DUNE FD/Prototypes Software Status

Tom Junk

DUNE Software Architecture and Management

October 25, 2024

Welcome Andrew Olivier!

- He's already made two releases of dunesw
  - v09_92_00d00 (September 20)
  - v10_00_02d00 (October 18) (!)
- Lynn Garren has announced LArSoft v10_00_03 on Oct 18 v10_00_04 on October 21
- no dunesw corresponding to larsoft v10_00_00, v10_00_01, and we'll likely skip v10_00_03.
- We'll head straight on to v10_00_04. Incremental changes in the point releases.

LArSoft v10 – Breaking Geometry Changes

- Main change – separation of the wire readout geometry from the geometry service, into its own service.
- The old geometry service "did everything".
- Readout geometry is already parameterized – wire lookups are done formulaically instead of looking them up in the GDML representation.
- Including pixels is a further step along this direction – A service that defines wires should not be required to work with a pixel detector.
- Each pixel need not be known to GEANT (memory explosion).

LArSoft v10 Story

- Four release candidates of larsoft v10, starting late 2023
- Kyle Knoepfel did the coding work, with regular design discussions with Erica. Tom and Tingjun participated in design meetings last year
- Kyle provided PRs for each affected experiment upgrading their code to the new API
- Request went out late 2023, early 2024 for experiments to test the new refactored geometry.
- Tom was busy, but built it in the spring, and got a FD workflow running with Dom Brailsford's help.
- Dom and Tom ran with the new v10 release candidates and identified several issues.

LArSoft v10 testing with DUNE

- Dom had a FD 1x2x6 single-muon workflow test that at least didn't crash.
- Wire intersection calculator did not return correct optional false value when wires don't intersect (new use of std::optional)
- Parts of tracks not reconstructed in the right place – problem traced to drift sign convention inconsistency. Kyle fixed it quickly.
- Separate problem found with event display – the UPS product range seemed to want c++20 conventions but it wasn't strictly necessary.
- VD workflow also simply crashed. Problem traced to FCL not being updated.
  - Problem was the AuxDet geometry was not set and inherited a value from a non-overrided @local that set the geometry for SP.
  - TGeoManager can handly only one geometry at a time – once you read in a GDML file, there's no going back. So we had the wrong GDML file for the VD workflow. Problem not entirely obvious but later we knew what to look for.

More v10 issues, discovered after signoff and release of LArSoft v10

- With the two workflows working, and Erica eager to get the release out, Tom and Dom agreed for v10 to come out and we'd chase remaining problems.
- CI integration tests are very useful! Tom ran them interactively to speed debugging.
- PANDORA was throwing exceptions in the ProtoDUNE-DP workflow
  - Only two planes, but the view numbers are not kU and kV (0 and 1), but rather kX and kY (2 and 3).
  - Code in PANDORA (and a few other places in larreco) was using "view" instead of "plane", which works "most of the time"
  - Simple fixes found to look up things based on view and not plane.

More v10 issues, discovered after signoff and release of LArSoft v10

- The ProtoDUNE detector properties service separately defined drift directions – the initial bugfix that Kyle made to the LArSoft detector properties service was not propagated to the ProtoDUNE one.
- Hits and tracks were reconstructed at the wrong x locations
- Also T0 data products were empty (consequence of wrong x locations)
- Disambiguation by space points had a new bug introduced (looking in wrong TPC).
- Problems were identified via the CI integration tests
- After fixing the sign bug, anode-piercing T0's were still missing
- simple coding bug (forgotten variable set) in updated routine spotted and fixed.

A Small Problem with PDSP Decoding

- PDSP data reco integration test results were not exactly reproducible from one run to the next.
- Problem traced to FEMB 302 (the one with the broken clock line).
- Decoder returns 5996 ticks instead of 6000 because clock is slow.
- David Adams resampled the 5996 to stretch them into 6000 in dataprep
- Problem is, ticks 5992-5995 contained corrupt data that changed when the program was rerun. So we are missing 8 ticks and not just 4.
- Stretching already validated. Didn't want to disturb existing physics, so ADC(5991) is now copied in to 5992-5995 if nticks==5996
- PDSP CI reco data product counts now exactly reproduce, and also compare exactly between v9 and v10.

Other Loose Ends

- Some code was newer than Kyle's PRs and worked with the old API.
- Not too much of it, and the biggest piece was actually copied from SBND, where Kyle had put a modified copy in his PR.
- User analysis code and scripts may now be out of date
  - Example wire endpoint dumper code I copy from the geometry wiki in Redmine needs another refresh.
- People should be extra vigilant!
- José Soto found a problem with the PDHD AuxDet configs

Even though the right GDML was used (didn't clobber the TPC geometry), additional FCL configs were needed to get the number of optical detector channels right. José fixed the problem himself!

Better test coverage would help in this matter!

dunetrigger

- Wes Ketchum gave me a heads-up about a DUNE trigger simulation product he's been working on:

  https://github.com/wesketchum/dunetrigger

- It depends on dunecore, though some of the example fcls require other things (duneprototypes, dunesw)
- It would need updating to the new geometry API – just a couple of calls need to be adjusted.
- Wes would like to include it in the dunesw stack.
- Would need a dune repo (and protected develop branch), a Spack recipe, a home in the SciSoft web server, and updates to some scripts that loop over dune products.

Spack Status

- Kyle Knoepfel announced the beta test version of Spack MPD at the collaboration meeting in Santa Fe
- Tom tried out the commands from Kyle's talk and ran into a problem with an undefined environment.
- Kyle helped – explained how to create your own environment

- Patrick Gartung has updated the DUNE recipes in dune_spack to v09_92_00 (larsoft spack releases are v09_90_00 and v10_00_03)

https://github.com/dune/dune_spack

- And Tom tried building a stack using regular Spack tools (not Fermi ones) on dunebuild03 starting with lardataobj – a long compile ensued, and stopped after undefined symbol for libintl_gettext compiling cxpm.c

Putting data files in Git

- A "minor infraction" https://github.com/DUNE/duneopdet/pull/61

contains seven data files, each with 1024 lines with one floating-point number in text form. Around 8 kB/file.

But ... it may be the beginning of a large set – one per optical detector, perhaps. Though Laura says we don't have plans for one per opdet in the "near future".

- Alternatives: StashCache, or a database
- Still have some IFDH transfers in framework jobs (like the PDSP event display!) These can break when the external environment evolves and a new IFDH fixes it but we set up an old version in an old DUNE release

PDSP use of IFDH

- spdp::DetectorPropertiesProtoDUNEsp::UpdateReadoutWindowSize calls ifdh_ns::ifdh::getMetadata
- Hangs on my desktop in dunesw v09_91_04d00 which sets up ifdhc v2_7
- Setting up dunesw v10_00_02d00 which sets up ifdhc v2_7_2 lets that work.
- setting up dunesw v09_91_04d00 and unsetting up ifdhc, setup ifdhc v2_7_2 gives a sqlite error when running a program.