32nd International Symposium on Lepton Photon Interactions at High Energies (2025, University of Wisconsin)

Towards rapid and efficient analyses at scale

Philip Chang[2], Lindsey Gray[1], Kelci Mohrman[2], Nick Smith[1], Ben Tovar[3]

1 Fermilab, 2 University of Florida, 3 University of Notre Dame

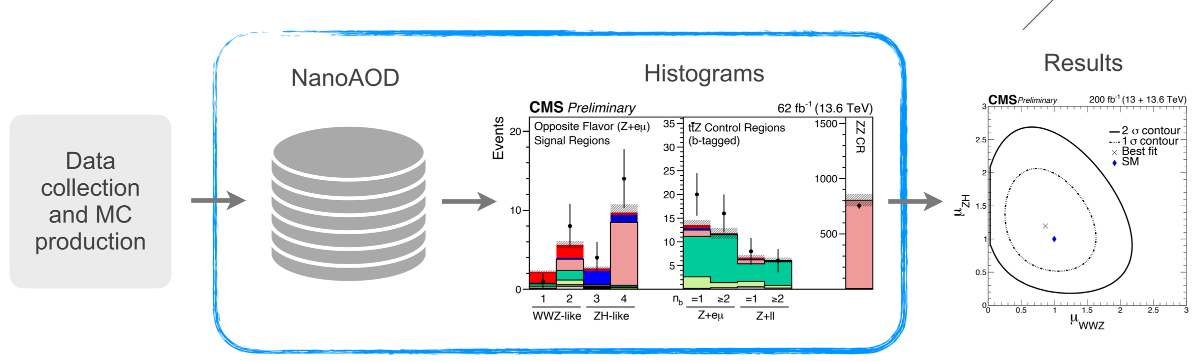
Kelci Mohrman
(k.mohrman@ufl.edu)

Introduction: Aiming for fast turnarounds with columnar approach

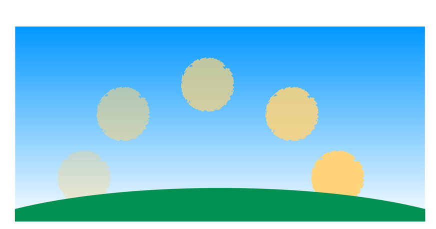
The challenge of efficiently analyzing our mounting data will continue growing with increasing luminosity

A particular challenge: End-user analysis

- Usually run many times iteratively as an analysis is developed
- Innovation proceeds no faster than the wall time
- Important to minimize turnaround

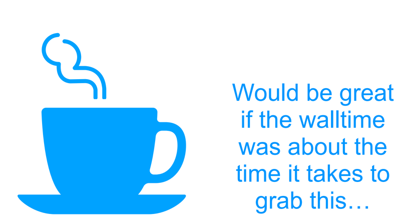


Without a focus on performance, analysis frameworks can often lead to turnarounds of "about a day" →



To work towards a fast turnaround goal, the coffea framework:

- Uses a columnar approach
- Aims to provide an easy to use, scalable, fast solution →



Coffea used by many LHC analyses, e.g.:

Search for new physics impacting t(t)X using SMEFT (CMS):

- Input: 2-lepton skim of R2 samples (~2TB, ~1B events)
- Processing: Full analysis with systematics (output hists ~1GB)
- Performance: Process in ~1.5 hr (~900 CPU cores, `WorkQueue`)

Measurement of WWZ at 13 and 13.6 TeV (CMS):

- Input: 4-lepton skim of R2 + R3 samples (~60GB, ~30M events)
- Process: Full analysis with systematics (output hists ~20MB)
- Performance: Process in ~30m (~80 CPU cores, `futures`)

This project aims to explore two of the computing challenges that lie ahead as we scale up

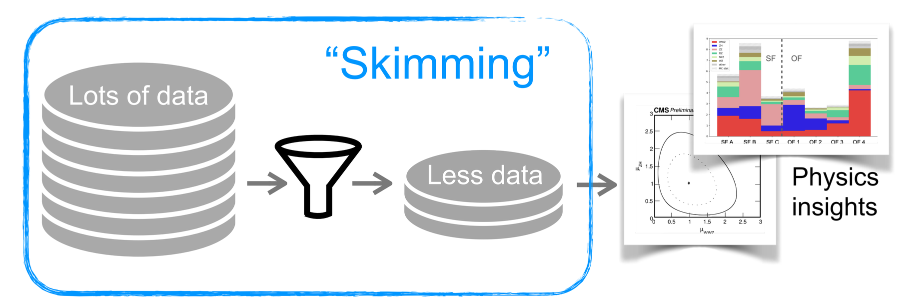
1: Challenges with increasing data volumes (20x with HL-LHC)

