Reconstruction for Liquid Argon TPC Neutrino Detectors Using Parallel Architectures

S.Berkman, G.Cerati, A.Reinsvold Hall, M.Wang (Fermilab);
B.Gravelle, B.Norris (UOregon)

CHEP2019 - Nov. 07, 2019

Project Goals

SciDAC
Scientific Discovery through Advanced Computing

- "HEP event reconstruction with cutting edge computing architectures" project supported by the **DOE SciDAC program**
  - https://computing.fnal.gov/hepreco-scidac4/; https://www.scidac.gov/
- Collaboration between physicists at Fermilab and computer scientists at UOregon
- Mission: **accelerate HEP event reconstruction using modern parallel architectures**

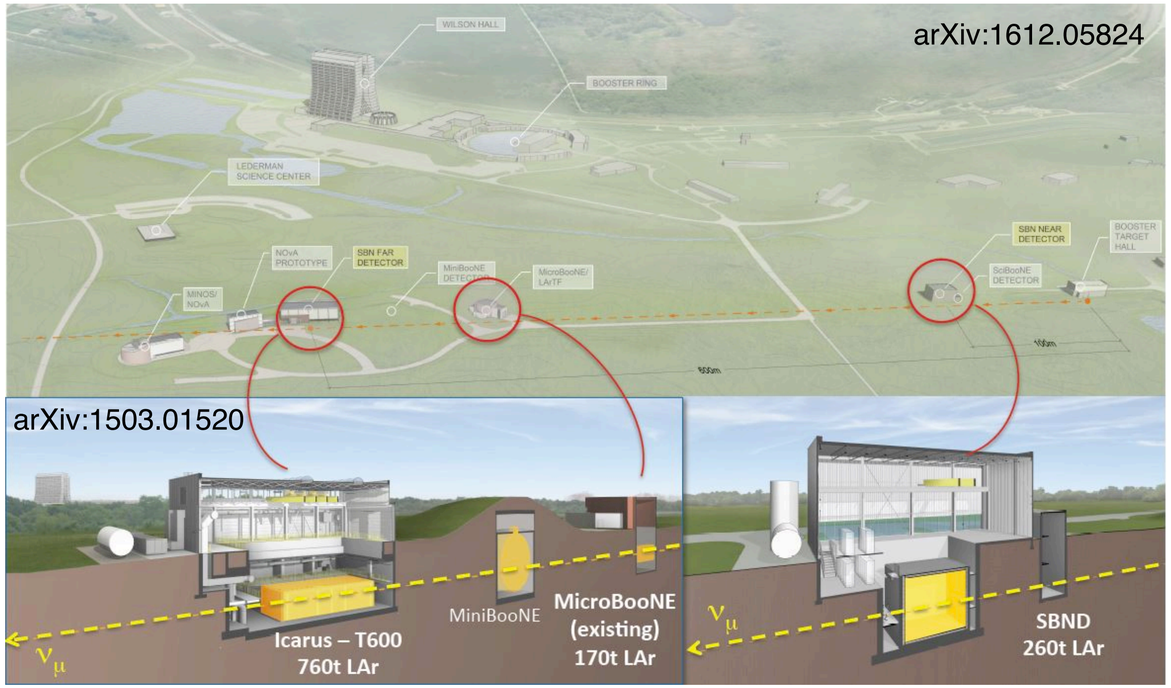
- Focus on two areas:
  - Novel parallel algorithm for charged particle **tracking in CMS** (larger collaboration, see later talk!)
  - Pioneer similar techniques for **reconstruction in LArTPC detectors**

- Goals of the project are the following:
  1. Identify key algorithms for the outcome of the experiments that dominate reconstruction time
  2. Re-design the algorithms to make efficient usage of data- and instruction-level parallelism
  3. Deploy the new code in the experiments' framework
  4. Explore execution on different architectures and platforms

