Charting a course toward rapid turnaround of HL-LHC scale analyses: Benchmarking current capabilities and exploring the acceleration of columnar processing via heterogeneous architectures

PI: Prof. Philip Chang (University of Florida)
Postdoc: Kelci Mohrman (University of Florida)

1 Project goals

This project aims to benchmark the performance of the step of late-stage data analysis (in which nanoAOD formatted data is transformed into histograms) for realistic CMS analyses in order to understand current capabilities, scaling, and bottlenecks for columnar analysis workflows; acceleration of the columnar processing via GPU offloading will also be explored. The results of these studies will help to illuminate the challenges and opportunities that lie ahead as CMS pushes towards rapid and efficient turnarounds of HL-LHC physics analyses. An ongoing CMS multi-boson analysis will be used as the example application for the proposed explorations. The analysis is fairly representative of a mature CMS analysis studying Run 2 and early Run 3 data, and is implemented in the `coffea` [1] framework. We will aim to benchmark the performance that is able to be achieved under various configurations in order to understand where the bottlenecks lie and how the analysis scales towards skimming and processing larger data volumes. We will also aim to demonstrate the feasibility of running a portion of the analysis on GPUs and to enumerate the developments that would remain in order to run the analysis fully on GPUs.

2 Motivation and background for the project

As CMS ventures towards unprecedented luminosities at the upcoming runs of the HL-LHC, the challenge of efficiently analyzing the rapidly mounting data will become increasingly crucial to the pursuit of new physics at the LHC. One step of the data-analysis workflow that presents an interesting challenge is the selection and histogramming performed on nanoAOD-formatted data. This processing step (sometimes referred to as "late-stage data analysis" [2] or "end user analysis" [3]) is generally run many times iteratively as an analysis is explored, developed, and optimized; since innovation can proceed no faster than the wall time of one iteration of this step, it is important to minimize the turnaround.

The `coffea` framework [1] is designed to help confront this challenge. Based on a columnar approach that leverages the scientific python ecosystem and HEP-specialized packages (e.g., `awkward` [4]), `coffea` aims to provide a set of user-friendly tools that enable rapid turnarounds of the late-stage analysis step without requiring analyzers to spend significant time optimizing their code. There exist examples in which `coffea` has been observed to facilitate rapid turnarounds of Run 2 scale analyses (e.g., [5]), and extrapolations using a `coffea`-based analysis have shown that there are no fundamental limitations preventing HL-LHC scale analyses from being executed in as little as 10 minutes [2]. However, further studies of current bottlenecks and additional development of analysis cyberinfrastructure would be required in order to achieve this potential. It would thus be beneficial to invest further study into characterizing the current capabilities in order to understand how different variables impact the performance and to identify challenges that may be encountered as

we scale towards processing larger volumes of data. We propose to perform such studies in order to help chart a course towards rapid and efficient HL-LHC physics analyses.

In addition to benchmarking current capabilities, it is also important to explore how new approaches could help to improve performance. For example, the use of GPUs to accelerate computationally expensive tasks represents an exciting opportunity to achieve faster turnarounds. For late-stage data analysis, previous studies [6] have demonstrated that several functions commonly used in columnar-based analysis workflows are able to be accelerated substantially when executed with a GPU (up to about 30 times faster than a single CPU thread). Since the time of [6], the developers of the `awkward` package have continued to add GPU implementations for array-based functions [7]. We propose to test these kernels, measure their performance, and enumerate the remaining work that would be required to run a realistic CMS analysis fully on GPUs.

3 Project details

This project would consist of two main parts: benchmarking current performance, and exploring the acceleration of analysis processing with GPUs. The `coffea`-based analysis that will serve as the example application for these studies is detailed in section 3.1. The performance studies are discussed in section 3.2, and the GPU explorations are described in section 3.3; we propose to spend approximately 6 months on each part.

