LArIAT on the Open Science Grid

Justin Hugon
Louisiana State University

LArIAT Analysis Meeting
April 21, 2017

Motivation

- All of LArIAT has 100 guaranteed slots at Fermilab
- Currently also using unclaimed slots available in Fermilab GPGrid
  - If those are claimed, we only have 100 slots
- Fermilab and LArSoft have already done the heavy lifting to connect us to unclaimed slots on the Open Science Grid (OSG) – computer clusters all over the world
  - That many more slots that we can use!

Lets start using the OSG resources!

Overview of Steps to Use the OSG

1. Setup your LArSoft environment to use a filesystem accessible on the OSG: /cvmfs
2. Pull tweaked versions of larbatch and lariatutil to make things run smoothly on the OSG
3. Modify a line of your grid .xml file to allow access to “OFFSITE” resources
    - Modify a few more lines so your jobs get started quicker

Kirby, Hans, Jason, and others helped me with all of this!

Setup your LArSoft to use /cvmfs

- Previously, we used software in /grid/fermiapp
- /grid/fermiapp is going away soon and can't be used on the OSG
- The following slides show how to make LArSoft use /cvmfs

All of the following slides' code is on:
https://redmine.fnal.gov/redmine/projects/lardbt/wiki/Setting_up_the_Offline_Software_CVMFS

Setup a LArSoft Environment from Scratch

Make a directory for your code and cd into it, then run:

```
source /cvmfs/lariat.opensciencegrid.org/setup_lariat.sh
setup ninja v1_6_0b
version=v06_15_00 # replace with your version
qual=e10:debug # replace with your qual
setup larsoft $version -q $qual
mrb newDev
source localProducts*/setup
cd srcs
mrb g -t $version lariatsoft
mrbsetenv
nice mrb i --generator ninja -j8
mrbsetenv
setup lariatsoft $version -q $qual
setup lariatsoft $version -q $qual
```

Converting from a directory that uses /grid/fermiapp

1. Move your old directory to a backup location: mv mylariatdir mylariatdir.bak
2. Create a new directory and go inside: mkdir mylariatdir; cd mylariatdir
3. Run this code:

```
source /cvmfs/lariat.opensciencegrid.org/setup_lariat.sh
setup ninja v1_6_0b
version=v06_15_00 # replace with your version
qual=e10:debug # replace with your qual
setup larsoft $version -q $qual
mrb newDev
source localProducts*/setup
cd srcs
```

Continued on next slide...

Converting from a Directory that Uses /grid/fermiapp, Continued

4. Copy the directories from your old working area's srcs here (don't copy CMakeLists.txt)
5. Run this code:

```
mrb uc
mrbsetenv
nice mrb i --generator ninja -j8
mrbsetenv
setup lariatsoft $version -q $qual
setup lariatsoft $version -q $qual
```

Setting up LarSoft Directory on a New Login

Whenever you login again, go to the working directory you created earlier and then run (you could put this in a script in your directory that you can source):

```
source /cvmfs/lariat.opensciencegrid.org/setup_lariat.sh
setup ninja v1_6_0b
version=v06_15_00 # replace with your version
qual=e10:debug # replace with your qual
setup larsoft $version -q $qual
source localProducts*/setup
mrbsetenv
setup lariatsoft $version -q $qual
setup lariatsoft $version -q $qual
```

Extra Tweaks to Code to Make OSG Work

For now, you need special versions of larbatch and lariatutil to make things work. Go to your srcs directory and run:

```
mrb g -t feature/jhugon_CVMFSWorks lariatutil
cd lariatutil
git checkout $version ups/product_deps
cd ..
mrb g -t develop larbatch
cd larbatch
git checkout LARSOFT_SUITE_$version ups/product_deps
cd ..
nice mrb i --generator ninja -j8
```

Running on the OSG

- Now that you have done all of that stuff, only one change is required to run on the OSG.
- Change the line in your .xml file from:

```
<resource>DEDICATED,OPPORTUNISTIC</resource>
```

to

```
<resource>DEDICATED,OPPORTUNISTIC,OFFSITE</resource>
```

It's that simple!