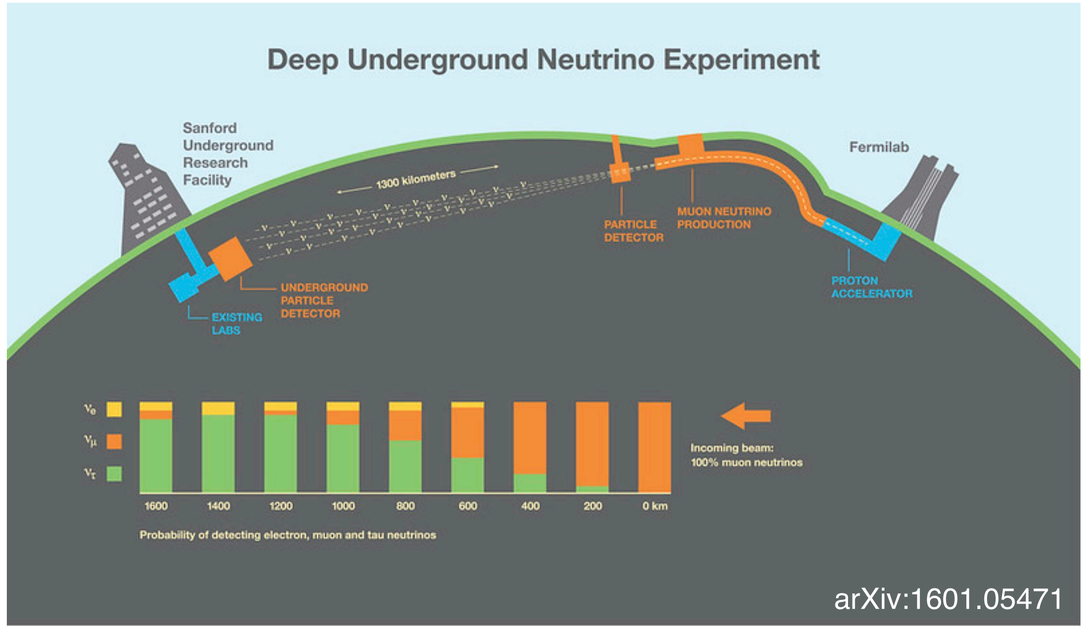
Fermilab

LArTPC Neutrino Experiments

- US-based neutrino physics program relies on present and future experiments using the **Liquid Argon Time Projection Chamber** (LArTPC) technology
- Addressing fundamental questions like: are there sterile neutrinos? do neutrinos violate CP symmetry? what is the neutrino mass hierarchy?



ICARUS MicroBooNE SBND



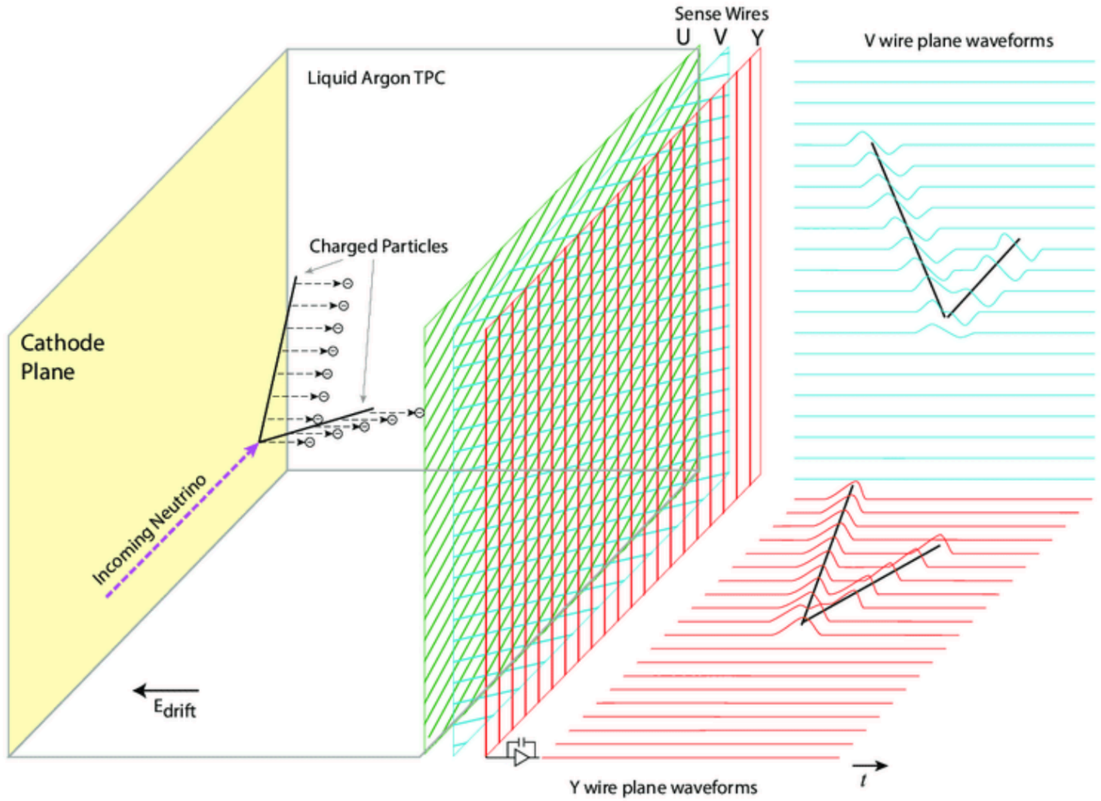
DUNE

Fermilab

LArTPC Working Principle

- Charged particles produced in neutrino interactions ionize the argon
- Ionization electrons drift in electric field towards anode planes
  - typically 2 induction planes, one collection plane
- Sense wires detect the incoming charge
  - waveforms have bipolar signal on induction planes, unipolar on collection plane
  - wires are oriented at different angles in the planes for 3D reconstruction

