DUNE Software Management Projects

Tom Junk

DUNE Core Computing Meeting

June 14, 2021

A Sampling of Projects

- Mostly just maintenance and service list.
- Other projects, like moving the FD simulation to use the refactored LArG4 and using better disambiguation are software development projects rather than software management projects.

Projects:

- Bring ND_Production under version control and distribute in CVMFS
- DAQ data-format integration
- Split DUNETPC (we keep saying this...)
- Move code repositories and wikis to GitHub

ND_Production

- Becoming important on a short timescale
- Mat Muether (via Slack): "It seems one final step is to tag the relevant code, and build into a cvmfs area."
- https://github.com/DUNE/ND_Production
- Production scripts and configuration files (.xml, .cfg). No build required.
- Job script run_everything.sh sets up some products in UPS, copies five tarballs from persistent dCache for others.
- No versioning of input tarballs – reliance on the right ones in the input directory.

Input tarballs for ND run_everything.sh

ls -lrth /pnfs/dune/persistent/users/LBL_TDR/sw_tarballs/

total 59M

-rw-r--r-- 1 marshalc dune 24M Apr 23 2020 edep-sim.tar.gz
-rw-r--r-- 1 marshalc dune 15M Apr 23 2020 nusystematics.tar.gz
-rw-r--r-- 1 marshalc dune 2.2M Apr 23 2020 nusyst_inputs.tar.gz
-rw-r--r-- 1 marshalc dune 6.6M May 8 2020 DUNE_ND_GeoEff.tar.gz
-rw-r--r-- 1 marshalc dune 988K Dec 3 2020 sim_inputs.tar.gz
-rw-r--r-- 1 marshalc dune 9.6M May 14 09:09 larcv2.tar.bz2
-rw-r--r-- 1 marshalc dune 1017K Jun 4 14:05 sim_inputs_larval.tar.gz

ND_Production Versioning Options

List ordered easiest to hardest.

1. Make a new ND_Production UPS product out of the contents of the ND_Production repository and tag and release it in CVMFS and scisoft.fnal.gov. Use as-is, meaning relying on the right tarballs being in the persistent dCache directory. There are issues with long-term reproducibility if the tarballs change. We could version the tarball directory.

2. Make a new ND_Production UPS product with the contents of the ND_Production repo as above, but include the matching tarballs in it.

3. Make a new ND_Production UPS product that depends on specific versions of other products so UPS sets them up automatically when ND_Production gets set up. These would include dk2nugenie, genie_xsec, genie_phyopt, geant4, edepsim, and nusystematics, which are already UPS products but possibly not in the desired versions, and some further explanation of what's in nusyst_inputs and DUNE_ND_GeoEff. Figure out why the MakeProject in dumpTree.py won't work when edep-sim is in LD_LIBRARY_PATH, so the environment doesn't have to change during a batch job, or so that the production workflow can work with just one environment.

4. Do the above, but with Spack. At least there is a UPS2Spack tool but only Marc Mengel has run it to my knowledge.

Personal Repositories vs. DUNE-owned?

- DUNE_ND_GeoEff is a repository owned by Cristovao Vilela and is used for DUNE-PRISM work. Built and tarred up and included in run_everything.sh
- Fine in the (very) short term, just getting things hacked together, but code run in production for DUNE ought to be in a collaboration-owned repository.
- Shields us from repository owners deleting their repo, or removing old tags.

DAQ Data-format Integration

- From an e-mail of Kurt's:
  For the higher-level data structures that the DAQ software adds to the raw data (i.e. pieces that are in the *dataformats* package), we are providing the necessary information to do that (and we can explain how to access the information, etc).
- I am not familiar with the mechanism(s) that the data produced by the electronics use to report missing data. I'll try to update my knowledge on that (but probably not before the meeting on Wed).

It might be worthwhile to separate our discussion tomorrow into parts:

- the packaging, building, dependencies, and ownership of the different software components
- the contents and behavior of those components

Some News about Data Formats

- Software-test hdf5 files are being written by the DAQ group
- Less header information in the HDF5 attributes – more information is being packed in the datasets themselves. Such as the run number. No subrun number for now.

