# HEP Event Reconstruction with Cutting Edge Computing Architectures

Giuseppe Cerati, Boyana Norris

SciDAC-4 PI meeting

July 24, 2018

# HEP Event Reconstruction with Cutting Edge Computing Architectures

- Pilot project, funded for 3 years. Website: http://computing.fnal.gov/hepreco-scidac4/

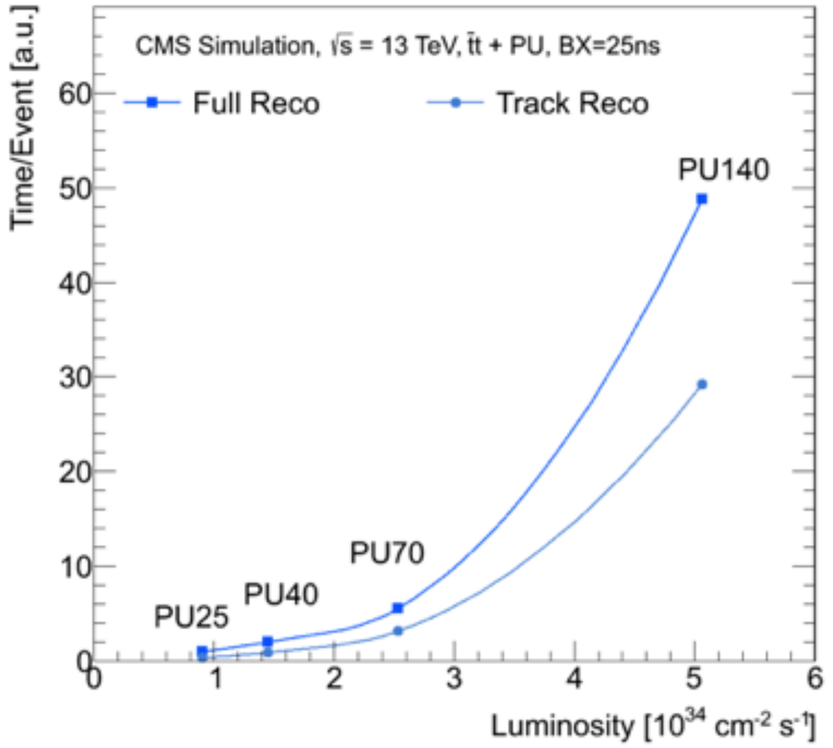
- Fermilab:
  - G. Cerati (PI), M. Wang (staff), A. Hall (postdoc), S. Berkman (postdoc, starting 09/2018)
  - expertise in reconstruction algorithms for HEP experiments
- University of Oregon:
  - B. Norris (PI), B. Gravelle (grad. student)
  - expertise in analysis and optimization for high performance computing

- HEP Areas of interest:
  - CMS tracking:
    - expand UCSD/Princeton/Cornell “Parallel Kalman Filter” collaboration
    - work on optimization (physics+computing) and deployment of the new algorithm
  - LArTPC reconstruction:
    - brand new effort, build on top of knowledge acquired through the CMS tracking effort

# Introduction

- P5 panel recommendation: Higgs as a tool for discovery, physics of $\nu$ mass
  - HL-LHC, LArTPC program (SBN, DUNE)
- Reconstruction for both LHC and LArTPC experiments is challenging
- Future experiments even more challenging:
  - larger sizes or more granular: more detector channels
  - higher beam intensities: more data to process
- Reconstruction compute time does not scale well

- Need large speedups in reconstruction to reach design detector sensitivity and enable discoveries!






Fermilab

# The current model

gigantic multi-author sequential C++ code:
worked great for past discoveries
but inadequate for future challenges

# A new model?


exploit different architectures and parallel computing to conquer new territories

