2020 LArSoft Work Plan

Last updated: December 10, 2020 - Final 2020 update

Introduction

This document records the LArSoft work plan for 2020. Progress on specific items going forward are given quarterly at Steering Group meetings.

Erica and Katherine discussed priorities with each of the Offline leads in September of 2019. The experiments detailed their plans for the next year, the implied requirements for LArSoft, and how LArSoft could help, as well as what the experiments might be able to contribute to LArSoft code. Some items may be raised by LArSoft, not a particular experiment, to address general problems faced by all LArSoft users.

Based on those discussions, LArSoft proposes a plan of work for 2020 along with the relative priority of the various items. These were reviewed at the October Offline Leads meeting. A revised work plan was discussed and approved by the experiments at the December LArSoft Steering Group meeting.

The resources responsible for executing parts of the plan is listed in cases where this is known. It should be noted, however, that the LArSoft project team does not have sufficient staff at present to carry out the entire program. Some projects can and should be performed by the experiments. Allowing experiment service credit for contributions to items that benefit an experiment, but that are of a more general nature would help to fill some of this gap in effort. The more this happens, the more all experiments benefit.

Short term priorities for 2020

We define short term priorities as those items that should receive sufficient effort to be completed in CY2020 due to the urgency or timeliness of the need. The meaning of priorities listed is discussed in Appendix A.

1. **Re-architecture of LArSoft code to ensure thread safety and to implement multi-threading as needed for critical production workflows**. (ongoing from 2018)
   Priority: high
   Re-architect code in experiment production workflows to ensure thread safety, and where appropriate and cost effective, to implement multi-threading in order to control memory scaling and improve resource utilization on production platforms. Collaborate with efforts to run multi-threaded workflows on HPC resources.
   https://cdcvs.fnal.gov/redmine/issues/19288
   Resources: Kyle Knoepfel, Saba Sehrish, Mike Wang, HEPReco SciDaq

   Feb 20:
   - Various services have been made thread-safe. An overall strategy for converting more has been agreed upon. Along with these services, a number of modules have been made thread safe. The targets for both the services and modules are all in DUNE production workflows. (Kyle Knoepfel, Saba Sehish)
   - Hit finding has been multi-threaded in a demonstration by the HEPReco group, and partly integrated into LArSoft. (Sophie Berkman)
   - Additional code for an ICARUS signal processing workflow has been demonstrated, but has not yet been integrated into the head of LArSoft. (Mike Wang)

   Apr 9:
   - LArSoft was migrated to a new version of art (3.05) that allows an incremental approach to implementing thread-safe services. This change will expedite the process of adapting services, and unlike the previous version of art, allows experiments to use each service as it is adapted,
   - The balance of multi-threaded hit-finding code was integrated into LArSoft (HEPReco group)
   - GPU as a service has been demonstrated to work within LArSoft, along with basic scaling properties. (Mike Wang)
   - Continued progress on making ICARUS and DUNE production workflows thread-safe (Kyle Knoepfel, Saba Sehrish, Mike Wang)

   May 14:
   - Significant progress has been made re-architecting production critical services to be thread safe since the introduction of art 3.05. Some of the changes provide new models

for how event-specific updates to services should be handled in a multi-threaded environment. (Kyle Knoepfel, Saba Sehrish)

- The project continues to work closely with ProtoDUNE and ICARUS to make their production workflows thread-safe. (Kyle Knoepfel, Saba Sehrish, Mike Wang)

Aug 6:

- Successfully delivered a completely thread safe production workflow for ICARUS
- Lost effort from Saba Sehrish from July through the end of FY20 due to other SCD commitments

Aug 25:

- The v09 series of LArSoft was released on Aug 19. With this release, the changes needed for a complete production workflow, first demonstrated in ICARUS, are now integrated into the head of LArSoft development.
- Work continues on making more code thread safe, and in exploring GPUaaS to execute particular LArSoft workflows

Nov 17:

- This item is being carried forward into 2021 Work Plan, with an expanded scope that connects directly to experiment efforts to use LArSoft on HPC resources.
- In pursuing multi-threading upgrades in LArSoft, the need for a concurrent caching system for conditions information became apparent. In response, support for a new concurrent caching model was developed in art, and will be released in art v3.07. This will be relevant for services that manage conditions data.

