2023 LArSoft Work Plan

Last updated: September 12, 2023

Introduction

This document gives the LArSoft work plan for 2023. Progress on specific items going forward are given quarterly at Steering Group meetings.

Erica and Katherine discussed priorities with each experiment in a series of meetings in October and November of 2022. The experiments detailed their plans for the next year, the requirements for LArSoft, and how the LArSoft Project Team (“the Project”) could help, as well as what the experiments might be able to contribute to LArSoft code. Some items may be raised by the Project to address general problems faced by all LArSoft users. Major observations from these discussions are listed in Appendix B since they may be of interest to other experiments and the project as a whole.

Based on those discussions, LArSoft proposes a plan of work for 2023 along with relative priorities of the various items. The Project resources responsible for executing parts of the plan are listed in cases where this is known. It should be noted, however, that the Project does not have sufficient staff at present to carry out the full program of work in this plan, or the entirety of the short term or long term programs individually. The Project will collaborate with experiments to prioritize and coordinate the work to best meet experiment needs. We also note that some work related to the plan can be performed by the experiments. Allowing experiment members to receive service credit for contributions to work plan items that benefit their 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.

1. Short term priorities for 2023

We define short term priorities as those items that should receive sufficient effort to be substantially completed in CY2023 due to the urgency or timeliness of the need, and where the Project will drive the timeline. The meaning of priorities listed is discussed in Appendix A.

1. **Thread safety, multi-threading, and High Performance Computing (HPC) for critical production workflows:**
    - **Re-architect LArSoft and experiment code to ensure thread safety.**
    - **Implement multi-threading where appropriate to improve resource utilization or to enable execution on HPC resources.**
    - **Provide support for hardware acceleration where appropriate.**
    - **Coordinate / collaborate with efforts to use LArSoft on HPC platforms.**

   Priority: high

   This is a multi-year effort that started in 2018 working on thread safety and multi-threading to improve resource utilization on existing grid resources, and to prepare LArSoft code to be effective in an HPC environment. A fully thread-safe production workflow and the ability to utilize GPU-as-a-Service for deep-learning inferencing within LArSoft were completed in 2021. The focus will remain on critical production workflows where the introduction of these advanced computing techniques will provide substantial added value through improved or expanded resource utilization, or gains in execution speed. A number of such targets have been identified.

   Resources: Kyle Knoepfel, Mike Wang, in collaboration with those previously working on the HEPReco SciDAC

   Feb 22 update
   - Making progress on DUNE super-nova production workflow. Identified and working to address the source of some discrepancies compared to single-threaded operation. Expect to be completed with this in a few weeks, at which point the entire workflow should be running in a multi-threaded mode. (Mike Wang)

   Mar 23 update
   - Two people will be joining part time to work in this area. One (Saba Sehrish) is returning to LArSoft, and will pick up where work left off in making those database services that are needed for production workflows thread safe. Afterward, Saba will work to enable the use of GPUs in production workflows. The other (Marc Paterno) will be focusing on replacing select algorithms with portable GPU-enabled algorithms.

   June 11 update
   - The work noted in the Mar 23 update has been mostly completed. Three services that access databases have been updated to allow execution in multi-threaded

environments (channel status, det pedestal, and electronics calib services), leaving one remaining (pmt gain service).

Sept 12 update

- All larevt services thread safe. Ones that access DB use concurrent caching. Now working on expt code. ArgoNeuT, SBND (work since spring)
- Discrepancy between single-threaded and MT results in SN pipeline demonstrator understood, but work still underway to fix (on-going since winter)
- geometry for pixels: factored into wire readout geometry and volume geometry. Adapting expt code. After validation, will introduce pixel readout geom (on-going since winter)
- GPUaaS: working to extend to Pytorch. Allows pytorch GNNs

2. **Multi-threading workshop - complete**

   Priority: high

   Organize and host a workshop centering multi-threading solutions for experiment-selected candidate pieces of production code. The work will further the goals of enabling fully thread safe production workflows based on core LArSoft code. Provide expertise to assist and relevant tutorials as part of the workshop. Planning to have this in February of 2023.
   Resources: Erica Snider, Kyle Knoepfel, Katherine Lato

