Coffea-Casa: Building composable analysis facilities for the HL-LHC

Sam Albin[1], Garhan Attebury[1], Kenneth Bloom[1], Brian Bockelman[2], Carl Lundstedt[1], Oksana Shadura[1,*], and John Thiltges[1]

[1]University of Nebraska-Lincoln, Lincoln, NE 68588
[2]Morgridge Institute for Research, 330 N. Orchard Street, Madison, WI 53715

Abstract. The large data volumes expected from the High Luminosity LHC (HL-LHC) present challenges to existing paradigms and facilities for end-user data analysis. Modern cyberinfrastructure tools provide a diverse set of services that can be composed into a system that provides physicists with powerful tools that give them straightforward access to large computing resources, with low barriers to entry. The Coffea-Casa analysis facility (AF) provides an environment for end users enabling the execution of increasingly complex analyses such as those demonstrated by the Analysis Grand Challenge (AGC) and capturing the features that physicists will need for the HL-LHC.
We describe the development progress of the Coffea-Casa facility featuring its modularity while demonstrating the ability to port and customize the facility software stack to other locations. The facility also facilitates the support of batch systems while staying Kubernetes-native. We present the evolved architecture of the facility, such as the integration of advanced data delivery services (e.g. ServiceX) and making data caching services (e.g. XCache) available to end users of the facility. We also highlight the composability of modern cyberinfrastructure tools. To enable machine learning pipelines at Coffea-casa analysis facilities, a set of industry ML solutions adopted for HEP columnar analysis were integrated on top of existing facility services. These services also feature transparent access for user workflows to GPUs available at a facility via inference servers while using Kubernetes as enabling technology.

1 Introduction

The HL-LHC will be a challenging analysis environment compared to the LHC experiments of today. The data volumes will go up by a factor of 100 and to achieve the desired physics reach of results, analysts will need new techniques and approaches as well as new software infrastructure, see Figure 1. An analysis that physicists are able to do on a laptop today (e.g. the typical computing resource available for the user) by simply having files with physics events stored on a local disk, instead would be expected to be handled on a dedicated facility during the HL-LHC era, leveraging complex large-scale computing hardware, advanced data delivery services, and ML training and inference see Figure 1. It is also expected that the analysis target scan turnaround time will decrease from weeks to hours. The analysis specifics

* e-mail: oksana.shadura@cern.ch

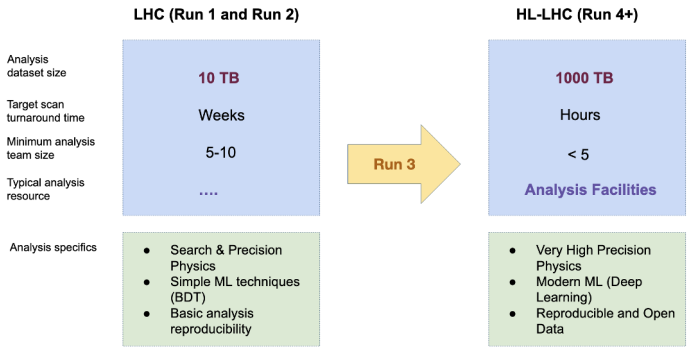

Figure 1. Requirements for the physics analysis and computing infrastructure for HL-LHC

will be also changed including very high precision physics with modern machine learning methods and which should be reproducible and physics data samples used for analysis should be in open-access.

Existing facilities provide the back-bone setups for physicists doing analysis, but can be complex for users who need to track different configurations, different ways to access data, or trying to easily move from one facility to another. Managing these different interfaces and different scaling mechanisms on various facilities, together with the lack of documentation, could be challenging experience especially for new users. Also, not all existing facilities are suitable for interactive analysis use.