2. **Migrate to GitHub and pull requests.** (Started in June 2019)

Priority: high

Investigate moving the LArSoft central repositories from Redmine to GitHub, along with those collaboration services offered by GitHub that make the migration cost effective. There are three phases.

- ○ Phase I: Migrate central repositories to GitHub
- ○ Phase II: Develop necessary infrastructure to support pull requests
- ○ Phase III: Transition the current model of merging to develop by developers and the Code Management team to a pull-request system where authorized experiment personnel handle requests after mandatory CI testing prior to merging.

This item was mostly completed in 2019. The outstanding work includes getting CI jobs to run that are triggered by GitHub; and completing the documentation needed for testing and general use.

Resources: Patrick Gartung

Feb 20:

- The migration to GitHub on Jan 23 went smoothly and was completed on schedule. An mrb issue caused a delay in re-opening the repositories, but the problem was resolved before the scheduled contingency time expired.

- Following the migration, guidance was requested for collaborating on feature branches. These and other issues were discussed at the LArSoft Coordination meeting on Jan 28.
- This and other feedback is being used to update the documentation which is available on the LArSoft wiki page, Working with GitHub.
- There have been no significant issues reported related to the GitHub migration so far, only minor and easily fixed failures with workflow scripts. Some general guidance was requested for how to collaborate on feature branches, which was provided at the Jan 28th LArSoft Coordination Meeting. We are now attempting to improve the usability of the system through improved tools. A first project in this direction is to provide users and L2 approvers a simple, visual way to assess the status of all pull requests within their projects, or that are otherwise of interest to a developer. The “project” view within GitHub, and OctoBox.com have been discussed as possible solutions.

Apr 9:

- Consulting support is being provided to Mu2e and the SBN experiments as they migrate to GitHub (SBN) and deploy the LArSoft-adapted pull-request approval workflow code. (Patrick Gartung)

May 14:

- SBN experiments migrating to GitHub and production use of the LArSoft-adapted pull-request approval workflow within the next week. We continue to provide consulting support, including discussions of how to collaborate within the new model (Lynn Garren, Patrick Gartung)

Nov 17:

- This item was completed, and is now closed.

3. **Provide a support mechanism for running LArSoft in containers**

Priority: high

Develop the procedures, processes and documentation needed to allow users to develop and run LArSoft using a well-supported containerized solution. Provide on-going support for producing the required images.

Resources: SciSoft

Feb 20: no update

Apr 9: no update

May 14:

- Have identified a Docker image solution, though we are still working through the details of how to adapt the development workflow to use Docker. (Lynn Garren, Kyle Knoepfel)

Aug 25:

- This work has concluded, with a Docker image solution having been provided, and all necessary changes made across LArSoft and experiment code. As of the LArSoft v09 series releases, Mac OS builds are no longer supported. Mac users who wish to continue developing on Mac OS will use Docker to run LArSoft.

Nov 17:

- Documentation is available at the LArSoft wiki at: https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/Developing_LArSoft_on_Unsupported_Operating_Systems_with_Docker
- This item was completed, and is now closed.

4. **SPACK - new build system for art and LArSoft** (deferred from 2017, continued from 2018 and 2019).
   Priority: high
   Purpose:
   - To migrate LArSoft to the Spack packaging tool, and the SpackDev development system, a standard set of packaging and build tools that have broad community support
   - Via this migration, to address portability and configurability issues raised by experiments with the current build system

   https://cdcvs.fnal.gov/redmine/projects/spack-planning/issues
   Resources: Chris Green

   Feb 20:
   - Sophie Berkman has been working on a native build of LArSoft appropriate for running on the ALCF at Argonne. This work was performed within the context of the MVP.

