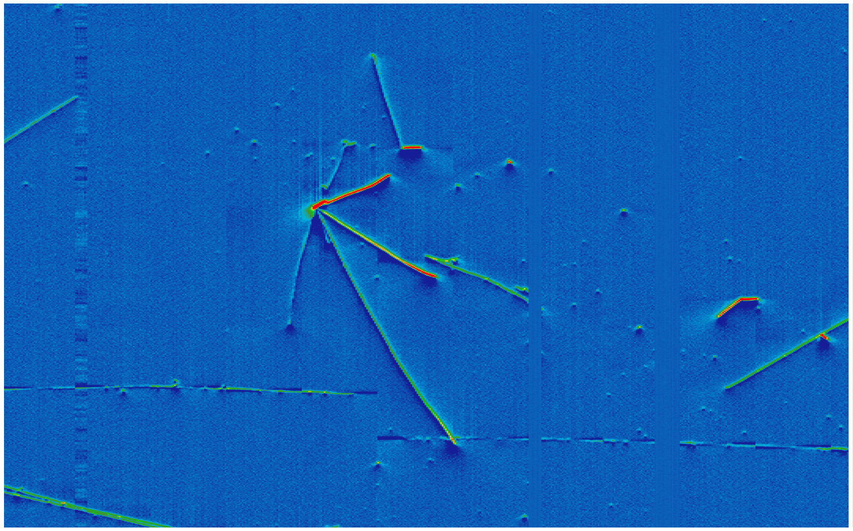
Reconstruction for LArTPC Neutrino Detectors Using Parallel Architectures



Sophie Berkman

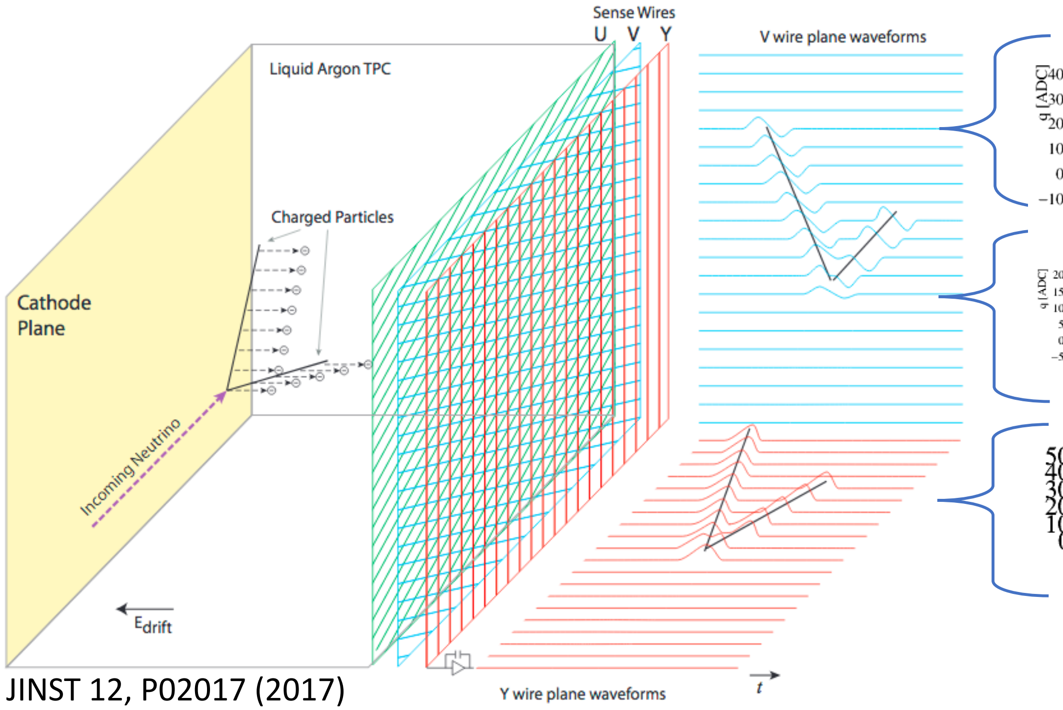
for the SciDAC HEP Reconstruction and HEP at HPC groups

