Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

SciDAC Projects

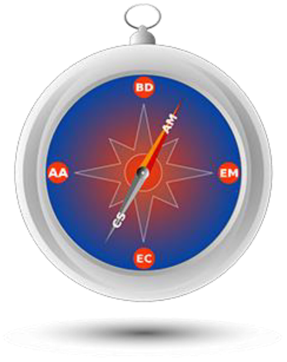
Marc Paterno
SCD Projects Meeting
2 July 2020

Scientific Discovery Through Advanced Computing (SciDAC)

- Overall program: https://www.scidac.gov/
  - DOE Program managers are Lali Chatterjee and Randall Laviolette
  - Combines ASCR and domain science programs
- Five HEP projects
  - ComPASS4: Accelerator Science and Simulation (Jim Amundson)
    - Conventional Beam Dynamics, Plasma-based acceleration
  - Accelerating HEP Science - Inference and Machine Learning at Extreme Scales (Salman Habib)
    - Focus areas: Cosmology, Stats/ML at Scale, Accuracy
    - https://press3.mcs.anl.gov/cpac/projects/scidac
  - HEP Data Analytics on HPC (Jim Kowalkowski)
    - Accelerate HEP analysis on HPC platforms with help from ASCR FASTMath and RAPIDS
    - https://computing.fnal.gov/hep-on-hpc/
  - HEP Event Reconstruction with Cutting Edge Computing Architectures (Giuseppe Cerati)
    - Accelerate HEP event reconstruction using modern parallel architectures
  - Event Generation on HPC (Stefan Hoeche) **(this was on your agenda already)**
    - Short-distance cross section calculations on HPC

ComPASS-4 Project

Community Project for Accelerator Science and Simulation 4

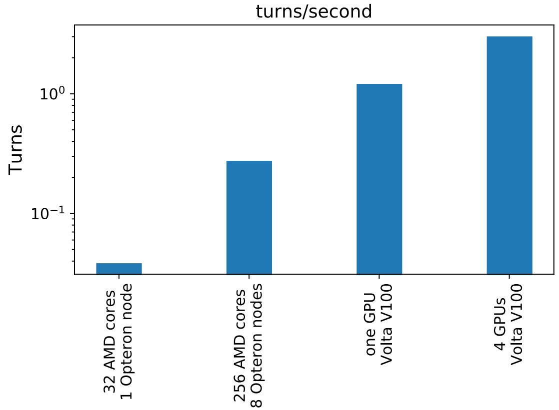


- Multi-institutional SciDAC-4 project combining the best algorithms in math and computer science to efficiently simulate and optimize detailed conventional and plasma based accelerator systems using HPC resources.
- Project is led from Fermilab
- PI: James Amundson (FNAL), deputy-PI: Eric Stern (FNAL)
- Co-PIs: Stefan Wild (ANL), Warren Mori (UCLA), Esmond Ng (LBNL), Ann Almgren (LBNL)

Progress with Synergia

Synergia is our accelerator modeling framework optimized for HPC resources

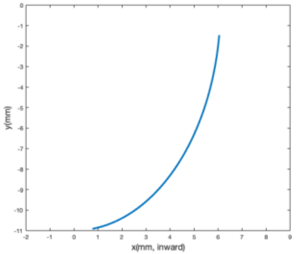
- In the process of upgrading Synergia to use Kokkos for particle data storage and parallel computing primitives for transparent operation on both CPUs and GPUs.
- Using GPUs provides substantial computing performance boost and is essential to utilize the next generation of HPC facilities coming online at NERSC and ALCF.
- Completed all the beamline elements necessary for a Booster simulation.
- The 2.5D space charge solver converted and validated for Kokkos.
- Identified and fixed Kokkos memory handling bugs which were reported to the Kokkos developers.



The performance of a model accelerator simulation run with the identical Synergia codebase on similar cost CPU or GPU hardware that might be available to university researchers. The performance of the code running on four GPUs is about 500 times faster than running on CPUs.

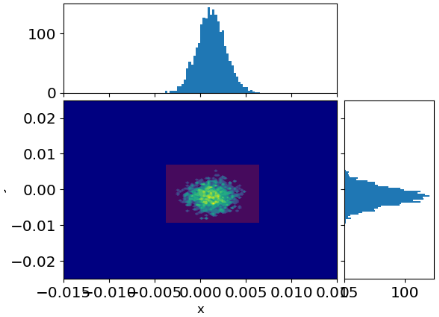
Synergia simulations of PIP-II Injection Painting

To minimize space charge during injection, the beam is injected in linac bunchlets over the course of 285 turns to achieve a more uniform charge distribution.