The Coffea-Casa analysis facility [1] starts with a base of the "Coffea" processing framework [2] for low-latency columnar analysis and uses a modular approach, adding other services such as ServiceX [3] or a simple scale-out of tasks into a traditional batch system. The facility provides an interactive experience for physicists that is closer to working on a laptop as opposed to a traditional batch system-based facility. The facility adopts an approach that allows transforming existing computing facilities into composable systems using Kubernetes as the enabling technology.

In this article we will describe the detailed overview of design approaches behind the Coffea-Casa analysis facility. We will also describe the experience of using the facility as a testbed for early adopters to investigate the Python analysis ecosystem and additional services. In this way we hope to reach the goal of handling HL-LHC analysis requirements and expected data rates.

2 Building blocks for Coffea-Casa analysis facility

Over the last few years, the Coffea-Casa analysis facility development has validated essential design features, enumerated in Figure 2, for use in future facilities. In this figure, features are enlisted from top-to-down showing gradually the ones on the top that are crucial for users and down - the most important for the service managers.

Coffea-Casa's basic framework is an extension of the Zero-to-JupyterHub project [4] with custom container images, services and classes containing extended functionality. This creates a basic framework on which to build custom functionality to satisfy the design features of Coffea-Casa. Columnar analysis and the Pythonic ecosystem is supported in the custom containers spun up in the Kubernetes [5] instance of Coffea-Casa. Efficient data delivery, data management and data caching solutions are facilitated using an XCache[6] service. Machine Learning (ML) services and tools are provided by Kubernetes based MLFlow [7] and

Triton Inference services [8]. Custom modified Dask-Jobqueue [9] classes allow for scalable integration with large scale compute resources. All of these are managed via modern CI/CD techniques. These modules will be outlined in more detail in the rest of this article.

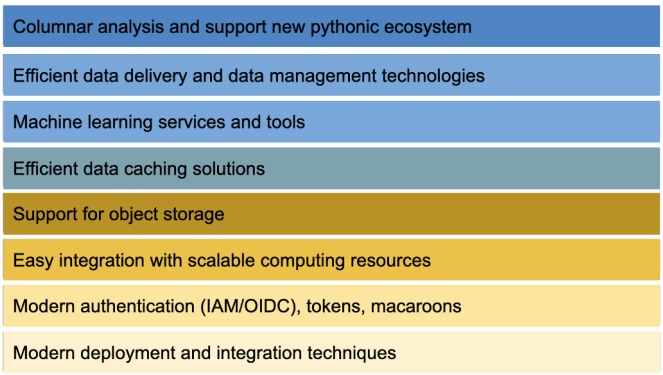

Figure 2. Essential design features for future analysis facility

2.1 Easy integration with scalable computing resources

One of the key design concepts in Coffea-Casa was to leverage currently deployed compute and storage infrastructure to facilitate analysis work. Experiments have invested heavily in the creation and deployment of compute infrastructure designed to serve current workflow paradigms. In order for Coffea-Casa to be readily adopted, it must be able to utilize this infrastructure without modifying the existing infrastructure.

To facilitate this design imperative, Coffea-Casa was created to be easily integrated with existing batch resources. Figure 3 shows how a Coffea-Casa instance integrates with an HTCondor-based resource. Here the HTCondor [10] resource is entirely external to the Coffea-Casa instance. As the Coffea-Casa workload scales up, it submits Dask [11] jobs to the HTCondor queue. These jobs start a container with a Dask worker service that connects back to the Dask scheduler created for the user's instance when the user logged in. In this way the HTCondor resource can be easily utilized to do Dask work for the Coffea-Casa user. The integration with other types of batch resources is readily accomplished by the modular nature of Coffea-Casa.

2.2 Transition to tokens

In past analysis workflows, authentication and authorization was handled through the use of x509 proxy certificates and complex mapping mechanisms. This use of proxy certificates has a couple of inherent drawbacks. First, x509 proxies are generally created with a long lifetime to survive for the entire analysis workflow. Second, there is no capabilities associated with a given proxy; the bearer of the proxy is given access and authorization based solely on identity and not on desired functionality. There does not exist any way for an x509 proxy to have its authorized capabilities stored within the proxy itself. The transition to token based authentication seeks to remedy these two shortcomings.