   March 23 Update

   - The workshop was held Mar 2–3 with the goals
     - Learn MT and acceleration capabilities of frameworks and common toolkits
     - Share experiences about existing utilization and throughput problems
     - Explore how MT and acceleration is being used to address them
     - Discuss results, opportunities opened by applying these techniques
   - A total of 43 people registered for the workshop, with 20–30 attending the hybrid sessions. The program consisted of one day of presentations spread across two mornings, plus an open working session on the first afternoon. The full program can be found at: https://indico.fnal.gov/event/57914/
   - Some observations from the workshop include:
     - The audience was highly engaged with the material, with robust discussion throughout. It was clear from comments that people were learning and adjusting plans based on things they heard, thereby meeting one of the primary goals of the workshop.
     - All experiments reported on work related to multi-threading, high-performance computing (super-computers) and GPUs, at varying stages of development. Few had demonstrated production scale. The work on ND-LAr, which is not currently integrated into LArSoft, was of particular note as a third generation set of algorithms with native GPU support.
     - Complementary strategies for addressing memory consumption issues were also discussed. These methods can work in conjunction with multi-threading to optimize resource utilization. The low-level data algorithms discussed are also good targets for GPU acceleration.

- Writeup with links to slides and video recording are available as part of LArSoft training at: https://larsoft.org/multi-threading-workshop-2023/
- This work plan item is now completed.

3. **SPACK - migrate to a new, community-supported build system for art and LArSoft**
   Priority: high
   This is a continuation of a multi-phased project to migrate LArSoft to a Spack-based packaging and build system, which has broad HEP community and industry support. Via this migration, we will address a number of long-standing portability and configurability concerns raised by the experiments that are inherent to the current system. The overall strategy centers on minimizing disruption to the experiments by providing backward compatible changes until the final migration. The first major milestone, preparing LArSoft for the transition to a system with a Spack-based back end, was completed in Q4 of 2021. The final phase will end support for UPS and the cetbuildtools-based build system. An aggressive education campaign is part of this project.

   Details on Spack at Fermilab can be found in a presentation by Marc Mengel at the LArSoft Coordination meeting. Further information on the build system can be found in the following presentations by Chris Green: Feb 23, 2020 LArSoft Coordination Meeting, Nov 2, 2020 LArSoft Coordination Meeting.
   https://cdcvs.fnal.gov/redmine/projects/spack-planning/issues
   Resources: Patrick Gartung, Chris Green

   March 23 update
   - The team continues to make steady progress on creating a system usable by release managers that is capable of building and releasing reproducible distributions of code via Spack in an optimal manner. There is not yet a timeline for completion of this work.
   - Progress on making the external dependencies of LArSoft buildable by Spack has stalled due to resource limitations.
   - The SciSoft team assists with migrating experiment code to the pre-requisite state for migration as requested.

   June 11 update
   - Since the last update, work on making LArSoft externals buildable with Spack has been suspended due to effort limitations. Additional effort with LArSoft expertise from the SciSoft team will be needed to resume work. We are working to arrange this. In the mean time, the Spack team is improving the Spack codebase and developing a tutorial that will be useful to LArSoft once work progresses.

   Sept 12 update
   - Effort is again being put into making LArSoft external buildable dependencies. While there is a large set of products to deal with headway has been made. LArSoft believes building recipes for most products will be straightforward with a couple exceptions, notably Tensorflow.

- The tutorial has been given to Mu2e. We are working with Mu2e to produce a clean usable build.
- CI's first nightly build of the art suite completed! It was built on both SLF 7 **and AL 9**! The build scripts are invoked independently from Jenkins. We need to move to using Jenkins on the backend to reduce boilerplate.

4. **Pixel detectors within LArSoft**
    - **Re-architect geometry systems to improve integration of multiple readout schemes**
    - **Identify and implement common low-level and 3D data products needed to represent data from LArTPCs with pixel readouts**
    - **Support development of pixel-based anode simulations**

    Priority: high

    Work with experiments to develop requirements and design solutions, and to carry out the work necessary to adapt reconstruction and simulation codes. Conceptually, the solution will abstract and separate anode descriptions from the TPC volume descriptions. Details on the progress of this work will be presented at the LArSoft Coordination Meeting.
    https://cdcvs.fnal.gov/redmine/issues/19328
    Resources: Erica Snider, Kyle Knoepfel

    Feb 22 update
    - Holding weekly meetings since before the end of CY2022. Progress so far:
        - Elimination of deprecated methods, streamlining and refactoring of iterators and other clean-ups of the Geometry interface (Kyle Knoepfel)
        - Refactoring of readout geometry interface and volume geometry interface into separate classes (Kyle Knoepfel)
        - Refactoring of object sorting classes to eliminate interdependency between ChannelMap and volume geometry initialization. (Kyle Knoepfel)
        - Draft specification of pixel readout geometry requirements based on existing code (Tom Junk)

    March 23 update
    - Weekly meetings continue.
        - Refactoring of readout geometry and volume geometry interfaces is nearly complete. Design work needed to complete the wire geometry description is under way. Once complete, we will be in position to validate all the changes on data.
        - Completion of the pixel readout geometry will allow creation of a full ND-LAr simulation within LArSoft.