3.1 About the `ewkcoffea` analysis framework

Studying electroweak processes and implemented with the `coffea` framework[1], the `ewkcoffea` analysis [8] will be used as the example application for these studies. This analysis is a standard model search for the WWZ process in four-lepton final states using Run 2 and early Run 3 (2022) data. The analysis is relatively mature, with several corrections and systematics included, and features a variety of elements common to many CMS analyses, e.g.:

- Filtering and sorting arrays
- Computing quantities based on the kinematics of objects and combinations of objects
- Evaluating BDTs per-object and per-event
- Filling and accumulation of histograms

The analysis processes ∼100M skimmed events and ultimately fills approximately 1-50 histograms in approximately 20 channels (depending on the configuration). The input data is approximately 500GB, and the resulting histograms (saved in a gzip pickle file format) can be up to O(1GB), depending on the configuration. There exist CMS analyses that are substantially different from `ewkcoffea`[2] , and the performance considerations for these analyses

[1]The analysis is implemented in both the `coffea` 0.7 paradigm and the so-called "calendar-`coffea` " paradigm (which is more heavily integrated with Dask, and has superseded `coffea` 0.7 as of late 2023).

[2]Some analyses may be computationally lighter (e.g. a significantly simpler event selection, or without any evaluation of ML models) or heavier (e.g. with a computationally expensive neural network evaluation for each event). Some analyses may process significantly more/less data. Other analyses may process data in a different format (i.e. a format different from NanoAOD, though it should be noted that the fraction of CMS analyses that use NanoAOD formatted data is $\sim 50\%$ and growing).

may vary from the results obtained in this proposed study. However, overall, the `ewkcoffea` analysis is fairly representative of a sizable fraction of CMS analyses, so the performance capabilities that this study aims to benchmark will be relatively general.

3.2 Project details: Benchmarking current capabilities

As introduced in section 1, the goal of this part of the project is to understand the current performance capabilities of columnar analyses. Using the University of Florida (UF) T2 site, we will benchmark the `ewkcoffea` analysis (with both `coffea` 0.7 and the current calendar-`coffea` implementation) and explore the scaling. To that end, we propose to study:

- The feasibility and performance of different scale-out mechanisms. We will study the TaskVine [9] executor (which is developed by the Notre Dame CCL group [10] and is designed to provide performant scale-out for large-scale `coffea` analyses [11]) as well as potentially the Dask distributed executor [12].
- The performance of the compute step with varying amounts of resources (i.e. scale up the number of CPU cores and track the performance in order to understand the maximum number of CPU cores that can be efficiently utilized concurrently, and to identify the bottleneck(s) that prevent further scaling).
- The impact on performance of different types of file access (e.g. remote access via XRootD, and direct local access of files stored on the UF T2).
- The effect of different storage systems (e.g. the standard spinning disk storage at the UF T2, vs. small-scale tests of NVMe storage).
- The scaling with multiple concurrent instances of the analysis
- The portability of the analysis (by running at a different site than the UF T2, e.g. the Fermilab LPC).

These performance studies will be based on skimmed data. However, we will also aim to benchmark the feasibility and performance of the usage of `coffea` for the skimming step. Here we use "skimming" to refer to a preliminary event selection process in which a loose requirement is applied to remove events that will not be studied by the analysis. The events that pass the skim requirements are saved to NanoAOD formatted data files. In subsequent iterations of the analysis processing, the skimmed files (rather than the full samples) are processed. Thus, in contrast to the analysis processing step, a skim only needs to be performed once (in principle). However, in practice, the skimming step is usually rerun multiple times for a variety of reasons (e.g. if the analyzers wish to obtain a tighter skim, or if the analysis selection changes significantly requiring a looser skim to be applied). Since skims are run less often than the analysis step, the turnaround time is usually a less pressing concern. However, as data volumes increase, analyzers will likely move towards tighter skims, which increases the likelihood of needing to rerun the skimming step multiple times throughout the lifespan of the analysis. Thus, the skimming step represents an increasingly important consideration in an analysis workflow. The `coffea` framework has the ability to write to Root files. We will aim to test this functionality and understand the performance and scaling of this skimming step (by studying the $\sim$10TB of unskimmed inputs to the

ewkcoffea analysis) in order to build a more complete and realistic picture of the current performance capabilities of full analysis workflows.

3.3 Project details: Exploring GPU acceleration

The offloading of columnar-analysis tasks to GPU resources offers an exciting opportunity to significantly accelerate the processing step. We propose to demonstrate the feasibility of running a portion of ewkcoffea with GPU resources and to enumerate the remaining developments that would be required to execute ewkcoffea entirely on GPUs.