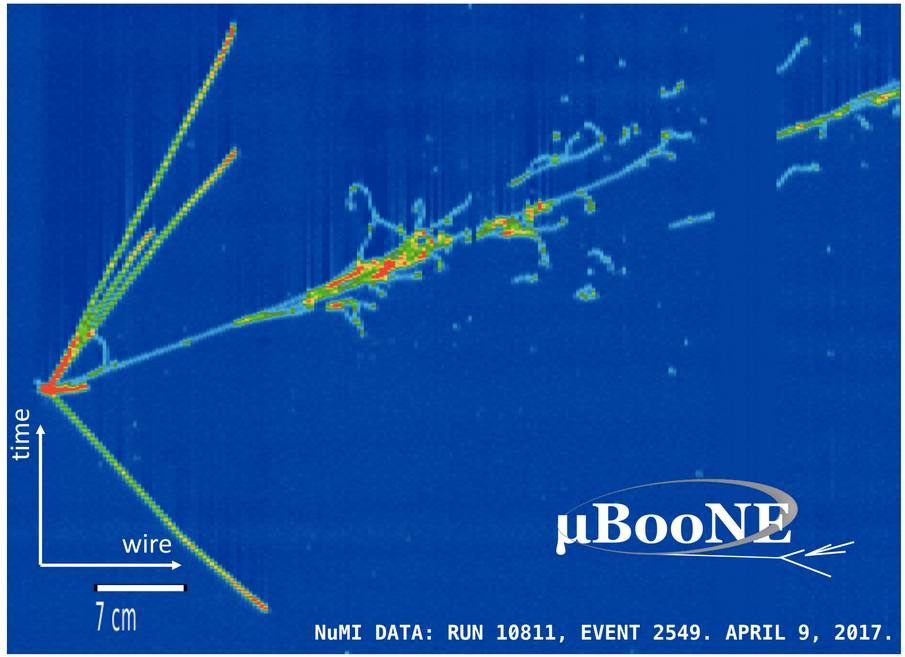
arXiv:1612.05824

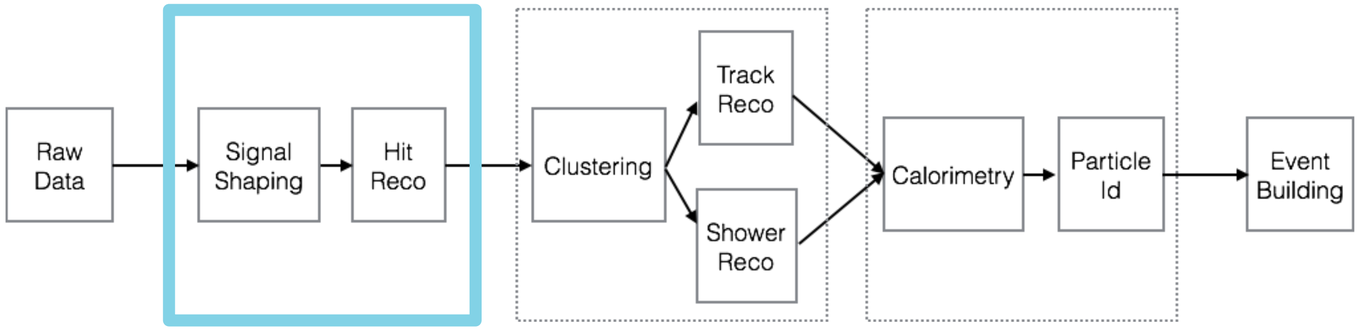


Fermilab

LArTPC Reconstruction

- Reco in LArTPC experiments is **challenging** due to unknown interaction point, many possible topologies, noise, contamination of cosmic rays
  - Takes **O(minutes)/event** in MicroBooNE
  - ICARUS ~5x bigger, DUNE Far Detector O(100)x bigger
- LArTPC detectors are modular in nature ➔ **parallelism!**
- LArTPC is a 'young' technology: reconstruction is a field of active development
  - traditional 'hit-based' reconstruction approach
  - image-based reconstruction: DL techniques or tomographic approach
  - early parts of the reco chain are more mature and common between different approaches