# Project goals

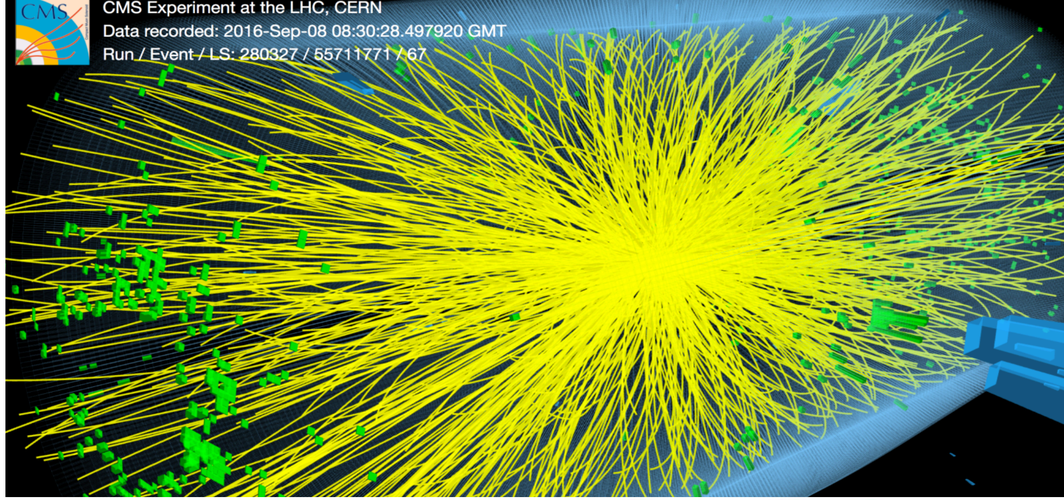
- Accelerate HEP event reconstruction using modern parallel architectures.

- Focus on two areas:
  - novel parallel algorithm for charged particle tracking in CMS
  - pioneer similar techniques for reconstruction in LArTPC detectors

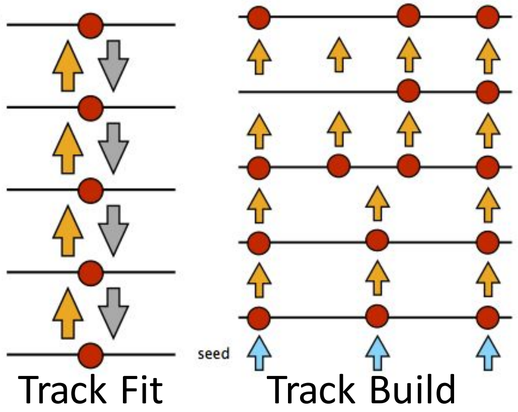
- Goals of the project are the following:
  - Identify key algorithms for physics outcome which are dominant contributions for the experiments' reconstruction workflows
  - Characterize and re-design the algorithm to make efficient usage of parallelism, both at data- and instruction-level
  - Deploy the new code in the experiments' framework
  - Explore execution on different architectures and platforms

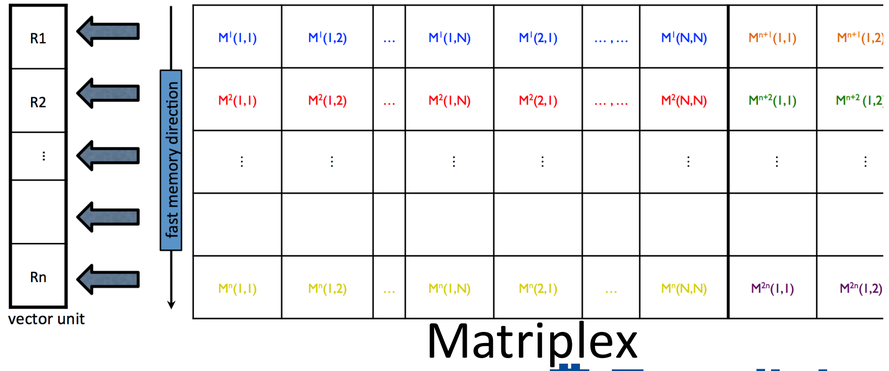
# CMS tracking prototype

- Collaboration with UCSD/Cornell/Princeton
- Reconstruct trajectories of charged particles through thin silicon layers
  - image: data with PU~50, HL-LHC will have PU~200




- Kalman filter-based fit or build
  - Build is combinatorial search for compatible hits along the track: drives the increase in processing time
- Inspired by CMS version but with large differences:
  - Avoid resolving fine grained geometry structures
  - Vectorized using Matriplex: SIMD processing of multiple candidates
  - Thread-level parallelization: distribute seeds in TBB tasks
    - can handle concurrent processing of multiple events



