Operating the 200 Gbps IRIS-HEP Demonstrator for ATLAS

Robert W. Gardner Jr[1,*], *Douglas* Benjamin[2], *Lincoln* Bryant[1], *Matthew* Feickert[3], *Farnaz* Golnaraghi[1], *Alexander* Held[3], *Fengping* Hu[1], *David* Jordan[1], *Judith* Stephen[1], *Ilija* Vukotic[1], *Ofer* Rind[2], *Gordon* Watts[4], and *Wei* Yang[5] on behalf of the ATLAS Computing Activity

[1]Enrico Fermi Institute, University of Chicago, Chicago, IL, USA
[2]Brookhaven National Laboratory, Upton, NY, USA
[3]University of Wisconsin-Madison, Madison, WI, USA
[4]University of Washington, Seattle, WA, USA
[5]SLAC National Laboratory, Palo Alto, CA, USA

Abstract. The ATLAS experiment is currently developing columnar analysis frameworks which leverage the Python data science ecosystem. We describe the construction and operation of the infrastructure necessary to support demonstrations of these frameworks, with a focus on those from IRIS-HEP. One such demonstrator aims to process the compact ATLAS data format PHYSLITE at rates exceeding 200 Gbps. Various access configurations and setups on different sites are explored, including direct access to a dCache storage system via Xrootd, the use of ServiceX, and the use of multiple XCache servers equipped with NVMe storage devices. Integral to this study was the analysis of network traffic and bottlenecks, worker node scheduling and disk configurations, and the performance of an S3 object store. The system's overall performance was measured as the number of processing cores scaled to over 2,000 and the volume of data accessed in an interactive session approached 200 TB. The presentation will delve into the operational details and findings related to the physical infrastructure that underpins these demonstrators.

ATL-SOFT-PROC-2025-027
11 February 2025

1 Motivation and Context

The HL-LHC introduces unprecedented challenges in data processing, storage, and access [1–3]. Efforts are underway to develop innovative solutions for high-throughput analysis, addressing both software [4] and infrastructure needs [5]. This study demonstrates an HL-LHC analysis pipeline capable of sustaining a data transfer rate of 200 Gbps between storage and CPU. The analysis was based on a model approximating the processing of a 200 TB dataset within 30 minutes, a projection for 2030 [6]. For 2024, the goal was scaled to 25% of this target, corresponding to the 200 Gbps benchmark. Testing was performed at the UChicago Analysis Facility, part of the U.S. ATLAS Shared Analysis Facility [7], taking advantage of its close proximity to storage servers at the ATLAS Midwest Tier 2 center (MWT2), which is a component of the Worldwide LHC Computing Grid (WLCG) [8].

*e-mail: rwg@uchicago.edu

1.1 UChicago Analysis Facility

The UChicago Analysis Facility (AF) supports diverse analysis workflows for the ATLAS collaboration [9], integrating traditional batch systems like HTCondor with interactive tools such as Jupyter notebooks with GPU support. At its core are approximately 35 "hyperconverged" nodes, each equipped with substantial disk space, memory, and CPU resources, making them well-suited for hosting storage, job slots, and other critical services. The facility also includes four dedicated login nodes and six GPU nodes, alongside additional machines optimized for running Jupyter notebooks, enabling interactive analysis workflows. Co-located with MWT2, the facility employs a flexible Kubernetes-based infrastructure that enables dynamic reconfiguration to adapt to evolving analysis requirements. This Kubernetes foundation facilitates the integration of advanced services like ServiceX [10] and Coffea-Casa [11], as well as other container-based applications. During the 200 Gbps challenge, the facility expanded its resources by adding 75 servers to its original 35 hyperconverged nodes to meet the benchmark's requirements.

2 Shape of the Challenge

As indicated schematically in Figure 1, the challenge was structured around two distinct data workflows: (1) direct data flow from XCache [12] servers to Dask workers using the Uproot [13] and Coffea libraries for high-performance ROOT file handling, and (2) transformed data flow through ServiceX which converts ROOT files into columnar formats for storage in S3-like object stores before ingestion by Dask. The two workflows reflect modern trends in high-energy physics, emphasizing efficient data handling and interoperability between diverse tools and formats. This dual-path approach also provided flexibility to explore bottlenecks inherent in each method.