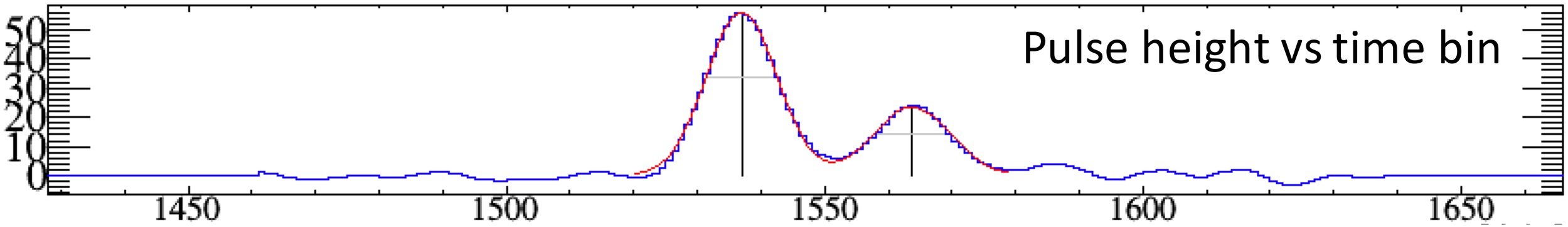


Typical reconstruction chain for LArTPC experiments

Fermilab

Feasibility Study: Hit Finding in LArTPC

- MicroBooNE: ~8k wires readout at 2 MHz, deconvolved wire signals are **Gaussian pulses**
- Hit finding: **identify** pulses and determine their **peak** position and **width**
- It takes a **significant fraction** of the reconstruction workflow
  - few percent to few tens of percent depending on the experiment
- Wires can be **independently processed**:
  - algorithm suitable to demonstrate speedup potential by parallelizing LArTPC reconstruction



Standalone Implementation

- Replicated LArSoft hit finder as **standalone code** for testing and optimization
  - LArSoft is the shared codebase for LArTPC experiments: https://larsoft.org/

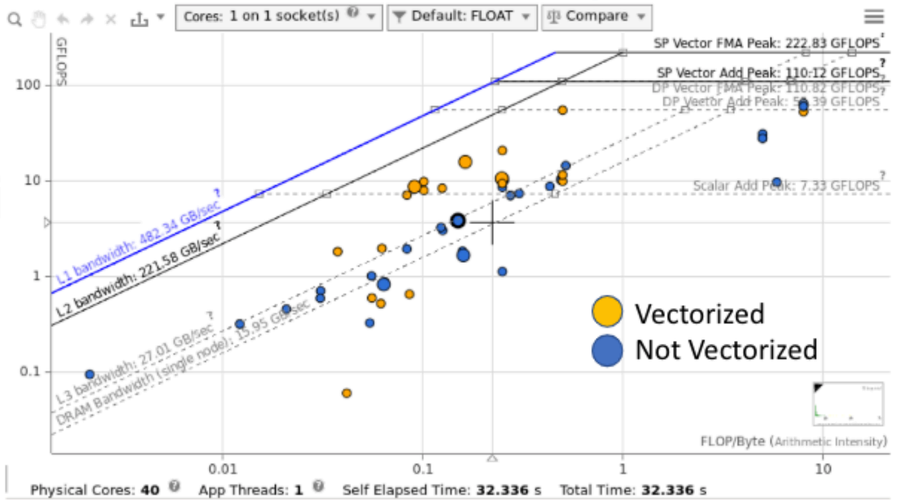
- Replaced Gaussian fit based on Minuit+ROOT with a local implementation of **Levenberg-Marquardt minimization**
  - gradient descent when far from minimum and Hessian minimization when close to it
    - implementation based on “Data Reduction and Error Analysis for the Physical Sciences”
  - include boundaries on fit parameters for better fit stability

- Early tests showed that standalone implementation is **~8x faster** than default
  - **before optimizations** and without any vectorization or multi-threading

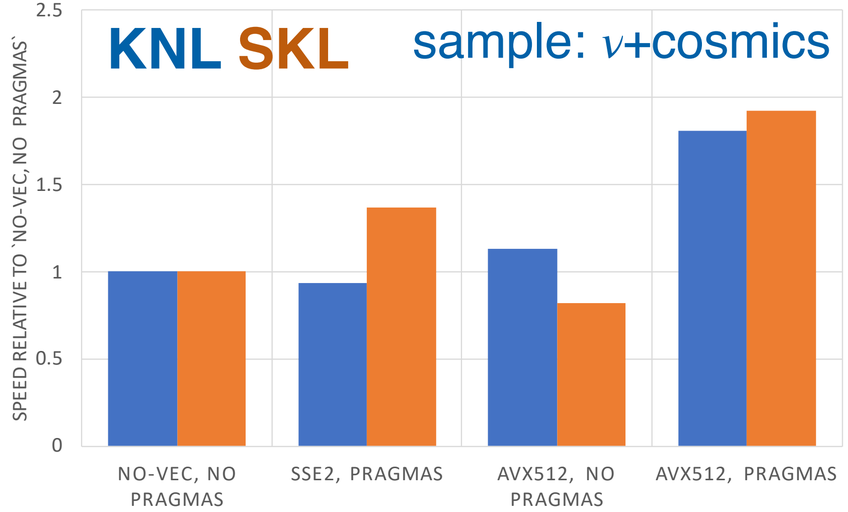
Vectorization Results

- Profiling the code (e.g. roofline) shows that most of the time is still spent in the minimization algorithm
  - number of iterations needed to converge is variable: difficult to vectorize across multiple hit candidates.
- We choose to **vectorize specific loops** within the algorithm, typically **across data bins**
  - main limitations: only a subset of the code is vectorized, number of bins is same order as vector unit size
- Close to **2x speedups**, both on Skylake Gold (SKL) and KNL when compiling with icc+AVX-512
  - Compared to Intel Math Kernel Library: our fitter is faster