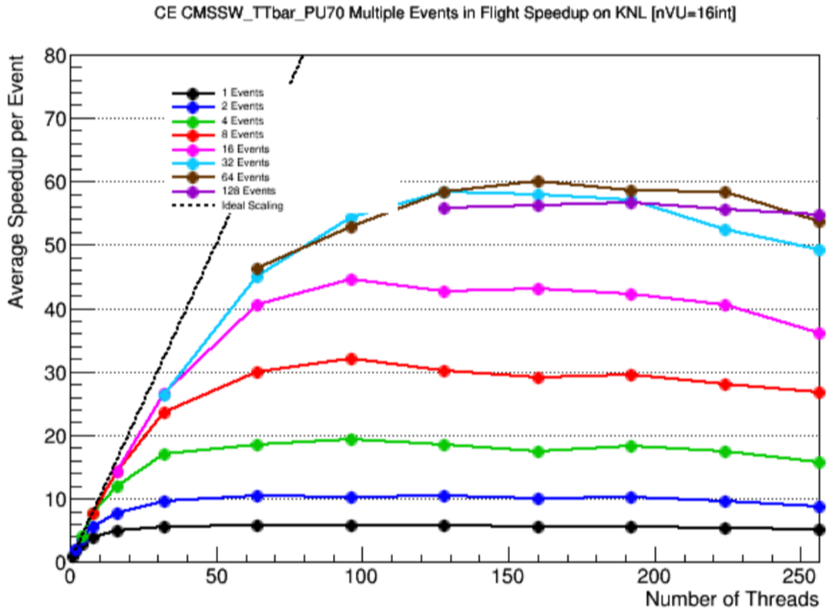

- Challenges: branching points, low arithmetic intensity
  - quick processing of many small objects