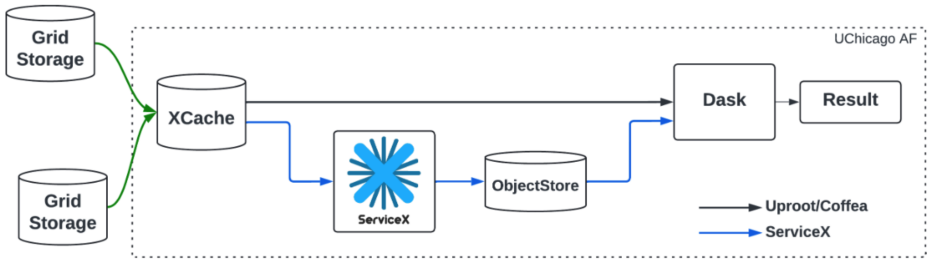

Figure 1. Both processing paths started from a warm XCache. The path that involves ServiceX is expected to have higher scaling limit but requires optimization in splitting resources between ServiceX and Dask workers.

3 Baseline

To establish a performance baseline, a 192 TB data sample in the ATLAS PHYSLITE format [14] was replicated to the dCache storage system at MWT2 using Rucio [15]. The data was distributed across 62 pool nodes, each connected at either dual 10 Gbps or dual 25 Gbps to the local area network. Care was taken to place this data on newer storage servers, chosen for their superior read performance. Initial tests focused on measuring raw data read rates using the XRootD `xrdcp` client utility [16] writing to /dev/null. These tests achieved an aggregate throughput of nearly 300 Gbps (storage to client) while concurrently supporting

ATLAS production workloads, Figure 2. The test setup employed 250 `xrdcp` clients distributed across the Analysis Facility (AF) cluster worker nodes. The results confirmed that the storage system and network infrastructure were capable of supporting the benchmark's aggregate throughput target for simple read operations from storage, ensuring readiness for further high-throughput analysis benchmarks. However, the shape of the analysis challenge, with its focus on using "nearby" caches, was not as topologically simple as we discuss below.

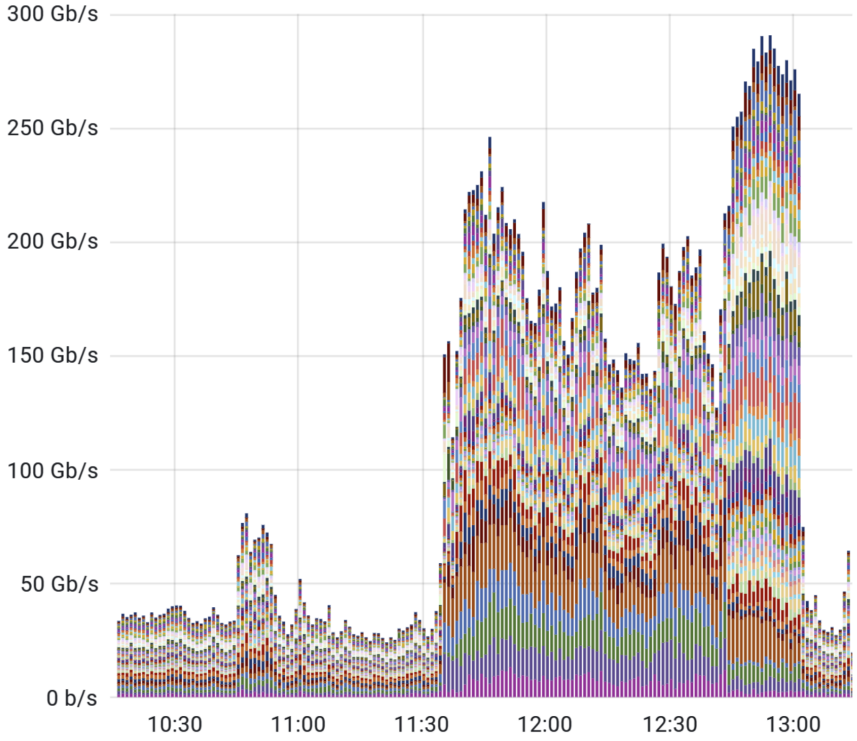

Figure 2. Aggregate throughput between AF workers and MWT2 dCache storage servers. Each color represents the read rate from an individual dCache pool. Most testing occurred during a 90-minute period, where competition for client workers (due to HTCondor scheduling on the AF) and shared usage with MWT2 production I/O resulted in the observed variability.