Roofline analysis



Vectorization performance

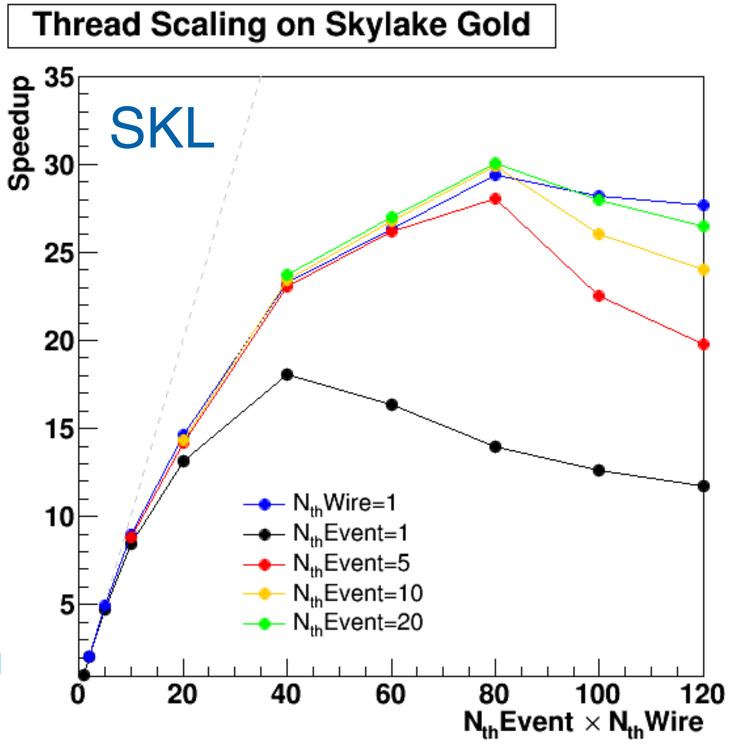


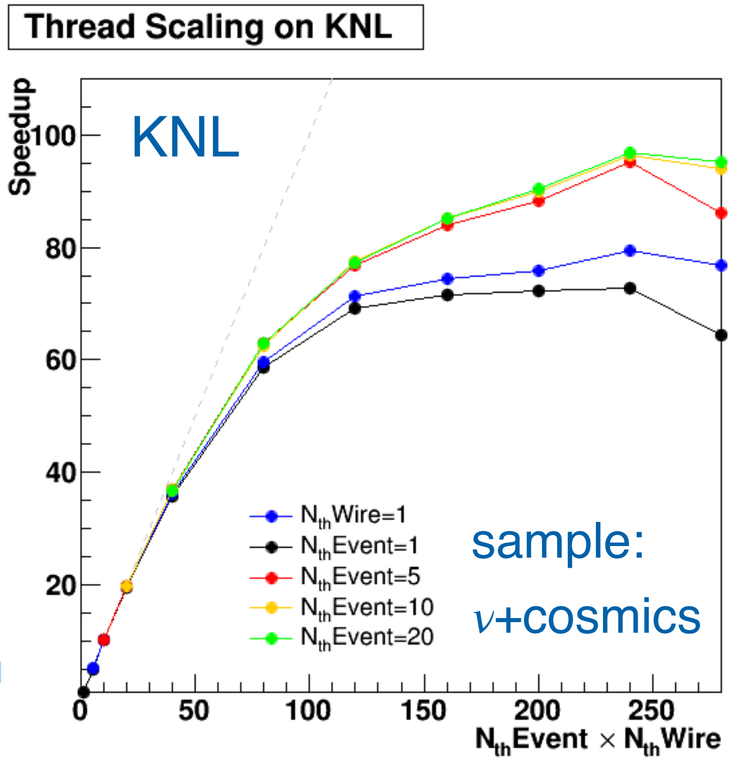
Fermilab

Multi-threading Results

- Implemented using **OpenMP** with dynamic scheduling
- Best performance achieved with **two-level nested parallelization**
  - *parallel for* over **events**
  - regions of interest on **wires**: *parallel region* with *omp for+critical* (output synchronization)

