

## NOvA-ART - Nova Simulation Problem #7087

### Structure of input to Geant greatly influences simulation time

09/30/2014 05:07 AM - Alec Habig

<b>Status:</b>	Rejected	<b>Start date:</b>	09/30/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Nova Simulation Component:</b>			

#### Description

There are two ways one could feed particles to the detector simulation to be simulated.

The data structure is a vector of MCTruth's, a generation module produces `< std::vector<simb::MCTruth> >()`;

The way we were doing it at first for Supernova (stolen from SingleGen) is to use the Add() method of the MCTruth object to stack up a number of positrons in one MCTruth object, push that one object into the vector, and go. This simulates just fine, but doesn't result in a good truth tree for backtracker.

The approved way (which makes backtracker a bit happier, although the BT tools still don't work right for me) is to have one particle per MCTruth object, and push multiple MCTruth's into the vector.

Both methods result in the same simulation output in terms of generated hits. The approved method makes a file that's ~10% larger for the same events, presumably from including more truth objects?

However, the approved way (one particle per MCTruth, big vector of truths) takes an order of magnitude more time to generate the same hits output. Either there's some unnecessary setup happening every time a MCTruth is popped from the vector, or creating the truth objects (or whatever's in that 10% more space) is really super slow.

I don't have the inside knowledge of the simulation needed to figure out what's going on (heck, I can't even get the truth products through to my analysis!), but Brian suggested documenting it with an issue in case someone later takes a crack at simulation execution time problems.

#### History

##### #1 - 09/30/2014 10:35 AM - Alec Habig

- File *times.txt* added

- File *order.patch* added

Timing summary (complete timing report attached below):

"Fast" way: SingleGen and SupernovaGen as in the repository. Truth info suffers. 3.89s/event, of which 3.85s is spent in "simulate". O(10s) of positrons per event, 10-20 meV per particle. One MCTruth item used, with positrons Add()'d to that object.

"Slow" way: modified SupernovaGen (see attached patch) to have one positron per MCTruth item, stacked into a vector. 49.0s/event, of which 48.9 is spent in "simulate".

Release S14-09-05 used.

##### #2 - 10/01/2014 07:52 AM - Alec Habig

For what it's worth, the way cosmic MC is generated (according to the prodcosmics.fcl file in the CRY directory) is affected by this issue. I'm racing two jobs, one with the default setup, and one with "physics.producers.generator.BreakMCTruth: false" set. This switch changes between the two truth organization modes outlined above. The non-default mode is making muons factors of several times faster than the default mode. Will post times later when they come in.

##### #3 - 10/02/2014 12:23 AM - Nathan Mayer

The interesting thing is a little further down in the time reports attached. The first method generator, which builds the MCTruth object(s) runs at about the same speed in both cases. And in both cases this is pretty fast. Presumably because you're just putting positrons in to the detector at with rand four vector position and momentum. But GEANT is 10 times slower between the two methods. Adding Robert Hatcher as a watcher because he's in the FNAL generator group at Fermilab.

Do you have the ART output for these two jobs? I'd like to look at the two files.

#### #4 - 10/02/2014 05:31 AM - Alec Habig

Two SN-induced positron files here: /nova/ana/users/habig/sngen. The one named "-faster" is the one which went, well, faster. Will get timing details and the cosmic files showing the same problem posted in a bit here.

#### #5 - 10/02/2014 06:29 AM - Alec Habig

- File *cosmics-time.txt* added

ok, cosmic sim test complete. This is using the default prodcosmics.fcl (slower) vs. the same with BreakMCTruth set to false (faster). 10 cosmics events are created per run (multiple muons in each event's time window, of course). Report attached (cosmics-time.txt).

Summary: very similar conclusions as to the positron stuff above, but more time of course spent generating the things, doing photon transport, and converting to DAQ hits. The time inflation happens only in the geantgen stage. Output files are in /nova/ana/users/habig/cosmiccheck vs. /nova/ana/users/habig/cosmiccheck-nobreak. They look very similar, will move them over to a local machine and hand scan them: will add a note if anything looks appreciably different.

Faster way: 1104s/event (11.6 generator, 618 geantgent, 61.7 photrans, 413 DAQ)  
Slower way: 2378s/event (11.5 generator, 1891 geantgent, 59.4 photrans, 403 DAQ)

#### #6 - 10/02/2014 08:07 AM - Robert Hatcher

It's late and I'm not quite ready to look at detailed dumps, so let me do some speculating about code who's design I wasn't involved in, so I'm guessing at rationales.

I think this observed effect is due to the way NOvA/LArSoft have structured how they interface to Geant4. In NUTTOOLS/G4Base the G4Helper class essentially feeds each MCTruth to Geant4 and then does a "run" of 1 event (i.e. /run/beamOn 1). I suspect that it was structured this way due to the (at the time) restrictions on how one could interact with the Geant4 run manager and trying to limit the clash between two "event loops" (ART & Geant4).

So each MCTruth is a separate run with whatever overhead that setup entails. For MCTruth's that are tiny little positrons (as opposed to several GeV of hadrons, muons or electrons) I could imagine that this wouldn't scale very well and the overhead is dominating the time. It might be possible to restructure the way ART and Geant4 interact (I know there have been some changes to the G4 RunManager class from pressure by mu2e), but that would be a relatively major undertaking.

```
bool G4Helper::G4Run(const simbc::MCTruth* primary)
{
    // Get the event converter ready.
    fConvertMCTruth->Reset();

    // Pass the MCTruth to our event generator.
    fConvertMCTruth->Append( primary );

    // Start the simulation for this event. Note: The following
    // statement increments the G4RunManager's run number. Because of
    // this, it's important for events to use the run/event number
    // from the EventDataModel Header, not G4's internal numbers.
    fUIManager->ApplyCommand("/run/beamOn 1");

    return true;
}
```

#### #7 - 10/02/2014 08:50 AM - Alec Habig

Looked at the cosmics in the evd. Both files are fine. The truth info is there: the only difference is that the faster, "nobreak" version appears to get confused as to which particles are primaries or not: maybe it's just off by one level in the family tree?

Anyway: if what Robert says about the structure is true, but we want to gain a factor of 2x back on our cosmics simulation, maybe some work on how we sort out the MC truth can get us our cake and eat it too.

#### #8 - 10/02/2014 09:01 AM - Nathan Mayer

There were two things that motivated the change from 1 MCTruth per/spill to 1 MCTruth/muon for cosmics.

1) Overlaying cosmic MC onto beam MC at the FD. If you want to shuffle up the muons when doing these overlays, so that you can generate many "unique" spills out of fewer muon tracks, then the mixing is greatly simplified if every muon track gets its own MCTruth.

2) There was a desire to have cosmic CAFs. And the structure of the CAFs was not easily compatible with 1 MCTruth/spill.

#### #9 - 10/04/2014 02:53 PM - Christopher Backhouse

Given the times Alec gives this seems well worth getting into, especially since geant dominates our generation time. For ND beam and FD cosmics this is going to be a big effect.

Simulation packages in the first-analysis branch are frozen for generation now, so it's more or less an ideal time to go digging around in the guts of EventGenerator.

**#10 - 11/15/2016 04:58 PM - Alexander Himmel**

- Status changed from New to Rejected

#### Files

---

times.txt	10.3 KB	09/30/2014	Alec Habig
order.patch	2.06 KB	09/30/2014	Alec Habig
cosmics-time.txt	10.2 KB	10/02/2014	Alec Habig