Things DUNE Users Need To Be Able To Do with SPACK

Tom Junk

February 13, 2024

Time-Ordered List of Tasks

- Log on to an AL9 node
- Check to see if user environment is "clean"
- Verify installed environment setup script works
- Load the (an?) installed larsoft package in CVMFS

source /cvmfs/larsoft.opensciencegrid.org/spack-packages/setup-env.sh
spack find larsoft
spack info larsoft
spack load larsoft@09.81.00

  - **These all work!**
  - A problem already:

    spack load larsoft

    also works! Old UPS problem with "current" versions causing non-reproducibility in workflows. "current" versions apply to system-level lossless services, like ifdhc, *not* to physics code where versions affect calibration and results.

    See Junk and Lyons, "Reproducibility and Replication of Experimental Particle Physics Results", Harvard Data Science Review Fall 2020, Sec. 6.2
    https://hdsr.mitpress.mit.edu/pub/1lhu0zvn/release/4?readingCollection=c6cf45bb

    also: what to do about setting up the compiler. The compiler that comes with AL9 will someday be "old", like the SL7 one is now; gcc 4.8.5

Time-Ordered List of Tasks

- Verify if the *art* suite of tools works with the installed packages
  - art --help works!
  - product_sizes_dumper works!
  - cetskelgen does not work -- Can't locate Module/Pluggable.pm in @INC
- Check a basic larsoft workflow
  - lar -c eventdump.fcl <LArSoftartROOTfile>
  - Does not work – FHICL_FILE_PATH and the locations of fcl files are misaligned. Either the makefiles or the FHICL_FILE_PATH settings need to be changed. makefiles if we want to preserve the old "job" directory names from the LArSoft UPS products.

Time-Ordered List of Tasks

- Rebuild an *art* or a LArSoft package in a user's own disk space
  - Tutorial has examples: py-black, watch and art-root-io
  - Struggling with these
  - Can attempt this even if workflows are missing fcl files.
- Insert a print statement, build and run the user-rebuilt version of the package (and any other packages that come along for the ride, such as the larsoft environment).

- Make a new package in a user's own disk space.
  - Questions: Where to put the package.py and patch files? Do we need patch files? What yaml files need construction, editing, placing in the right places>

Time-ordered List of Tasks

- Make a DUNE package (in a DUNE environment)
  - start with duneutil
  - Also can work on header-only packages dunedaqdataformats, dunedetdataformats, or the data-only package dune_pardata
- Test distribution of this package in CVMFS – make a RC tag to not clobber actual production releases.
- Test the spack-ported DUNE package for functionality
  - duneutil has several standalone scripts (e.g. setup_fnal_security, which is relied upon by many collaborators)
- Test LArSoft workflows if the FHICL_FILE_PATH problem is resolved. (can happen at any time before testing DUNE LArSoft functionality). Use CI scripts to test more than just eventdump.fcl.

