LArSoft and Future Framework Directions at Fermilab

or Preparing for DUNE’s data-processing needs

Chris Jones for Kyle J. Knoepfel
CHEP 2019 in Adelaide, Australia
Preparing for DUNE’s data-processing needs

• Significant effort at Fermilab’s Scientific Computing Division going into understanding and planning for the data-processing needs of:
  – CMS and LHC Run 3
  – DUNE
Preparing for DUNE’s data-processing needs

• Significant effort at Fermilab’s Scientific Computing Division going into understanding and planning for the data-processing needs of:
  – CMS and LHC Run 3
  – DUNE

• This talk focuses on Fermilab-supported aspects of DUNE’s data-processing:
  – Framework
  – LArSoft reconstruction software
• Near detector at FNAL
• Far detector in South Dakota
  – Up to 4 underground modules
    • Several different technologies
  – Each module contains ~17kt of liquid Argon
  – E.g. single phase modules
    • has 150 Anode Plane Assemblies (APAs)
    • APA has 2560 wires (i.e. channels).
    • Each channel sampled every 500 ns, with 12-bit precision.

Single phase module readout: 0.55 MB per sample
DUNE’s data readout at the far detector

• Readout for full module is a trigger record.

• Neutrino beam (rate of 0.83 Hz)
  – For single phase modules
    • 41.5 MB for one APA
    • 6.22 GB for full module

• Supernova burst
  – For single phase modules
    • 300 MB for single channel
    • 750 GB for single APA
    • 120 TB for full module

• LArSoft and the reconstruction framework must be able to handle a very wide range of data rates.
DUNE’s framework computing needs

• The atomic processing unit for DUNE is fluid.
  – DUNE thinks in terms of a trigger record, which may contain multiple regions of interest.
  – Each region could be considered an “event,” but the shape of the event is not necessarily consistent from one processing stage to the next.
  – Instead of rigid hierarchy of Run ⊃ SubRun/Luminosity Block ⊃ Event, the hierarchy might be *user-defined and dynamic.*
DUNE’s framework computing needs

• The atomic processing unit for DUNE is fluid.
  – DUNE thinks in terms of a trigger record, which may contain multiple regions of interest.
  – Each region could be considered an “event,” but the shape of the event is not necessarily consistent from one processing stage to the next.
  – Instead of rigid hierarchy of Run \supseteq\ SubRun/Luminosity Block \supseteq\ Event, the hierarchy might be user-defined and dynamic.

• Collider-based frameworks are an awkward fit.
  – Can one of the existing frameworks be adjusted to support such processing?
  – Is a new framework required?

Fermilab’s future frameworks initiative (FFI) seeks to answer these questions.
FFI item 1: A convergence of two frameworks?

“In many areas it is recognised that different experiments could have adopted common solutions, reducing overall development effort and increasing robustness and functionality. That model of duplicated development is not sustainable. We must endeavour to achieve better coherence within HEP for future developments to build advanced, open-source projects that can be shared and supported in common.”


• art was born in 2010 as a fork of the CMSSW framework.
• Since that time, both art and CMSSW framework developments have proceeded according to the needs of the experiments each framework supports.
In many areas it is recognised that different experiments could have adopted common solutions, reducing overall development effort and increasing robustness and functionality. That model of duplicated development is not sustainable. We must endeavour to achieve better coherence within HEP for future developments to build advanced, open-source projects that can be shared and supported in common.”


- art was born in 2010 as a fork of the CMSSW framework.
- Since that time, both art and CMSSW framework developments have proceeded according to the needs of the experiments each framework supports.
- Sustainability and maintenance concerns have triggered discussions regarding the feasibility of consolidating the art and CMSSW frameworks into one.
- Discussions are ongoing.
- **Bottomline:** Fermilab takes the HSF-mentioned concerns very seriously.
LArSoft

“The Liquid Argon Software (LArSoft) Collaboration develops and supports a shared base of physics software across Liquid Argon (LAr) Time Projection Chamber (TPC) experiments.”

- larsoft.org

• C++ toolkit of experiment-agnostic LArTPC reconstruction algorithms.
  – Interfaces with other third-party libraries (ROOT, G4, WireCell, PandoraPFA, etc.)
  – Relies on the art framework
**LArSoft**

“The Liquid Argon Software (LArSoft) Collaboration develops and supports a shared base of physics software across Liquid Argon (LAr) Time Projection Chamber (TPC) experiments.”

- larsoft.org

• C++ toolkit of experiment-agnostic LArTPC reconstruction algorithms.
  – Interfaces with other third-party libraries (ROOT, G4, WireCell, PandoraPFA, etc.)
  – Relies on the *art* framework

• A software project supported primarily by volunteers in the neutrino community
  – Started in 2008 by a postdoc; formal support now provided by Fermilab
  – Most contributors are graduate students, postdocs, and staff scientists from multiple experiments (including DUNE) and many institutions.
  – They have their own research to do, with generally no funding for LArSoft development.
The Liquid Argon Software (LArSoft) Collaboration develops and supports a shared base of physics software across Liquid Argon (LAr) Time Projection Chamber (TPC) experiments.