How to Make Your Jobs Start Faster: Memory

Minimize the memory you require

- Some nodes might have a free CPU, but only a smaller amount of free memory
- Putting this in your fcl's services section will print out the memory it uses:

```
MemoryTracker:    {}
```

- The max I've found in MC reconstruction is 1200 MB, so I put this in my .xml file:

```
 <memory>1500</memory>
```

How to Make Your Jobs Start Faster: Time

Use 3 hr or less queue times

- Many OSG sites only have short queues–reducing queue times gives you access to more slots
- Estimate how many events/job can make your longest stage fit in a 3 hr window
- Put a line like this in each stage of your .xml file:

```
<jobsub>--expected-lifetime=3h</jobsub>
```

- I do 1h for analysis and 3h for reco

Job CPU Efficiency

- OSG sites are more strict about CPU efficiency
- One trick I've found for MC generation is to put multiple .fcl's in one stage:

```
<stage name="gensimdigireco">
    <fcl>prodsingle_lariat.fcl</fcl>
    <fcl>RecoMC.fcl</fcl>
    <outdir>&myoutdir;/&release;/gensimdigireco/&name;</outdir>
    <workdir>&myworkdir;/&release;/gensimdigireco/&name;</workdir>
    <numjobs>&njobs;</numjobs>
    <datatier>simulated</datatier>
    <jobsub>--expected-lifetime=3h</jobsub>
</stage>
```

It will do lots of processing and only send home the result of the last fcl in the stage

See my example .xml here:
https://github.com/jhugon/lariatgridconfigs/blob/tutorial/osg_test.xml

Data Reconstruction on the OSG

- I haven't tried data reconstruction on the OSG
- It's a good idea to stage your dataset to disk before starting jobs. It might take some time. Run:

```
samweb prestage-dataset --defname=<dataset name>
```

- The only handle we have on job runtime is

```
<maxfilesperjob>10</maxfilesperjob>
```

- We'll have to adjust maxfilesperjob and the total number of jobs
- Have to make sure that $N_{jobs} < 10000$ and $N_{jobs} \times (\text{Job Time [hr]}) < (100 \text{ slots}) \times (1 \text{ week})$

LArIAT dE/dx Calibration with Cosmics

Cosmic Data & MC

Data

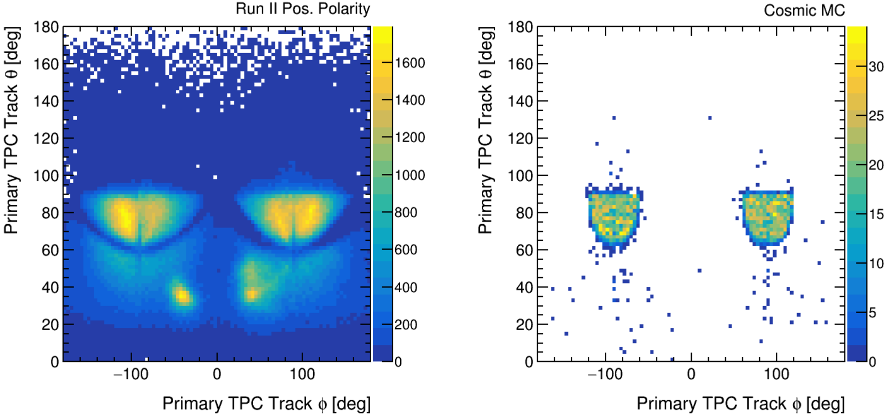
- Run II+ data
- No WC tracks or TOFs
- Select trigger bit 10, which I think is the COSMICON trigger

MC Sample

- Generated μ^+ starting at -15 cm $< x <$ 15 cm, $y =$ 50 cm, -30 $< z <$ 120 cm
- Flat energy from 1.5 GeV to 2.5 GeV
- Going downward with angle to the zenith of up to 30°

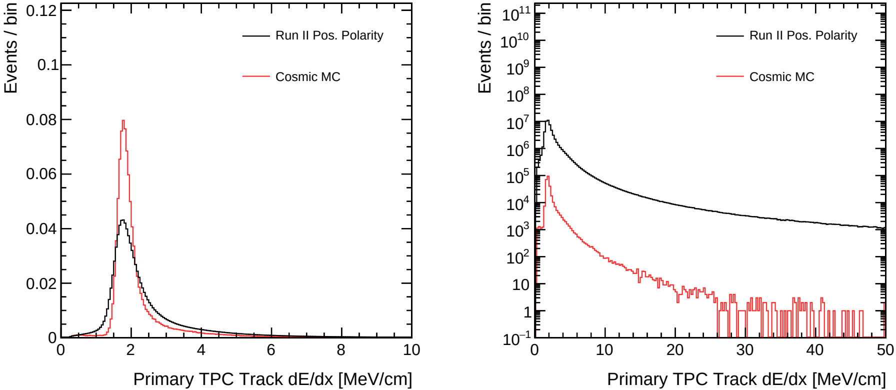
Selected the longest TPC track as “primary track”