    June 8 update
    - A full draft refactoring has been completed, including the definition of the necessary readout geometry for LArTPC with wire readouts, though some minor non-technical details remain. Work is currently in progress to update the experiment code to the new refactored scheme. (Kyle Knoepfel)

- Work on the pixel readout class is awaiting completion and validation of the refactored volume and wire readout geometries.

Sept 12 update

- Final changes needed to restructure the geometry system to allow support for a pixel readout geometry are nearing completion. Will first validate with existing wire readout geometries, then introduce a pixel readout geometry.
- Continued work on the updated documentation.

5. **Neutrino event generator refactoring**

   Priority: high

   Refactor the interface to the GENIE neutrino event generator such that the GENIE version can be selected through the runtime environment configuration. One side effect will be to provide a template for how to interface other event generators so as to simplify integration and maximize flexibility. The work will commence after the final phase of the Spack migration is completed.
   Resources: Robert Hatcher

   March 23 update

   - No activity

   June 8 update

   - Robert Hatcher has raised this issue again with the GENIE collaboration. No estimate yet as to when work might resume or be completed

   Sept 12 update

   - No significant work has occurred during the past quarter. Further analysis of the dependencies suggested a more simple approach than was previously discussed, but no work currently scheduled.

6. **Review and update / augment LArSoft documentation in key areas as requested by experiments.**

   Priority: high

   Continuing project to provide improvements covering areas noted by experiments, and new developments in LArSoft capabilities and the surrounding ecosystem. Initiatives will include:

   - Add a Machine Learning section to LArSoft.org and to LArSoft wiki pages.
   - Maintain and enhance an up-to-date introductory slide deck on LArSoft - https://indico.fnal.gov/event/49621/#1-see-attached-source-material, ensuring alignment with other sections of LArSoft.org
   - Continuously review, enhance and update this documentation as needed.

   Resources: SciSoft team

   March 23 update

   - Adding workshop updates and videos.

June 8 update

- Documentation effort has been directed at the refactored geometry system. Drafts for a significant fraction of the documentation required exists.

Sept 12 update

- Updated Docker information on LArSoft wiki to reflect licensing changes.

7. **Provide consultative support to experiments for transitioning to the refactored LArG4 detector simulation framework, or developing simulations for new detectors**
   Priority: high
   Provide consultative support for efforts within the experiments to migrate to the new simulation framework. Advise and provide guidance in refactoring experiment code to be compliant with the workflows and interfaces within that framework.
   Resources: Hans Wenzel for LArG4 / artg4tk, SciSoft team for geometry, other infrastructure

   Feb 22 update

   - Facilitated introduction of fixes written by experiment collaborators that enable the full optical simulation to run within the refactored LArG4.

   June 8 update

   - Have formed a dedicated team to work on the DUNE ND LAr simulation. This work is currently aimed at the first step in the refactored LArG4, and at understanding differences between the ND LAr developed EDEPSIM framework and LArG4.

   Sept 12 update

   - No update.

8. **Provide support for multi-experiment event display capable of dedicated and integrated displays of TPC charge collection and photo-detector systems, along with external detectors such as CRTs, spectrometers, calorimeters, etc.**
   Priority: high
   Decide whether LArSoft will provide support for Titus. Support entails re-architecting Titus to make proper, extensible use of the LArSoft Geometry system, provide a library of primitives to assist in displaying LArSoft geometry and data product information and in controlling the display, and a system of plug-ins for experiment-specific elements. Agreements on long-term maintenance and code ownership are part of this work as well.

   March 23 update

   - No activity

   June 8 update

   - No activity

Sept 12 update

- No activity.

