CI test in LArSoft
Status report

LArSoft Coordination Meeting

September, 22nd 2015

Vito Di Benedetto

Overview

- **LarSoft CI tests goal.**
- **Status of LArSoft CI tests for all experiments.**
- **LArSoft CI tests statistics.**
- **Future plans.**

LArSoft CI tests goal

- **Test LArSoft to ensure it**
  - fulfills the requirements for which it was designed (for all target experiments).
  - meets quality expectations.
    - reliability, performance, interoperability...
  - meets experiments expectations.
    - does it do what we want? Will it produce the physics results we want?
- **Get all target experiments involved on this activity**
  - Test LArSoft and experiment code through regression test.
  - Use the experiment specific workflow.
- **Regression test strategy**
  - Existing tests run against modifed/updated code.
  - Check whether changed code breaks anything that worked prior of the change.
- **Independent tests can run in parallel**
  - Where needed, split the experiments workflow making it the most granular possible.
  - Each “elementary” step of the work flow can be tested independently.
  - Users can get fast feedback from the test results.

LArSoft CI tests goal

- **Get all target experiments involved on this activity**

Experiments contacted:

- **ArgoNeuT**
- **DUNE35T**
- **LArIAT**
- **MicroBooNE**
- **SBDN**

- Each of them provided their production chain.
- Some work done with ArgoNeuT and SBDN to reorganize the FhiCL files used to implement their workflow in the CI test.

LArSoft CI regression test

- I Implemented a regression CI test suite
- It is published in the feature branch

feature/vdb_ci_regression_test_suite for all experiment repositories (argoneutcode, dunetpc, lar1ndcode, lariatsoft, uboonecode)

- Implementation details:
  - Run updated/modified experiment code to generate “current” data files
  - Use official FHiCL files with some seed option added for CI test
    - Use fixed seed and/or read seed from an external file to ensure data production reproducibility (not always possbile!)
  - Compare “current” data files against “reference” data files generated for this purpose (reference data filename contains LArSoft tag used to generate them)
    - Reference files are sitting on the experiment dCache area.
    - Usually I run tests on the experiment NFS area.

Status of CI regression test

➢The test process a **single event** using the default experiment configuration

Experiments	test implemented	#tests passed
● **ArgoNeuT**	✔	**3/4 (*) [reco]**
● **DUNE35T**	✔	**4/5 (*) [reco]**
● **LArIAT**	✔	**2/3 (*) [reco2D]**
● **MicroBooNE**	✔	**4/6 (*) [detsim, reco1]**
● **SBDN**	✔	**3/4 (**)(reco step can't run)**

(*) The reference data files were created with LArSoft v04_20_00. The test has been performed using LArSoft v04_22_00.

- In the indicated stages the experiment code has some modified and/or added data product(s).
  - ➡ In DUNE35T there are added data products due to new pandora (v02_01_00) used in LArSoft v04_22_00.

(**) SBND recently updated their code to LArSoft v04_00_00. At the moment there is some issue in the reco stage. They are aware of this issue.

CI regression test statistics

Experiment	qualifier	stage	Time (min)	Memory (Mb)
MicroBooNE	prof v04_14_00	prodgenie	2:04	591
		g4	2:36	1665
		detsim	0:59	1190
		reco1	1:13	1338
		reco2	1:55	1410
		analysis	0:47	1410
	prof v04_22_00	prodgenie	2:15	609
		g4	2:53	1840
		detsim	1:17	1392
		reco1	1:35	1464
		reco2	2:46	1506
		analysis	0:56	1508

Tests executed on uboonegpvm03

- Simulated a ~1.8GeV ν_μ interaction with cosmic rays.
- MicroBooNE code uses similar resources in both LArSoft releases.
- Each test runs well within the target time upper limit (10 min)

CI regression test statistics

Experiment	qualifier	stage	Time (min)	Memory (Mb)
DUNE35T	prof v04_14_00	prodantimu	0:09	89
		g4	0:18	336
		detsim	0:13	164
		reco	0:15	252
		analysis	0:27	4543
	prof v04_22_00	prodantimu	0:09	89
		g4	0:13	215
		detsim	0:13	152
		reco	0:27	329
		analysis	0:14	7664

Tests executed on lbnegpvm03

- Simulated a ~1.6 GeV single $\bar{\mu}$ interaction.
- DUNE35T code uses similar resources in both LArSoft releases.
- Each test runs well within the target time upper limit (10 min)

CI regression test statistics

Experiment	qualifier	stage	Time (min)	Memory (Mb)
LArIAT	prof v04_14_00	fragment	0:11	204
		reco	0:16	211
	prof v04_22_00	slicer	0:14	165
		beamlinereco	0:09	101
		reco2D	0:12	134

Tests executed on lariatgpvm03

- Procesed 1 real data event taken at FTBF, MCenter beamline, 16GeV beam.
- LArIAT changed FhiCL files between v04_14_00 and v04_22_00.
- Each test runs well within the target time upper limit (10 min).