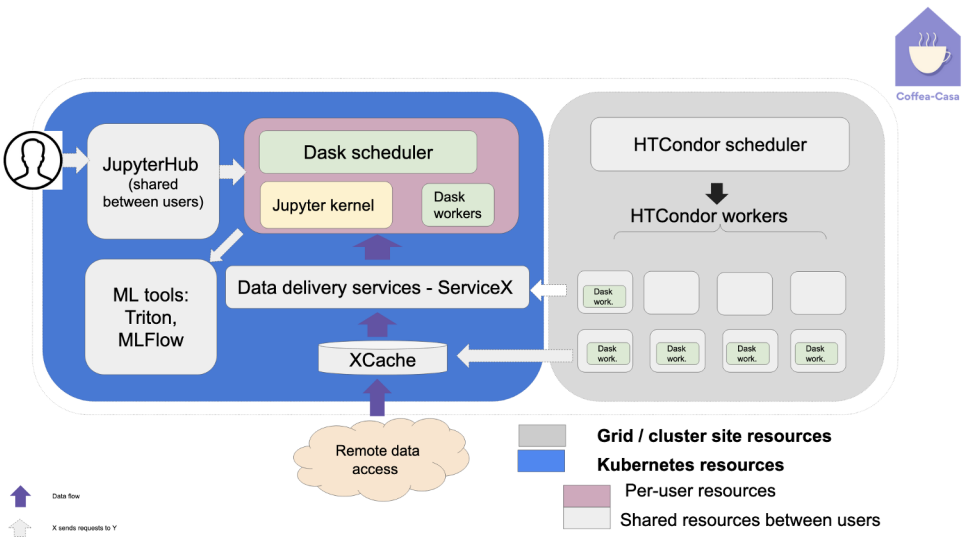

Figure 3. The schema of Coffea-Casa Analysis Facility

First, token lifetimes are created with a much shorter valid lifetime and thus present a smaller window of opportunity for being compromised. A token is able to be renewed to allow for authentication and authorization to persist through the lifetime of the analysis workflow while any individual token remains short lived.

Second, tokens can be constructed to have only specific permissions and capabilities, allowing for fine-grained authorization without the need for complex mapping infrastructure. For instance, a token may be constructed to have read privileges for a storage element but lack any write privileges, thus creating a more secure workflow.

Coffea-Casa seeks to remove, where it can, any use of x509 proxies by the user and instead facilitate authentication and authorization via tokens. These tokens are constructed upon user login based on their identity returned by the OpenID Connect (OIDC) layer [12] of an OAuth service to which the Coffea-Casa instance is registered. In this way a user's identity is verified once at login and the user need not handle identity management manually. Numerous tokens are constructed and managed by Coffea-Casa including tokens to allow reading from an XCache [6] instance and tokens for submitting to an HTCondor queue.

2.3 Integrating data delivery services and advanced caching

The XRootD software framework is foundational for efforts across the LHC. It is used as a reference platform in IRIS-HEP [13] for data streaming and bulk data transfers, as the basis for data federations in CMS, and, in its "XCache" configuration, as a data caching service for both production and analysis. All instances of Coffea-Casa have integrated XCache support for experimental datasets. This allows for data access speedup during the interactive analysis in cases where the user needs to access the dataset multiple times.

For columnar analysis, which is one of the growing approaches to physics analysis, the ServiceX service was developed to run inside analysis facilities. ServiceX is designed to create and cache columns to meet an analyst's data delivery needs. ServiceX is easily deployed in Kubernetes as a part of facility deployment, see Figure 3, and available for each user directly from their session through pre-generated configuration files and tokens.