   Apr 9:
   - Work on using spack for ALCF builds continues
   - A long-standing pull-request to spack has been integrated that is part of a scheme to allow spack and UPS to work together, which is part of the foundation for the spack migration strategy

   May 14:
   - The second long-standing, blocking pull-request to integrate Fermilab adaptations has been accepted and merged. This clears the path for completion of the plan to migrate to the spack / spackdev build system.

   Aug 25: no update
   Nov 17:
   - Though significant progress has been made on the Spack-based build system, a production system is not expected in 2020. This item will therefore be rolled into the 2021 work plan.

   Dec 10:
   - The art suite has been transitioned to use the new Spack-based system in a UPS backwards compatibility mode. A plan to extend this work to include the LArSoft software stack is now being developed. This high level plan will include not only the steps necessary to make LArSoft build in this UPS backwards compatibility mode, but also those needed to transition from that point to a build that uses Spack / SpackDeve only, and that is completely free of UPS. The plan is due by the end of CY2020.

5. **Pixel-detectors within LArSoft**. (Considered in 2018, carried forward from 2019)
   Priority: high
   Work with experiments to identify and implement common solutions that include LArTPCs with pixel readout. The integration should include the following:
   - Adaptation or development of low-level data products
   - Shared data products for 3D hits
   - Geometry adaptations
   - Development of pixel-based anode simulations.

   Expect experiments to carry out a significant fraction of the reconstruction and simulation work.
   https://cdcvs.fnal.gov/redmine/issues/19328
   Resources: Erica Snider & ??

   Feb 20: The SLAC group has presented proposed updates to the LArSoft Geometry system that provide compatibility with the DUNE pixel detectors (ArgonCube) at the most recent DUNE Collaboration Meeting. The project is meeting with them to discuss various issues and next steps.

   Apr 9:
   - Approaching a critical time, since developers at DUNE are actively working on software for pixel detector simulation and reconstruction, and some preliminary changes to the LArSoft geometry system have already been made

   May 14:
   - No update.

   Aug 25:
   - Preparing a plan to re-factor the geometry system to accommodate pixel readout descriptions. The new system will accommodate both pixel and wire readouts with natural terminology without the need to overload nomenclature or strain descriptions across different readouts.

   Nov 17:
   - This item will roll into the 2021 work plan.

6. **Existing Event Display**
   Priority: medium
   Implement improvements to the existing LArSoft Event Display to make it faster and easier to navigate through a large number of APSs, and to identify TPCs or detector regions with data of interest. The implementation should be applicable to ProtoDUNE, DUNE FD and ICARUS.
   Resources: not yet identified

Feb 20: no update

March 11:

- ICARUS is not as interested in this item since they are happy with TITUS - combination of python (3) and C++ and is based on the PyQT5 interface. Marco Del Tutto and Gianluca Petrillo updated the event display originally authored by Corey Adams for MicroBooNE. ProtoDUNE, however, is still a major consumer of the LArSoft event display, saying it is still the easiest to modify, provides the best concentration of information on the screen, and allows viewing of all relevant levels of detail in the images.

Apr 9:

- No update.

May 14:

- No update.

Nov 17:

- This item will be closed. A new item in the 2021 work plan will reflect the change in strategy to support running community-supported event displays in art and gallery.

7. **New Event Display development plan.**
Priority: medium
Phase 1: Create a plan for developing a new LArSoft event display in light of the results of the 2019 ED technology review (see https://cdcvs.fnal.gov/redmine/issues/19038), and the ED requirements document with identified resources to execute the plan. The new ED may leverage existing code and effort from outside institutions.
Phase 2: Design and implement an event display framework following the requirements and investigation of tools described above, consulting with the experiments at regular intervals to ensure that evolving needs are met.
Resources:

Feb 20:

- At the behest of SCD management, we have investigated a non-Fermilab partner to take the lead in providing a new LArSoft event display (ED), but were unable to get a commitment to work toward a common ED. We are now working on next steps, and are considering a completely different strategy to leverage the most prevalent non-LArSoft EDs to provide a solution in LArSoft/gallery environment.