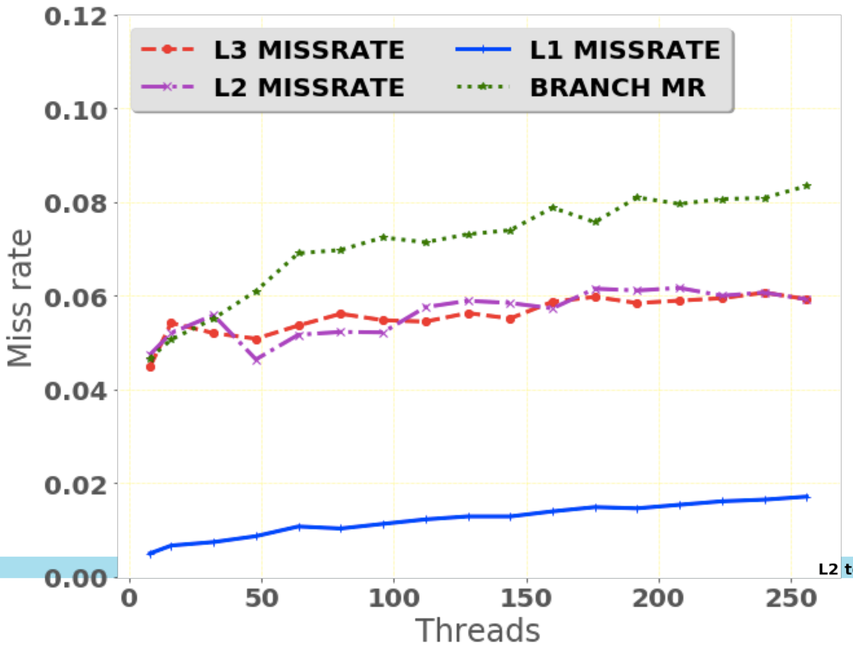
Fermilab

# Profiling with TAU Commander

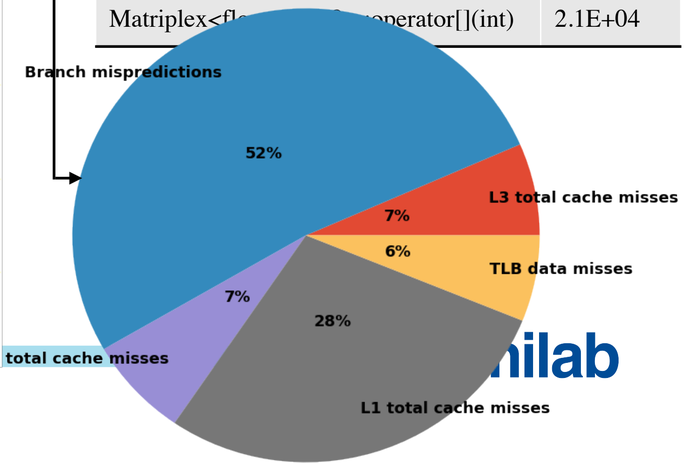
- Code profiled with TAU Commander (sampling) and analyzed with Pandas
  - @NERSC Cori, configuration: 1 NUMA domain, MCDRAM in direct mapped cache mode
  - our feedback lead to improvements to TAU Commander (e.g. TBB support)
- Speedup after first analysis: ~8%
  - mostly from improved caching in propagation
- Deeper analysis ongoing
  - Overall scaling
  - Memory efficiency and scaling
  - Correlation between metrics
  - Per-function analysis with stall breakdown
- Indications for possible improvements:
  - improve vectorization (preparatory functions), L3 misses, branch mispredictions




| Correlations between metrics (Spearman) for 256 threads | Stall Cycles | Total Cycles |
|---|---|---|
| Scalar vector ops | 86.01% | 82.78% |
| Conditional branches | 81.92% | 77.70% |
| L3 misses | 81.20% | 77.25% |
| Branch instructions | 80.67% | 77.00% |
| Branch mispredictions | 79.81% | 74.99% |
| Load/store instructions | 79.68% | 74.60% |
| L3 accesses | 78.61% | 73.99% |
| L2 total cache misses | 77.84% | 73.86% |
| L2 accesses | 77.55% | 72.79% |
| Memory acces stalls | 77.02% | 74.78% |
| L1 total cache misses | 76.43% | 71.66% |
| Unconditional branches | 76.31% | 71.99% |
| Vector operations | 74.15% | 68.52% |
| TLB data misses | 73.86% | 68.70% |




| Least scalable functions: Increase in total cycles (from 8 to 256 threads) | |
|---|---|
| MkBuilder::map_seed_hits() | 2.6E+06 |
| Hit::Hit() | 1.6E+05 |
| Matriplex<fl operator[](int) | 2.1E+04 |