Primary Track Angle



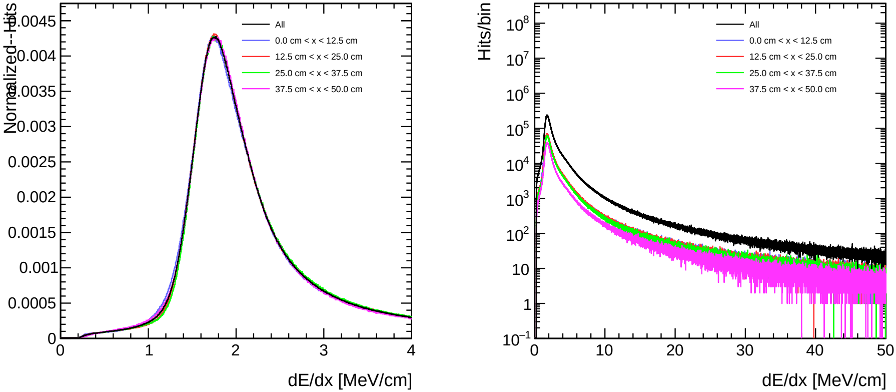
MC doesn’t reproduce events going through cosmic paddles

Cosmic dE/dx



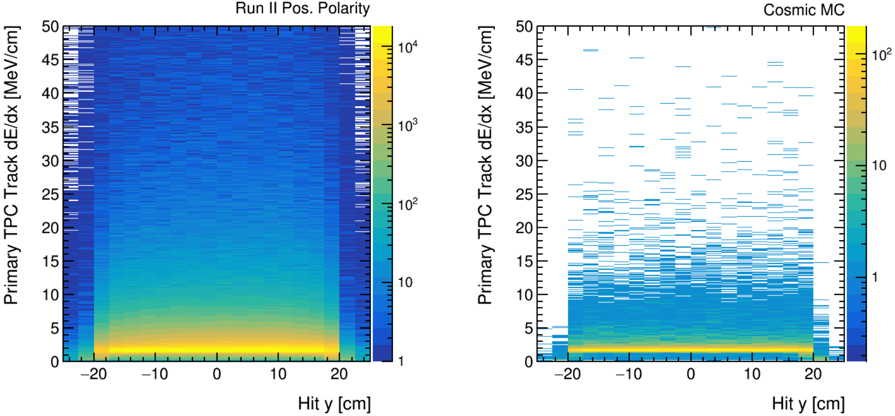
Wider in data as others have seen

Cosmic dE/dx for x slices



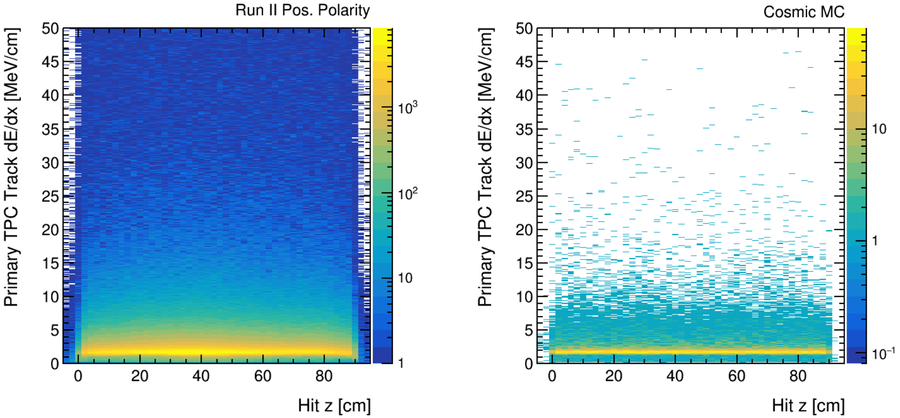
I thought this might be causing the smearing in data, but no