Example TPC fragment header:

```
--------------------------------------------------------------------------------
Path : TriggerRecord00973/TPC/APA000/Link03
Size : (22352, 1)
Data type : int8
Magic word : 0x11112222
Version : 2
Frag Size : 22352
Trig number : 973
Trig timestamp : 81158439679669294 (2021-06-08 18:13:13.593386)
Window begin : 81158439679669294 (2021-06-08 18:13:13.593386)
Window end : 81158439679670294 (2021-06-08 18:13:13.593406)
Run number : 333
Error bits : 0
Fragment type : 1
GeoID type : TPC
GeoID region : 0
GeoID element : 3
--------------------------------------------------------------------------------
```

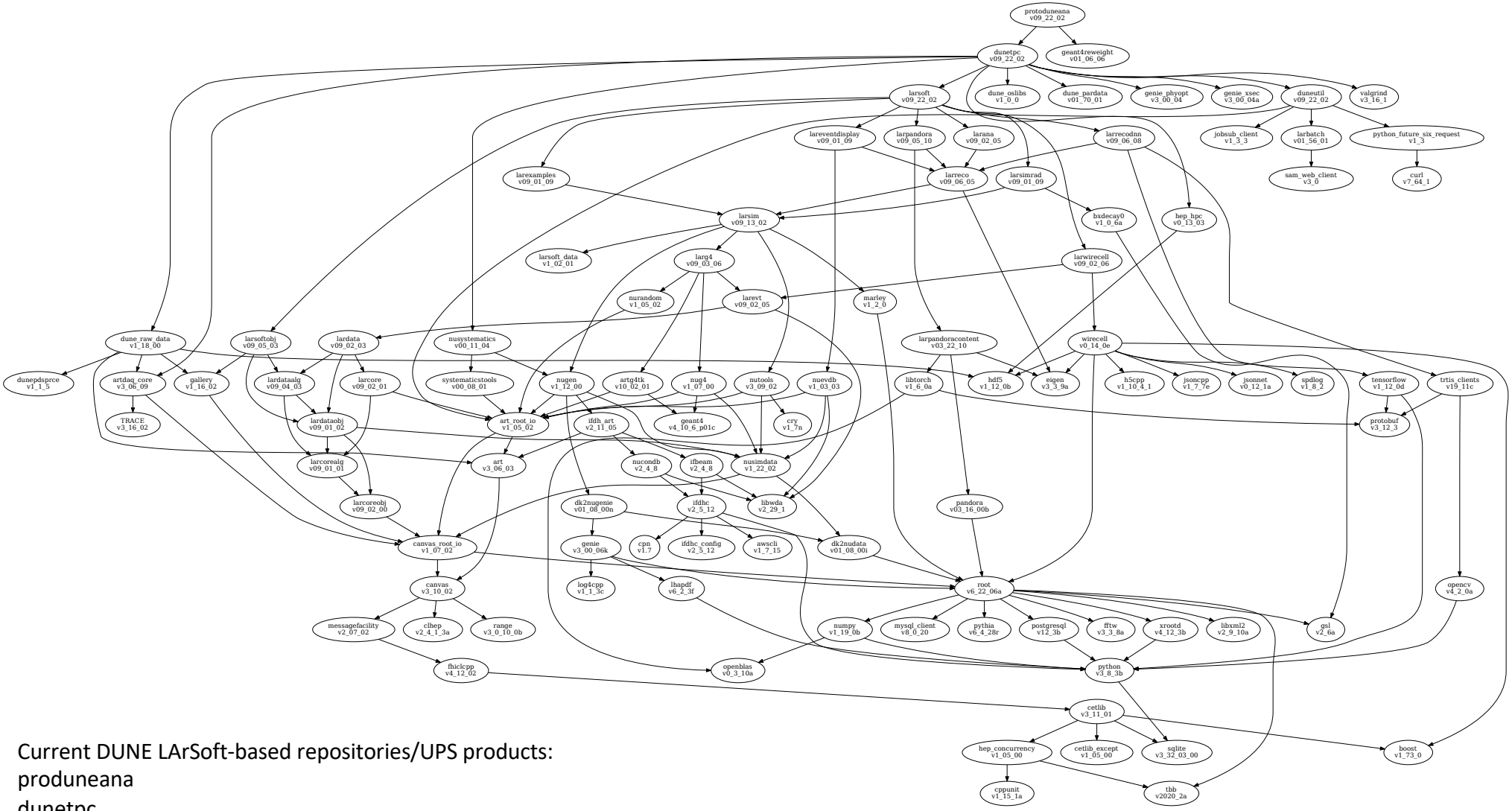
People Interested in Writing Interfaces to HDF5