2. Long term and continuing priorities for 2023

We define long term priorities as those that we do not anticipate completing within CY2023 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. **Provide technical expertise, advice and leadership in matters related to LArSoft and software development**

2. **Develop and execute support plans as needed in areas where assistance from the Project has previously been requested by experiments, or advised by the Project.**
   Priority: per request
   a. Align Wire-cell reconstruction and simulation with LArSoft workflows and interfaces (eg, factoring electron drift from anode response simulations via drifted electrons)
   b. Extend the NuWro direct integration model to include the GiBUU event generator framework and the NEUT event generator.
      i. June 2021: Work proceeding within Genie Collab to provide event library solution to integrating external generators. This requires stand-alone running of the generators to produce the libraries, so is not the direct integration sought. Discussing with Genie team about the direct alternative
   c. 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.
   d. Coordinate between GArSoft and LArSoft as needed to allow GArTPC design and development
      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

   e. SBND data reduction strategies.
   f. 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.
      i. Sept update: In light of Fermilab Frameworks Workshop held in June 2023 and on-going work to develop a prototype framework that can meet DUNE framework

requirements, we consider that this item is now out of scope and being addressed by the larger framework effort within CSAID.

g. Architecture work to support a common framework for data preparation
h. Education and assistance with optimization of critical production workflows, particularly for the SBN experiments.
i. Provide a solution suitable for running TextFileGen at production-scale

3. **Architecture work to facilitate integration of machine learning into LArSoft algorithms**
   Priority: per request
   Possible targets include standard modules to provide translation and downsampling for image-based algorithms. A work plan item can be opened once a set of deliverables is defined.

4. **Support for integration of community-supported event displays within art and gallery based LArSoft jobs.**
   Priority: per request

5. **Work on user support, bug fixes, emergency feature requests, software builds, releases.**
   Priority: per request
   Resources: SciSoft team

6. **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: per request

Other considerations on work planning

In response to input received from the experiments, the Project will pursue the following topics as time allows:

- Improving informal channels of communication, which have suffered during the Covid era. To facilitate more efficient communication, community members are encouraged to engage directly with SciSoft team members before and after opening issue tickets. In addition, SciSoft team members understand that greater visibility at experiment software meetings would be helpful.
- Seeking information on recent LArG4 migration experiences, then using this information to enhance the available documentation.
- Seeking information on recent experiences using LArSoft on HPC resources, then using this information to create a common resource for others interested in using HPC.
- Developing a plan to ensure LArSoft is compatible with reading and writing data in HDF5 format, as well as other data formats.
- Tracking progress on DUNE and related community-based data processing framework developments

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.
- Per request: The priority will be set in consultation with the requesting experiment or ticket author.

Appendix B: Major observations from one-on-one meetings with each experiment in October and November of 2022

(Topics in common across multiple experiments)

1. **Common**
    a. Multi-threading
        i. Services that access databases
        ii. Algorithm code in question differs across expts
        iii. Module label that only allows multi-threading within the module (SBN)
    b. HPC
        i. Dispatching to GPUs (beyond NuSonic)
        ii. For SBN, implies Spack and multi-threading
    c. Spack
        i. DUNE and SBN, though not obviously for the same reasons
        ii. A need to accelerate pace for delivering a serviceable version for LArSoft. Do not believe all features that are currently the bottlenecks are needed. DuneDAQ, for instance, is using spack to build releases

2. **DUNE** - Met with Tingjun Yang, Tom Junk, Heidi Schellman
    a. Pixels:
        i. Functional geometry system that includes pixels, wires, +
        ii. Get basic simulation and reconstruction working for pixels. Use data from new detectors to drive that development in a way that will create the foundation for the harder problems further down the line. Pursue all related infrastructure work.
    b. Multi-threading (related to services + non-WC algorithm code; focus though is trigger frame serialization)
    c. Support for Near Detector technologies and integration
    d. Support and integration for external tools (WireCell, ML, Pandora)
    e. Evolution to HPC resources/GPUs, etc
        i. For instance, though we can dispatch DL inference to GPUs, we cannot currently dispatch deconvolution tasks.
        ii. Noted that some efforts related to HPCs are being funded under DOE funding opportunities. Work is both within experiments and at the lab.
    f. Continued evolution of build systems (Spack!)
    g. Support for multiple neutrino generators
    h. Improvements to underlying framework
        - Support for HDF5
        - Support for metacat
        - [possible support for event building in offline]
        - In the long term: support for SN data stream

