© 2025 CERN for the benefit of the ATLAS Collaboration. CC-BY-4.0 license.

ATL-SOFT-PROC-2025-040
25 February 2025

Investigating Data Access Models for ATLAS: A Case Study with FABRIC Across Borders and ServiceX

Ilija Vukotic[1,*], *Fengping* Hu[1], *Lincoln* Bryant[1], *Robert William* Gardner Jr.[1], *Shawn* McKee[2], *Judith* Stephen[1], and *David* Jordan[1]

[1]Enrico Fermi Institute, University of Chicago, Chicago, IL, USA
[2]University of Michigan, Ann Arbor, Michigan, USA

Abstract. This study explores enhancements in analysis speed, WAN bandwidth efficiency, and data storage management through an innovative data access strategy. The proposed model introduces specialized 'delivery' services for data preprocessing, which include filtering and reformatting tasks executed on dedicated hardware located alongside the data repositories at CERN's Tier-0, Tier-1, or Tier-2 facilities. Positioned near the source storage, these services are crucial for limiting redundant data transfers and focus on sending only vital data to distant analysis sites, aiming to optimize network and storage use at those sites. Within the scope of the NSF-funded FABRIC Across Borders (FAB) initiative, we assess this model using an "in-network, edge" computing cluster at CERN, outfitted with substantial processing capabilities (CPU, GPU, and advanced network interfaces). This edge computing cluster features dedicated network peering arrangements that link CERN Tier-0, the FABRIC experimental network, and an analysis center at the University of Chicago, creating a solid foundation for our research. Central to our infrastructure is ServiceX, an R&D software project under the Data Organization, Management, and Access (DOMA) group of the Institute for Research and Innovation in Software for High Energy Physics - IRIS-HEP. ServiceX is a scalable filtering and reformatting service, designed to operate within a Kubernetes environment and deliver output to an S3 object store at an analysis facility. Our study assesses the impact of server-side delivery services in augmenting the existing HEP computing model, particularly evaluating their possible integration within the broader WAN infrastructure. This model could empower Tier-1 and Tier-2 centers to become efficient data distribution nodes, enabling a more cost-effective way to disseminate data to analysis sites and object stores, thereby improving data access and efficiency. This research is experimental and serves as a demonstrator of the capabilities and improvements that such integrated computing models could offer in the HL-LHC era.

1 Introduction

In high-energy physics, data processing tasks can be broadly categorized into two groups with distinct computing requirements: "production" and "analysis", as they are informally referred to in ATLAS [1]. The production tasks include Monte Carlo (MC) generation, simulation,

[*] e-mail: ivukotic@uchicago.edu

reconstruction, calibration, and similar activities. These tasks typically involve processing vast datasets, consuming millions of CPU hours, and have turn-around times measured in weeks. The PanDA workflow management system [2], designed for global grid computing, efficiently handles such workloads.

In contrast, analysis tasks involve one or more event-filtering steps, followed by detailed analysis and systematic studies. For these tasks, optimal processing demands turn-around times measured in hours, minimal data management overhead (e.g., merging small files, dataset movement, and cleanups), and a streamlined workflow that delivers only the necessary filtered data. This distinction highlights the differing operational needs between production and analysis in high-energy physics.

ServiceX [3], developed by the Data Organization, Management, and Access (DOMA) group within IRIS-HEP [4], enables rapid filtering of large datasets using func_adl [5], with the option to download or stream the resulting data in a user-preferred format. However, its performance heavily relies on the input data being locally accessible.

To address this requirement, two strategies can be employed: implementing a highly efficient caching system or performing the filtering directly at most grid sites where the data resides. We evaluated both approaches and compared their performance against the commonly used download-then-process workflow.

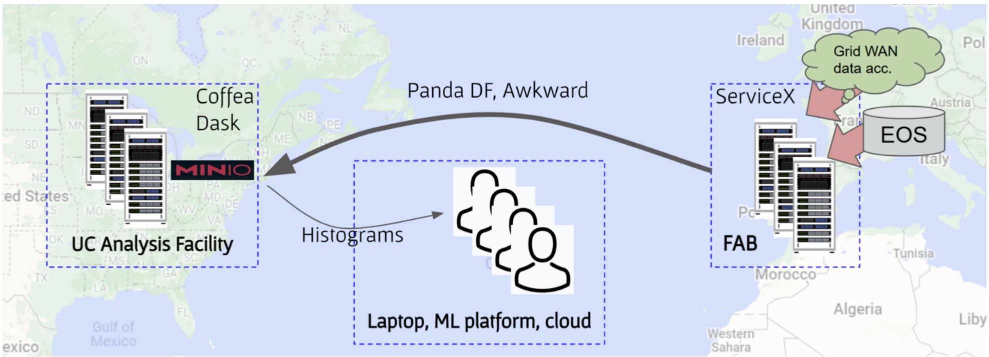

Figure 1. Server-side filtering strategy processes data locally available at ATLAS Tier-0 site (CERN) EOS storage. Outputs are transported via fast FABRIC links to University of Chicago Analysis Facility for a parallel processing using up to a thousand Dask workers.