2: Challenges with increasing analysis computational complexity

Goals: benchmark current capabilities and explore novel techniques to help CMS chart a course towards rapid and efficient turnarounds at the HL-LHC

1: Addressing increasing data volumes

With 20x the data with HL-LHC, many analysis will focus on increasingly rare/ unique features



Use skimming to quickly identify subset of events with those features

Skimming is most effective if used aggressively

By the same token, must be fast and painless to remake skims

Skimming often O(1 day) for Run 2, not viable at HL-LHC scales

We aim to benchmark skimming performance at scale, understand current capabilities and bottlenecks as we scale

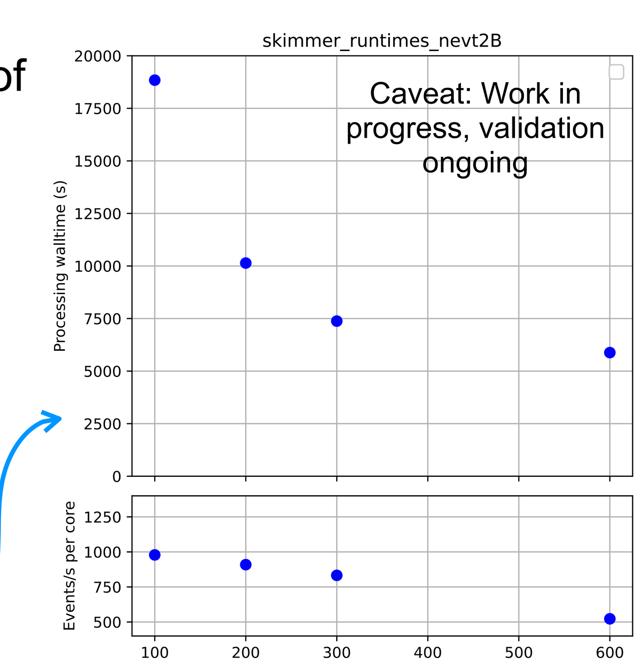
Developing and benchmarking a skimming tool with coffea 202X: Cortado

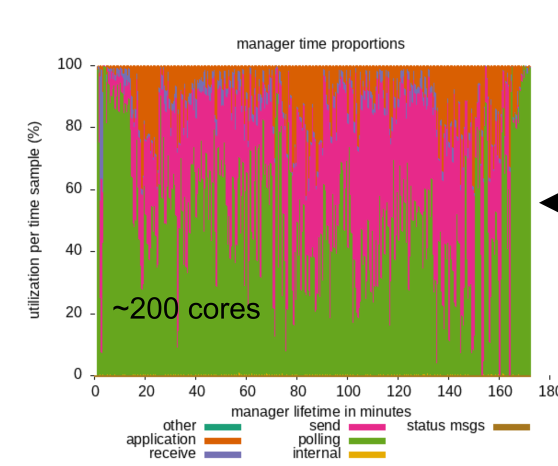
Inputs: Unskimmed data/MC for an analysis of scale ~200 fb^{-1} (i.e., Run 2 + early Run 3)

- Totals: ~11B events, ~13TB, ~20k nanoAOD root files, 418 datasets
- Files located at the University of Florida (where the tests are run from)

Preprocess: ~30m w/ 100 cores (`TaskVine`)

Process (tests with ~1.8B events, `TaskVine`): "~couple billion events in a ~couple hours with a ~few hundred cores"





← TaskVine produces diagnostic plots to better understand the performance, e.g. here see significant portion of the runtime is spent in "waiting" (green), indicating we can increase throughput by using more CPU cores

Directions to develop and to explore

Understand the bottleneck(s) limiting the event/s per core rate

Explore speedup with higher performance storage

Study the effect of different kinds of file access

Study scaling with multiple concurrent instances of the application

2: Addressing increasing analysis complexity

Traditionally, compute step is often not the main bottleneck in HEP analyses (main bottleneck often I/O)

However, as we search for BSM in larger phase space and with more parameters (e.g. global EFT searches), analyses complexity grows

Thus accelerating the compute step with GPUs may also be important for achieving rapid and efficient HL-LHC analyses

Testing columnar operations with GPUs

GPU implementations of many `awkward` functions are available already

Aim to test these to understand the performance and acceleration for realistic analyses, using the IRIS-HEP ADL benchmark queries:

1. Plot the p_T^{miss} for all events
2. Plot the p_T of all jets from all events
3. Plot the p_T of all jets with $|\eta|$<1 from all events
4. Plot the p_T^{miss} for all events with at least 2j with p_T>40 GeV
5. Plot the p_T^{miss} for events with an OS muon pair with an invariant mass between 60 and 120 GeV
6. For events with at least 3j, plot p_T of the trijet system with invariant mass closest to top and plot the maximum b-tagging score among the trijets
7. Plot the sum of the p_T of jets with p_T>30 GeV that are not within 0.4 in ΔR of any lepton with p_T>10 GeV
8. For events with at least 3ℓ and a SFOS pair, find the SFOS pair closest to the Z mass and plot the transverse mass of the p_T^{miss} and leading other ℓ

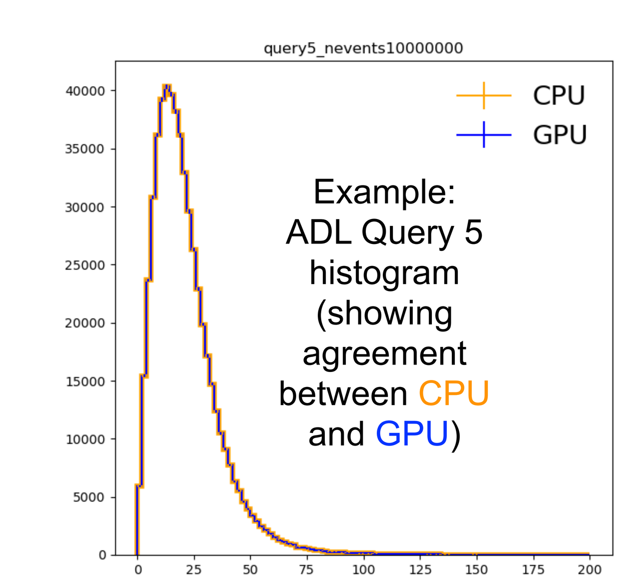
1-5 already implemented and validated on GPUs

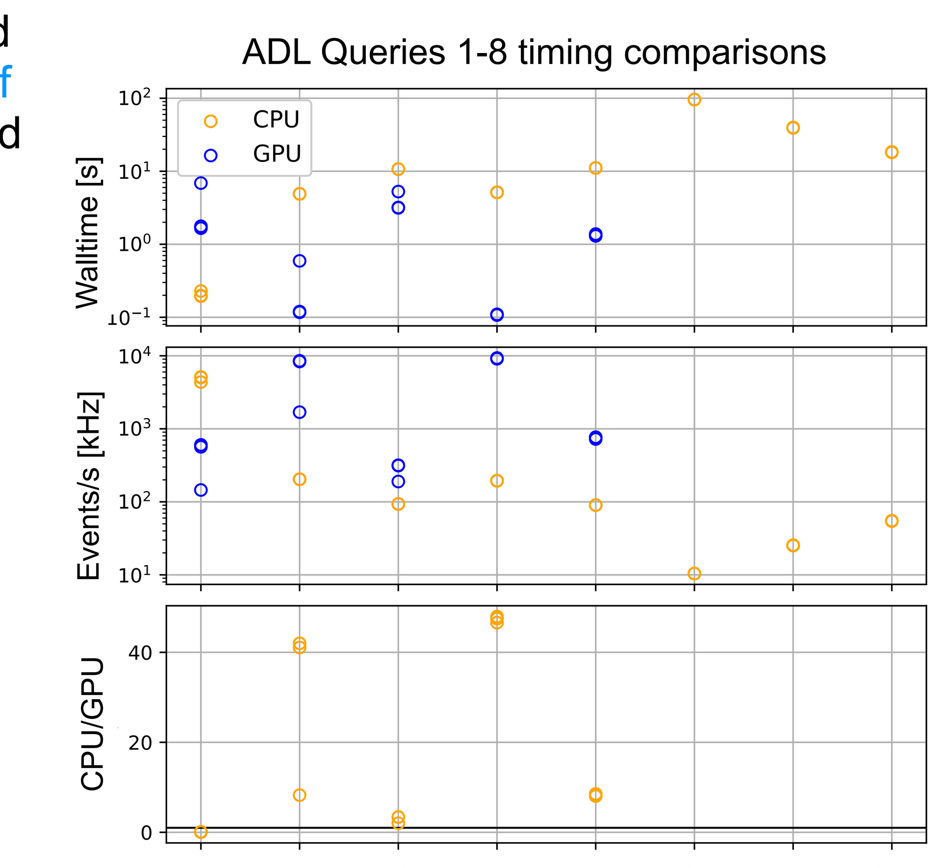
Aim to test if these queries runnable on GPUs, validate the outputs against CPU, and compare performance

Input: ~10M data events from 2012 open data (stored in parquet format)

Hardware: L4 GPU, located at University of Florida Hipergator

Performance: the GPU-based queries show improvements of factors of up to ~40x compared to the CPU -based query





Acknowledgments: Research supported by the USCMS S&C R&D Initiative