Apr 9:

- WebEVD is in the process of being integrated into the LArSoft distribution. The author is a DUNE collaborator

May 14:

- WebEVD integration was completed. No other updates.

August 25:

- The redmine ticket to track this, https://cdcvs.fnal.gov/redmine/issues/19034, has been closed since LArSoft event display strategy has changed from developing a common tool to providing adequate support for community developed and supported displays sufficient to allow community members to choose on their own. There are currently six options which are given in Appendix B in this work plan.

Nov 17:

- This item is closed. See notes for item (6).

8. **NuTools product re-factoring**

Priority: medium

Re-factor NuTools and the relevant LArSoft packages so as to separate the explicit dependence of LArSoft versions from explicit versions of Genie. The re-factoring should introduce a new product that interfaces LArSoft to Genie via common, stable data products in NuTools, and sources of geometry information in LArSoft. Upon completion, this framework will be used to integrate the NuWro and other generators into LArSoft. At lower priority, the same technique should be used to decouple LArSoft versions from explicit versions of Geant4.
Resources: Robert Hatcher

Feb 20: no update
Apr 9: no update
May 14: no update
Aug 25:

- Initiating discussions on how to address decoupling geometry initialization from the base LArSoft release. A solution to this problem will open a pathway forward, since the output-side solution is in hand.

Nov 17:

- The remaining parts of this item have been rolled into a new work item for 2021.

Long term and continuing priorities for 2020

We define long term priorities as those that we do not anticipate completing within CY2020 due to the nature of the work involved. Unless explicitly indicated, a project defined as a long-term priority should not be construed as implying it is "low priority". A "continuing priority" is a set of tasks that by construction represent a continual, recurring stream of work that may or may not span more than a calendar year.

1. **Places where 2020 support from LArSoft team has been requested**
   Priority: high as needs arise. Low otherwise
   a. Support for migrating to the refactored LArG4
      i. Feb 20: work on-going to on-board ProdoDUNE SP
      ii. May 14: additional requests from ProtoDUNE SP and DP are being addressed
      iii. Aug 25: Offered as a solution to a DP problem raised by ProtoDUNE in June. Working with the author, but have no recent updates as to progress.
   b. Wire-cell integration
   c. Geometry (e.g., cases where ICARUS breaks implicit assumptions: non-vertical collection wires, existence of horizontal wires, multiple channels per geometric wire)
      i. May 14: Geometry updates and utilities offered by ICARUS are being integrated into LArSoft
   d. Extend the NuWro direct integration model to include the GiBUU event generator framework and the NEUT event generator.
   e. Investigate need for magnetic field map within LAr volume. Design and develop the interface for such a service. -- Make this a named project. Assign architect, etc.
   f. Coordinate between GArSoft and LArSoft
      i. Ensure sharing of data products and geometry (and with other detectors)
      ii. Ensure sharing of any services, should that be needed (eg, for magnetic field)
      iii. Possible introduction of non-uniform magnetic field to LAr volume

      Aug 25: learned recently that GAr detector is not yet part of the ND baseline design, and is currently competing against other design options

   g. SBND data reduction strategies
   h. Participate in the discussion of how the DUNE data streams (SN + possibly others/all) will be handled within *art* / LArSoft. May involve moving away from root-based I/O.

2. Support for running select LArSoft workflows on HPC resources.
   a. SBND has time allocations on Theta at Argonne, and are looking to run LArSoft there
      i. May 14: The project has been consulting on this project, and has contributed some changes to address spack build-system issues.
   b. LArIAT has requested that they be able to run LArG4 on GPUs.

