First performance measurements with the Analysis Grand Challenge

Oksana Shadura[1], **Alexander Held**[2]

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

E-mail: `oksana.shadura@cern.ch`

Abstract. The IRIS-HEP Analysis Grand Challenge (AGC) is designed to be a realistic environment for investigating how analysis methods scale to the demands of the HL-LHC. The analysis task is based on publicly available Open Data and allows for comparing the usability and performance of different approaches and implementations. It includes all relevant workflow aspects from data delivery to statistical inference. The reference implementation for the AGC analysis task is heavily based on tools from the HEP Python ecosystem. It makes use of novel pieces of cyberinfrastructure and modern analysis facilities in order to address the data processing challenges of the HL-LHC. This contribution compares multiple different analysis implementations and studies their performance. Differences between the implementations include the use of multiple data delivery mechanisms and caching setups for the analysis facilities under investigation.

1. Introduction

The Analysis Grand Challenge (AGC) started out as an integration exercise for IRIS-HEP [1], connecting various areas of work within the institute and the surrounding ecosystem. It quickly transformed from that into a project that we positioned to be useful to the broader community. The AGC provides a testbed for physics analysis software developers to explore user experience, interfaces, and performance [2].

The goal of the analysis exercise is to demonstrate the handling of data pipeline requirements of the HL-LHC, including large data volumes, bookkeeping, and handling of different types of systematic uncertainties. It also includes investigations of the use of reduced data formats (e.g. PHYSLITE [3] or NanoAOD [4]), aligned with the goals of the LHC experiments. In addition to that, the project aims to engage users to explore columnar analysis concepts.

The AGC also aims to explore the concept of fast "interactive analysis" with a turnaround time of minutes or less. We are testing the feasibility of this idea by employing highly parallel execution in short bursts, which furthermore needs to happen with low latency. This also requires heavy use of caching to improve performance in subsequent executions of similar analysis tasks.

The HL-LHC will introduce new computing challenges surrounding the adaption of existing analysis paradigms at facilities to handle more data-intense end-user data analysis. To address this, there are ongoing efforts from different groups to study and prototype new facilities capable to assist in the upcoming physics analysis challenges.

Another important target we envision for the AGC is to prepare analysis facilities for execution of analyses towards the HL-LHC and to provide new concepts and services for end users.

2. IRIS-HEP Analysis Grand Challenge components

The AGC project includes efforts on a number of related items. While IRIS-HEP is involved in all of them, the project is structured in a way to allow for contributions on specific aspects of the project without having to interact with everything at once. Aspects of the work include:

(i) defining a physics analysis task of realistic HL-LHC scope and scale, allowing to easily implement and re-implement it;

(ii) developing analysis pipelines that implement said physics analysis task;

(iii) finding and addressing performance bottlenecks and usability concerns for the pipelines implemented.

These proceedings describe first performance measurements obtained, following the definition of an analysis task and its implementation with a specific pipeline as described in reference [2]. We will briefly summarize the task and implementation here.

2.1. AGC analysis task description

The physics analysis task consists of a cross-section measurement of top quark pair production in final states with a single charged lepton. This task is chosen to capture relevant workflow aspects of a typical physics analysis. The analysis phase space is also somewhat generic, allowing to convert the setup into other types of analyses, such as searches for beyond the Standard Model physics phenomena. The analysis task features prominently the handling of different types of systematic uncertainties, including the handling of associated metadata and bookkeeping aspects. The analysis logic itself includes simple kinematic top quark candidate reconstruction.

The input data to this task is derived from Run-2 CMS Open Data [5], with around 400 TB available in MiniAOD [6] format. The implementation described here makes use of inputs in an ntuple format, pre-converted from the MiniAOD format and consisting of about 1 Billion events (around 3.5 TB) made available publicly in XRootD-accessible storage at the University of Nebraska–Lincoln.

The custom ntuple format used here is similar to the NanoAOD format used by CMS and the PHYSLITE format used by ATLAS. The analysis task probes a workflow that is applicable to both ATLAS and CMS.