4 Bottlenecks

The demonstrator called for caching rather than directly reading from the MWT2 dCache system since future analysis facilities may not be co-located with a large storage facility and performing analysis over wide area networks introduces additional challenges. The AF initially employed only a single XCache server with a 50 Gbps network connection for its physics community. This was obviously going to be the first bottleneck. To remove it, the cache capacity was expanded to eight servers, each equipped with ten 3.2 TB NVMe drives configured as JBOD arrays. With dual 25 Gbps interfaces, which we bonded, the system thus was capable of a total bandwidth of 400 Gbps for read (egress) traffic from "fast" storage. After deploying these additional servers we conducted another round of low-level read tests, this time targeting the caches, and again using simple `xrdcp` clients from AF workers. The tests utilized 85 AF nodes, with each node running one `xrdcp` client for each cache (a total of

85 × 8 clients). Figure 3 shows that these tests achieved a peak aggregate throughput of 350 Gbps, comfortably above the benchmark requirement of 200 Gbps. However, initial runs of the demonstrator pipelines (both data flow paths) yielded throughput measurements far below that target and thus "the network" drew further scrutiny.

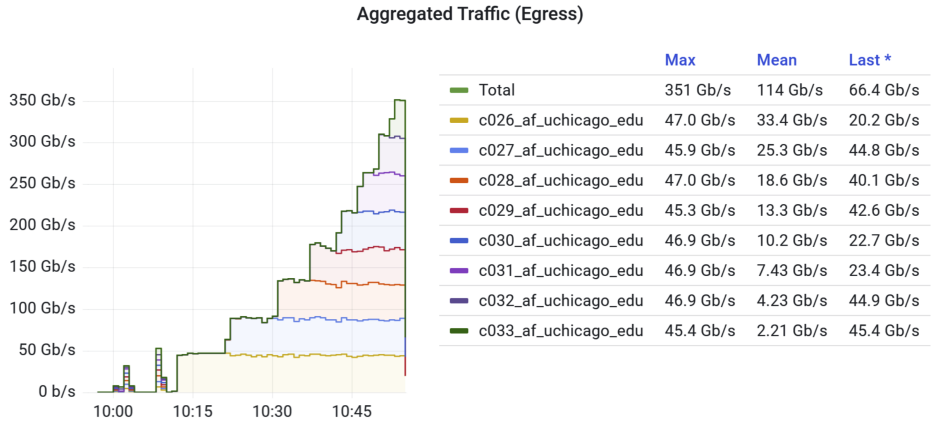

Figure 3. Network egress traffic for each XCache server during simple read tests.

4.1 Caching and Network Topology

The network topology, specifically the relative placement of workers and caches within the AF, was a critical factor in achieving high throughput. A detailed examination of all host and switch uplink interfaces for errors and I/O rates revealed that the placement of caching servers within the AF worker node racks was the primary bottleneck. These caching servers, originally repurposed worker nodes with large, fast disk volumes, were connected to top-of-rack switches with uplinks to the spine network limited to 80 Gbps, as shown in Figure 4. This configuration resulted in multiple caching servers on the same top-of-rack switch competing for limited bandwidth when serving clients distributed across the cluster. To address this issue, the XCache nodes were redistributed across multiple top-of-rack switches, which localized traffic and reduced cross-rack contention. While this redistribution partially mitigated the bottleneck, it underscored the importance of cache-aware network planning for future deployments. File distribution for the entire 192 TB dataset across the eight caching servers was managed by Rucio, which used filename hashing to assign files to specific XCache nodes. This file placement information was utilized by both Dask and ServiceX clients during dataset reads, ensuring efficient access paths for their respective workflows.

4.2 Cluster Optimizations

Running a production analysis facility supporting hundreds of users alongside the 200 Gbps challenge posed significant challenges. Resource contention arose as Dask workers prioritized Kubernetes job slots over HTCondor workloads. To alleviate this, 75 MWT2 worker nodes were allocated to the AF, adding 3,000 hyperthreaded cores. A Kubernetes Horizontal Pod Autoscaler [17] was configured to dynamically balance HTCondor and Dask workloads, ensuring equitable resource allocation. Additional operational tuning was necessary to address performance issues during the challenge. The Kubernetes `etcd` database [18] required reconfiguration to handle the large-scale launch of Dask workers. Additionally, networking

