Pandora CI

Using the CI

First, do some basic setup (since the CI bits are in their own repo)

```
# Source the setup script and setup lar_ci
source /cvmfs/fermilab.opensciencegrid.org/products/common/etc/setups.sh
setup lar_ci

# Get a valid kerberos ticket for jobs, and a cert for CI jobs
kinit
setup cigetcert
cigetcert -s fifebatch.fnal.gov

# Setup voms proxy...
voms-proxy-init -rfc -noregen -voms dune:/dune/Role=Analysis

# Finally, authenticate and store the CI cert
export CI_CERT=/tmp/ci_cert.pem
kx509 -o ${CI_CERT}
```

Now, the current shell env should be setup and be able to kick off a CI job. The syntax for this is:

```
trigger --cert ${CI_CERT} \ # Use the CI cert we got previously
        --build-delay 0 \ # Use no build delay
        --version rcross_pandora_dunefd \ # What branch of the lar_ci repo to use.
        --workflow CI_VALIDATION_DUNE \ # Which of the workflow files to run
        --jobname dune_ci \ # Jenkins job name
        --gridwf-cfg cfg/grid_workflow_DUNE_pandora_reco_eff.cfg \ # Config to use
        # Optionally
        --revisions "dunetpc@feature/123" \ # Repository resolutions, info below.
        --testmode \ # Run in the Test CI, not the live one.
```

In the future, the `--version` flag will not be needed, once the Pandora specific portions are merged into `lar_ci` proper. This will kick off a build that can then be tracked over at http://lar-ci-history.fnal.gov/LarCI/app/ns:dune/view_builds/.

The `--testmode` flag, will run the CI in the test CI instance (https://dbweb6.fnal.gov:8443/TestCI/app/ns:dune/view_builds/index), rather than the "real" CI system.

The `--revisions` flag allows other remotes to be used. It accepts wildcards and more, but the most useful part is resolving against GitHub as needed, like so:

```
--revisions "PandoraPFA/LArContent>larpandoracontent@feature/larpandoracontent_v03_22_07"
```

I.e. resolve `larpandoracontent` to `PandoraPFA/LArContent` on GitHub, using the `feature/larpandoracontent_v03_22_07` branch.

Full Setup / Making New Reference Files

The Pandora CI utilises pre-processed files. This means we can run over more events, for higher statistic plots, but also uncouples us somewhat from upstream changes. That is, a change in a plot should only really come from a Pandora change, not from a change to the inputs to Pandora. Upstream changes should be picked up outside of this CI workflow.

To this end, a little bit of file preparation is required, to get the pre-processed files in place, as well as getting the reference file sorted as normal. The full process is as follows. Example config files can be found at the end.

1. Kick off 3 sets of jobs: nu, nue, nutau. An example XML is below, but really the important part is to do the normal chain, up to the reconstruction step. Then, run the `up_to_pandora.fcl` file, and `after_pandora.fcl` file independently, so you have the results from the before Pandora bit saved. These files are the basic reco configuration file, but overwritten to only run every step before Pandora, and then only Pandora. It is also possible to trim the files down to remove any other modules that are not needed to run Pandora. Finally, run the analysis module to get the actual analysis TTree out.

2. Move the partially reconstructed files into their correct location. This location is set in the `[reco]` bit of the CI config file, plus `sim`. At the moment, that is `/pnfs/dune/persistent/users/rcross/ci_validation/pandora/v09_15_00/sim`,

but that will change when the reference files are re-made. Additionally, these files need a very specific name, of `prodgenie_dune10kt_1x2x6_up_to_pan_X.root` where X is a 0 padded number, padded to the length of the number of jobs. With the default config this is 400 jobs, so 000, 001, 002, 003, etc. The current convention has been to have 150 nu, 150 nue, 100 nutau, but this does not fully matter (outside of more taking longer and this is running in the CI so nothing too crazy).

3. With the partially reco'd files in place, next the reference file is needed. `hadd` together all the outputs you care about (i.e. the fully reco'd and analysed set corresponding to the files used in step 2). With this large combined file, run it through the `PandoraRecoEffPlots.C` to produce a set of plots and a `ci_validation_histos.root` file. This is the reference histogram that the CI will compare against, and it lives in a `validation` folder, next to the `sim` folder from step 2.

At this point, the pre-processed files are in place with the correct names, and the reference file should be in the correct place as well. Finally, if you have updated the reference version, make sure to update the reference version defined in the CI configuration file in `lar_ci`, so that any paths are generated correctly.

