PROCEEDINGS OF SCIENCE

The IRIS-HEP Analysis Grand Challenge

Alexander Held[a,*] **and Oksana Shadura**[b]

[a] *University of Wisconsin–Madison,*
447 Lorch St., Madison, WI, United States
[b] *University of Nebraska–Lincoln,*
1400 R St, Lincoln, NE, United States

E-mail: alexander.held@cern.ch, oksana.shadura@cern.ch

Analysis workflows commonly used at the LHC experiments do not scale to the requirements of the HL-LHC. To address this challenge, a rich research and development program is ongoing, proposing new tools, techniques, and approaches. The IRIS-HEP software institute and its partners are bringing together many of these developments and putting them to the test in a project called the Analysis Grand Challenge (AGC). The AGC aims to demonstrate how novel workflows can scale to analysis needs at the HL-LHC. It is based around a physics analysis using publicly available Open Data and includes the relevant technical requirements and features that analysers at the LHC need. The analysis demonstration developed in this context is heavily based on tools from the HEP Python ecosystem and makes use of modern analysis facilities. This talk will review the state of the ecosystem, describe the status of the AGC, and showcase how the envisioned workflows look in practice.

41st International Conference on High Energy physics - ICHEP2022
6-13 July, 2022
Bologna, Italy

*Speaker

PoS(ICHEP2022)235

1. Introduction

The Institute for Research and Innovation in Software for High Energy Physics, IRIS-HEP [1], develops software cyberinfrastructure that aims to address the computing challenges of the HL-LHC. IRIS-HEP involves a number of physicists, computer scientists, and engineers from a range of institutes across the United States who perform research and development for the HL-LHC in multiple areas. Those areas include innovative algorithms for data reconstruction and triggering, analysis systems to reduce time-to-insight and maximize physics potential, as well as data organization, management, and access systems. As new tools, techniques and approaches are arising from this program of work within IRIS-HEP and the broader community, it becomes important to put these new ideas to the test.

The IRIS-HEP Analysis Grand Challenge (AGC) started out as an integration exercise with the idea of testing an end-to-end analysis pipeline designed for use at the HL-LHC in the context of a physics analysis of realistic scope and scale. It was designed to provide the environment where technologies being developed within IRIS-HEP and the adjacent ecosystem could be connected together. There are two components to the AGC for this purpose:

- the definition of a physics analysis task representative of HL-LHC requirements,
- the implementation of an analysis pipeline addressing this task.

With a pipeline implemented, the AGC allows to identify and address performance bottlenecks and usability issues.

1.1 Analysis in the context of the AGC

Analysis in the AGC starts from centrally produced common data samples that are assumed to be available for physicists to use. The term refers to all subsequent steps that are needed to produce the outputs needed for publishing a result. Figure 1 visualizes these workflow steps.

Analysis includes: extraction of relevant data, filtering of events, calibration of objects and the calculation of systematic variations, construction of observables, histogramming (in case of binned analyses), construction of statistical models and statistical inference, visualization of results and all relevant information to study analysis details. These steps need to be performed in a reproducible way, and are expected to be run many times to go from a first idea to a full publication-ready result.

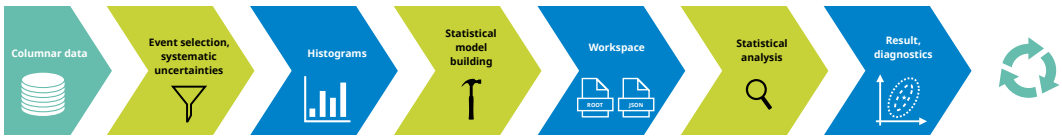

Figure 1: Schematic analysis workflow in the AGC context.

1.2 Beyond an integration exercise

Following its inception as an integration exercise, the AGC is now also meant to be a resource that can be useful to the broader community. It provides a testbed for analysis software developers

to explore user experience, interfaces, and performance. The AGC analysis task allows for prototyping analysis workflows, implementations of AGC analysis pipelines allow for functionality and integration testing for analysis facilities. Facilities based on the coffea-casa [2] model provide the environment and services to execute analysis pipelines at scale. Users can interact with a facility through a JupyterLab [3] interface and seamlessly scale to additional resources provided by a batch system via Dask [4].

Another aspect being investigated in the context of the AGC is the feasibility of "interactive" analysis with a turnaround time of minutes or less. Achieving this with the data volumes expected from the HL-LHC requires highly parallel execution in short bursts, with low latency and good use of caching for intermediate results. The feasibility of this can be studied with AGC analysis pipeline implementations.

Two dedicated workshops [5, 6] were organized so far in the context of the AGC to showcase AGC pipeline implementations and demonstrate the components being used, but to also interact and receive community input.

2. The analysis task

The main analysis task in the AGC is a $t\bar{t}$ cross-section measurement in the single lepton channel. By focusing on a process like $t\bar{t}$ production that is so ubiquitous at the LHC, in a region of phase space where a large number of physics analyses are situated, this analysis task is well positioned for extensions and alterations. It allows for example a conversion into a search for phenomena beyond the Standard Model. The analysis task is set up to capture relevant workflow aspects encountered in physics analysis.