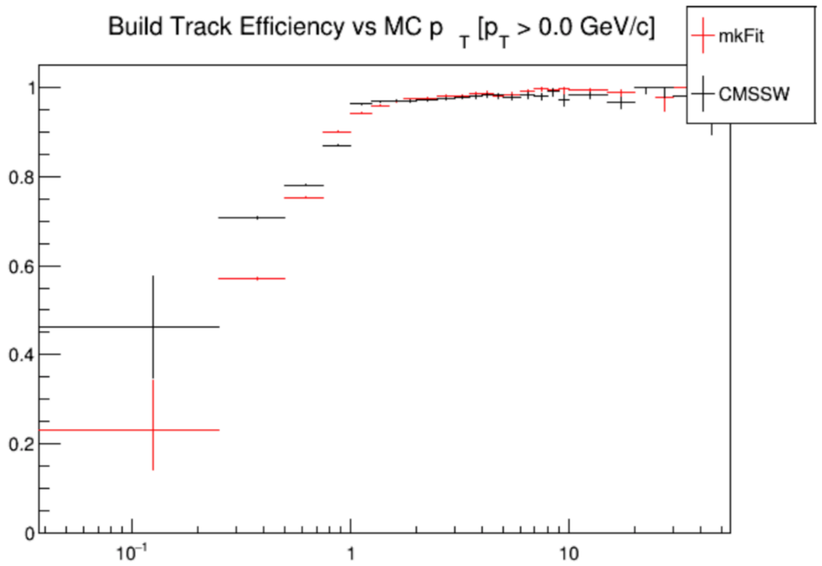
# Comparison with Current CMS Version

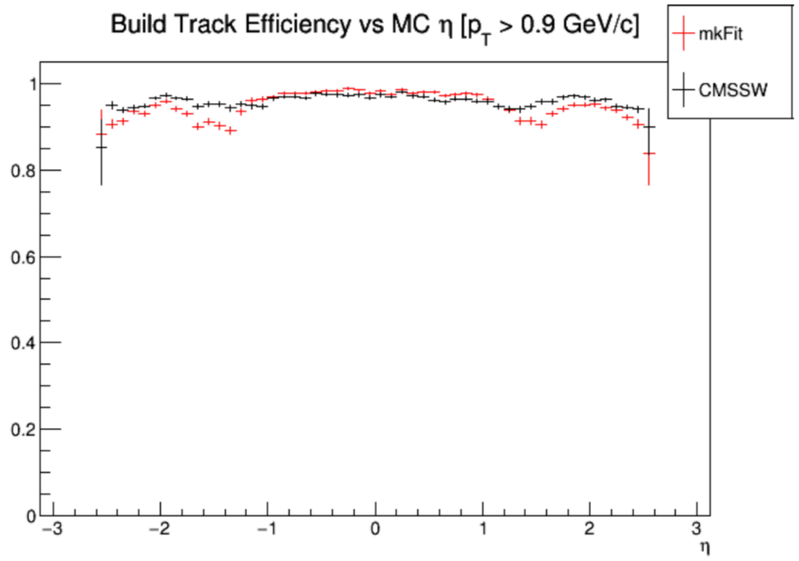
- Comparison of physics and timing performance with CMS version
  - both using standalone tools and CMS validation
  - start from same inputs (hits and seeds from first iteration of offline tracking)

- Overall comparable results
  - same resolutions, similar efficiencies except for:
    - low momentum ($p_T$<0.5 GeV)
    - |η|>1.2 (geometry transition from barrel layers to endcap disks)

- Ongoing investigations target efficiency recovery
  - testing hit rejection handles used in CMS version and not in prototype

- Timing performance of prototype clearly better than CMS version
  - Single threaded time 10-20x faster than CMS on KNL (20x for AVX512)
  - Highest event throughput using Skylake Gold: ~250 events/s
  - Possible use in High Level Trigger for LHC Run3!