Open Data plays a crucial role in this project, as it allows anyone to participate without requiring specific access permissions. The analysis task focuses on demonstrating realistic workflows, but is not concerned about getting all physics details fully correct: it includes the use of simplified tools for calibrations and systematic uncertainties to probe the workflow aspects.

2.2. An AGC implementation: software stack and analysis pipeline

The implementation of the AGC analysis task employed for the results shown in these proceedings makes use of a set of tools developed by IRIS-HEP and partners, with figure 1 depicting the software stack being used and tested.

The pipeline setup includes the *ServiceX* [7] service providing the delivery of columns following a declarative *FuncADL* [8] request. The *coffea* [9] framework then orchestrates distributed event processing and histogram production using *uproot* [10] and *awkward-array* [11], with *hist* [12, 13] providing histogramming functionality. Visualization is provided by the *mplhep* [14] package. The statistical model construction is done by *cabinetry* [15], while statistical inference is performed using the *pyhf* [16, 17] package.

Implementations for AGC analyses task are openly developed in the IRIS-HEP AGC repository [18], including the specific implementation used for the performance results in these

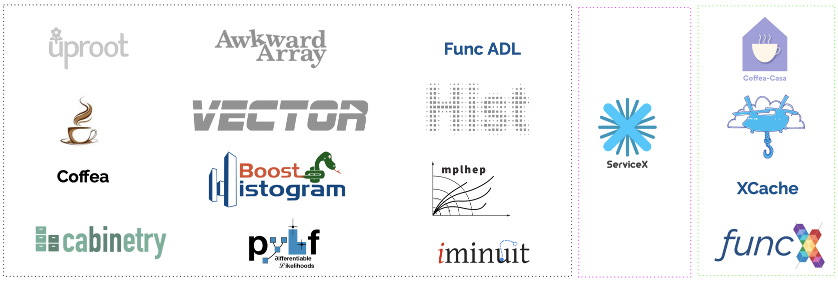


Figure 1: Software packages employed and considered in the AGC implementation described here, including the ones focused on end-user physics analysis (left box), data delivery service (*ServiceX*, middle box) and additional services provided by analysis facilities (right box).

proceedings. The repository also includes the categorization of datasets in terms of their role in the AGC analysis task and where to find them.

2.3. An AGC implementation: R&D on data management tools

An ongoing research and development effort focuses on improved techniques for delivering physics events to analysts. This includes the effort on the development of dedicated data delivery services (such as *ServiceX*) and integrating them together as one coherent ecosystem. All of this is intended to be available on analysis facilities, offering a user-friendly access and experience.

2.4. An AGC implementation: R&D on analysis facilities

As one of the prototype facilities targeting the HL-LHC, the Coffea-casa facility [19], developed by the University of Nebraska–Lincoln (IRIS-HEP), brings new, interactive paradigms for users from R&D into production. This facility is used as a testbed for the IRIS-HEP Analysis Grand Challenge, offering the possibility for end-users to execute analysis at HL-LHC-scale data rates. This facility is adopting an approach that allows it to transform existing facilities (e.g. LHC Tier-2 sites) into composable systems, using Kubernetes as the enabling technology as described in figure 2.

The Coffea-casa facility provides modularity and portability offering various configurations. The Coffea-casa team demonstrated the ability to port and customize the analysis facility setup to another site, co-locating it with the existing ATLAS Tier-3 analysis facility at the University of Chicago. The configuration required adjusting the Coffea-casa facility setup to become more Kubernetes-native, providing an HTCondor batch queue directly in Kubernetes.

3. Results

The performance measurements employ two AGC analysis setups to test different scalability issues: many-core scalability and distributed scaling. Two different facilities were used as testbeds, one at the University of Nebraska–Lincoln, a second at the University of Chicago.

The University of Nebraska–Lincoln hosts a US-CMS Tier-2 site as well as resources dedicated to IRIS-HEP for development of the Coffea-Casa CMS analysis facility. This facility was used for an AGC scale-out performance benchmarking setup, with available resources including 12x Dell R750 each with dual Xeon Gold 6348 28C/56T CPUs, 512GB RAM, 200Gb networking, and 10x 3.2TB NVMe, providing in total 672 cores.