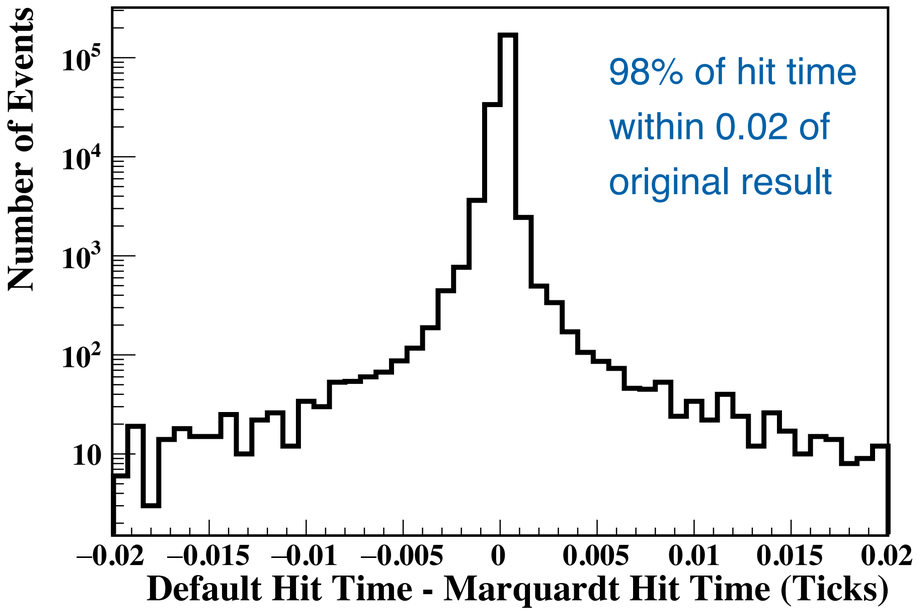
- Results show near ideal scaling at low thread counts
  - **speedup** increases up to **30x (95x)** for 80 (240) threads on Skylake Gold (KNL)

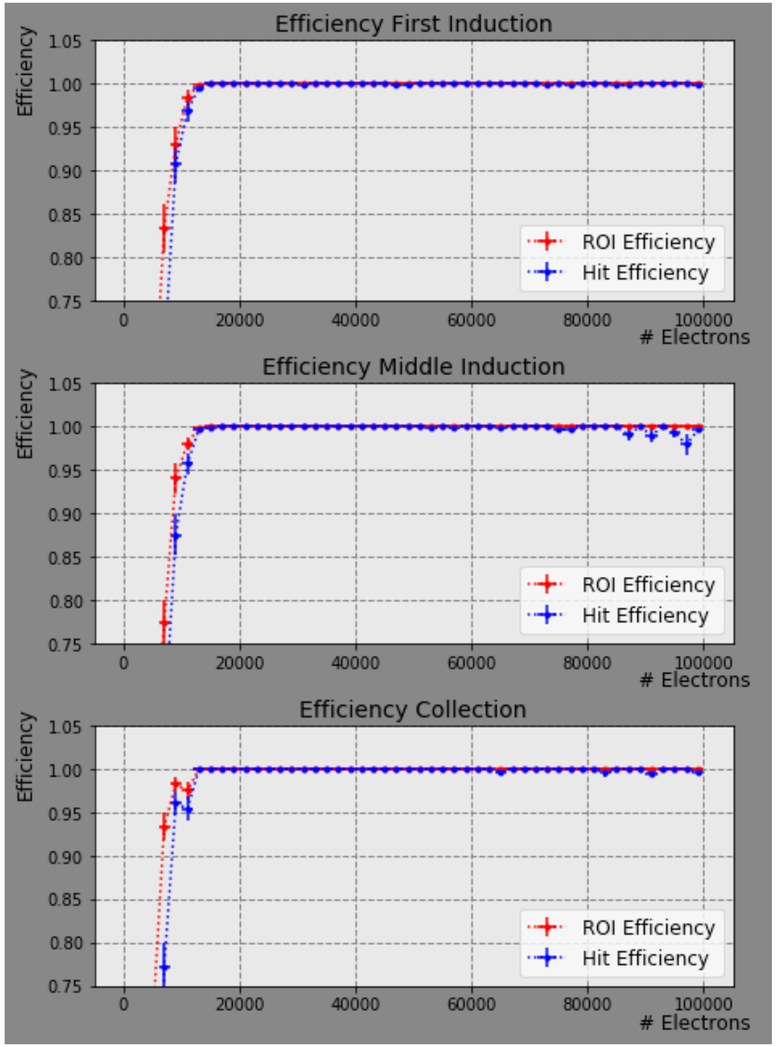




Validation of Algorithm Output

- Physics output **validated against original algorithm**
  - one to one comparison of hit parameters shows little difference
- Algorithm is **fully efficient** across all planes both in MicroBooNE and ICARUS
  - detectors with large differences in signal-to-noise ratio
  - waveforms with low S/N need fit parameters limits





Fermilab

LArSoft Integration

- Minimization algorithm **integrated** and used as a **plugin in LArSoft**
  - currently compiled with gcc by default
  - implementation of wire level parallelization with TBB is ongoing

- Testing the Levenberg-Marquardt hit finder in **MicroBooNE** and **ICARUS** reconstruction shows **speedups** of **12x** and **7x** respectively (single thread)
  - Old hit finder on ICARUS takes ~40% of total reconstruction time: ready to use the new algorithm in production!