2 System Architecture and Setup

To evaluate the download-then-process workflow, we utilized the University of Chicago Analysis Facility (UChicago AF) [6]. For testing central filtering with cached remote data access, we employed a production ServiceX instance also hosted at the UChicago AF. Server-side filtering was tested using ServiceXLite, deployed on the FAB cluster at CERN, with output data transported via FABRIC [7] and subsequently processed on a large Dask [8] cluster (Figure 1). In the following sections, we provide a detailed description of each component.

2.1 University of Chicago Analysis Facility

This facility is one of the largest ATLAS Analysis Facilities, supporting over 500 users and offering a wide range of resources, including HTCondor [9] processing queues, JupyterLab [10] environments, GPUs, and Dask clusters. It boasts more than 7,500 CPU cores distributed across 115 nodes, providing robust computational capabilities for its users. All of the resources are a part of single Kubernetes cluster. A Rook-orchestrated Ceph [11] cluster provides filesystem, block device and S3 storage.

2.2 ServiceX

ServiceX comprises multiple interconnected services that require a Kubernetes cluster for deployment. It is managed through a Helm[12] chart and demands careful configuration and maintenance. Given these complexities, deploying and supporting a large number of independent instances is impractical. Additionally, expecting users to log into multiple instances or determine which instance is best suited for specific data introduces significant inefficiencies.

To address these challenges, we developed ServiceXLite—a streamlined, single Kubernetes deployment designed solely for data filtering. ServiceXLite relies on a central ServiceX instance to assign workloads, simplifying deployment and ensuring a more user-friendly experience. The architecture of that setup is shown in Figure 2.

When used to filter data delivered over WAN, the central ServiceX deployed at the UChicago AF was configured to use eight XCache [13] instances with NVMe-based backing storage.

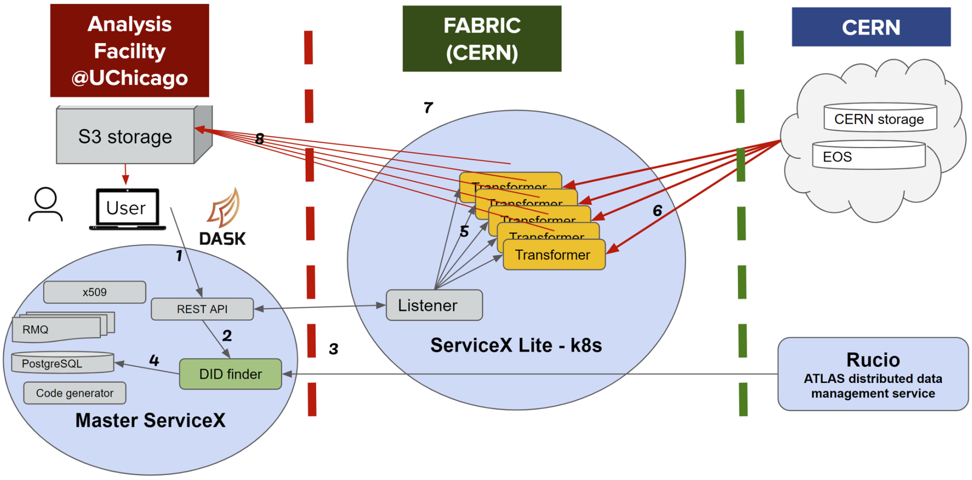

Figure 2. System architecture of the server-side filtering approach. A user at the UChicago AF submits request to a local ServiceX, and uses a local Dask cluster to analyze data from local S3 storage. Numbered arrows show order of steps and direction of information flow. Filtered data has been delivered by a ServiceXLite instance running on the FAB cluster at the CERN Tier-0 site.

2.3 FABRIC and FAB

FABRIC is an international infrastructure that enables cutting-edge experimentation and research at-scale in the areas of networking, cybersecurity, distributed computing, storage, vir-