The facility at the University of Chicago was used to test the scaling performance when using locally available input files. The university hosts a US-ATLAS Tier-3 site and additional resources dedicated to IRIS-HEP for the development of a Coffea-Casa ATLAS analysis facility.

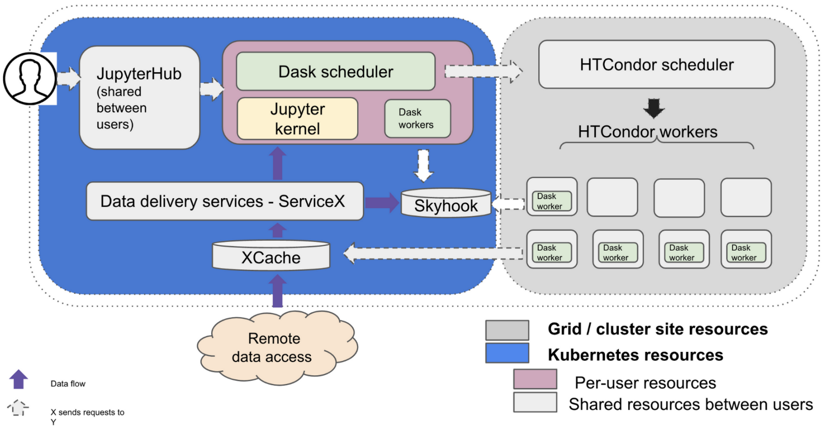
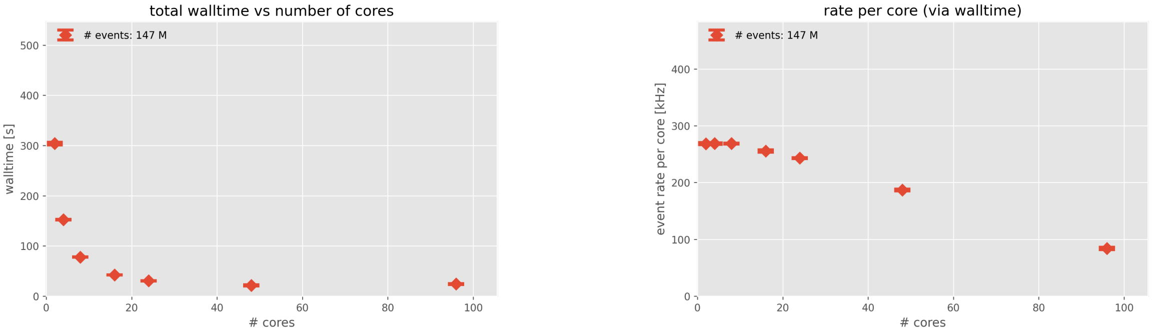


Figure 2: Coffea-casa facility, developed by the University of Nebraska-Lincoln (IRIS-HEP).

Resources available for testing included 16 nodes with dual Xeon Gold 6348 56C/112T CPUs, 384 GB RAM, and 10x 3.2 TB NVMe, providing in total 1792 cores.

3.1. AGC setup with dataset stored on local disks

The goal of these performance tests was to check the multi-core scalability for an AGC setup using *coffea* as analysis framework and processing the AGC dataset stored on local disks to avoid network overhead. For efficient scaling over multiple local cores we used Python futures via the *FuturesExecutor* in *coffea*.



(a) Walltime measurements as a function of the number of cores used.