2.4 Machine learning services and tools for analysis facilities

There are many stand-alone tools to aid with building, training, optimization, and deployment of machine learning (ML) models (e.g. MLFlow or KubeFlow [14]), along with more traditional approaches commonly used in the community. When moving from running individual machine learning workflows on a laptop to running on facilities able to run them efficiently on scale, ML Operations (MLOps) [15] can help. It is a methodology for enabling collaboration across multiple scientists. MLOps helps to gain control over different model versions, multiple experiments within the same problem, and model management and deployment. Adopting MLOps in the analysis workflows allows admins to automate all the steps and incorporate CI/CD practices. Providing MLOps infrastructure requires expertise to deploy and provide relevant MLOps software infrastructure and services. While technically challenging in some cases, it contributes a significant added value for turnaround times on physics analyses that rely heavily on ML approaches. There are multiple industry solutions for various use cases, but as a part of the Analysis Grand Challenge analysis pipeline [16], the main focus is to provide a platform for handling the ML life-cycle and ML inference server often used in HEP analysis.

Two services, MLFlow and Nvidia Triton [8] were deployed and integrated in the Coffea-Casa analysis facility, again see Figure 3, and tested using a typical columnar data analysis workflow. MLFlow is an open-source platform that provides experiment and model run management at the core of their platform and can be easily deployed via Kubernetes. NVIDIA's Triton Inference server provides the transparent access to high-speed, GPU-based inference as part of a user's application.

2.5 Coffea-Casa GitOps development strategy

While developing Coffea-Casa we are relying on a GitOps strategy [17]. GitOps is defined as a model for operating Kubernetes clusters or cloud-native applications (e.g. Coffea-Casa AF). The main advantage of it is that it implements the “infrastructure-as-a-code” concept. It allows for rapid collaboration, better quality control, and higher level of automation (CI/CD).

Using GitOps for the analysis facility development allowed us to easily handle configuration of the facility via a collaborative group of administrators in a deterministic manner. It also allows the team easily to maintain two production facility instances and two different facility sandboxs for development purposes. It also allows us to easily package the core infrastructure as a Helm charts [18] for redeployment in other facilities.

3 IRIS-HEP Analysis Grand Challenge

The IRIS-HEP Analysis Grand Challenge (AGC) [19] started out as an integration exercise for IRIS-HEP to provide mechanisms for testing an end-to-end analysis pipeline. It is designed to prepare infrastructure for the HL-LHC in the context of a physics analysis of realistic scope and scale and to develop flexible, easy-to-use, low latency analysis facilities. In addition, it allows evaluation of the new Python ecosystem for analysis.

There are two components to the AGC [20] for this purpose: the definition of a physics analysis task representative of HL-LHC requirements and the implementation of an analysis pipeline addressing this task. The AGC provides a well-defined physics analysis task with a pipeline implementation. It allows scientists to identify and address performance bottlenecks and usability issues. There have been multiple AGC implementations developed, which use coffea [21] (the IRIS-HEP implementation), RDataframe [22], Julia [23], and columnflow [24].

The goal of the AGC is to bridge analysis at scale gap towards HL-LHC and provide closer connections to LHC experiments. An additional goal is providing extended functionality and testing data preservation pipelines on the coffea-casa facility. We also plan to stress test our facility with a series of AGC benchmarks with incremental data rate goals for throughput.

4 Conclusions

Coffea-casa is a prototype analysis facility delivering the extra functionality needed for improved user experience. Coffea-casa Analysis Facility benefits from the idea of modular architecture and integrates in itself multiple services as dependencies. It allows us to rethink established design patterns and integrate new advanced services with traditional facilities enabling the possibility of quick interactive analysis turnaround, allowing end-users to worry only about physics. We believe focusing on enabling ML-based analysis for facilities together with the ability to handle HL-LHC data volumes is the right path to future analysis facilities.

The facility already accommodates the first users who have been testing the facility to execute their physics analysis using the Coffea analysis framework [21] as well as others. At the beginning 2024, the facility accommodated multiple CMS and Opendata user trainings as well more than 200 users in total used its services.