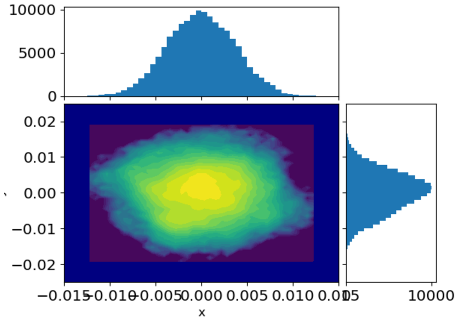


X-Y particle density during injection

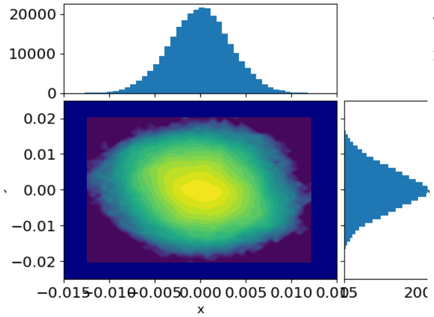
Turn 1



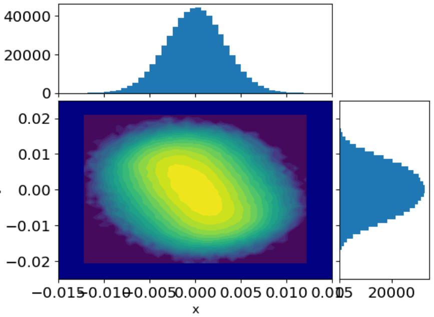
Turn 71



Turn 142



Turn 285



HEP Event Reconstruction *with Cutting Edge Computing Architectures*

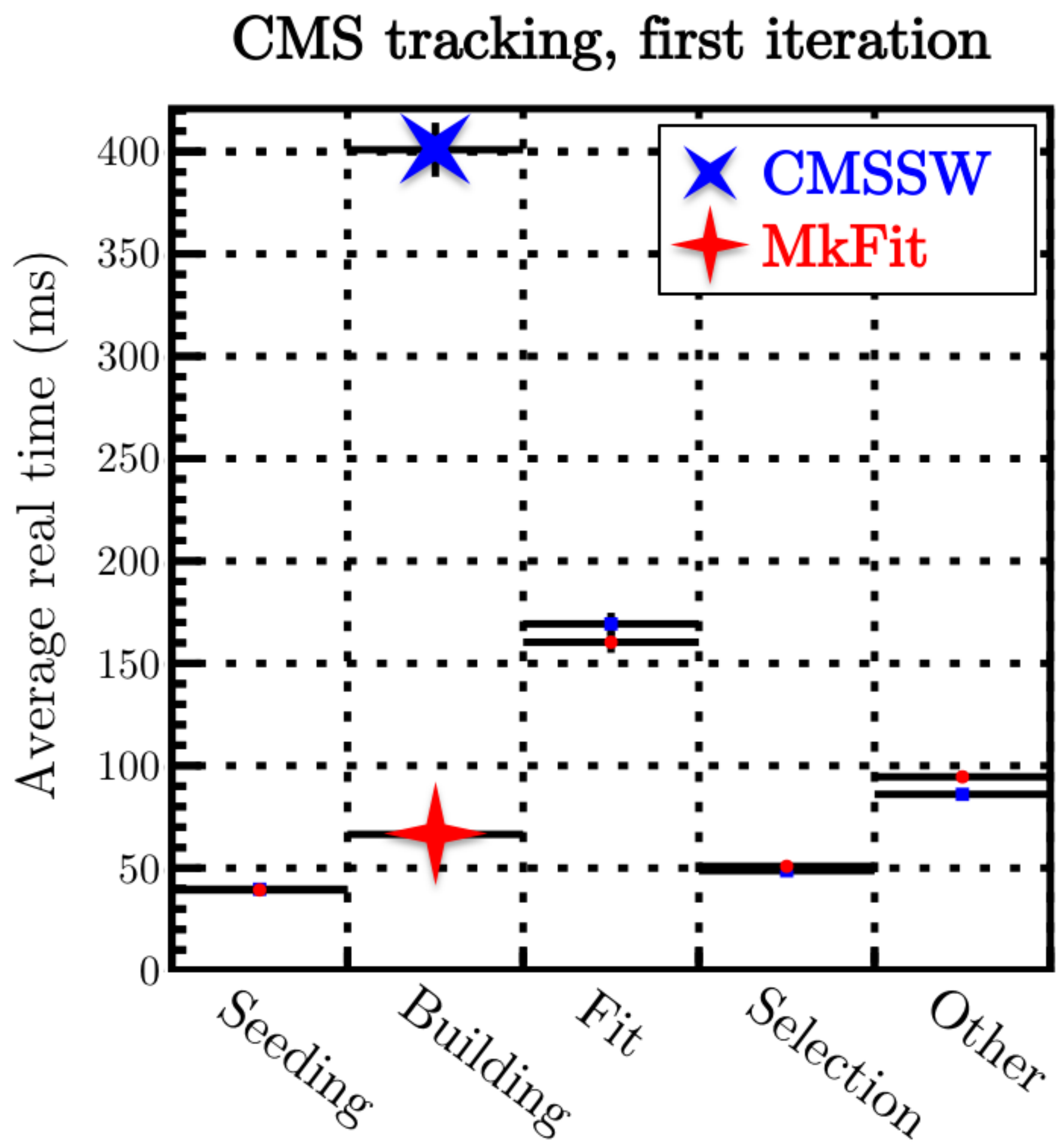
- Main goal of the project is to accelerate HEP event reconstruction using modern parallel architectures. Project initially funded for 3 years, was recently extended for 2 more.
- Team:
  - Sophie Berkman, Giuseppe Cerati, Allison Hall (FNAL)
  - Brian Gravelle, Boyana Norris (UOregon, CS)

