

Software Architecture News and To-do List

Tom Junk

DUNE Software Architecture Meeting

February 17, 2023

News – CI and Tokens

- From Vito: x509 certificates are being replaced by SciTokens for authentication
- CI reference files and input files in persistent dCache are still readable
- Need a way to update these however
- Vito proposes a new Capability Set that gives permission to update reference files. Vito, Jake, Tom, Gavin, David Adams proposed to be members
- Vito also says a shared duneci account will need to be created to use the CI token role
- CI Validation (automated physics plot comparisons) seem to have fallen by the wayside. Time to revive them?
- Other CI news from last year – the CI system now can automatically adjust UPS product versions in case out develop branch falls behind larsoft's, breaking the CI builds for a trivial reason. We asked Vito to enable this for us.

Jobsub_Lite

Currently, duneutil sets up the default ("current") version of jobsub_client

https://github.com/DUNE/duneutil/blob/develop/ups/product_deps

And here are some alternatives we could imagine setting up instead:

```
<dunegpvm09.fnal.gov> ups list -aK+ jobsub_lite  
"jobsub_lite" "beta2" "NULL" "" ""  
"jobsub_lite" "beta3" "NULL" "" ""  
"jobsub_lite" "beta4" "NULL" "" ""  
"jobsub_lite" "beta5" "NULL" "" ""
```

Next OS to Support

- no, not NeXT
- Looks like it's going to be Alma Linux 9.x
- DAQ group has already started working with it
- No intention to support CentOS 8.x
- Some institutions have installed CentOS 8.x on their nodes.
 - Kerberos support?
 - Containers if not?
- I have AL9 on my desktop and run SLF7 in a container. Most things work (arm debugger runs when logged in as root, and kx509 gives me a python error, but so far so good).

To-Do List

- Switch to mandatory pull request model
 - Needs ongoing coordination or some pull requests may languish. I was slow with one from Laura Paulucci.
 - Reviewers identified
 - Useful example – Andy Chappell’s dunereco PR is tied to one in larpandoracontent
- Copyrighting our software – done (at least for centrally-managed stuff).
 - Instructions in DUNE-doc-27141-v3 – Thanks, Mike! Uses the Apache 2.0 Open-Source license.
 - Deployed in DUNE’s larsoft-based stack and a few others: dunesw, dunecore, duneopdet, dunesim, dunecalib, dune prototypes, dunedataprep, dunereco, duneana, duneexamples, protoduneana, duneutil, dunepdlegacy, duneanaobj, garsoft, garana, dunendlar, dune_raw_data
 - Some software is already copyright by others. Policy point #5 says those copyrights take precedence. I added an “unless otherwise noted within the source files” clause in the README.md files.
 - A few source files I copied from dune-daq/hdf5libs (and modified) refer to a nonexistent COPYING file. Fermilab copyrighting these files ought to be fine.

To-Do List (cont'd)

- Retire dune_oslibs
- SPACK/CETModules
 - Some work we can do on our part while waiting for the ecosystem to evolve: remove all `FIND_UPS_PRODUCT()` instances in CMakeLists.txt files and replace with `FIND_PACKAGE()`. Unfortunately, this means rewriting a lot of library link lists, especially for ROOT dependencies which had been grouped together with convenience symbols.
- Wiki Migration away from Redmine to GitHub and MediaWiki
 - Good news! Nearly all of the info on the Redmine wiki for dune_tpc is obsolete already.
 - I migrated the Sam4Users page some time ago.
 - One important page left: DUNE Geometries User Guide
https://cdcvs.fnal.gov/redmine/projects/dunetpc/wiki/DUNE_Geometries
 - I still add new info to this one.. Probably should go in MediaWiki but GitHub may be okay too.
 - Not trivial as there are figures and uploaded files
 - Started migrating on Tuesday.

To-Do List (cont'd)

- Tag and release dunesw v09_67_00d00, including the Pandora PR in dunereco
- HDF5 static interface to XRootD. (VFL or VOL?)
- Wiki page explaining how to use the dynamic interface – done; added to the dCache how-to wiki page.
- Coding guidelines wiki page
- Doxygen/LXR. Is GitHub search adequate? Probably not... If anything, searching across repositories might be needed.

Versioning Software Shared with the DAQ

- `dunedaqdataformats` -- starting with `v3_6_1` now contains prior versions of `dune-daq/detdataformats` inside of it.
 - If you want to `#include` a header file from it, you need to use the appropriate version – in a directory of the UPS product. All versions have to be visible simultaneously to offline jobs as we don't want to lock data to a particular software version.
 - So far, we use the headers only. If we had to compile code with re-used names, we would have to rename objects to keep versions from clashing.
 - Use of HDF5 keeps ROOT from automatically trying to convert an object from one version to another.
- Not yet done with `dunedetdataformats` – data formats have not evolved yet, but we get new headers. e.g. `WIBFrame.hpp` and `WIB2Frame.hpp` are already versioned inside that. This was not true for `Fragment.hpp` in `daqdataformats` which just changed out underneath us.
- I misspoke at the CM about `detdataformats` being versioned like this – we may have to do this in the future if we get two versions of the same header we need to support simultaneously.

New Trigger Data in ProtoDUNE-HD

- We now have Trigger Primitives, Trigger Activities and Trigger Candidates in the ProtoDUNE-HD data files.
- There is a “subsystem” enum: Detector Readout is one, and Trigger is another. New trigger data is labeled as Trigger
- Barnali was looking for a project to do – can develop three new data products:
 - DUNETriggerPrimitive.h
 - DUNETriggerActivity.h
 - DUNETriggerCandidate.h
- And a module to produce them. Grab the HDF5RawInputFile pointer from the Service like the raw input tool and unpack as usual. Per-APA delayed reading not needed here because trigger data are small.

Versioning DAQ Data

- Trigger data is formatted as described in header files here: <https://github.com/DUNE-DAQ/detdataformats/tree/develop/include/detdataformats/trigger>
- These define structs that may evolve with version.
- We may need to support multiple versions of these simultaneously in the offline.
- Would like fewer offline versions so analysis can proceed uniformly – convert older versions of data to same offline data product as newer.
- ROOT has schema evolution tools but I'd prefer to do it ourselves. So don't directly use DAQ struct defs in offline data products, but rather use a struct with the same structure but basic types (short, int, enum, etc).

ROOT limitation on TBranch Entries

- 1 GB limit on TBranch entry size (not on the TTree entry size as I erroneously believe previously)
- We can save raw::RawDigits per APA (separate instances for each APA).
- Test ROOT script shows a TTree with 1 entry and five TBranches each with 900 MB of data can be output (4.5 GB output file, data randomized so it didn't compress well). Called TTree::Fill() like normal.
- Interactive ROOT took 14 GB of RAM to do that, however. 4.5 GB of it was user copy of memory, and two for ROOT -- Output and compression.
- Use TBranch::Fill() instead, made user copy of memory go out of scope after calling TBranch::Fill(), and turn off compression: memory usage down to 4.5 GB. No way to do this (yet) in *art*. Kyle knows we want “eager writing”.

Advertisement – LArSoft Multithreading and Acceleration Workshop

<https://indico.fnal.gov/event/57914/>

And another challenge: Memory usage task force: Tom, Jake, Barnali, Andrew, Tejin