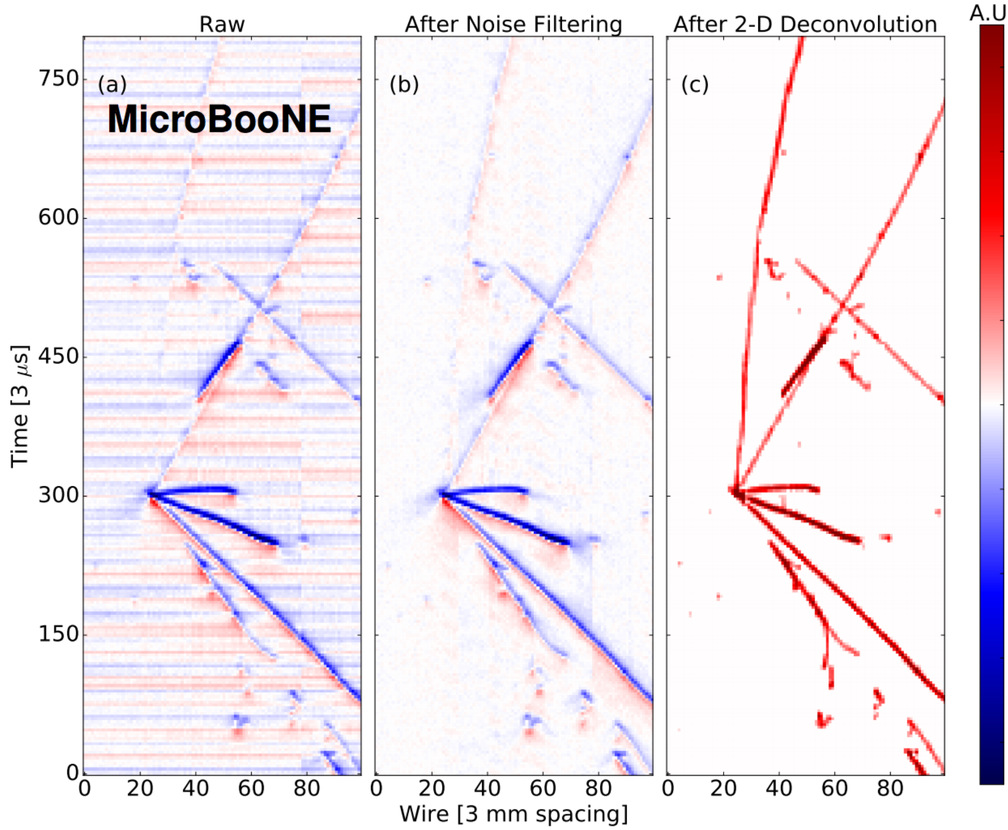
- First vectorized and multi-threaded algorithm for LArTPC!

Signal Processing

[1] JINST 12, P08003 (2017)
[2] JINST 13, P07006 (2018)
[3] JINST 13, P07007 (2018)