- The project operates in the following way:
  - Identify key algorithms for the physics outcomes of each experiment, that also are dominant contributors to their reconstruction workflows' time
  - Characterize and re-design the algorithms 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

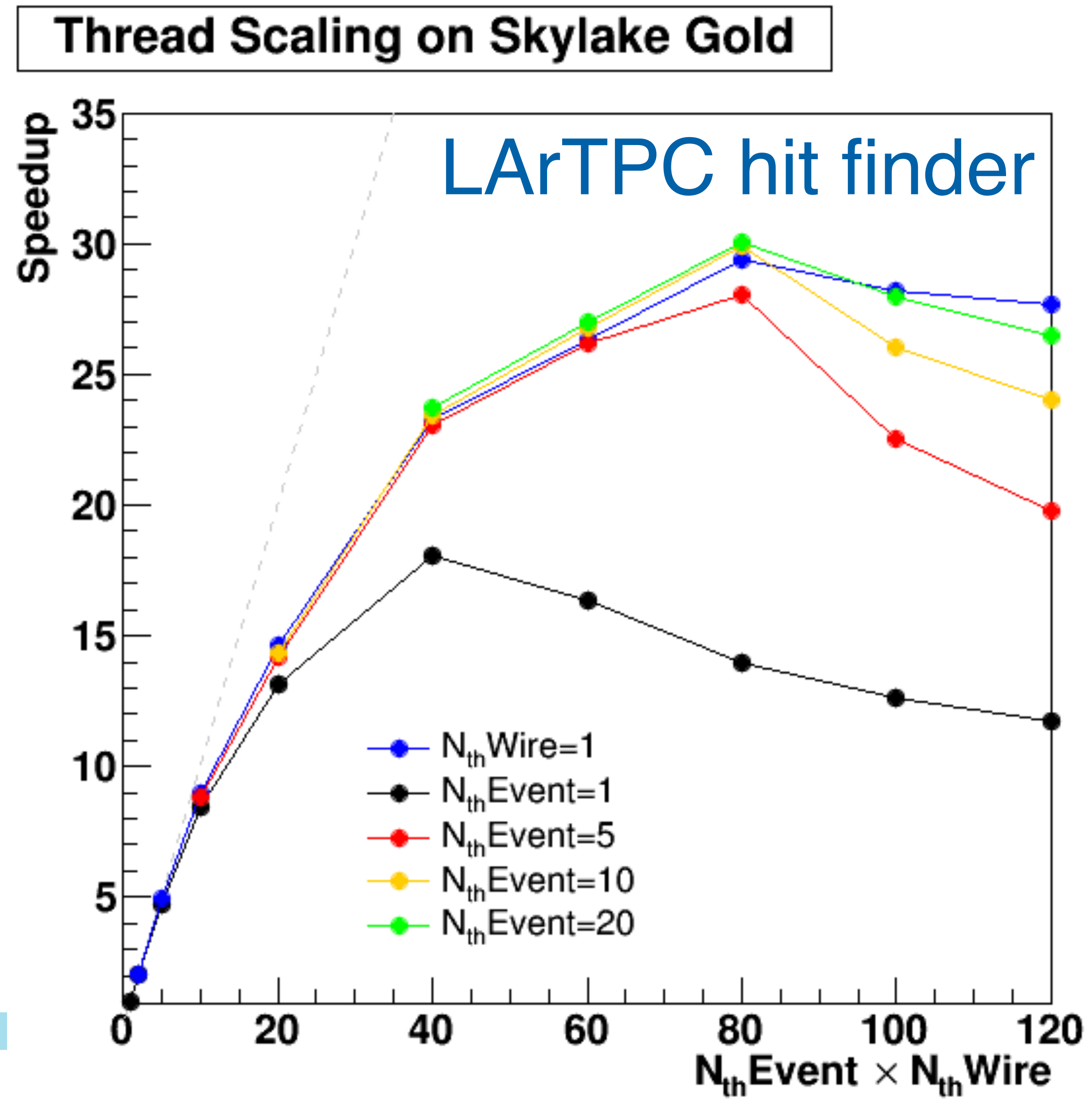
- Focus on two areas:
  - Novel parallel algorithm for charged particle tracking in CMS
    - part of mkFit collaboration (Cornell/FNAL/UCSD/UOregon/Princeton, non-SciDAC funding from NSF IRIS-HEP and USCMS)
  - Pioneer similar techniques for reconstruction in LArTPC detectors