The use of CMS Open Data from 2015 [7] as the input for the analysis task means that anyone can participate in the AGC without the need for special permissions. The CMS Collaboration made this data available in MiniAOD format. The AGC uses 4 TB of data converted to a custom ROOT ntuple format, corresponding to roughly 1 billion events. Exploration of other formats, such as the CMS NanoAOD format, is planned in the context of the AGC. The AGC analysis task, and implementations of it, are strictly aimed at demonstrating workflows and functionality instead of physics results. As workflows are the focus, made-up tools can be used for detailed aspects such as the exact object calibrations or evaluations of systematic variations as long as the relevant workflow can still be captured.

2.1 Focus on user experience

As denoted in the Second Analysis Ecosystem Workshop report [8], the three biggest pain points when it comes to the physics analysis user experience are dealing with systematic uncertainties, handling metadata, and scaling a pipeline from a prototype to a sufficiently powerful facility. The AGC analysis task includes the handling of various types of systematic uncertainties to probe the user experience in this aspect. These include weight-based uncertainties, object-based variations that change kinematics of jets and leptons (thereby affecting the selection of events and calculation of observables), as well as other types of uncertainties that do not need to be evaluated at the stage of event processing (but only when constructing the statistical model, such as cross-section uncertainties). Handling of metadata is addressed by various bookkeeping aspects that are required

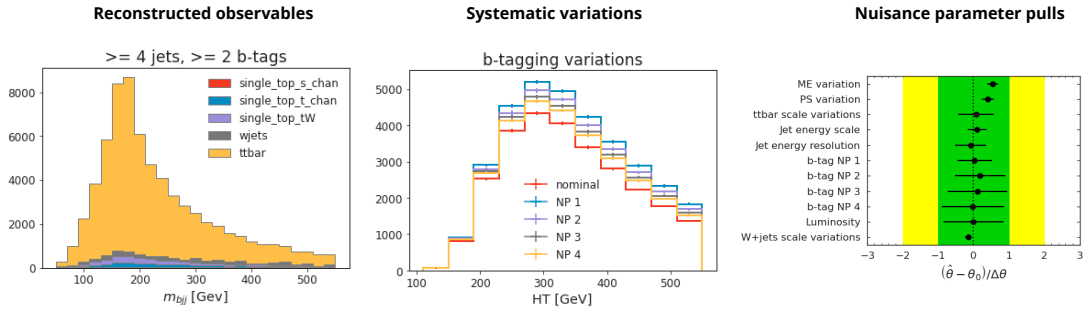

Figure 2: Examples of figures produced in a reference implementation of the $t\bar{t}$ analysis task.

to successfully address the AGC tasks. The last point mentioned in the report, scale-out user experience, can be studied for any AGC implementation by investigating how seamlessly users can move from prototyping on their laptop to running at full scale on a large analysis facility.

3. A reference implementation

IRIS-HEP provides a reference implementation of a suitable analysis pipeline to address the $t\bar{t}$ analysis task. It is based on the coffea [9] framework, which provides the tooling for columnar analysis and facilitates scaling of the pipeline via a range of supported execution engines. This allows for running on a broad set of computing resources without the need to modify the analysis code itself. The AGC makes heavy use of libraries from the Scikit-HEP [10] project to provide relevant functionality in conjunction with coffea. ROOT files are read with uproot [11] and awkward-array [12] is used for columnar data processing. The boost-histogram [13] and hist [14] libraries are used for histogram handling. Construction and handling of statistical models are performed using cabinetry [15], with pyhf [16, 17] used for statistical inference.

ServiceX [18] is optionally used as a data delivery service. It serves data following a declarative request, returning the desired columns and filters. The ServiceX instance can be co-located with the input data for optimal performance, and the output of a request is cached locally to speed up subsequent analysis executions. A range of additional optional services is also being investigated.

The analysis implementation, alongside information about where to find the relevant input data, is provided on GitHub [19]. Figure 2 shows a few of the figures produced when running the analysis: a histogram of the reconstructed top quark mass, different systematic variations related to b-tagging, and results of a maximum likelihood fit.

4. Current status and next steps

The pipeline developed to address the AGC analysis task generally works well, with well-defined interfaces between the various components in the workflow. The implementation and accompanying performance testing revealed various areas for further improvement. Several performance bottlenecks are being addressed and significant gains are expected. Aspects related to user experience were also uncovered in this process, both related to the interfaces between different

libraries and services, and also in the context of handling systematic uncertainties. These points are also being followed up upon to further improve the analysis implementation.

The $t\bar{t}$ analysis task is foreseen to be expanded further in the future, featuring an expanded set of systematic uncertainties to handle and a larger amount of data to process. Another extension will be a machine learning component, which has been frequently requested. In addition to this, the AGC project plans to provide a complete description of the $t\bar{t}$ analysis task that is fully decoupled from any implementation and to work on addressing the performance and usability issues that have been observed. An implementation of the data processing part of the $t\bar{t}$ analysis task has also been developed in ROOT's RDataFrame [20]. As new approaches appear, comparisons between them will be another focus.