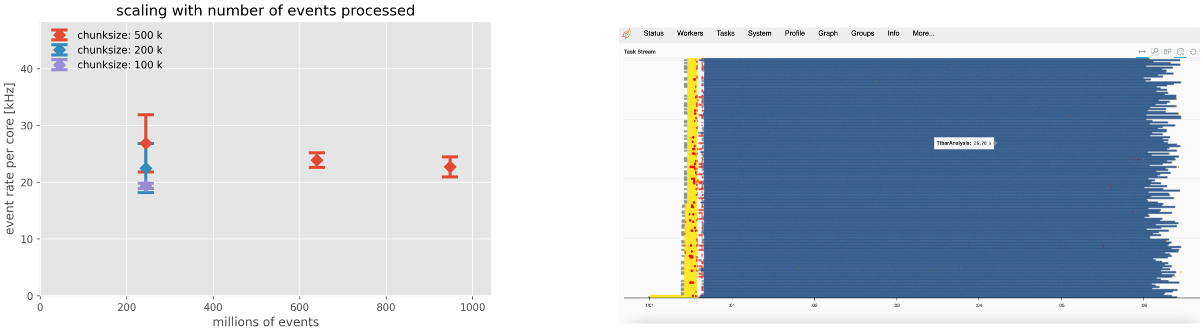
(b) Event rate as a function of the number of cores used.

Figure 3: Reading locally stored files and scaling on local machine at the University of Chicago Coffea-casa AF.

Figure 3 shows the performance in terms of total walltime and event rate per core when scaling the analysis to use more cores. The slight degradation in efficiency may point towards some remaining overhead in the parallelization.

3.2. AGC: scale-out to distributed resources

Tests of the scale-out AGC implementation performance in a distributed setup were performed at the Coffea-casa facility at the University of Nebraska–Lincoln. We used a *coffea* setup employing the *DaskExecutor* to allow running tasks in a *Dask* [20] *Distributed* cluster. The resulting jobs ran on the HTCondor Tier-2 batch queue.



(a) Event rate scaling as a function of the number of events processed.

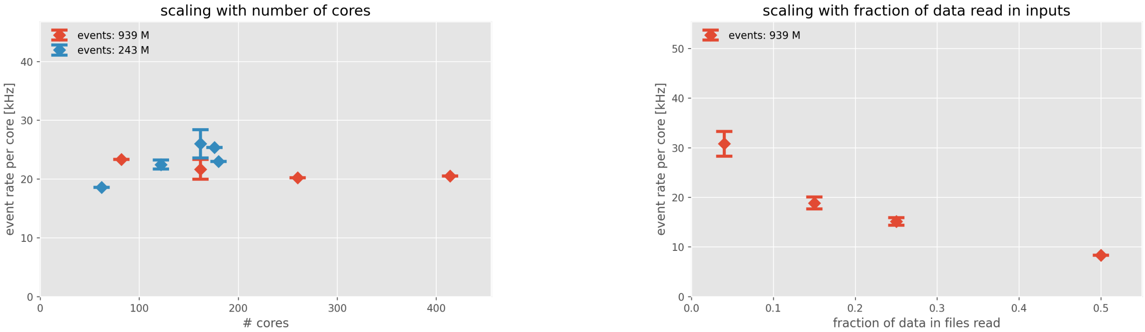
(b) Dask task graph showing efficient scheduling.

Figure 4: Using the Coffea-casa facility at the University of Nebraska–Lincoln CMS Tier-2 (*coffea* with *DaskExecutor*): stable scaling up to 1B events on the Tier-2 HTCondor job queue with efficient scheduling.

Figure 4 shows the event rates measured in scaling tests as a function of the number of events processed. This setup employs files read over the network. The event rate is stable, independent of the size of the dataset being processed. The *Dask* task graph shows efficient scheduling of jobs performing the data processing.

3.3. Scaling with I/O and number of cores

Further scaling tests performed at the University of Nebraska–Lincoln Coffea-casa setup focused on the effect of using an increased number of cores and reading various fractions of the data in the input datasets.



(a) Event rate scaling as a function of number of cores

(b) Event rate scaling as a function of the fraction of data being read.

Figure 5: Using the Coffea-casa facility at the University of Nebraska–Lincoln CMS Tier-2 (*coffea* with *DaskExecutor*): stable scaling to 400 cores, event rates as a function of the fraction of data read.