Past Achievements

- CMS Tracking
  - Developed a vectorized, multi-threaded track building algorithm (mkFit) for CMS that achieves significant speedups and comparable physics performance to the nominal algorithm.
  - Vectorized and multi-threaded implementation of the mkFit track building algorithm has been integrated into the CMS software framework.



- LArTPC Reconstruction
  - Vectorized and multi-threaded implementation of hit finder algorithm for LArTPC experiments
  - Fully integrated into experiments' framework; in ICARUS 7x faster (single thread) than previous version.
  - Explored Fast Fourier Transform (FFT) libraries vectorized and multithreaded on both CPU and GPU.



Ongoing work

- CMS Tracking
  - Working towards deployment of mkFit code in CMSSW, initial goal is HLT
  - Developed a test setup (based on code extracted from the tracking algorithm) that is used to explore tools for code portability and targeting efficient execution on GPUs and multicore CPUs
  - Paper just submitted to JINST: https://arxiv.org/abs/2006.00071

- LArTPC Reconstruction
  - Collaborating with HEP-on-HPC SciDAC project to develop an ICARUS reconstruction workflow on Theta at ALCF that leverages the algorithms optimized by our project
    - currently beta-testing Spack installation of LArSoft/icaruscode
  - Planning to extend portability studies to LArTPC code as well
  - Paper in preparation about hit finder

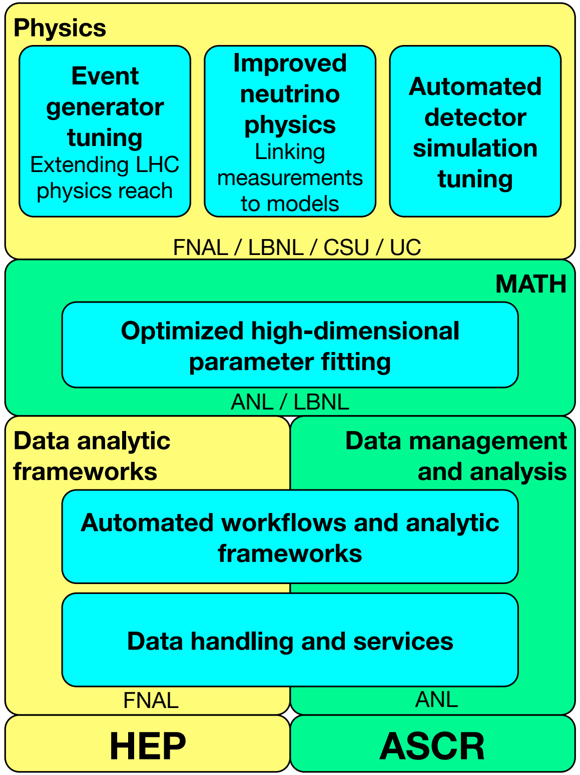
Status of Spack build of ICARUS code

- We are currently testing the spack build of icaruscode on a machine at UOregon; this is a non-FNAL system, so it allows to check which steps in the installation actually rely on FNAL-specific configurations. Progress has been slow in the last few weeks due to the Neutrino2020 conference (first the preparation, then the attendance). Hopefully this will be completed soon.
- The next step will be building on Theta. For this step it would be great if we could coordinate the effort, including members of the spackdev team and also alert people at ALCF so that they are ready to help.

