



# Spack Progress

Stephen White

August 2022



# Software Build Status

- CI LArSoft
  - Vito is working on the converting LArSoft dependencies to Spack.
    - 22 packages in LArSoft stack and ~35 dependent products (excluding art and its dependent packages)
    - Possibly a portion of them are already used by other spack users and we can use their spack recipes from upstream.
- LArSoft
  - The v09\_54\_00 build is fully using Cetmodules. All LArSoft-using experiments except LArIAT are using v09\_56\_XX.
- Cetmodules
  - The development branch of FNAL's Spack fork is now capable of building all Cetmodules' Perl/documentation dependencies. This is being incorporated into Marc's spack-infrastructure system for use in the LArSoft CI.

# Spack Software Status

- Spack 0.19-dev.fermi branch
  - Chris G. and Marc M. are working on this branch,
  - Includes features from main Spack repo not yet released upstream
  - Testing art/larsoft builds, especially externals needed.
- The biggest issue that we are dealing with are AI based products:
  - Three (at least) AI products: PyTorch, TensorFlow and Triton, all with problematic build systems and integration issues.
  - Can we (AI Group, mgmt?) evaluate and pick one to put effort into?
  - Dedicated expert emphasis on these AI issues?

# Experiment Build Status

- ICARUS
  - We met, on the 19th, with ICARUS about starting the conversion to Spack.
  - Patrick presented the technical roadmap on starting the conversion.
  - ICARUS will meet with their Release Production team to begin the conversion to Cetmodules.
    - ICARUS will be updating to the latest LarSoft (comprising Cetmodules-using packages) for their upcoming production.
    - Use of the automatic migration tool prior to the production freeze will provide ICARUS with the ability to incorporate Spack:
      - supporting existing UPS packages and MRB development going forward
      - avoiding delays to production schedules
      - preserving the ability to incorporate developments in experimental code back to the production stream