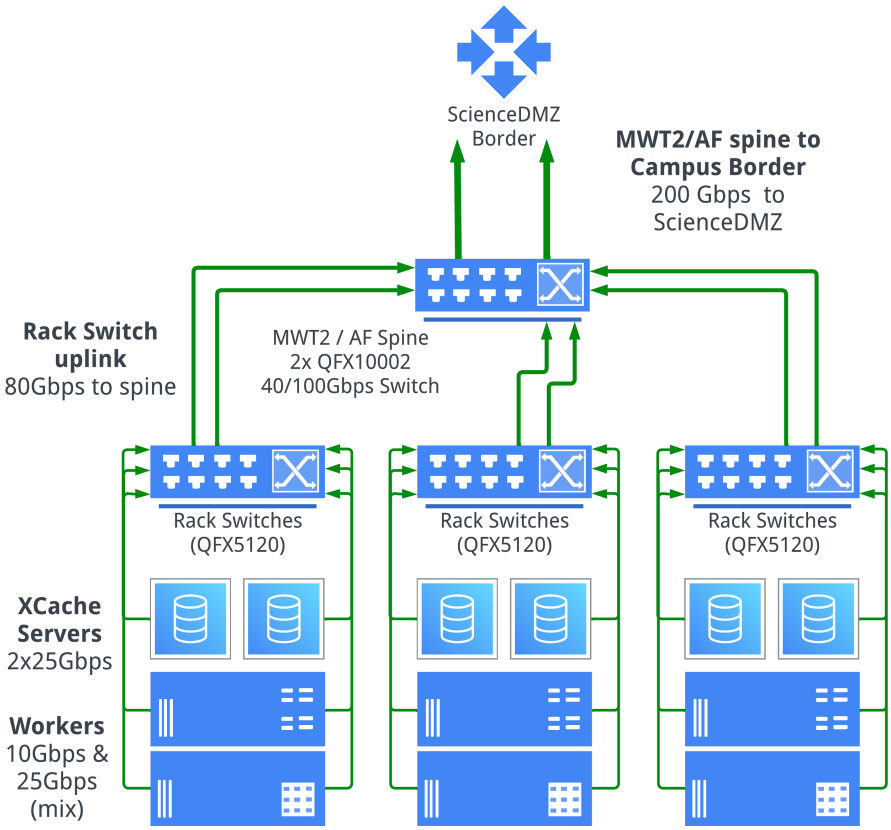

Figure 4. The AF network topology during the 200 Gbps demonstrator showing the relative placement of caches and worker nodes within the network. Juniper model numbers and link capacities are shown.

inefficiencies, such as suboptimal MTU settings and DNS resolver timeouts, were identified and partially mitigated. These adjustments demonstrated the complexity of operating a high-performance analysis system at scale. Network bottlenecks described above were a recurring theme, with significant traffic passing through multiple layers of indirection, including Kubernetes ingress controllers and load balancers. Two potential solutions were proposed: (1) developing smarter, rack-aware services to optimize traffic flows and (2) simply upgrading the network infrastructure to eliminate bottlenecks. The latter was identified as the more practical short-term solution.

4.3 Object Store Bottlenecks

The UChicago Analysis Facility (AF) employs a Rook-based Ceph storage system [19] within its Kubernetes cluster to meet diverse storage requirements, including POSIX-compliant file systems, block devices, and S3-compatible object stores. For the 200 Gbps challenge, the RADOS Gateway (RADOSGW) was critical, managing the S3-like object storage utilized by ServiceX and Dask. The RADOSGW relied on an all-NVMe disk pool with 3x replication, which ensured data redundancy but incurred significant write amplification.

Initially, a single RADOSGW instance was deployed behind an Ingress controller for TLS termination. This configuration quickly became a bottleneck, funneling all ServiceX

traffic through a single server. A high volume of *503 Slow Down* errors indicated that system components were overloaded under peak loads. Moreover, the reliance on a single ingress point and MetalLB's Layer 2 mode [20] for traffic distribution further constrained throughput. Finally, the limited number of Ceph placement groups (32) resulted in poor data distribution across the storage pool, exacerbating performance bottlenecks.

To address these challenges, several steps were implemented, with before and after configurations depicted in Figure 5:

1. **Multiple RADOSGW Instances:** A total of 19 additional RADOSGW instances were deployed across 35 nodes, each equipped with either 25 Gbps or 10 Gbps connectivity, to better handle the increased S3 traffic.