SciDAC HEP Data Analytics on HPC

HEP Data Analytics on HPC: Reminder of Goals

- **Physics component:**
  - Participate in getting new physics results out
  - LHC and neutrino experiments
- **Accelerate targeted HEP analysis on HPC platforms**
  - Working within experiment code and operations
  - In some cases by orders of magnitude
- **Transform how physics tasks are carried out using ASCR tools and techniques**
  - Parallel storage and access for experiment data
  - Data-parallel programming and advance mathematical techniques for HEP analysis



http://computing.fnal.gov/hep-on-hpc/

HEP Data Analytics on HPC: Interactions with HEP Community

SBND

- Feldman-Cousins algorithm rewritten with vectorization & MPI, Increasing performance more than 300x.

NOvA

- Extensive work on the Feldman-Cousins analysis on HPC architectures. Adopted as primary technique for oscillation results.
- Tools for producing HDF5 analysis formats for physics applications as alternative data format for use in HPC facilities and ML tools.
- Developed Pandana framework for highly scalable event filtering

ICARUS

- Parallel production workflow defined for event processing using art & LArSoft frameworks coupled to HEPnOS and DIY usable designed for HPC systems

ATLAS analysis

- Enabled H->μμ analysis using HPC systems and our scalable software technology to generate billions of background events

CMS & ATLAS

- Working to integrate Pythia HDF5 tools into CMS infrastructure

DUNE

- Adopted Feldman-Cousins software based on NOvA work

SciDAC: HPC Framework for Event Generation at Colliders

- Developed new event generator framework for unprecedented accuracy, tying Sherpa and Pythia8 together by means of HDF5 and DIY
- Explored application of Machine Learning for Monte-Carlo simulations with Sherpa.

HEP Event Reconstruction with Cutting Edge Computing Architectures

- Partner project in the design and use of the ICARUS workflow.

Theory and Phenomenology

- Leverage closed form solutions to neutrino oscillation problem from FNAL and BNL theory to accelerate computation
- Assisted with Dark Matter Direct detection phenomenology studies
- Helped conduct reinterpretation of ATLAS data to constrain new physics models
- Enabled early-universe research through technical assistance with complex differential equation solving

FASERnu

- Our expertise in generator tuning helped in the successful proposal of the "Fasernu“ experiment to be built at CERN shortly

HEP Data Analytics on HPC: SciDAC Institute Interactions

RAPIDS

DIY

- Used extensively for data parallelism and wrapping MPI in our C++ applications
- Integrated directly into existing analysis applications for scaling

HEPnOS

- Distributed memory object store designed for use at HPC facilities, using Mochi
- Developed for accessing and storing HEP physics objects with an API that matches access patterns in end-user analysis, production data processing, and simulation applications

HDF5

- Working in collaboration with the Northwestern University group to optimize use of parallel HDF5 in HEP applications.
- Parallel and scalable concatenation procedure from ensemble of grid-generated HDF5 files usable at HPC facilities.

Workflows with DECAF and PYCOMPSS

- Explorations of workflow automation with feedback loop using several MPI-capable physics applications for high-level parameter optimization.
- Includes Pythia event generation, Rivet, and Apprentice

FASTMath

Numerical optimization expertise

- Provided expert advice on use of state-of-the-art optimization techniques
- Introduced the project team to using AMPL solvers
- Introduced constrained optimization techniques

Robust and Bi-level optimization

- Produced software packages and tools together with the HEP project team
- Automated data selection (choosing relevant observables) for numerical minimization
- Automates a labor-intensive, manual procedure carried out by domain scientists

Surrogate optimization

- Created a novel multi-variate pole-free rational approximation algorithm targeted specifically at problem facing the HEP application space.
- Demonstrated superior performance and quality compared to pure polynomial approximation techniques together with the HEP project team

Major subprojects

- PandAna: Using python Pandas with MPI for parallel analysis
- HEPnOS: Distributed memory object store for HPC systems
- HDF5 for analysis and generator event storage
- ATLAS background estimation studies
- SBNfit Feldman-Cousins correction contour calculations upgrade
- NOvA Feldman-Cousins performance upgrades
- Automatic selection of observables

Summary (mainly management and planning)

- **We passed our midterm review without any problems or issues**
- **As expected, experimented are slow to adopt new tools and methods**
- **Our relationship with the Northwestern CS group continues and has expanded into the Exa.TrkX project**
- **The job req for a position shared with COMPASS still needs to be submitted**
  - Hope is that the process of getting approval will be easier than is was in March through May
- **We will be loosing one of our main developer in September (Holger)**
  - I started talking with Alex Sousa from University of Cincinnati about a replacement. Holger is currently employed through University of Cincinnati as a visiting scholar.
- **We are behind with the Icarus workflow**
  - Completing the demo of HEPnOS with NOvA event selection has taken longer than expected to complete. There have been lots of technical hurdles to overcome.
  - Waiting for the spack build of LArSoft to be completed as a major step
- **There are several software products that need to be reviewed, released and posted on the DOE code site.**