- C++ toolkit of experiment-agnostic LArTPC reconstruction algorithms.
  - Interfaces with other third-party libraries (ROOT, G4, WireCell, PandoraPFA, etc.).
  - Relies on the art framework.
- A software project supported primarily by volunteers in the neutrino community.
  - Started in 2008 by a postdoc; formal support now provided by Fermilab.
  - Most contributors are graduate students, postdocs, and staff scientists from multiple experiments, including DUNE.
  - They have their own research to do, with generally no funding for LArSoft development.

Experiments using LArSoft:
- DUNE
- μBooNE
- ArgoNeuT
- LArIAT
- SBND
- Short-Baseline Near Detector
C++ coding

- Project started before C++11
- 350K lines of code
  - Often many additions from C++ non-experts
  - Rarely code subtractions
- Code needs to be updated and maintained
LArSoft challenges

• C++ coding
  – Project started before C++11
  – 350K lines of code
    • Often many additions from C++ non-experts
    • Rarely code subtractions
  – Code needs to be updated and maintained

• Contribution model relies on a single code librarian merging features by hand.
  – Until now, LArSoft has not used a pull-request model
  – Breaking changes are presented (in summarized form) at a coordination meeting
  – No dedicated forum or process for formally reviewing code
Current LArSoft efforts

• Clean up
  – Fermilab guidance to LArSoft contributors: “Remove more lines of code than you add.”
  – Remove unnecessary files
    • Most effective, but hardest to do.
  – Remove unused/unnecessary functions and virtual tables
    • Effective, but still requires coordination.
  – Remove unnecessary header and library dependencies
    • include-what-you-use.org has been quite helpful (but not perfect) in reducing compile times, library sizes, and LOC.
Current LArSoft efforts

• Clean up
  – Fermilab guidance to LArSoft contributors: “Remove more lines of code than you add.”
  – Remove unnecessary files
    • Most effective, but hardest to do.
  – Remove unused/unnecessary functions and virtual tables
    • Effective, but still requires coordination.
  – Remove unnecessary header and library dependencies
    • include-what-you-use.org has been quite helpful (but not perfect) in reducing compile times, library sizes, and LOC.

• Move to pull request model
  – Almost ready to switch over to using GitHub
  – Many of CMSSW’s GitHub scripts were ported over to work for LArSoft
Current LArSoft efforts

• Thread-safety
  – Several important LArSoft facilities were not designed with thread-safety in mind.
  – Large, expansive campaign to make critical algorithms thread-safe
    • Remove reliance on global, mutable state
    • Adopt “const all the things” coding idioms
    • Favor declarative coding patterns over procedural ones
Current LArSoft efforts

• Thread-safety
  – Several important LArSoft facilities were not designed with thread-safety in mind.
  – Large, expansive campaign to make critical algorithms thread-safe
    • Remove reliance on global, mutable state
    • Adopt “const all the things” coding idioms
    • Favor declarative coding patterns over procedural ones

• Reduce coupling to the framework
  – Clarifies intention of library
  – Improves separation of concerns
  – Eases maintainability
  – Improves testability

“The most important single aspect of software development is to be clear about what you are trying to build.”
- Bjarne Stroustrup
DUNE working timeline

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Software used (related to this talk)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late 2021/Early 2022 ProtoDUNE II</td>
<td>Framework and DAQ system that satisfy the needs.</td>
</tr>
<tr>
<td>Aug. 2024 Installation of Module 1</td>
<td>LArSoft</td>
</tr>
<tr>
<td>Aug. 2025 Installation of Module 2</td>
<td>LArSoft</td>
</tr>
<tr>
<td>2026/2027 Start of DUNE data-taking</td>
<td>LArSoft</td>
</tr>
</tbody>
</table>

- According to DUNE computing, a goal is to have a “working prototype software environment” by the time ProtoDUNE-II is operational in 2021.
Conclusions

- Fermilab working hard preparing for DUNE’s computing needs
- LArSoft used currently for DUNE’s reconstruction steps
  - Undergoing changes to improve collaboration model and update code
  - Multi-threading effort underway to better utilize many-core CPUs
- Discussions underway regarding what requirements are necessary for DUNE’s data-processing framework
  - Future frameworks initiative begun to support DUNE’s computing tasks
  - Assessing feasibility of consolidating Fermilab-supported frameworks, and whether such consolidation delivers a framework that meets DUNE’s needs
- Goal is for software to be ready by the time DUNE needs it.

Thank you