- Tammy Walton
- Amit Bashyal
- Mike Kirby
- Tom Junk (using example from Kurt Biery)
- ROOT has some advantages (would like to find or write equivalents for HDF5):
  - xrootd can stream part of a file
  - browsability
  - schema evolution of data products
  - art's rootinput module has features we like (delayed reading)
- And some disadvantages
  - partial files are unreadable

Splitting dunetpc

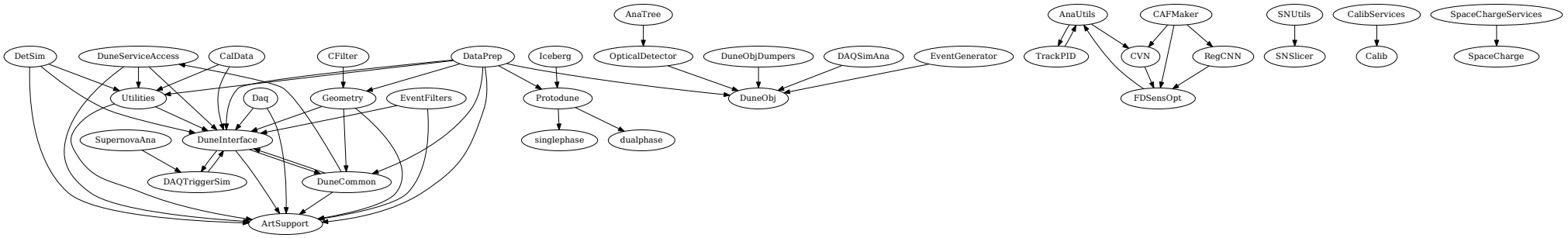
- We keep saying we'll do this.
- Doesn't add functionality but it would make the build faster.
- Factorized from Spack – we're not waiting for that to happen.
- Did some studies of which directories in dunetpc depended on which others, and which take the longest to build.
- David Adams's data preparation code is the biggest piece and also has the most dependencies.
- Removing analysis code from dunetpc does not speed the build up by much.

A Recent Dependency Graph



Current DUNE LArSoft-based repositories/UPS products:
produneana
dunetpc
duneutil
dune_raw_data
dunepdsprce
dune_pardata

#include dependency graph for dunetpc



- Not a DAG. There are circular dependences (at least with the includes)

Moving Repositories to GitHub

- GArSoft moved in May 24, 2021
- It was easier than I had expected.
- Some things that require attention:
  - Clean up old history: There was a 122 MB GDML file in there that had long been deleted, but it was in the history. Deleted.
    - larsim had this problem of cleaning old clutter but it got re-pushed back by users. I told garsoft people to start from a fresh test release. git complained about differing histories when pulling over a non-modified repo. github has a file-size limit and may reject mistaken push with old history. Let's see how this goes.
  - User list: you have to have a GitHub account to push. DUNE members can push, plus others we add by hand.
  - Update and transfer the wiki documentation (partly done).

And a Mini-Project – install forge_tools when it becomes available

- forge_tools v20_0_3 is incompatible with gcc v9_3_0 (e20)
  - Spotted first by Gianluca Petrillo
- dunetpc (and garsoft) now only builds e20 and c7
- New versions have been made available via upd – central creation of that product needed to get the licensing and UPS-ification right
- Experiments install forge_tools in their own CVMFS areas and declare a default (latest) version. Not hard to do – instructions are good. Backwards compatibility means we always want the latest version.
- There's a forge_tools v20_0_3 in the common db though.