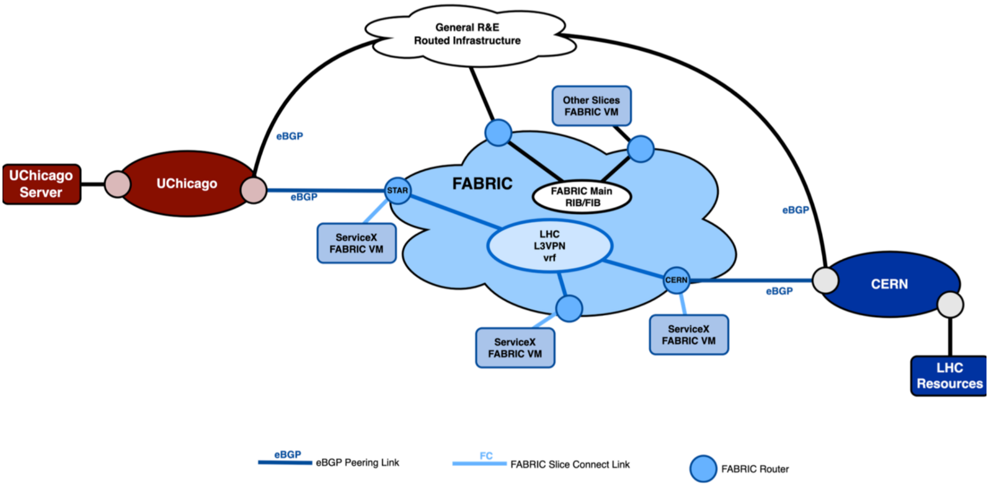

Figure 3. Network peering arrangement that links the CERN Tier-0 and the Analysis Facility at the University of Chicago.

tual reality, 5G, machine learning, and science applications. It has 29 sites interconnected by high speed, dedicated optical links.

FABRIC Across Borders (FAB) is an extension of the FABRIC testbed connecting the core North America infrastructure to four nodes in Asia, Europe, and South America.

The FAB infrastructure at CERN includes almost 1200 CPU cores, 5 GPU equipped nodes, and high-speed network interfaces. A FABRIC experiment (slice) is a virtual cluster composed of compute and networking services. After creating the slice using FABRIC APIs via the Jupyter Notebook service, we utilized Kubespray [14] to set up a Kubernetes cluster and Flux [15] to manage Continuous Deployment (CD) of infrastructure components such as CertManager, the ingress controller, Multus [16], and application components (e.g., ServiceX Lite).

The experiment slice consists of 10 nodes, each equipped with 64 cores and 128 GB of RAM. We leveraged FABRIC's external connections and peering services (based on interconnects via ESnet [17] and R&E networks) to provide the slice with access to both the LHCOne network [18] and the UChicago Analysis Facility network.

By running ServiceX Lite on FABRIC as a middleman, we required only two peering connections: one between the slice and LHCOne, and another between the slice and the UChicago Analysis Facility. This approach eliminated the need for a direct route between LHCOne and the Analysis Facility, preventing experiment traffic from interfering with production traffic and reducing overall complexity, as shown in Figure 3.

3 Performance Evaluation

Our analysis involved processing 3 TB of ATLAS data distributed across 21,000 ROOT [19] files, accessing approximately 5% of the data from each file. The results, including time-to-result and the volume of data transferred, are summarized in Table 1.

The conventional workflow, which relies on downloading the entire dataset using RUCIO [20] followed by HTCondor processing, required nearly 24 hours, with the vast majority of this time spent on data transfer and only minutes on processing.

Table 1. Time to completion and bandwidth used by the three approaches to data analysis

Configuration	Time to result (HH:MM:SS)	Transatlantic data transfer volume (GB)
Download and local processing	22:28:00	3057
ServiceX transformers local, reading over WAN	00:15:28	12
Process data using ServiceXLite on FABRIC@CERN	00:06:54	5
ServiceX transformers local, reading from cache	00:03:33	N/A

Using ServiceX in its standard configuration, input data is read over the WAN. However, the high latency between the University of Chicago and CERN makes reading small file segments inefficient. When leveraging XCaches, blocks of 256 KB are read and cached, significantly reducing the number of read operations and mitigating latency penalties. This approach, however, nearly doubles the amount of data transported. Despite this, the processing time of less than 16 minutes, while transferring only 12 GB across the Atlantic, marks a significant improvement.

The ServiceXLite-based approach, which performs local filtering near the data storage, completed the task in just seven minutes—a 55% improvement in processing time. Moreover, it further reduced the transfer volume by an additional factor of two, as only the filtered data was transported. However, this approach remains roughly twice as slow as the theoretical minimum achievable when processing data directly from a local hot cache.

4 Future Prospects

The study identifies avenues for further enhancement, including the integration of intelligent scaling, fair-share resource allocation, and location-aware data distribution. By redesigning the filtering service as a single distributed system and deploying it across most or all Tier-1 and Tier-2 sites that host analysis data, these facilities could become highly efficient data distribution hubs, strengthening the broader HEP computing ecosystem.

As the HL-LHC era approaches, characterized by exponentially increasing data volumes, the Server side filtering model offers a scalable, cost-effective solution to meet the computing demands of next-generation experiments.

