



2019 LArSoft Work Plan

Last updated: March 7, 2019

Introduction

This document records the LArSoft work plan for 2019. Progress on specific items going forward are recorded in redmine issues with updates to this document given quarterly at Steering Group meetings.

Erica and Katherine discussed 2019 priorities with the Offline leads and spokespeople in November of 2018. 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 were raised by LArSoft, not a particular experiment, and address general problems faced by all LArSoft users.

Based on those discussions, LArSoft proposed a plan of work for 2019 along with the relative priority of the various items. This was discussed and approved at the December 13th 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 2019

We define short term priorities as those items that should receive sufficient effort to be completed in CY2019 due to the urgency or timeliness of the need. The meaning of priorities listed is discussed in Appendix A.

1. **Finish investigating a new Event Display framework common to all experiments.**

Priority: high

(Investigation began in 2018.) Some important features:

- a) Provides a simple, intuitive means to navigate through large events and multi-TPC detectors.
- b) Ability to zoom and pan with a data density that matches the screen resolution.
- c) Provides basic interfaces to art for controlling runs and events, the geometry service, run conditions services. Re-running reconstruction workflows and displaying the result should be possible.

This work does not cover development of drawing routines. It is expected that the core event display users within the experiments will provide this code. Investigating is being tracked under <https://cdcv.sfnal.gov/redmine/issues/19034>

Resources: Erica Snider, Paul Russo & waiting on the division after SPPM meeting on 2/21/19.

3/7/19 status: Four potential technical solutions have been identified, along with some of the costs, benefits and trade-offs of each. Significant effort will be required to create prototypes with any substantial functionality. We are weighing how to proceed from here with the evaluation.

2. **Event Display development plan.**

Priority: high

Phase 1: Design and implement an event display framework following the requirements and investigation of tools above. Phase 2: Consult with the experiments as they implement the Event Display in their analysis.

Resources: Waiting on the evaluation and the division after SPPM meeting on 2/21/19.

3. **Migrate to GitHub and pull requests.** (Started in June 2018)

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 once a migration decision is made.

- 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.

Resources: ??

3/7/19 status: Chris Jones spoke at length with the offline leads on 2/21. He provided details of the roles, workflows and tools CMS uses to support their GitHub / pull request system, and described how the system evolved. An extensive discussion followed on how this might be applied to LArSoft. Deployment options include the bare system offered by GitHub, through a fully developed, multi-role system with well-defined workflows and approvals. Additional work will be required to develop a proposal for LArSoft.

4. **SPACK - new build system for art and LArSoft** (deferred from 2017, continued from 2018).

Priority: high

Purpose:

- To migrate to a standard set of build tools that have broad community support
- To address portability and configurability issues raised by experiments with the current build system
- To allow continued full support of Mac OSX as a LArSoft development platform within the context of Apple's System Integrity Protection system

<https://cdcv.sfnal.gov/redmine/projects/spack-planning/issues>

Resources: Chris Green

5. **NuWro Integration.**

Priority: medium

Provide an architecture for integrating NuWro into LArSoft using the direct integration model. NuWro should take configuration data via data conversion from LArSoft, and provide output data to LArSoft. A data conversion layer provides an interface between the two. Input data sources should include FHICL files, LArSoft services, and LArSoft compatible flux files. The results should decouple LArSoft from specific NuWro version so that experiments can select the NuWro version best suited to their needs without changing LArSoft releases.

Resources: Erica Snider, and Genie experts

6. **Provide training sessions and/or workshops on using LArSoft.**

Priority: medium

Provide and record a workshop on the architecture, design choices, and why we do the things we do. More advanced topics as well--such as contributing algorithms, multi-threading, code analysis.

- Make a plan for addressing the need for local training sessions in different geographic regions

Resources: Erica Snider, Katherine Lato, & presenters

3/7/19 status: This task will be addressed via a session at the summer LArSoft Workshop. Current planning for this includes a rough outline of a curriculum, and examination of existing materials from DUNE tutorials that could be used in part or in whole.

7. **Re-architecture of art services in LArSoft to ensure thread safety.** (from 2018)

Priority: medium

Based on a discussion in December of 2017, a plan was developed to re-architect a certain class of services (such as those that tag event-by-event transients in various electronics channels, and the calculation of event trigger times used in the global clock service) such that they are inherently thread safe, while still providing the capability of on-the-fly, event-by-event updates.