Stable scaling is observed up to 400 cores as depicted in figure 5. The fraction of data in the input files being read (which changes depending on the number of columns accessed in the file) can have a significant effect on the event rates, indicating that the time spent on data processing is not a significant contribution to the event rate when only reading a small fraction of the data in the files.

4. Future directions for further performance improvements

We expect that the following projects can improve the performance of AGC implementations beyond the results shown in section 3:

(i) XCache [21] — XRootD file-based caching proxy used for regional and site caches to store and serve datasets, helping to reduce latency and WAN traffic;

(ii) ServiceX — data extraction and data delivery service, offering "column-on-demand" functionality;

(iii) Skyhook DM [22] — an extension of the Ceph distributed storage for scalable storage of tables and for offloading common data management operations (selection, projection, aggregation, and indexing, as well as user-defined functions).

5. Conclusion and outlook

The first performance measurements obtained in the context of testing a specific pipeline implementing the AGC analysis task at various facilities show promising results. We plan to extend the analysis pipeline to include additional methods for data delivery and compare their performance in future work. We also expect to extend measurements to additional hardware configurations on various CMS and ATLAS analysis facilities.

Acknowledgments

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

References

[1] Elmer P, Neubauer M and Sokoloff M D 2017 Strategic Plan for a Scientific Software Innovation Institute (S2I2) for High Energy Physics (*Preprint* 1712.06592)

[2] Held A and Shadura O 2022 *PoS* **ICHEP2022** 235

[3] Elmsheuser J *et al.* 2020 *EPJ Web Conf.* **245** 06014

[4] Rizzi A, Petrucciani G and Peruzzi M (CMS) 2019 *EPJ Web Conf.* **214** 06021

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

[6] Petrucciani G, Rizzi A, Vuosalo C and on behalf of the CMS Collaboration 2015 *Journal of Physics: Conference Series* **664** 072052 URL https://dx.doi.org/10.1088/1742-6596/664/7/072052

[7] Galewsky B, Gardner R, Gray L, Neubauer M, Pivarski J, Proffitt M, Vukotic I, Watts G and Weinberg M 2020 *EPJ Web Conf.* **245** 04043

[8] Proffitt M and Watts G 2021 *EPJ Web Conf.* **251** 03068

[9] Gray L *et al.* coffea URL https://doi.org/10.5281/zenodo.3266454

[10] Pivarski J, Schreiner H, Hollands A, Das P, Kothari K, Roy A, Ling J, Smith N, Burr C and Stark G Uproot URL https://doi.org/10.5281/zenodo.4340632

[11] Pivarski J, Osborne I, Ifrim I, Schreiner H, Hollands A, Biswas A, Das P, Roy Choudhury S, Smith N and Goyal M Awkward Array URL https://doi.org/10.5281/zenodo.4341376

[12] Schreiner H *et al.* boost-histogram URL https://doi.org/10.5281/zenodo.3492034

[13] Schreiner H, Liu S and Goel A hist URL https://doi.org/10.5281/zenodo.4057112

[14] Novak A *et al.* 2022 mplhep URL https://doi.org/10.5281/zenodo.3766157

[15] Held A, Feickert M, Schreiner H, Henkelmann L, Hollands A and Simpson N cabinetry URL https://doi.org/10.5281/zenodo.4742752

[16] Heinrich L, Feickert M and Stark G pyhf URL https://doi.org/10.5281/zenodo.1169739

[17] Heinrich L, Feickert M, Stark G and Cranmer K 2021 *Journal of Open Source Software* **6** 2823 URL https://doi.org/10.21105/joss.02823

[18] Held A *et al.* Analysis Grand Challenge URL https://doi.org/10.5281/zenodo.7274936

[19] Adamec M, Attebury G, Bloom K, Bockelman B, Lundstedt C, Shadura O and Thiltges J 2021 *EPJ Web Conf.* **251** 02061

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

[21] Bauerdick L A T *et al.* (CMS) 2014 *J. Phys. Conf. Ser.* **513** 042044

[22] Chakraborty J, Jimenez I, Rodriguez S A, Uta A, LeFevre J and Maltzahn C 2022 *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)* pp 81–88