5 Conclusion

By deploying an efficient data filtering service close to the storage hosting frequently used datasets for analysis, we demonstrate the potential of server-side (or even in-network) data filtering and reformatting to revolutionize data access for high-energy physics analysis. This model achieves significant gains in processing speed and bandwidth efficiency, paving the way for more agile, responsive computing infrastructures within the HEP community.

6 Acknowledgements

We want to explicitly acknowledge the support of the National Science Foundation, which supported some of this work through FAB: NSF OAC-2029261, OSG: NSF MPS-1148698 and IRIS-HEP: NSF OAC-1836650 grants and, of course, the FABRIC project itself, which is funded by NSF grants CNS-1935966, CNS-2029176, CNS-2029200, CNS-2029235, CNS-2029260 and CNS-2330891. Furthermore, we acknowledge our collaborations with the CERN IT, WLCG sites, and LHCONE/LHCOPN communities, who also participated in this effort.

References

[1] The ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider,* JINST 3, S08003 (2008). https://doi.org/10.1088/1748-0221/3/08/S08003

[2] T. Maeno, A. Alekseev, M. Barreiro, et al., *PanDA: Production and Distributed Analysis System* (2024), https://doi.org/10.1007/s41781-024-00114-3

[3] B. Galewsky, R. Gardner, L. Gray, M. Neubauer, J. Pivarski, M. Prott, I. Vukotic, G. Watts, and M. Weinberg. *ServiceX A Distributed, Caching, Columnar Data Delivery Service.* EPJ Web Conf., 245:04043, 2020.

[4] IRIS-HEP website. http://iris-hep.org

[5] M. Proffitt and G. Watts, *FuncADL: Functional Analysis Description Language, EPJ Web Conf.*, vol. 251, p. 03068, 2021. DOI: 10.1051/epjconf/202125103068. Available at: https://arxiv.org/abs/2103.02432.

[6] O. Rind, D. Benjamin, L. Bryant, C. Caramarcu, R. Gardner, F. Golnaraghi, C. Hollowell, F. Hu, D. Jordan, J. Stephen, I. Vukotic and W. Yang EPJ Web of Conf., 295 (2024) 07043 DOI: https://doi.org/10.1051/epjconf/202429507043

[7] I. Baldin, A. Nikolich, J. Griffioen, I. Monga, K. Wang, T. Lehman, and P. Ruth. *Fabric: A national-scale programmable experimental network infrastructure.* IEEE Internet Computing 23, no. 6 (2019): 38-47, https://ieeexplore.ieee.org/document/8972790

[8] Dask Development Team (2016). *Dask: Library for dynamic task scheduling* http://dask.pydata.org

[9] T. Douglas, T. Todd, L. Miron (2005). *Distributed Computing in Practice: the Condor Experience*. Concurrency and Computation: Practice and Experience. 17 (2–4): 323–356.

[10] JupyterLab documentation. Accessed on 2025-12-14. https://jupyterlab.readthedocs.io/

[11] Ceph. *Ceph: The Future of Storage*. Retrieved January 27, 2025, from https://ceph.io/

[12] *Helm*, Available at: https://helm.sh/ (Accessed: 2025)

[13] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, *Xrootd - a highly scalable architecture for data access* (2005)

[14] Kubernetes-SIGs. *Kubespray: Deploy a Production Ready Kubernetes Cluster*. Retrieved January 27, 2025, from https://github.com/kubernetes-sigs/kubespray

[15] Kubernetes-SIGs. *FluxCD: GitOps operator for Kubernetes*. Retrieved January 27, 2025, from https://fluxcd.io

[16] Intel Corporation. (2018). *Multus: CNI Implementation in Kubernetes2*. Retrieved January 27, 2025, from https://github.com/k8snetworkplumbingwg/multus-cni

[17] J. Zurawski, B. Brown, G. Rai, E. Dart, C. Dawson, C. Hawk, P. Mantica, S. Margetis, K. Miller, N. Miller, A. Wiedlea (2024). *Nuclear Physics Network Requirements Review (Final Report)*. OSTI.GOV. https://doi.org/10.2172/2386941.ESnetTechnicalOverview

[18] E. Martelli, S. Stancu (2015). *LHCOPN and LHCONE: Status and Future Evolution*. Journal of Physics: Conference Series, 664, 052025. https://doi.org/10.1088/1742-6596/664/5/052025

[19] R. Brun and F. Rademakers. *ROOT An object oriented data analysis framework.* Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers,

Detectors and Associated Equipment, 389(1):8186, 1997. In New Computing Techniques in Physics Research V.

[20] M. Barisits, T. Beermann, F. Berghaus, B. Bockelman, J. Bogado, et al. *Rucio: Scientific Data Management*, Computing and Software for Big Science (2019) 3:11 https://doi.org/10.1007/s41781-019-0026-3