2. **Networking Configuration Update:** The system was reconfigured to use `NodePort`, enabling direct access to individual RADOSGW instances and bypassing the central Ingress controller, thereby reducing network contention.

3. **Increased Placement Groups:** The number of Ceph placement groups was increased from 32 to 512, significantly improving data distribution across the Ceph pool and reducing data skew.

These adjustments alleviated many of the performance bottlenecks, improving storage system efficiency. However, the 3x replication factor remained a challenge, amplifying network and storage loads during high-throughput operations.

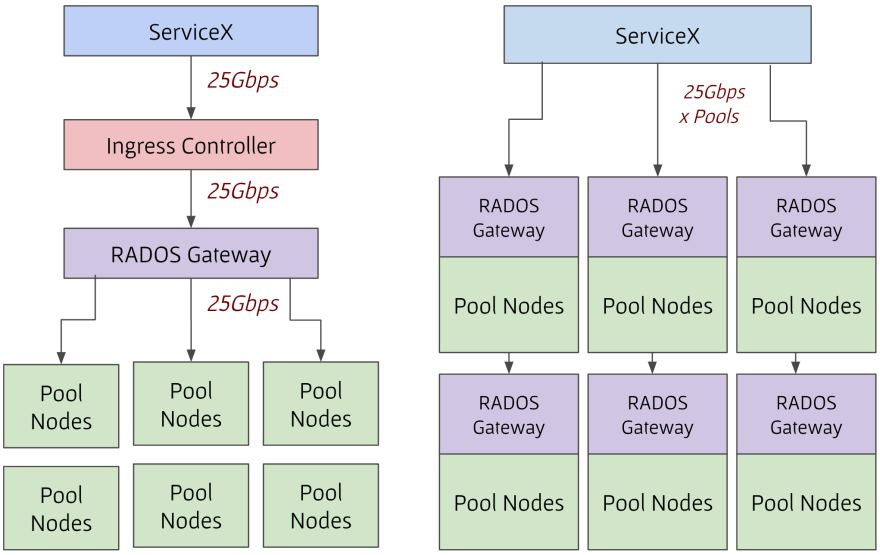

Figure 5. Deployments of RADOS Gateways before (left) and after optimizations (right).

5 Results and Summary

The results for the two pipeline methods after these facility reconfigurations and service deployments are shown in Figure 6. The demonstrator successfully achieved the 200 Gbps

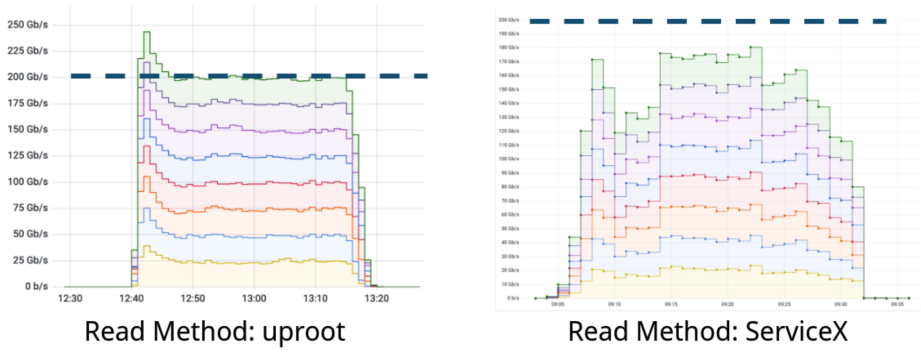

Figure 6. Aggregate read I/O traffic from from the two pipeline methods: Using Dask and Uproot reading from XCaches (left), and ServiceX reading from caches and object store (right).

benchmark target for direct Dask-cache reading, whereas the path using ServiceX with object storage approached a plateau value of 170 Gbps.

Looking ahead, we are interested to explore further optimizations of the object store for ServiceX output, including eliminating the 3x replication, which is unnecessary for these transient datasets. We are additionally exploring reorganizing the placement of caches within the existing network, and upgrading with 400/100 Gbps capable switches.

This work was supported in part by the National Science Foundation awards PHY-2120747, OAC-2115148, OAC-2029176, OAC-1836650 and PHY-2323298.

