Software Architecture/Management News and Announcements

Tom Junk
DUNE Software Management
November 11, 2022

New DAQ-Related UPS Products

- ers
  - ATLAS's Error Reporting System (or Service)
  - Commonly used by DUNE-DAQ software
  - Does not seem to have any external dependencies
  - Provides two kinds of functionality
    - Custom exceptions (and things like ERS_HERE so the exception prints out what threw the exception). Offline uses cet_exception. Portable code can use std exceptions
    - Logging: performs same functions as MF
- Status: DAQ group provided a Spack recipe and build. Last summer they were maintaining a UPS product.
- I took the Spack build and made a UPS product out of it using some of the files in the previous version and updating version numbers.
- Installed in DUNE CVMFS. Not tested yet. Product made "by hand" – i.e. no automatic build script. only e20:prof available.

New DAQ-Related UPS Products

- nlohmann_json
- Used by DUNE-DAQ's hdf5libs
- header-only product. No build required
- Used to make specifying JSON in C++ code look like JSON syntax.
- https://github.com/nlohmann/json
- Interest in DUNE for having a more full-featured JSON interface
- Status: DAQ version installed as a UPS product. (e20+c7) x (prof+debug) qualifiers are available
- Installed in CVMFS. Waiting for SciSoft directories still: RITM1529633

New DAQ-Related UPS Products

- highfive

- HDF5 interface from C++
- Useful because arranging to close objects in the C API requires extra work – C++ destructors can assure closure.
- Unclosed objects can cause grief – you can access HDF5 files and their contents even after they are closed (!) if the object inside is not closed. Causes mysterious problems with XRootd. HDF5 has a clean-up-on-closure option when you open a file that we now use.
- header-only product. Interfaces to C API using C++ tools https://github.com/BlueBrain/HighFive
- Requested a long time ago by DAQ people
- (e20+c7) x (prof+debug) qualifiers are available
- RITM1529633 for the SciSoft server directories

Alternatives to highfive

- hep-hpc
  - Fermilab-developed and supported
  - Limited functionality: ntuple-style I/O.
  - Missing attributes
- HDF5's C++ API
  - deprecated
  - not thread-safe
  - removed from the hdf5 ups product.
- h5cpp:

https://h5cpp.org (website times out when I tried to access it Nov. 8)

https://github.com/steven-varga/h5cpp

I had problems finding documentation and examples.

- Native C API: Lots of documentation and examples! Everything must be possible with this. Clumsy, though.

hdf5libs – not requested (yet)

- The main reason to include ers, highfive and nlohmann_json in the DUNE stack was to use code in hdf5libs
- https://github.com/DUNE-DAQ/hdf5libs
- See examples from Kurt's talk at the September CM https://indico.fnal.gov/event/53964/contributions/250113/attachments/159552/210100/HDF5_Evolution_KBiery_13Sep2022_V4.pdf
- Some issues:
  - product name is too general for what it does
  - HDF5RawDataFile wants to own the open file. We need delayed reading, distributed from the source to the delayed reader tool.
  - Kurt's example has the event and subdetector loops in it, while *art* wants to break those up and take over some of that looping.

From the August Computing Essentials Meeting

The To-Do List

- Lock down repositories in GitHub so that only pull requests can be made
- Currently people in the DUNE organization can push to DUNE's repositories. A risk if people force pushes that overwrite or delete other peoples' work.
- LArSoft model: CMS bot evaluates pull requests with CI system and enforces review
  - L2 managers perform review and approve PRs
  - L1 managers merge PRs and make releases
- We can do this and even have a list of possible L2 managers.
- A concern – PRs can languish in review, especially if no one feels qualified to perform the review. Reviewers need to be nagged sometimes. Usually, they are responsive, and they know that without them, the code would be pushed anyway.

The To-Do List

- Spack migration
  - Modernize CMakeLists.txt files. Currently passes the requirements of the migrate script, but there is more work to be done
  - Change find_ups_product to find_package (mostly done in GArSoft as an example)
  - ROOT library lists need to be separately specified. Currently we use ROOT_BASIC_LIB_LIST which goes away when we remove find_ups product
  - We tried out Marc Mengel's Spack instructions a couple of years ago.
  - We are keeping an eye on it.

To-Do List

- Improve software organization, build out the package list
- Where does FD-specific code go?
  - duneprototypes is a catch-all for the ProtoDUNEs, coldboxes, and ICEBERG
  - Need FD-specific products for similar things. Channel maps, data ingestion, interfaces with databases, analysis tools.
  - Several FD module types will make this more challenging
  - Would like to share code whenever possible.
  - Sometimes it helps to fork code for detectors as each detector evolves over time and we don't want to break one detector when another one needs a change.
- Improve fcl dependencies (services_dune.fcl and tools_dune.fcl still tie everything together). May need a separate head product for each detector to straighten that one out. Currenty need to set up dunesw regardless of what you're doing.

The To-Do List

- More software-development tasks rather than software management tasks:
  - Reduce memory consumption of FD sim/reco!
  - Write code that runs on GPUs as well as CPUs
  - Write code that is thread safe and runs in multi-threaded mode

- Of these, reducing memory consumption is the highest priority. Needed for optimal use of multi-threaded applications anyway.

The To-Do List

- Improve documentation
  - Centralize
    - People may question the newness of the Redmine wiki for dunetpc when the code has migrated away.
    - GitHub and the DUNE Wiki are currently being used.
    - GitHub wiki does not need authentication: search engines can index it.
  - Prune out-of-date instructions

The To-Do List

- Roll out a copyright policy
- Talked with Liz Sexton-Kennedy and Aaron Sauers about this.
- The FNAL Prime Contract actually has a few pages on this.
- We can choose any Free and Open Source license we like
- It is mainly there to protect us from ourselves. DUNE developers by default are granted copyright even if we do nothing. A developer may leave in a huff though, and deny us the use of DUNE software. Or try to sell it to us.
- LArSoft uses the Apache 2.0 license. Battle-tested in court.
- Need to assert copyright (FRA) over contributions.
- FNAL Prime Contract says we can do this for DOE-supplied code with 2 weeks notice. Silent on contributions from non-DOE-funded entities.

Style Guide Examples

- LArSoft: https://larsoft.github.io/LArSoftWiki/The_rules_and_guidelines
- DUNE-DAQ: https://github.com/DUNE-DAQ/docs/blob/develop/docs/packages/styleguide/README.md