SBNFit Improvements

- Replaced contour calculation fixed-grid algorithm with continuous function generated using MFA (Multivariate Functional Approximation).
  - See https://www.mcs.anl.gov/~tpeterka/papers/2018/peterka-ldav18-paper.pdf
- Comparison of grid and MFA techniques is underway
- This enables fast minimizers to intelligently sample the parameter space.
- Recall that it is not feasible to include more dimensions into the SBNfit framework using the grid appoach
- Overall speedup of this application from algorithm improvement exceeds 300x

SBNFit Improvements (detailed comparison from Maya)

- **MFA: Multivariate Functional Approximation (Tom Peterka, ANL)**

Discrete/grid-based

- Input:
- Root files containing precalculated frequency spectra taken at different mass-splittings points
- Each spectra:
  - 114 bins of combined SBND, uBooNE and ICARUS nue and numu spectra
- load the precalculated spectra and assemble into an actual oscillate spectrum
  - Doesn't allow for fluidity. The FC calculation is limited to the mass-splittings points.
- compute all oscillated spectra from precomputed frequencies

MFA

- Input:
- Transfer the precalculated frequency spectra to hdf5 file.
- The output file, approx_sin.out in the example, contains the output MFA data model.
- Points in bins at a desired mass value are evaluated. (does not need to match the number of input points used to generate the model).
- Mass values that are outside the minimum or maximum mass used to compute the model will be clipped to the minimum or maximum, respectively.
- compute all oscillated spectra from precomputed frequencies

NOvA F.C. performance improvements

- Replaced ROOT with data structures and functions from Eigen
- Using DIY (MPI) version with OpenMP, capable of scaling to HPC facilities
- Could not validate algorithm changes before the big physics run for Neutrino 2020
- Planning validation run now that repeats part of the physics results
- Recall that the Feldman-Cousins analysis requires about 40M CPU-hours at NERSC. We are expecting a 2.5-8x speed improvement with these changes

Configurable data analysis workflow for the ATLAS experiment

Scientific Achievement

Novel workflow designed to efficiently use HPC resources enables physicists to generate, simulate and analyze billions of Monte Carlo events in a time scale of hours.

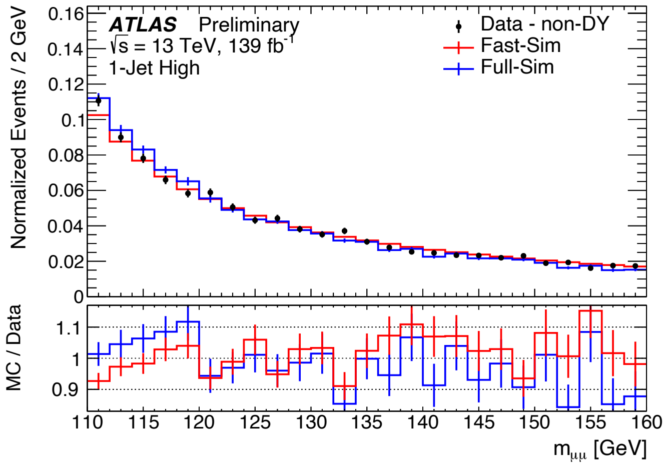
Significance and Impact

Boosted the discovery sensitivity for the search of the Higgs boson decaying to a pair of muons using the ATLAS detector, applicable to other data analysis at LHC too

Research Details

- The main challenge is the precise estimation of the background modeling uncertainty. Controlling the uncertainty requires massive Monte Carlo simulated events
- Traditional text-based storage of event files cannot be efficiently parallelized. We first introduced the HDF5-based storage
- Simulating the ATLAS detector takes very long time, making it impossible to generate billions of events; we introduced a fast detector simulation package enabled the task
- Data analysis is performed by using MPI-enabled parallelization

ATLAS-CONF-2019-028



Normalized dimuon invariant mass spectra, comparing data to ATLAS full simulation and fast simulation. Previously the fast simulation was generated by PowHeg.

Our fast simulation, using Sherpa generator, renders better agreements and will be used in the peer-reviewed publication. The better agreement improves the discovery sensitivity by 3-5%.