<https://cdcv.sfnal.gov/redmine/issues/19288>

Resources: Mike Wang, Paul Russo

3/7/19 status: Mike has demonstrated both intra-event and inter-event multi-threading and favorable memory scaling using a LArSoft signal processing algorithm in a stand alone environment. The same code has also been used to demonstrate inter-event and intra-event multi-threading within a LArSoft job. The memory scaling properties have not yet been measured in this case. Mike is now focused on identifying targets for further work within production workflow for ICARUS, DUNE and SBND.

8. **Pixel-detectors within LArSoft.** (2018)

Priority: medium

Work with experiments to identify and implement common solutions that include LArTPCs with pixel readout in cases where such integration is cost effective. It is possible that no such integration The reconstruction piece has two possible scenarios:

- Pattern recognition algorithms shared between pixel and SP/DP LArTPCs. This requires agreeing on a common data format to represent 3D data, whether obtained directly from a pixel detector, or inferred from 2D views in SP/DP detectors. There will likely be implications for the geometry interface, which will then need to support both conventional and pixel-based detectors. Some study will be needed to determine whether these constraints can be met. If code is shared, then support for the development of those algorithms will also be needed.
- No sharing of pattern recognition algorithms between pixel and conventional SP/DP LArTPCs. The code bases in this case diverge, so no further work is required aside from the normal support for data product, service and algorithm development.

<https://cdcv.sfnal.gov/redmine/issues/19328>

Resources: Erica Snider & ??

3/7/19 status: There has so far been little interest in collaborating or traction for this as a high-priority item, so it may not be pursued.

9. **Optimize LArSoft algorithms for Performance** (Was - Use of SIMD vectorization to optimize LArSoft algorithms (2018).

Priority: medium

Optimize performance of LArSoft algorithms with the goal of significantly improving CPU and memory performance on existing computing resources. This work should include studies and projects to introduce SIMD vectorization in those algorithms that might benefit from vectorization. The project should proceed by profiling real use cases and reviewing code to identify high-value targets; conducting vectorization studies, such as enabling compiler vectorization on minimally modified code, introducing vectorized data types coupled with re-design of select LArSoft data structures and algorithms; documenting performance comparisons and other studies; and preparing reports and documentation as needed. The work should include an assessment of the potential gains from further vectorization efforts. Resources: ??

3/7/19 status: The project and the LArTPC reconstruction SciDAC project have agreed to coordinate work plans and objectives, and have initiated weekly meetings to work toward this goal. An early goal will be perform tests using HPC resources to run LArSoft algorithms selected from high data density production workflows.

Long term and continuing priorities for 2019

We define long term priorities as those that we do not anticipate completing within CY2019 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 2019 support from LArSoft team has been requested

Priority: high as needs arise. Low otherwise

- a. Support for migrating to the refactored LArG4
- b. Provide a work-around for build system issues on OSX prior to Spack/SpackDev deployment - critical for code development at ICARUS, problem with install and cmake stages
- c. Wire-cell integration
- d. Global wire-coordinates at DUNE

- e. Memory and CPU profiling
- f. Geometry (e.g., cases where ICARUS breaks implicit assumptions: non-vertical collection wires, existence of horizontal wires, multiple channels per geometric wire)
- g. FLUKA integration
- h. Improving code location / browsing tools
- i. Improve the usability of the development environment and reduce the overhead required to maintain a stable and consistent development build
- j. Extend the NuWro direct integration model to include the GiBUU event generator framework and the NEUT event generator.
- k. A workshop to discuss event overlay requirements, techniques, solutions [Added Dec 20, 2018]

2. 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, Paul Russo, Saba Sehrish

3. Making LArSoft thread safe.

Priority: medium

After the services are thread safe, LArSoft will determine what else needs to be modified to ensure that LArSoft code is thread safe. As part of this work, documentation should help people in writing thread safe code.

Resources: Paul Russo & ??

3/7/19 status: The work on services being performed by Mike Wang will extend to LArSoft algorithms in production workflows, irrespective of whether the algorithms are currently services.

4. 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: ??

5. **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 meeting all other obligations.