November 24, 2020

SciDAC
Scientific Discovery through Advanced Computing

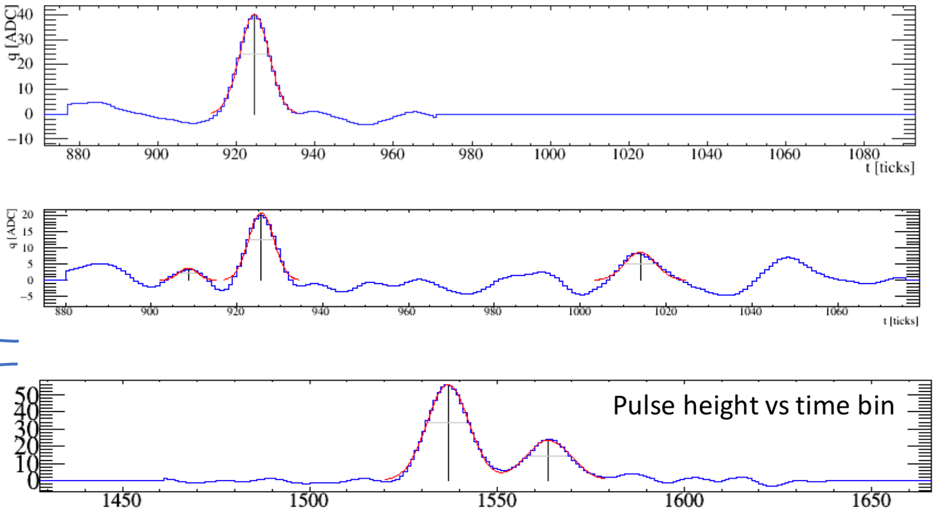
Fermilab

LArTPC Neutrino Experiment Reconstruction



JINST 12, P02017 (2017)

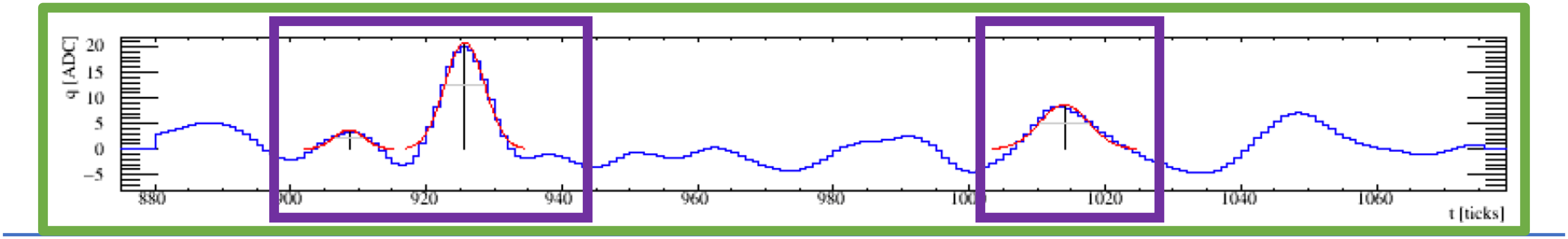
Example Wire Pulses



- Reconstruction time scales with the size of the detector: MicroBooNE: 8,256 wires DUNE: 384,000 wires/ 10 kTon cryostat
- Use parallelism and modern computing architectures to improve reconstruction time for an event
  - Many levels of parallelism in LArTPCs: wires, planes, cryostats