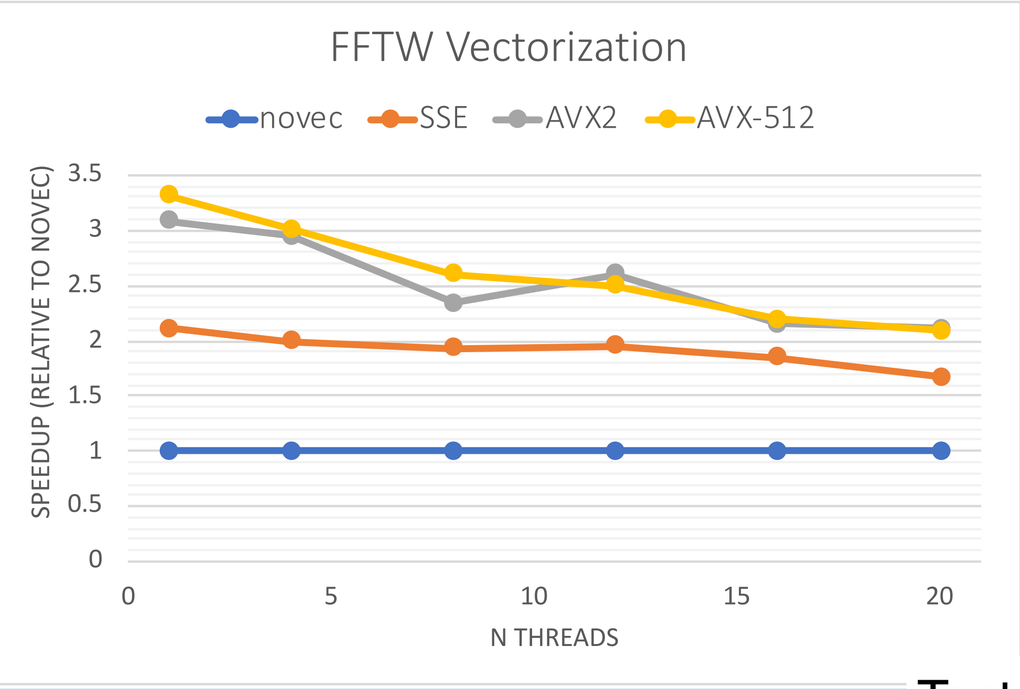
- Signal processing (SP) is the **most time consuming** step of LArTPC reconstruction
- It is composed of two steps [1,2-3]:
  - **noise filtering**: remove noise from waveform signal
  - **deconvolution**: remove electronics and field respons
- SP time scales with number of wires
  - large impact for DUNE

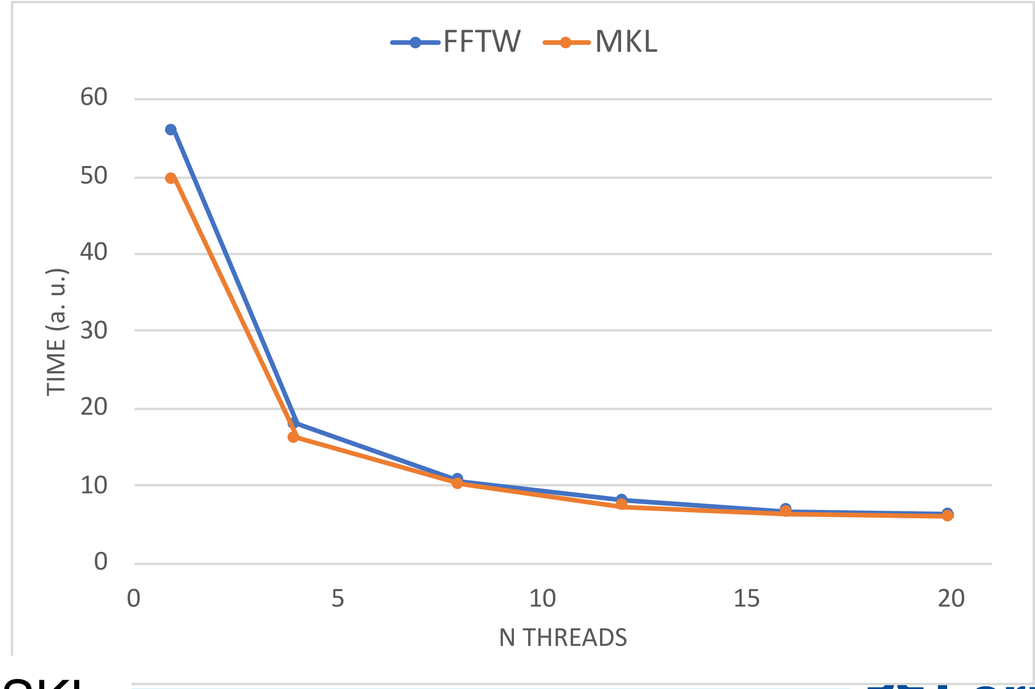
- LArSoft team working to make SP thread safe
- We focus on the core component of SP algorithms: the **Fast Fourier Transform** (FFT)



Tests with FFT Libraries

- Comparing state of the art **FFT libraries**: FFTW and MKL
  - **MKL** is precompiled with AVX-512; **FFTW** compiled with gcc and different vector extensions
- Test **single fwd+inverse FFT** on wire waveform data:
  - Up to **~3.5x speedup** for FFTW with AVX2 and **AVX-512**; **MKL ~10% faster** than FFTW
- Also starting to look into FFT implementations for GPU





Tests on SKL

Fermilab

Conclusions

- Project actively working to optimize LArTPC reconstruction algorithms for parallel architectures

- Completed optimization of hit finding algorithm:
  - large speedups from new minimization algorithm, vectorization, multi threading
  - code ported back in LArSoft, experiments ready to make use of it
- Work started to optimize signal processing: exploring FFT libraries

- Optimized algorithms open the possibility for efficient usage of supercomputers for LArTPC reconstruction… stay tuned!

UNIVERSITY OF
OREGON