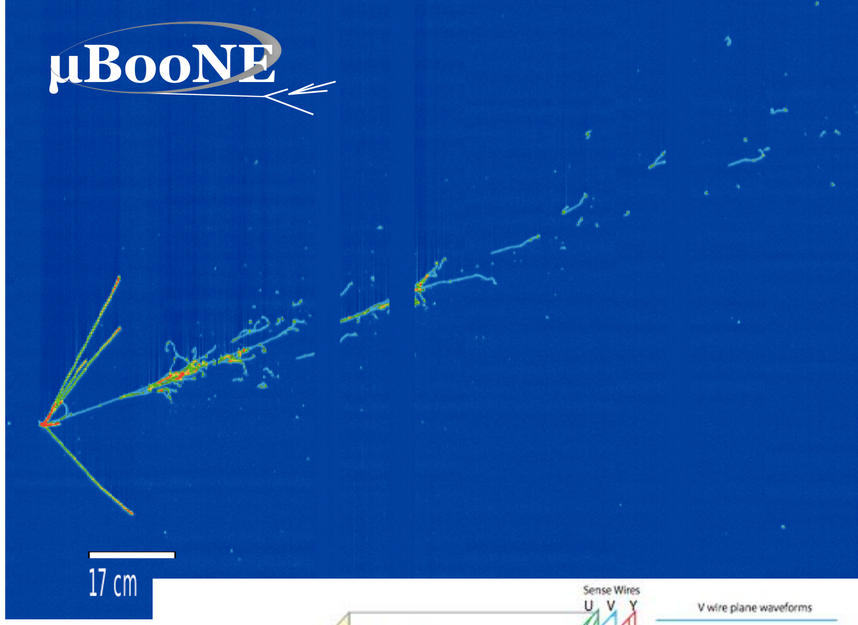





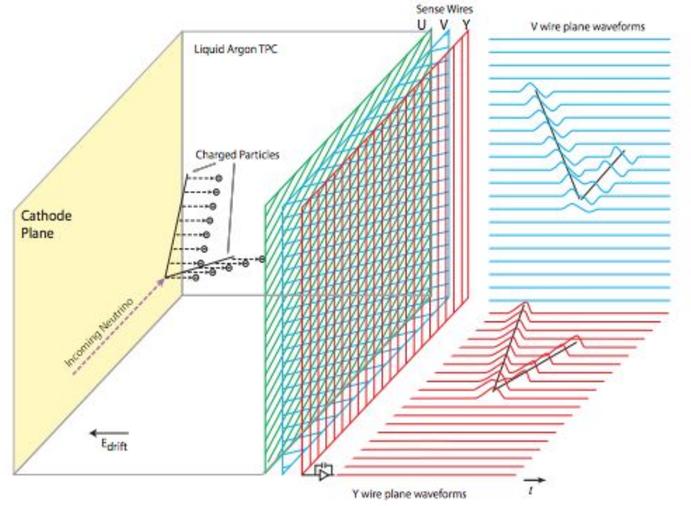

Fermilab

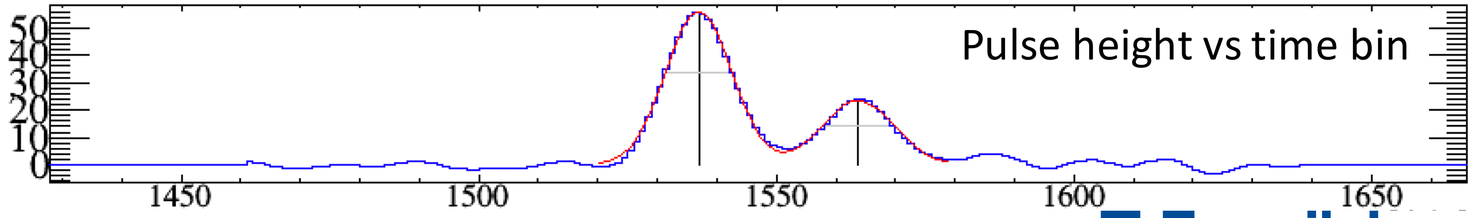
# Hit Finding in LArTPC

- Reconstruction in LArTPC neutrino experiments is challenging due to many possible neutrino topologies, noise, contamination of cosmic rays
  - Takes O(100) s/event in MicroBooNE
  - future experiments  much bigger and on more intense beams




- Feasibility study for LArTPC reconstruction: hit finding
  - MicroBooNE TPC is made of ~8k wires readout at 2 MHz
  - Signal from charged particles produces Gaussian pulses
  - Hit finding is the process of identifying such pulses and determine their properties (peak position and width)
    - currently takes ~15% of the MicroBooNE reconstruction workflow
  - Suitable for parallel processing: wires can be processed independently