Working to incorporate PandAna

Automated Parameter Tuning of HEP Event Generators

Scientific Achievement

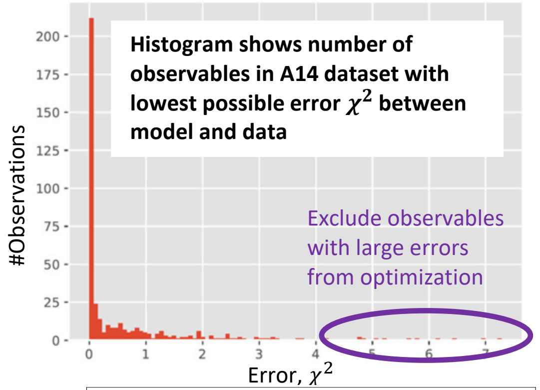
Developed mathematical tools for the automatic selection of observables used in event generator parameter estimation, optimization, and new comparison metrics.

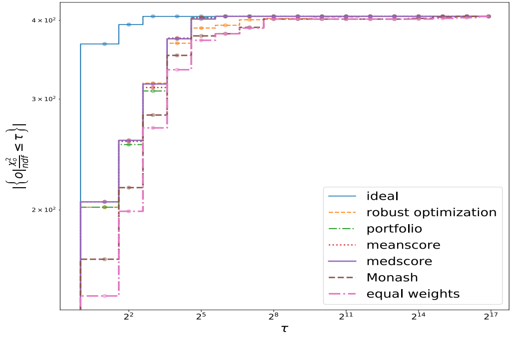
Significance and Impact

Automatic parameter estimation that accounts for model deficiencies, therefore greatly reducing time and resources needed for tuning by taking the human out of the loop.

Research Details

- Event generators must be tuned to fit observations from HEP experiments, but not all observations can be fit well
- Current methods for excluding data from tunes is replaced by two-step algorithm
- Step 1: Two filtering methods as a preprocessing step to down-select ‘’explainable” observations
- Step 2: Analyzed and compared different optimization formulations for solving the tuning problem





Comparison of best case (“ideal”), state-of-the-art (“Monash”) and optimized (others) tunes; Goal is to be as close as possible to Ideal

Objective Performance Metrics for Event Generator Tuning

Scientific Achievement

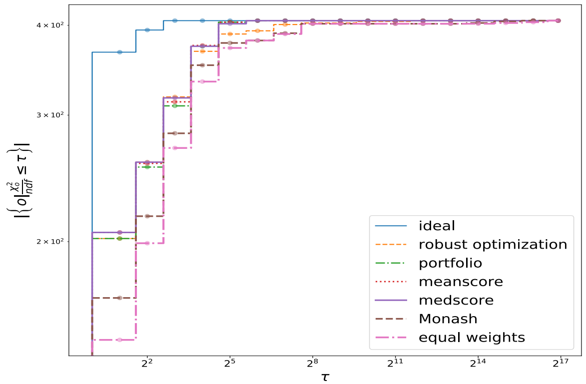
Developed objective metric for comparing the quality of different HEP event generator tunes that is based on multi-objective optimization theory.

Significance and Impact

Objective metric and automated optimizations replace subjective expert opinion and achieve improvements over state-of-the-art tunes.

Research Details

- Rigorous definition of ideal and worst-case tune using multi-objective optimization.
- Graphical representation based on performance profiles
- Definition of objective metrics based on area enclosed between (unattainable) ideal tune and optimized tune



Comparison of best-case ("ideal"), state-of-the-art ("Monash"), equal weight, and optimized (others) tunes in terms of multi-objective metric.

Model	Objective Metric Area
Medscore	4.2200e+02
Meanscore	4.2400e+02
Portfolio	4.3600e+02
Robust Optimization	4.7000e+02
Monash	6.5100e+02
Equal Weights	7.3300e+02

Objective metric that allows us to compare the quality of different tunes. Cells with lighter color are closer to the ideal case.

Bilevel Optimization for Tuning the Parameters of HEP Event Generators

Scientific Achievement

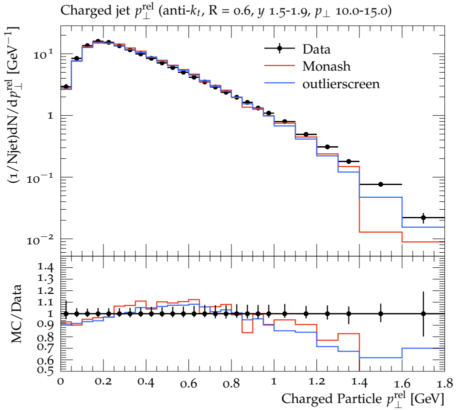
Developed bilevel optimization formulation to automatically determine the importance of observables in optimization and optimize the tuning parameters.