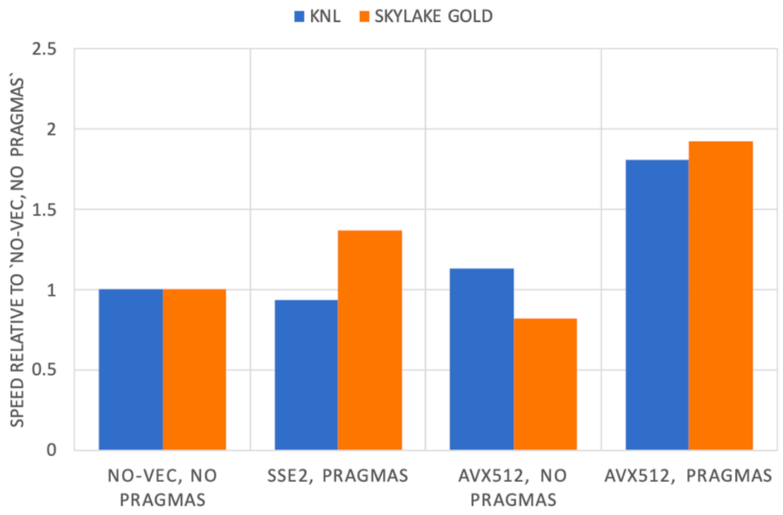
GausHitFinder Optimization

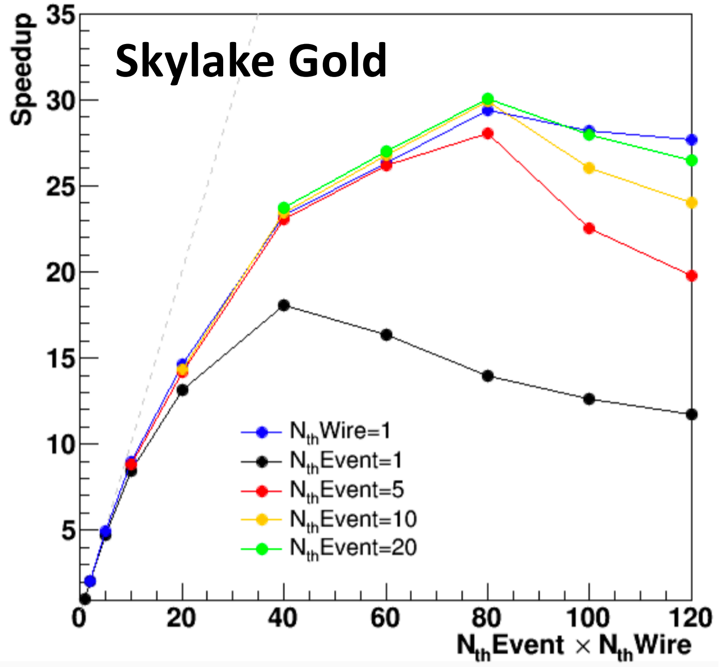
- GausHitFinder: part of wire signal processing
    - **Algorithm:** Charged particles produce pulses on wires. Identify and extract parameters associated with pulses (position, amplitude, width).
    - **Suited for parallel implementation:** Wires are independent; can be processed independently
    - **Potential for impact:** Up to 30% of reconstruction time, depending on the experiment
- Parallel developments tested using stand-alone code:
    - Implement Levenberg-Marquardt algorithm to do the fitting instead of ROOT/Minuit: ~8x faster than ROOT version
    - **Vectorization:** Compiler using avx512, explicit pragma calls on most time consuming loops as determined by profiling
    - **Multi-threading:** Nested OMP parallel for over events and regions of interest using dynamic scheduling



Optimized GausHitFinder Results

- Stand alone code:
  - Compiled with avx512 and icc
  - **Vectorization:** 2 times faster on KNL and Skylake
  - **Multi-threading:**
    - Near ideal scaling at low thread counts
    - Up to 30 times faster on Skylake, 100 times faster on KNL
- Integrated code into LArSoft for general use by LArTPC experiments
  - **Single threaded:** 12 times faster for MicroBooNE, 7 for Icarus
  - Both Icarus and ProtoDUNE use this version of the hit finder

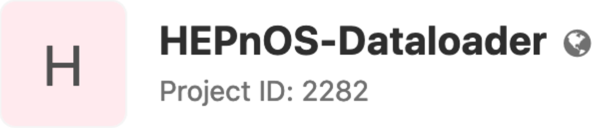




arXiv:2002.06291

HPC Workflow

- Developing workflow to run part of next Icarus production on Theta HPC at ALCF
- Strategy:
  - Local build of LArSoft with **spack** on Theta:
    - Take advantage of parallelism not available in a typical grid production environment.
    - Customize compiler options for specific pieces of code: ie. avx512+icc for optimal vector speed increases with GausHitFinder
  - **HDF5-HEPNOS** to organize the data
  - **DIY** to distribute the data across nodes
- First spack LArSoft/icaruscode build on non-Fermilab computer complete; work ongoing to compile on Theta


H
HEPnOS-Dataloader
Project ID: 2282

MATHEMATICS AND COMPUTER SCIENCE DIVISION

DIY: Do-it-Yourself Analysis DIY

Argonne NATIONAL LABORATORY

Conclusions

- Vectorization and multi-threading can be used to optimize LArTPC reconstruction algorithms
- GausHitFinder algorithm vectorized and multi-threaded:
  - Levenberg-Marquardt algorithm implemented to do the fitting instead of ROOT
  - Up to **200x faster** in stand alone version
  - **O(10x)** faster in LArSoft depending on the experiment, and before multi-threading
  - Paper in preparation
- Work ongoing to run production for Icarus at the Theta ALCF HPC to take full advantage of parallelization speed increases