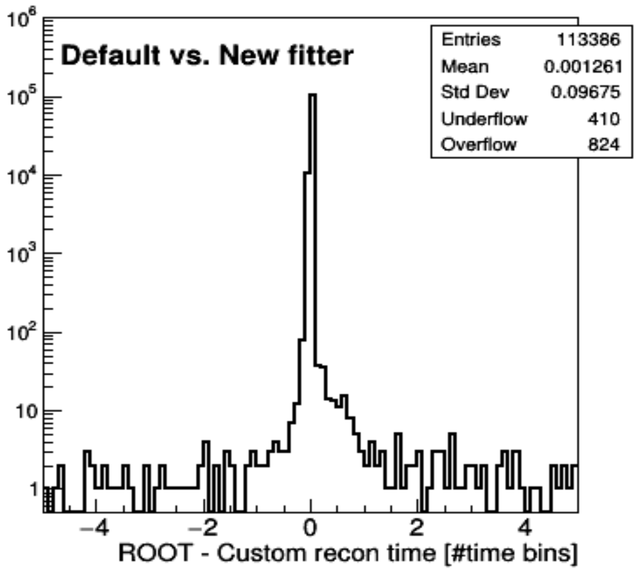



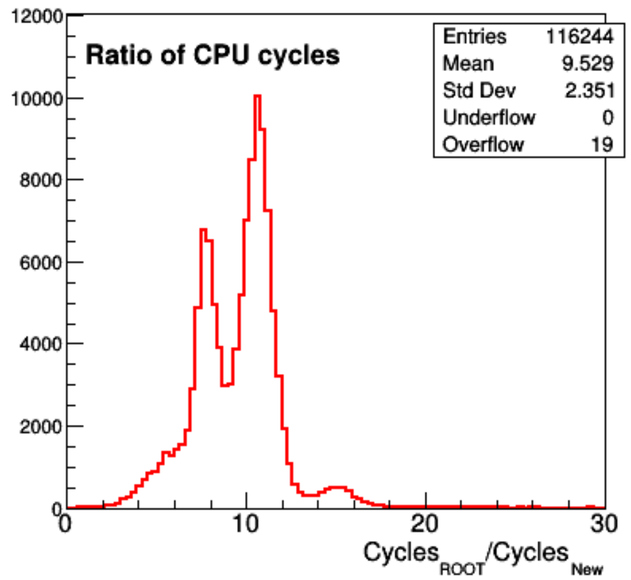



Fermilab

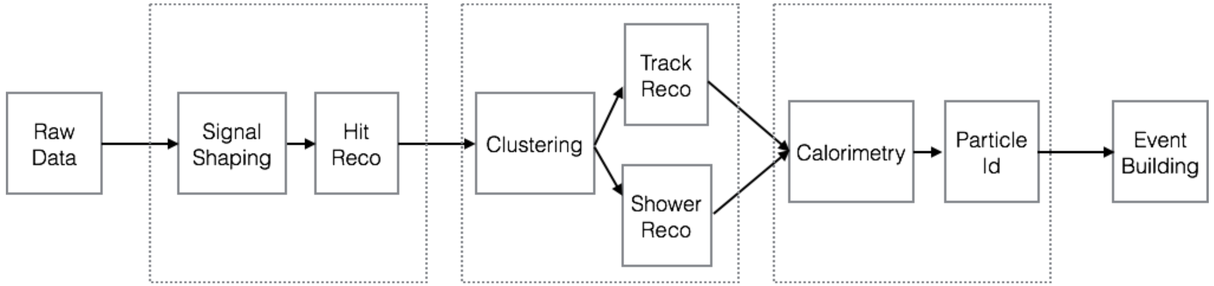
# Hit finding: first results and next steps

- Replicated implementation in standalone code for easier testing and optimization:
  - Replaced Minuit+ROOT Gaussian fit technique with local implementation of Levenberg-Marquardt minimization.
  - Results nearly identical in terms of hit properties but with large speedup (~8x).




- Next steps:
  - Hit finding:
    - parallelize at wire level
    - identify strategy for vectorization
    - port improved version into experiments' codebase
  - Identifying next algorithm:
    - exploring options upstream (signal processing) or downstream (pattern recognition) wrt hit finding




Typical reconstruction chain for LArTPC experiments

# Milestones for year 1

| CMS Tracking | | |
|---|---|---|
| Type | Description | Lead Inst. |
| Year 1 | | |
| M1.1 | Benchmark physics results against the current CMS algorithm; identify and start working on possible improvements | FNAL |
| M1.2 | Profile computing performance on KNL; identify bottlenecks and start working on solutions | UO |

| LArTPC Reconstruction | | |
|---|---|---|
| Type | Description | Lead Inst. |
| Year 1 | | |
| M1.1 | Setup working area; identify and port useful tools, data structures, approaches from CMS Tracking work. | FNAL, UO |
| M1.2 | Identify simple algorithm for feasibility study, produce standalone implementation | FNAL |
| M1.3 | Test different strategies for improved scalability of simple algorithm on parallel architectures | UO |

- Year 1 milestones completed or on track to be completed!

# Synergies

- Cornell/Princeton/UCSD collaboration: original authors of parallel tracking prototype for CMS. Our project is working in close contact with this collaboration.
  - http://trackreco.github.io/
- ASCR institutes: RAPIDS (Platform Readiness)
- HEP Experiments: LHC (CMS, Atlas), Neutrino experiments (DUNE, SBN)
- LArSoft, shared base of physics software across LArTPC experiments:
  - http://larsoft.org/
- Other HEP/ASCR DOE projects: Hep.TrkX (tracking with ML)
  - https://heptrkx.github.io/

# Thank you!


U.S. DEPARTMENT OF
ENERGY
Office of
Science


Fermilab


UNIVERSITY OF
OREGON