3. **ICARUS** - Met with Tracy Usher and Daniele Gibin
    - Multi-threading work. Known blockers have been database services, in particular for pedestals and channel status. May no longer need pedestals, since they will be calculated on the fly, so just need channel status, which will remain important.
        a) Also aware of another library that needs to be made thread safe – uses STL vectors – but this has been low priority pending a solution to the database services. Fixing the latter will motivate work on the former.
    - Event display support is needed.
        a) The LArSoft display is still useful, especially for reconstructed objects, but the support load currently exceeds the capacity of the two people who provide it.
        b) ICARUS preferred solution is Titus, though it currently only works for raw digits and wires. It would be useful to multiple experiments to have a unified Titus that works for all. The main problem with Titus is geometry. It currently reads in parameters from LArSoft services, but then uses internal representation for geometry, which leads to complicated calculations to turn channel numbers into positions used in the display. Fixing this would require digging into Titus to understand what has been done for geometry, and then figure out how to do it properly (ie, getting answers from geometry service and detector properties service, etc)
        c) Noted that it would also be useful to have CRT information displayed for debugging. Ditto for photo-detector systems. Integrated displays of this would be useful.
        d) LArSoft will continue to push for support in this area
    - LArSoft support is needed in creating the base releases and patchesICARUS needs as required.
        a) Expect SBN2022b release, to be used in stage 0 processing (signal processing, low-level reconstruction), in the next couple of weeks
        b) The next release, SBN2023a, will be early next year, and will be needed for stage 1 processing (high-level reconstruction)
        c) Supporting production releases will remain a priority for the SciSoft team, though noted that additional effort is needed in this area. Will again raise the issue with management.

4. **MicroBooNE** - Met with Herbert Greenlee.
    a. Supportive of ongoing multi-threading efforts.
    b. Supportive of making it easier to use HPC type computing within LArSoft.
        i. Would like to see working examples that people can use.
    c. For the workshop, would like a tutorial.
        i. The workshop scheduling should steer clear of the FCRSG timeframe

5. **SBND** - Met with Andrzej Szelc.
    a. Looks like SBND needs to run Corsica, not overlay. If they run with Corsica, they need a lot more computing power which is a problem. They have shown that they can generate cosmics on Theta and HPC centers. Don't need to transfer data in, just run simulation, generate a lot of events and copy them back. Anything that helps running LArSoft on HPCs, will help.
    b. Trying to have workflows that outsource part of SBND work to an HPC machine. They've been splitting each event to run on a separate core, which is not the way HPC centers intended. It runs, but took a long time to make it work.
    c. Interested in the prospect of being able to run photon ray tracing on GPUs. Might choose to run ray tracing on-the-fly if that were possible. Inquired about possibility of running particle ray tracing on GPUs. Is it? Will it be?
    d. Will need access to multiple event generators. Might need to use new versions of GENIE or tune variations, or be able to test different generators. NuWro at a minimum. Achilles possibly for BSM, but have not investigated. Currently using TextFileGen for all but GENIE, which is cumbersome. Supports LArSoft effort to make this easier and to decouple GENIE versions.
    e. Will continue to use Pandora for reconstructions, and WireCell for 2D deconvolution
    f. For an in-person workshop, suggested the possibility of one center in the US and one center in Europe. People In Europe are more likely to be able to go to CERN for two days than Illinois. Would need someone to host the CERN (or UK) site.
    g. If Titus becomes the go-to Event Display, request to make it easy to use.

6. **SBN Data/Infrastructure** - Met with Wesley Ketchum, Giuseppe Cerati
    - A big push for SBN going forward will be the ability to run on HPC, bringing that forward to production. So Spack and multi-threading continue to be major priorities. They should remain high in the work list. Is there a way to accelerate the work on the Spack side? Also most experiments don't require an understanding of how Spack works, or some of the features that seem to be taking the most time to develop on the Spack side. They need a recipe that clarifies how to proceed, plus help in real-time if issues occur in transitioning to Spack. Noted that DUNE DAQ code is a good example of success with a minimal Spack capability.

    - Would like a module label for multi-threading that allows for multi-threading only within the module. It would not, for instance, imply that the module itself could be run in multiple threads via art multi-threading. (It might depend on how multi-threading is being done within an event.)

    - Interested in architecture work to facilitate integration of machine learning into LArSoft algorithm. (Noted that this is item #3 from long term goals in the current work plan.). Need more examples and publicity about how to use NuSonic, for instance, which should be standardized across LArSoft. Generally, need to lower the bar for using NuSonic and ML in LArSoft.

- ICARUS is working on data overlay infrastructure. May need some support for this, but the work on the LArSoft side is straight-forward.
- Execution speed of simulation is an issue. Will be seeking some discussion of this and possible ways to accelerate some steps at the multi-threading workshop.