A test run can be kicked off as shown in the section above, using the `--testmode` flag as well. **Don't try to test against the same version of dunetpc. The current CI will fall over if you attempt to compare against the same version due to how file names and paths are generated.**

Example Grid Job File

Example config to run the 3 steps (sim/pre_pan/pan_ana). The `<local>` tag may not be needed if the code being used lives in `dunetpc` proper.

```xml
<?xml version="1.0"?>

<!-- Production Project -->

<!DOCTYPE project [
    <!ENTITY relsim "v09_15_00">
    <!ENTITY relreco1 "v09_15_00">
    <!ENTITY file_type "mc">
    <!ENTITY run_type "physics">
    <!ENTITY flavour "nu">
    <!ENTITY name "prodgenie_&flavour;_dune10kt_1x2x6">
    <!ENTITY tag "A">
]>

<job>

<project name="&name;">

  <!-- Group -->
  <group>dune</group>

  <!-- Project size -->
  <numevents>10000</numevents>

  <!-- Operating System -->
  <os>SL7</os>

  <!-- Batch resources -->
  <resource>DEDICATED,OPPORTUNISTIC,OFFSITE</resource>

  <!-- LArSoft information -->
  <larsoft>
    <tag>&relsim;</tag>
    <qual>e19:prof</qual>
    <local>/pnfs/dune/persistent/users/rcross/larsoft_tars/pandora_ana_dunetpc.tar</local>
  </larsoft>
```

```
  <!-- Validate on worker -->
  <check>0</check>

  <!-- Project stages -->

  <stage name="gen">
    <fcl>prodgenie_&flavour;_dune10kt_1x2x6.fcl</fcl>
    <fcl>standard_g4_dune10kt_1x2x6.fcl</fcl>
    <fcl>standard_detsim_dune10kt_1x2x6.fcl</fcl>

    <outdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relsim;/detsim/&name;</outdir>
    <logdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relsim;/detsim/&name;</logdir>
    <workdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relsim;/detsim_work/&name;</workdir>
    <output>prodgenie_&flavour;_dune10kt_1x2x6_${PROCESS}_%tc_detsim.root</output>

    <numjobs>200</numjobs>
    <disk>5GB</disk>
    <memory>1000</memory>
    <jobsub>--expected-lifetime=3h</jobsub>

    <datatier>detector-simulated</datatier>
    <defname>&name;_&tag;_detsim</defname>
    <schema>root</schema>
  </stage>

  <!-- file type -->
  <filetype>&file_type;</filetype>

  <!-- run type -->
  <runtype>&run_type;</runtype>

</project>

<project name="&name;_pre_pand">

  <!-- Project size -->
  <numevents>10000</numevents>

  <!-- Operating System -->
  <os>SL7</os>

  <!-- Batch resources -->
  <resource>DEDICATED,OPPORTUNISTIC,OFFSITE</resource>

  <!-- Larsoft information -->
  <larsoft>
    <tag>&relreco1;</tag>
    <qual>e19:prof</qual>
    <local>/pnfs/dune/persistent/users/rcross/larsoft_tars/pandora_ana_dunetpc.tar</local>
  </larsoft>

  <!-- Additional FCL Files -->
  <fcldir>/dune/app/users/rcross/batch_fcl/</fcldir>

  <!-- Validate on worker -->
  <check>0</check>

  <!-- Project stages -->
  <stage name="reco1">
    <fcl>reco_up_to_pandora.fcl</fcl>
```

```
    <outdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relreco1;/pre_pan/&name;</outdir>
    <logdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relreco1;/pre_pan/&name;</logdir>
    <workdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relreco1;/pre_pan_work/&name;</workdir>

    <numjobs>200</numjobs>
    <disk>5GB</disk>
    <memory>2000</memory>
    <jobsub>--expected-lifetime=10h</jobsub>

    <datatier>hit-reconstructed</datatier>
    <defname>&name;_&tag;_pre_pan</defname>
    <schema>root</schema>
  </stage>

  <!-- file type -->
  <filetype>&file_type;</filetype>

  <!-- run type -->
  <runtype>&run_type;</runtype>

</project>

<project name="&name;_post_pand">

  <!-- Project size -->
  <numevents>10000</numevents>

  <!-- Operating System -->
  <os>SL7</os>

  <!-- Batch resources -->
  <resource>DEDICATED,OPPORTUNISTIC,OFFSITE</resource>

  <!-- Larsoft information -->
  <larsoft>
    <tag>&relreco1;</tag>
    <qual>e19:prof</qual>
    <local>/pnfs/dune/persistent/users/rcross/larsoft_tars/pandora_ana_dunetpc.tar</local>
  </larsoft>

  <!-- Additional FCL Files -->
  <fcldir>/dune/app/users/rcross/batch_fcl/</fcldir>

  <!-- Validate on worker -->
  <check>0</check>

  <!-- Project stages -->
  <stage name="reco2">
    <fcl>reco_from_pandora.fcl</fcl>
    <fcl>run_pandora_ana.fcl</fcl>

    <outdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relreco1;/complete/&name;</outdir>
    <logdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relreco1;/complete/&name;</logdir>
    <workdir>/pnfs/dune/scratch/users/rcross/ci_complete/&tag;/&relreco1;/complete_work/&name;</workdir>

    <numjobs>200</numjobs>
    <disk>5GB</disk>
    <memory>2000</memory>
    <jobsub>--expected-lifetime=8h</jobsub>
```

```
    <datatier>full-reconstructed</datatier>
    <defname>&name;_&tag;_post_pan</defname>
    <schema>root</schema>
  </stage>

  <!-- file type -->
  <filetype>&file_type;</filetype>

  <!-- run type -->
  <runtype>&run_type;</runtype>

</project>

</job>
```