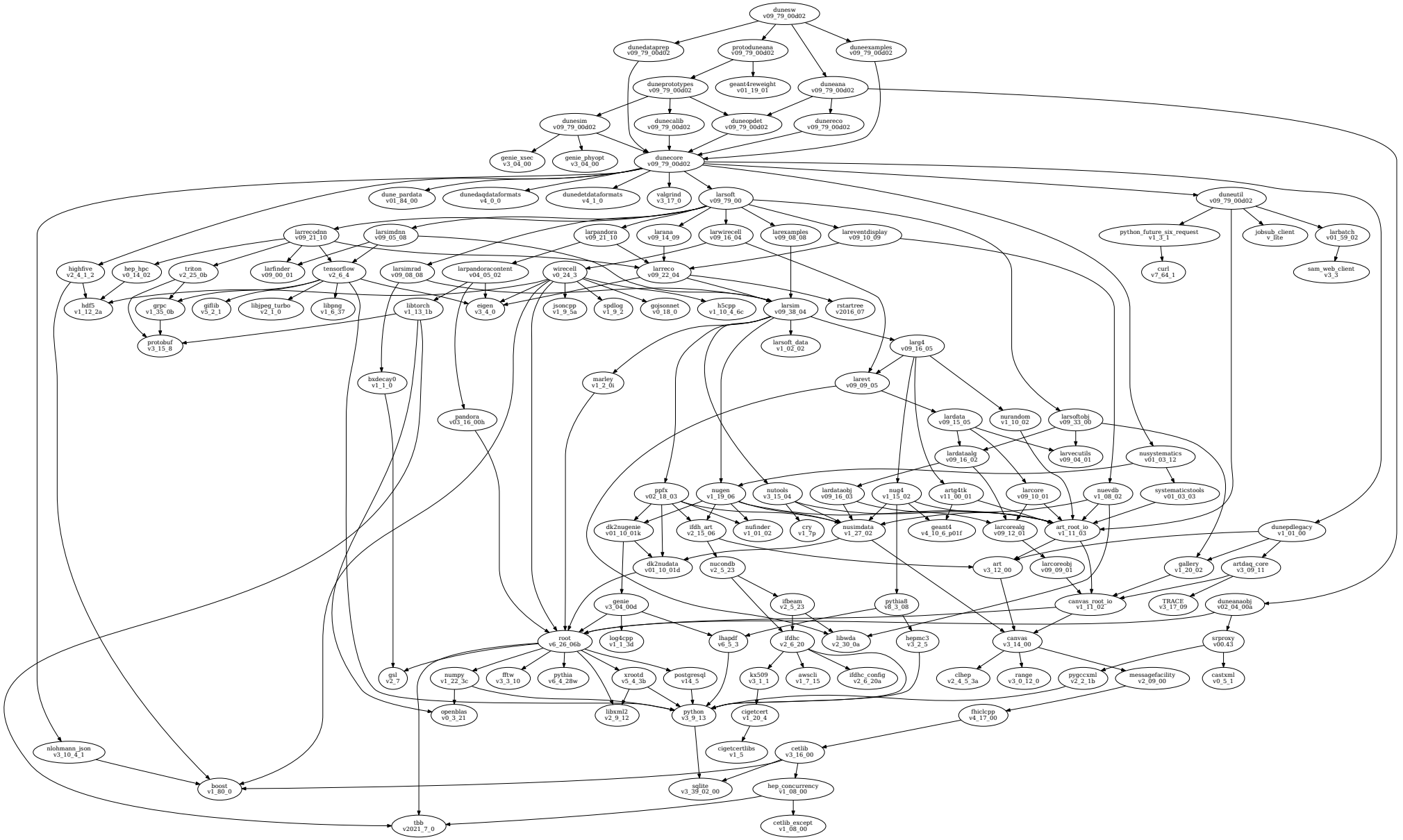
Time-Ordered List of Tasks

- Port remaining DUNE products to spack
- ups_to_spack can be useful, but a question – does this rely on the existence of a pre-built UPS product? We can only make these with SL7 and not AL9
- Test DUNE workflows using CI scripts
- DUNE physics groups should test performance – provide metrics for testing. Kirby has started the ball rolling on this one.
- Provide clang versions of packages. prof and debug (time to get rid of historical "prof" qualifier? "opt" and "debug" perhaps? "debug" = "noopt", but a grep for "opt" turns up "noopt" so distinct characters are best)
- Get the CI system back up for DUNE tests.

Documentation

- A continuous, parallel task! Every step should be documented, and documentation kept up to date.
- Tutorials for new and experienced DUNE users
- Tutorials need to be reproducible. Cut more releases as need be.

The dunesw tree from v09_79_00



Fermilab DUNE

Extras

Spack Resources

- https://fifewiki.fnal.gov/wiki/Spack
- https://spack.readthedocs.io/en/latest/basic_usage.html
- n.b. Spack documentation on the web is versioned. Some commands evolve and reading the wrong docs can cause confusion.
- LArSoft uses Spack 0.21.2dev0
- Get this with
  spack --version
- see output of spack –help
- Kyle gave a talk on Spack development at the LArSoft coordination meeting last week
  https://indico.fnal.gov/event/63013/

Work plan for Spack transition

- Test functionality of LArSoft Spack installation using AL9
- Test functionality of modifying a local copy of a LArSoft package
- Build the dunesw stack in Spack in a private disk area using the LArSoft installed packages
- Check basic functionality and iterate until known workflows work (e.g. MC jobs, ProtoDUNE data reco, and other workflows).
- Rebuild and install dunesw in CVMFS
- Perform physics validation
  - Kirby is contacting physics groups to ask them to think about and produce validation plots and criteria
  - It's not just the Spack transition that can benefit from this (compilers..)
  - Missing-file errors should be fatal, but code that falls back to an option may behave non-reproducibly

Setting up LArSoft in Spack

- Log on to dunegpvm16.fnal.gov

source /cvmfs/larsoft.opensciencegrid.org/spack-packages/setup-env.sh
spack find larsoft
spack info larsoft
spack load larsoft@09.81.00

n.b. /cvmfs/larsoft.opensciencegrid.org/packages contains old stuff that is only a distraction (tutorial is a bit out of date).

n.b. loading larsoft does *not* set up gcc like the UPS one does. Compiler may be needed by interactive ROOT sessions.

gcc version is 12.2.0, one point release newer than e26 = gcc 12.1.0

spack list gives a long list of packages, not all of which can be loaded.

Little Things ...

- I noticed Spack puts things in ${HOME}/.spack, as well as in /tmp/<username>/spack-stage

- These can be issues if $HOME is not defined (actually I didn't try undefining it – could be it uses ~), or if ones Kerberos ticket has expired. Or if /tmp doesn't exist or is not writeable.

- They seem not to be too important – I deleted my .spack directory and spack just made it again.
- Raises reproducibility questions if a build or a setup depends on the contents of these directories. (I am told they don't)

LArSoft Spack Environment

```
<dunegpvm16.fnal.gov> spack env list

==> 4 environments

    critic-2-12-04-gcc-12-2-0-cxx17-prof-gcc-11-4-1  larsoft-09-81-00-gcc-12-2-0-cxx17-prof-gcc-11-4-1
    gcc-12-2-0-gcc-11-4-1                            nulite-3-15-04-gcc-12-2-0-cxx17-prof-gcc-11-4-1
```

So.. what's the compiler version?

spack env activate larsoft-09-81-00-gcc-12-2-0-cxx17-prof-gcc-11-4-1

doesn't set up the compiler, but you have the option to load 11.4.1 or 12.2.0

I suspect the SciSoft team built gcc 12.2.0 using gcc 11.4.1

Executing the Work plan for Spack transition

1. Test functionality of LArSoft Spack installation using AL9

   e.g. lar -c eventdump.fcl <artrootfile>

This failed. FHICL_FILE_PATH points to nonexistent directories. Feedback to scisoft given. This may have been remedied with symlinks as a workaround for v09.82.00.

1a. Or more basically, check to see if these framework executables work (no LArSoft needed)

lar --help
product_sizes_dumper
cetskelgen

- These latter work! Except cetsklgen -- missing some Perl module.

Special dunsw products

- dunepdlegacy is a standard mrb-built product, just not released every week. Should be converted along with the rest.
- dunedetdataformats and dunedaqdataformats
  currently header-only products with some scripts and other non-built items. Should not be too hard to install these.
  Need to set up include paths so builds of other packages work
- dune_pardata is also a bespoke repo

- Not dunesw products but still in our management remit:
  edepsim
  nopayloadclient
  sandreco
  garsoft