

LArSoft Tools and Technology Workshop 2017

Tuesday, 20 June 2017 - Tuesday, 20 June 2017

Fermilab

Program

The topics the workshop will cover are:

Introduction to computing concurrency and vectorization within LArSoft

We will introduce the various types of parallelism in computing, and how they might be used, for instance, to address managing the large memory footprint needed to analyze LArTPC data. We will also cover the strategies that will soon be included in the art event processing framework, and what we need to know now in order to allow LArSoft algorithms to operate effectively in this environment.

All of our grid-based computing resources support some level of single-instruction, multiple-data processing. Utilizing these capabilities would bring currently unused resources to bear on those classes of problems the lend themselves to this type of processing model.

New and improved LArSoft Continuous Integration system

Continuous integration (CI) is an important tool that allows us to find problems quickly and greatly enhances our ability to share code across experiments. A number of improvements to the LArSoft CI system have been released that make it more intuitive and easier to use by non-experts. This session will discuss the new features of the system, and how you can use the system in your daily programming to help identify problems before they go into releases.

The SPACK and SPACKdev build system

LArSoft and the art event processing framework are moving to a new build system that has been adopted by the HEP Software Foundation, a community-based effort to coordinate efforts and find common solutions to the problems that are common across HEP software and computing. This hands-on session will provide the information and best practices you need in order to be able to configure and build your code using this new system.

Debugging and profiling tutorial

We all need to debug and profile our code at various times, right? This session will discuss some of the tools we have available on some of our supported platforms, and some techniques you can use to find bugs quickly, or find the places that are unnecessarily slowing your code down.