References

[1] R. Gardner, *Computing at the HL-LHC and beyond, PoS (LHCP2024) 333*, https://pos.sissa.it/478/333/pdf (2024)

[2] ATLAS Collaboration, *ATLAS Software and Computing HL-LHC Roadmap (2022)*, https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/UPGRADE/CERN-LHCC-2022-005

[3] O. Gutsche, T. Bose, M. Votava, D. Mason, A. Melo, M. Liu, D. Hufnagel, L. Gray, M. Hildreth, B. Holzman et al., *The U.S. CMS HL-LHC R&D Strategic Plan, EPJ Web of Conf. 295, 04050 (2024)*, https://doi.org/10.1051/epjconf/202429504050

[4] D. Shope, *Software Upgrades for High-Luminosity LHC, PoS (LHCP2024) 193*, https://pos.sissa.it/478/193/pdf (2024)

[5] D. Ciangottini, A. Forti, L. Heinrich, N. Skidmore, C. Alpigiani, M. Aly, D. Benjamin, B. Bockelman, L. Bryant, J. Catmore et al., *Analysis Facilities White Paper*, https://arxiv.org/abs/2404.02100 (2024), `arXiv:2404.02100 [hep-ex]`

[6] A. Held, B. Bockelman, O. Shakdura, *The 200 Gbps Challenge: Imagining HL-LHC analysis facilities*, https://indico.cern.ch/event/1338689/contributions/6009824/ (2024)

[7] O. Rind, D. Benjamin, L. Bryant, C. Caramarcu, R. Gardner, F. Golnaraghi, C. Hollowell, F. Hu, D. Jordan, J. Stephen et al., *The Creation and Evolution of the US ATLAS Shared Analysis Facilities, EPJ Web of Conf. 295, 07043 (2024)*, https://doi.org/10.1051/epjconf/202429507043

[8] I. Bird, P. Buncic, F. Carminati, M. Cattaneo, P. Clarke, I. Fisk, M. Girone, J. Harvey, B. Kersevan, P. Mato et al., *Update of the Computing Models of the WLCG and the LHC Experiments (CERN-LHCC-2014-014)*, https://cds.cern.ch/record/1695401 (2014)

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

[10] K. Choi, A. Eckart, B. Galewsky, R. Gardner, M.S. Neubauer, P. Onyisi, M. Proffitt, I. Vukotic, G.T. Watts, *Towards Real-World Applications of ServiceX, an Analysis Data Transformation System, EPJ Web Conf. 251, 02053* (2021), https://doi.org/10.1051/epjconf/202125102053

[11] M. Adamec, G. Attebury, K. Bloom, B. Bockelman, C. Lundstedt, O. Shadura, J. Thiltges, *Coffea-casa: an analysis facility prototype, EPJ Web of Conf. 251, 02061* (2021), https://doi.org/10.1051/epjconf/202125102061

[12] A. Hanushevsky, H. Ito, M. Lassnig, R. Popescu, A.D. Silva, M. Simon, R. Gardner, V. Garonne, J.D. Stefano, I. Vukotic et al., *Xcache in the ATLAS Distributed Computing Environment, EPJ Web of Conf. 214, 04008* (2019), https://doi.org/10.1051/epjconf/201921404008

[13] J. Pivarski, P. Das, C. Burr, D. Smirnov, M. Feickert, T. Gal, L. Kreczko, N. Smith, N. Biederbeck, O. Shadura et al., *scikit-hep/uproot3: 3.14.4* (2021), https://doi.org/10.5281/zenodo.4537826

[14] ATLAS Collaboration, *PHYSLITE (ATLAS Open Data)*, https://opendata.atlas.cern/docs/documentation/data_format/physlite/ (2024)

[15] V. Garonne, R. Vigne, G. Stewart, M. Barisits, T. Beermann, M. Lassnig, C. Serfon, L. Goossens, A. Nairz, on behalf of the ATLAS Collaboration, J. Phys: Conf. Ser. **513**, 042021 (2014)

[16] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, *XROOTD/TXNetFile: a highly scalable architecture for data access in the ROOT environment*, in *Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics* (World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2005), TELE-INFO'05, ISBN 9608457114

[17] *Horizontal Pod Autoscaling*, https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/

[18] *etcd*, https://etcd.io/

[19] *ROOK: Open-Source, Cloud-Native Storage for Kubernetes*, https://rook.io/

[20] *MetalLB is a load-balancer implementation for bare metal Kubernetes clusters, using standard routing protocols*, https://metallb.io/