CI regression test statistics

ArgoNeuT and SBND FHiCL files used for this CI test are splitted to allow independ checks for each simulation step

Experiment	qualifier	stage	Time (min)	Memory (Mb)
ArgoNeuT	prof v04_14_00	sim	1:33	682
		reco	0:08	190
	prof v04_22_00	gen	1:29	636
		geant	0:13	140
		detsim	0:09	101
		reco	0:10	192

Tests executed on argoneutgpvm03

- Simulated a ~2GeV ν_μ interaction.
- ArgoNeuT uses similar resources in both LarSoft releases.
- All tests run well within the target time upper limit (10 min).

Experiment	qualifier	stage	Time (min)	Memory (Mb)
SBND	prof v03_08_02	sim	2:07	540
		reco	0:35	713
	prof v04_00_00	gen	0:13	87
		geant	2:06	364
		detsim	0:14	278
		reco	Not Available	Not Available

Tests executed on lar1ndgpvm01

- Simulated a ~2GeV single **μ** interaction.
- SBND uses similar resources in both LarSoft releases.
- All tests run well within the target time upper limit (10 min).
- reco stage can't run because of some issue, reco statisctic not yet available.

Future plans

- Make sure of the event reproducibility for all experiments:
  - Make sure that
    services.SeedService.policy: "preDefinedSeed"
    works for all simulation stages.
  - Make sure that
    services.RandomNumberGenerator.restoreFrom: "<stage>RandomSeeds_Ref.dat"
    services.RandomNumberGenerator.saveTo: "<stage>RandomSeeds.dat"
    works for all simulation stages.

- Continue to work on a CI test in LarSoft to ensure that a new version of art code does not break LarSoft:
  - Build art with its dependencies.
  - Build LArSoft with its dependencies on the top of the fresh build of art.
  - Use the existing CI regression test to test the fresh build of LArSoft.

- Implement new CI test suites (?)

Backup slides

CI test implementation details

Test example:

srcs/uboonecode/test/ci/ci_tests.cfg excerpt

```
[DEFAULT]
STEPS = none gen geant detsim reco1 reco2 ana
LARSOFT_REFERENCE_VERSION=v04_20_00
BASEFILENAME=prodgenie_bnb_nu_cosmic_uboone
EXPCODE=uboonecode
EXPSCRIPT=ci_regression_test_uboonecode.sh
INPUTFILEDIR=/pnfs/uboone/scratch/users/vito/ci_tests_inputfiles
```

Experiment specific configuration

```
[test ci_gen_regression_test_uboonecode]
script=${UBOONECODE_DIR}/test/%(EXPSCRIPT)s
STEP=1
NEVENTS=1
args=%(NEVENTS)s %(STEP)s %(LARSOFT_REFERENCE_VERSION)s %(BASEFILENAME)s %(EXPCODE)s %(STEPS)s
inputfiles=%(INPUTFILEDIR)s/%(BASEFILENAME)s_Reference_gen_%(LARSOFT_REFERENCE_VERSION)s.root %
(INPUTFILEDIR)s/GenRandomSeeds_Ref.dat
parse_art_output=True
mem_usage_range=100:20000000
cpu_usage_range=10:60000
...
```

CI test section

```
[suite quick_test_uboonecode]
testlist=ci_gen_regression_test_uboonecode ci_geant_regression_test_uboonecode ci_detsim_regression_test_uboonecode
ci_reco1_regression_test_uboonecode ci_reco2_regression_test_uboonecode ci_ana_regression_test_uboonecode
```

CI test suite section

- The script to run the test is the same for all experiments.
- The “experiment specific section” in the ci_tests.cfg sets all required input to properly initialize the script.
- The “CI test section” sets further arguments for the specific CI test.
- The “CI test suite section” collects a list of tests to run all together.