A longer-term plan is the development of a differentiable analysis pipeline. This would allow for an investigation of end-to-end analysis optimization via gradient descent, and provide the possibility to evaluate the usefulness of gradient information in this context.

5. Conclusion

The AGC is an integration exercise to study HL-LHC analysis workflows. It defines a physics analysis task of relevant scope and scale, based on a $t\bar{t}$ cross-section measurement using 2015 CMS Open Data. An implementation addressing this task was developed, making use of coffea, many Scikit-HEP libraries, and ServiceX as an optional data delivery service. Coffea-casa analysis facilities provide the environment to execute this physics analysis at scale. All required data to perform the analysis task, as well as the code for a reference implementation, are accessible via a GitHub repository.

Acknowledgments

This work was supported by the U.S. National Science Foundation (NSF) Cooperative Agreement OAC-1836650 (IRIS-HEP).

The AGC is made possible thanks to the help of a large number of people working on many different projects. Thank you in particular to the teams behind: coffea-casa, Scikit-HEP, coffea, IRIS-HEP Analysis Systems, ServiceX, IRIS-HEP DOMA, IRIS-HEP SSL, and the CMS Data Preservation and Open Access (DPOA) group.

References

[1] IRIS-HEP, "Institute for Research and Innovation in Software for High Energy Physics (IRIS-HEP)." https://iris-hep.org/.

[2] M. Adamec, G. Attebury, K. Bloom, B. Bockelman, C. Lundstedt, O. Shadura et al., *Coffea-casa: an analysis facility prototype*, *EPJ Web Conf.* **251** (2021) 02061.

[3] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic et al., *Jupyter notebooks - a publishing format for reproducible computational workflows*, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Scmidt, eds., (Netherlands), pp. 87–90, IOS Press, 2016, https://eprints.soton.ac.uk/403913/.

[4] Dask Development Team, "Dask: Library for dynamic task scheduling." https://dask.org, 2016.

[5] A. Held, O. Shadura (organizers), "IRIS-HEP AGC Tools 2021 Workshop." https://indico.cern.ch/event/1076231/, Nov., 2021.

[6] A. Held, O. Shadura (organizers), "IRIS-HEP AGC Tools 2022 Workshop." https://indico.cern.ch/event/1126109/, Apr., 2022.

[7] CMS Data preservation and open access group, "Getting Started with CMS 2015 Open Data." https://opendata.cern.ch/docs/cms-getting-started-2015, 2022.

[8] G.A. Stewart, P. Elmer, G. Eulisse, L. Gouskos, S. Hageboeck, A.R. Hall et al., *HSF IRIS-HEP Second Analysis Ecosystem Workshop Report*, Aug., 2022. 10.5281/zenodo.7003963.

[9] L. Gray, N. Smith, B. Tovar, A. Novak, J. Chakraborty, P. Fackeldey et al., "coffea." 10.5281/zenodo.3266454.

[10] E. Rodrigues et al., *The Scikit HEP Project – overview and prospects*, *EPJ Web Conf.* **245** (2020) 06028 [2007.03577].

[11] J. Pivarski, H. Schreiner, A. Hollands, P. Das, K. Kothari, A. Roy et al., "Uproot." 10.5281/zenodo.4340632.

[12] J. Pivarski, I. Osborne, I. Ifrim, H. Schreiner, A. Hollands, A. Biswas et al., "Awkward Array." 10.5281/zenodo.4341376.

[13] H. Schreiner, H. Dembinski, A. Goel, J. Gohil, S. Liu, J. Eschle et al., "boost-histogram." 10.5281/zenodo.3492034.

[14] H. Schreiner, S. Liu and A. Goel, "hist." 10.5281/zenodo.4057112.

[15] A. Held, M. Feickert, H. Schreiner, L. Henkelmann, A. Hollands and N. Simpson, "cabinetry." 10.5281/zenodo.4742752.

[16] L. Heinrich, M. Feickert and G. Stark, "pyhf." 10.5281/zenodo.1169739.

[17] L. Heinrich, M. Feickert, G. Stark and K. Cranmer, *pyhf: pure-python implementation of histfactory statistical models*, *Journal of Open Source Software* **6** (2021) 2823.

[18] B. Galewsky, R. Gardner, L. Gray, M. Neubauer, J. Pivarski, M. Proffitt et al., *ServiceX A Distributed, Caching, Columnar Data Delivery Service*, *EPJ Web Conf.* **245** (2020) 04043.

[19] A. Held, O. Shadura, M. Feickert, J. Chakraborty, M. Proffitt, K. Choi et al., "Analysis Grand Challenge." 10.5281/zenodo.7274936.

[20] D. Piparo, P. Canal, E. Guiraud, X.V. Pla, G. Ganis, G. Amadio et al., *RDataFrame: Easy Parallel ROOT Analysis at 100 Threads*, *EPJ Web Conf.* **214** (2019) 06029.