Previous studies have demonstrated that it is possible to execute some columnar functions on GPUs, with speedup (compared to a single CPU core) of up to about a factor of 30 (though the factor depends significantly on the particular function) [6]. The developers of awkward are working to make available GPU implementations for many common functions used in HEP analyses. Using the ewkcoffea analysis as an example application, we thus propose to work in collaboration with the awkward team to test these functions, measure their performance, and identify the remaining work that would need to be completed (on the awkward side and/or analysis code side) in order to run the ewkcoffea analysis fully on GPUs. This information would help to elucidate the feasibility of performing analyses with heterogeneous resources and quantify the potential gains of this approach in order to help inform further research in this area. To that end, we propose to:

- Break down ewkcoffea into a set of one-line (or few-line) functionalities and communicate this list to the awkward team.
- Identify the functions that are already available in awkward, and test these individually with GPU resources in order to verify their functionality.
- Define a method of characterizing GPU unit-performance, and measure the performance of functions using GPU resources and the impact this would have on analysis turnaround time.
- Identify the work that would remain to run ewkcoffea fully on GPUs.

4 Details regarding the postdoc who would perform these studies

Kelci Mohrman graduated from the University of Notre Dame in 2023. During her graduate studies, she worked with the Notre Dame HEP computing group (comprised of CMS physicists, computing professionals, and computer scientists from the Notre Dame CCL team) to contribute to the development of a coffea-based analysis (TOP-22-006 [13]), which achieved relatively performant turnarounds [5] [14] on full Run 2 scale data. Exploration and development performed during this collaboration also resulted in a paper at the 2022 IEEE International Parallel and Distributed Processing Symposium [15]. The experiences in developing, scaling up, and benchmarking a large-scale coffea-based workflow (along with Mohrman's continued collaboration with the CCL team) would be beneficial to the studies proposed here. During Mohrman's first year as a postdoc at UF, she worked on a project to implement a tracking algorithm [16] with the SONIC [17] framework (through the USCMS R&D Initiative) [18], gaining experience with GPU-based workflows. This experience will be useful for the exploration of GPU acceleration of columnar processing.

References

[1] Coffea. https://github.com/CoffeaTeam/coffea.

[2] Kevin Lannon et al. Analysis Cyberinfrastructure: Challenges and Opportunities. In *Snowmass 2021*, 3 2022.

[3] Gavin S. Davies, Peter Onyisi, and Amy Roberts. CompF5: End User Analysis Topical Group Report. 9 2022.

[4] Awkward. https://github.com/scikit-hep/awkward.

[5] USCMS presentation "Towards fast analysis turnaround". https://indico.cern.ch/event/1269436/contributions/5403694/, 2023.

[6] Joosep Pata and Maria Spiropulu. Processing Columnar Collider Data with GPU-Accelerated Kernels. 6 2019.

[7] Awkward documentation "Awkward Arrays on GPUs". https://awkward-array.org/doc/main/user-guide/how-to-math-gpu.html.

[8] ewkcoffea. https://github.com/cmstas/ewkcoffea.

[9] TaskVine. https://ccl.cse.nd.edu/software/taskvine/.

[10] Notre Dame CCL. https://ccl.cse.nd.edu/.

[11] PyHEP 2023 "Executing Analysis Workflows at Scale with Coffea+Dask+TaskVine", Ben Tovar. https://indico.cern.ch/event/1252095/contributions/5592413.

[12] Dask.distributed documentation. https://distributed.dask.org/en/latest/.

[13] Aram Hayrapetyan et al. Search for physics beyond the standard model in top quark production with additional leptons in the context of effective field theory. *JHEP*, 12:068, 2023.

[14] "TOP-22-006 Analysis Experience", CMS CAT meeting. https://indico.cern.ch/event/1315863/contributions/5534554, 2023.

[15] Ben Tovar et al. Dynamic task shaping for high throughput data analysis applications in high energy physics. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 346–356, 2022.

[16] Philip Chang et al. Segment Linking: A Highly Parallelizable Track Reconstruction Algtorithm for HL-LHC. *J. Phys. Conf. Ser.*, 2375(1):012005, 2022.

[17] Aram Hayrapetyan et al. Portable acceleration of CMS computing workflows with coprocessors as a service. 2 2024.

[18] USCMS R&D Proposal "Deploying GPU algorithms through SONIC". https://uscms-software-and-computing.github.io/assets/pdfs/Kelci-Mohrman.pdf.