2 GB per physics event
 - Default stepper (RK4) accesses the B-field 10x / step
- New solutions to old problems
 - New stepper from ATLAS re-implements RK4 with fewer field accesses. Should be the same otherwise.
 - New G4CachedMagneticField adds field caching in G4
 - Suggest those be included in geant4's next releases(!)
 - Alloc / dealloc in G4Transportation an open issue
 - When we have solved the bugs, perhaps this could be attacked (again) with an eye towards optimization?
 - In particular G4Transportation::PostStepDoIt()
- Also: how is parallel navigation these days??

Physics

- A few problems/new ideas uncovered recently
- Constructing EM processes in a vector
 - Saved us 3.5 minutes/job in initialization time
 - No apparent memory or CPU cost
 - Request it be included in future G4 releases
- Bertini physics model allocates/deallocates quite a bit of memory (1.5 GB/event for ATLAS) - translated FORTRAN
 - To be rewritten, but some changes already patched
 - Look forward to having new code to try!
 - A recent reduction on the ATLAS side of 1.3 GB/event alloc/dealloc saved us 10% of our CPU time...
- New-ish problem with accessing indices out of range in hadronics (pointers, so no crash)
 - Not clear if this is causing any serious problems yet
 - How much valgrind-like testing is done by the G4 team?

Summary

- LOTS of help from the G4 developers - thank you!!
- Very fast response to any problem reports, and several quick 'private' patches
- It would be nice to have more frequent *public* patches to we ensure our private builds do not diverge either from each others or from the 'official' G4 version
 - ATLAS is planning to stick with G4 9.2 through first data, so having bug fixes back-ported from 9.3 etc would be helpful!
 - We will continue to test new releases to provide feedback...
- Production proceeding constantly
 - Around 400M events in 2009 with G4 8.3/9.1
 - About to launch the major 2009 production (several 100M events) with G4 9.2 - around 2M events a day
- We are *really* looking forward to data and a chance to battle-test G4 physics!!