Cosmic dE/dx versus y



Maybe we should be careful of hits at edge of TPC

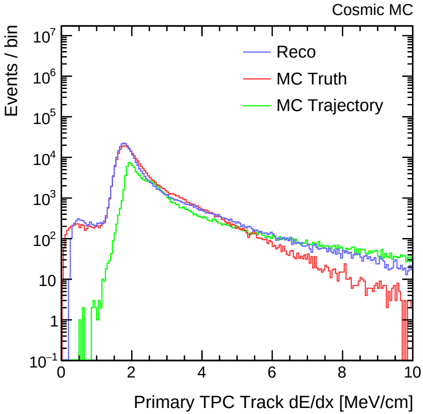
Cosmic dE/dx versus z



Maybe we should be careful of hits at edge of TPC

MC Cosmic dE/dx Reco v True

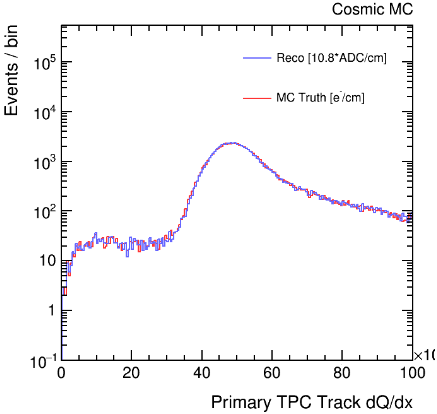
- Went and got the true energy deposit from sim::IDEs (lots of trouble)
- I called this “MC Truth” on plot
- Doesn’t seem much smearing from true to Reco
- I think computing dE/dx from MCTrajectory doesn’t work (points are too far apart)



Is our MC not smeared enough?

MC Cosmic dQ/dx Reco v True

- Went and got the true charge arrive on wires from sim::IDEs (lots of trouble)
- I called this “MC Truth” on plot
- Doesn’t seem that it is smeared at all



Are our MC electronics perfect?

Conclusions

- With a little work, you can have access to more computing resources on the OSG. See:

  https://redmine.fnal.gov/redmine/projects/lardbt/wiki/Setting_up_the_Offline_Software_CVMFS

  https://github.com/jhugon/lariatgridconfigs/blob/tutorial/osg_test.xml

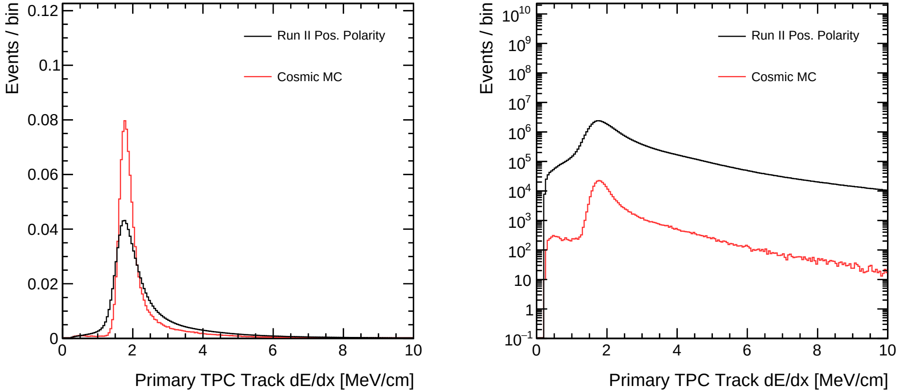
- Cosmic dE/dx: data wider than MC as others have shown
- MC: are we neglecting some smearing of charge in our simulation?

Backup

Cosmic MC Generation fcl

```
physics.producers.generator.PDG: [-13]
physics.producers.generator.X0: [25]
physics.producers.generator.Y0: [50]
physics.producers.generator.Z0: [45]
physics.producers.generator.T0: [0]
physics.producers.generator.SigmaX: [40]
physics.producers.generator.SigmaY: [0]
physics.producers.generator.SigmaZ: [75]
physics.producers.generator.SigmaT: [0]
physics.producers.generator.PosDist: 0
physics.producers.generator.P0: [2.0]
physics.producers.generator.SigmaP: [0.5]
physics.producers.generator.PDist: 0
physics.producers.generator.Theta0XZ: [0]
physics.producers.generator.Theta0YZ: [-90]
physics.producers.generator.SigmaThetaXZ: [180]
physics.producers.generator.SigmaThetaYZ: [30]
physics.producers.generator.AngleDist: 0
```

Cosmic dE/dx



MC doesn't reproduce events going through cosmic paddles