Significance and Impact

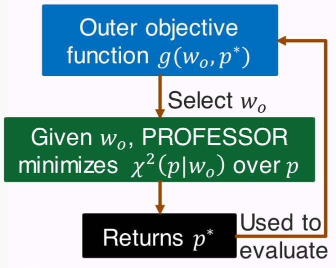
Bilevel optimization takes the expert out of the optimization loop, thus reducing the amount of human hours spent on hand-tuning the parameters while finding better solutions.

Research Details

- Assigning weights to observables before parameter optimization requires high degree of expert knowledge and experience; human is slowest part in this optimization loop and process is prone to mistakes/disagreements
- Upper level: optimize weights for each observable using derivative-free optimization
- Lower level: PROFESSOR software solves tuning problem for given weights



Example of an optimized observable; Monash = fit with current tune; outlierscreen = fit with bilevel approach after filtering



Optimize over observables' weights

Optimize over physics parameters

Flowchart of bilevel optimization

Event Generators: Parameter Tuning and χ^2 Optimization

- Studying use of Gaussian process models to include MC variance.

Fitting parameters, $\mathbf{p}$, in MC simulation to experimental data, $\bar{c}_i$

$$\min_{\mathbf{p}} \ \chi^2 := \sum_{i=1}^{\mathrm{N_b}} \frac{(m_i(\mathbf{p}) - \bar{c}_i)^2}{\sigma_i^2}$$

where

- $\bar{c}_i, \sigma_i$ are experimental mean/variance for bin i
- $m_i(\mathbf{p}) \in \mathcal{M}$ space of polynomial or rational approximation to fit MC-mean values

$$m_i(\cdot) \leftarrow \underset{m(\cdot)\in\mathcal{M}}{\operatorname{argmin}} \sum_{k=1}^{\mathrm{N_e}} (m(\mathbf{p}_k) - \mathrm{MC}_i(\mathbf{p}_k))^2$$

for parameter/MC-generator data pairs, $(\mathbf{p}_k, \mathrm{MC}_i(\mathbf{p}_k)),\ k = 1, \ldots, \mathrm{N_e}$

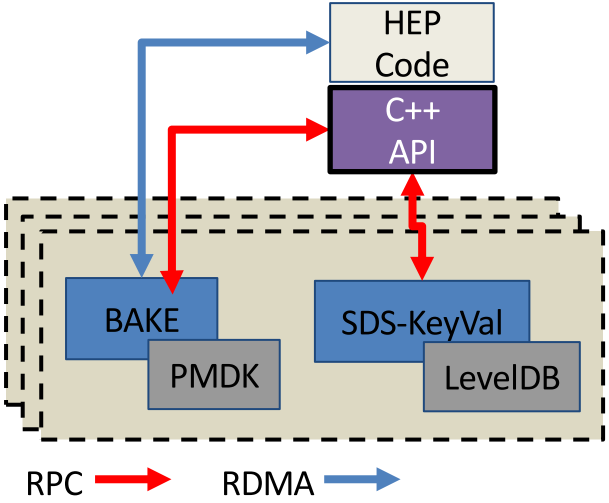
HEPnOS: Fast Event-Store for High-Energy Physics (HEP)

Goals

- Manage physics event data from simulation and experiment through multiple phases of analysis
- Accelerate access by retaining data in the system throughout analysis process
- Reuse components from Mochi ASCR R&D project

Properties

- Read in data at start of run and write results to persistent storage at the end of a campaign
- Hierarchical namespace matching physics concepts (datasets, runs, subruns)
- C++ API (serialization of C++ objects)
- Write-once, read-many



The HEPnOS event store is built using Mochi, a framework for developing specialized data services for use in HPC. Mochi allows use of state of the art libraries (e.g., LevelDB for key/value storage) while providing convenient APIs to scientists.

PandAna: An environment for scalable high-level HEP analysis on HPC

Achievements

Demonstration of scalable parallelization of an analysis code from NOvA by replacing serial IO mechanism with parallel IO.

Significance and Impact

Allows existing analysis code developed by experimenters to be deployed at HPC sites for processing of large datasets.

Research Details

- Provide an easy-to-use environment for fast and scalable HEP high-level data analysis
  - Users can develop on laptops or local clusters and deploy code to HPC
- **Use HDF5** for fast parallel reading of large amounts of data
- **Use Python** and popular Python data science tools (numpy, pandas)
- **Introducing to HEP the "tidy data" analysis model**, using large data matrices and distributed data parallelism
  - Use MPI for distributed parallelism
  - The parallelism in user code is implicit