5 Acknowledgements

This work was supported by the U.S. National Science Foundation (NSF) Cooperative Agreement OAC-1836650 (IRIS-HEP) and NSF-2121686.

References

[1] M. Adamec, G. Attebury, K. Bloom, B. Bockelman, C. Lundstedt, O. Shadura, J. Thiltges, Coffea-casa: an analysis facility prototype, in EPJ Web of Conferences (EDP Sciences, 2021), Vol. 251, p. 02061

[2] N. Smith, L. Gray, M. Cremonesi, B. Jayatilaka, O. Gutsche, A. Hall, K. Pedro, M. Acosta, A. Melo, S. Belforte et al., EPJ Web Conf. **245**, 06012 (2020)

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

[4] Project Jupyter Contributors, "Zero to JupyterHub with Kubernetes", JupyterHub for Kubernetes, 2023, https://z2jh.jupyter.org/

[5] E.A. Brewer, Kubernetes and the path to cloud native, in Proceedings of the sixth ACM symposium on cloud computing (2015), pp. 167–167

[6] L. Bauerdick, K. Bloom, B. Bockelman, D. Bradley, S. Dasu, J. Dost, I. Sfiligoi, A. Tadel, M. Tadel, F. Wuerthwein et al., XRootd, disk-based, caching proxy for optimization of data access, data placement and data replication, in Journal of Physics: Conference Series (IOP Publishing, 2014), Vol. 513, p. 042044

[7] MLflow Project, a Series of LF Projects, LLC, mlflow, https://mlflow.org/

[8] NVIDIA Development Team, NVIDIA Triton Inference Server, https://developer.nvidia.com/triton-inference-server

[9] Dask Development Team, Dask-Jobqueue, https://jobqueue.dask.org/en/latest/

[10] D. Thain, T. Tannenbaum, M. Livny, Concurrency - Practice and Experience **17**, 323 (2005)

[11] Dask Development Team, Dask: Library for dynamic task scheduling, https://dask.org (2016)
[12] N. Sakimura, J. Bradley, M. Jones, B. De Medeiros, C. Mortimore, The OpenID Foundation p. S3 (2014)
[13] IRIS-HEP, Institute for Research and Innovation in Software for High Energy Physics (IRIS-HEP) https://iris-hep.org/
[14] E. Bisong, E. Bisong, Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners pp. 671–685 (2019)
[15] S. Alla, S.K. Adari, S. Alla, S.K. Adari, Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure pp. 79–124 (2021)
[16] E. Kau man, A. Held, O. Shadura, Analysis Grand Challenge ReadTheDocs, https://agc.readthedocs.io/en/latest/
[17] F. Beetz, S. Harrer, IEEE Software **39**, 70 (2021)
[18] Co ea Casa AF Developers, Repository with con guration setup of a prototype of analysis facility - co ea-casa, 2023, https://github.com/CoffeaTeam/coffea-casa
[19] A. Held, O. Shadura, PoS **ISGC2022**, 035 (2022)
[20] A. Held, E. Kau man, O. Shadura, E. Guiraud, M. Feickert, J. Chakraborty, M. Pro , A. Wightman, K. Choi, E. Chavez et al., Analysis Grand Challenge, https://doi.org/10.5281/zenodo.7274936
[21] L. Gray, N. Smith, B. Tovar, A. Novak, J. Chakraborty, P. Fackeldey, N. Hartmann, G. Watts, D. Thain, G. Stark et al., co ea, https://doi.org/10.5281/zenodo.3266454
[22] D. Piparo, P. Canal, E. Guiraud, X.V. Pla, G. Ganis, G. Amadio, A. Naumann, E. Tejedor, Rdataframe: Easy parallel root analysis at 100 threads and, in EPJ Web of Conferences (EDP Sciences, 2019), Vol. 214, p. 06029
[23] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, SIAM review **59**, 65 (2017)
[24] Column ow development team, Column ow, https://columnflow.readthedocs.io/en/stable/