Aug 25: SciSoft team members have been engaged in exploring GPU as a Service for executing suitable LArSoft workflows. This work is still in exploratory and prototyping phases, but is engaged in testing at scales relevant for production processing

3. Architecture work to support a common framework for data preparation

4. Architecture work to provide full integration of DL networks
   a. May 14: created a new repository, larrecodnn (for Deep Neural Network) to host ML-related code. Migrated relevant code plus recent updates previously in larreco to larrecodnn

5. **Ongoing work on bug fixes, emergency feature requests, building software, helping users.**
   Priority: high, or as indicated by the request
   Resources: SciSoft team, primarily Erica Snider, Saba Sehrish

6. **Architecture-dependent libraries (2018).**
   Priority: low
   Follow-on work to SIMD vectorization or other identified use case. Support is needed to allow coexisting builds that include / exclude, support for different computing backends under a given OS flavor.

   a. An immediate goal will be to allow the standard setup procedure to support a generic library as well as one built with a single SIMD extension set
   b. More generally, there are a number of areas that might benefit from this feature, such as allowing use of GPU backends when available for external libraries, or selecting the optimal SIMD instruction set for a given computing node.The types of backend support required needs to be studied, and appropriate tools adapted or developed to allow the required setup procedures.

   Resources: ??

   11/5 20 Status: Closed because the immediate need is gone. It may come up again within the context of other work, but then should be included as part of those projects.

7. **Other topics under consideration to work on include the list of accepted, but not assigned, redmine issues.** These can be found under accepted redmine issues.
   Priority: low or as indicated in the respective ticket

Appendix A: Definition of priorities

The priorities listed above have the following meaning. The impacts noted are those negotiated with the experiments.

- High: the most urgent projects that have high impact to the community, and that should be fully staffed at all times until completion
- Medium: a project that has a significant impact to the community, and that should be staffed sufficiently to ensure completion within the calendar year
- Low: a project that does not have immediate impact to the community, could be rolled into the next planning period without imposing undue burdens, and should be staffed after ensuring all other obligations are met.

Appendix B: Event Display Options

The LArSoft event display strategy has changed from developing a common tool to providing adequate support for community developed and supported displays sufficient to allow community members to choose on their own. Here are the currently available options:

a. LArSoft Event Display (ROOT based)
   i. In principle can display everything both in 2D and 3D
   ii. Very slow for large detectors.
   iii. Works for all detectors. Can display multiple data items and labels at once

b. TITUS (Qt based)
   i. Combination of python (3) and C++ and is based on the PyQT5 interface.
   ii. Updated version (by Marco Del Tutto and Gianluca Petrillo) of the event display originally authored by Corey Adams (QT based) for MicroBooNE
   iii. Runs in gallery. Being used by ICARUS. Works for MicroBooNE, ArguNeut and possibly DUNE.
   iv. Native 2D displays of low-level data

c. WebEVD (using three.js/WebGL for access to OpenGL)
   i. Chris Backhouse developed
   ii. In use at ProtoDUNE. Has been updating based on feature requests.
   iii. Runs in LArSoft & Gallery.
   iv. Doesn’t have a native 2D display.
   v. Requested to be a UPS product distributed with LArSoft.

d. EVE based for SBND
   i. Umut Kose developed.
   ii. 3D Oriented,
   iii. Study of using EVE for event display started after SBN workshop in 2019 at Fermilab. It aims to provide the functions of 3D and 2D projection views, animations, interactive display for users.

    iv. Eve is a ROOT module based on experiment-independent part of the ALICE event display developed in cooperation between ALICE offline project and ROOT.

e. Bee event display (based on WebGL)

    i. Developed by Chao Zhang for use with wire-cell.
    ii. 3D oriented.
    iii. Uses three.js.
    iv. Doesn’t run in LArSoft and Gallery, uses wire-cell data format, so requires conversion step

f. Phoenix event display - Ed Moyse developing

    i. Presentation: https://indico.cern.ch/event/916410/contributions/3852667/attachments/2051044/3437881/Phoenix_HSF_04062020.pdf
    ii. Experiment-agnostic display that relies on event serialization via JSON. It allows for a shared code base.