

Evolution of the open-source data management system Rucio for LHC Run-3 and beyond ATLAS

Martin Barisits^{1,*}, *Thomas Beermann*², *Joaquin Bogado*³, *Vincent Garonne*⁴, *Tomas Javurek*¹, *Mario Lassnig*¹, *Andrea Manzi*¹, *Edoardo Martelli*¹, *Cedric Serfon*², *Tobias Wegner*¹, and *Xin Zhao*⁵ on behalf of the ATLAS Collaboration

¹European Organization for Nuclear Research (CERN), Switzerland

²University of Innsbruck, Austria

³National University of La Plata, Argentina

⁴University of Oslo, Norway

⁵Brookhaven National Laboratory (BNL), USA

Abstract. Rucio, the distributed data management system of the ATLAS experiment already manages more than 400 Petabytes of physics data on the grid. Rucio was incrementally improved throughout LHC Run-2 and is currently being prepared for the HL-LHC era of the experiment. Next to these improvements the system is currently evolving into a full-scale generic data management system for application beyond ATLAS, or even beyond high-energy physics. This contribution focuses on the development roadmap of Rucio for LHC Run-3, such as event level data management, generic meta-data support and increased usage of networks and tapes. At the same time Rucio is evolving beyond the original ATLAS requirements. This includes additional authentication mechanisms, generic database compatibility, deployment and packaging of the software stack in containers, and a project paradigm shift to a full-scale open source project.

1 Introduction

Managing large volumes of research data is a major challenge for any scientific project or experiment. The data requirements of these experiments are ever growing, leading to an unprecedented amount of required storage space and data organisation. These storage systems are typically heterogeneous and are distributed at multiple geographical locations under different administrative domains. The data workflows involved in managing and analyzing the data are also increasing in complexity, as the scientific data is usually produced, stored, and analyzed in multiple locations.

The ATLAS Experiment [1] at the Large Hadron Collider (LHC) [2] at CERN [3] is a typical example for such massive data requirements. To manage these large amounts of data, ATLAS developed the distributed data management system Rucio [4], which is a service taking care of the full data life cycle of the experiment. Rucio is now used in production at ATLAS, AMS [5], and Xenon1T [6] and is under evaluation by more scientific collaborations.

*e-mail: martin.barisits@cern.ch

In this article we describe the evolution of Rucio to a generic, open-source, data management system for application beyond ATLAS. This article is structured as follows: In Section 2 we present the generic metadata support recently added to Rucio. We continue in Section 3 describing the plans for event level data management. Section 4 presents a preliminary study and plans about the increased usage of tape based storage systems and the support needed from the data management system. In Section 5 we present the plan for introducing new authentication and authorization methods beyond the current WLCG standard with X.509 certificates and proxies. Section 6 continues with new ways of deploying Rucio, specifically on Kubernetes, as well as the evolution of Rucio to a full-stack open source project. The article concludes in Section 7 with a summary.

2 Generic metadata support

Rucio already supports a limited set of metadata that can be associated with each data identifier. This includes system internal metadata (timestamps, states, access count, ...), data-management metadata (checksums, file sizes, dataset length, ...) as well as a set of pre-defined user metadata (number of events, lumiblocknumber, task id, campaign, ...). However, with the increased usage of Rucio in other experiments, even outside of the high-energy physics domain, it became very apparent that a more flexible way of managing metadata is needed. This is especially true for smaller experiments that do have the possibility to operate a dedicated metadata catalog next to their data management system.

This development was guided by two design principles: The system must be able to add generic metadata to each data identifier. Thus, no pre-defined columns or data structures limiting the metadata are allowed. Secondly, registering a new data identifier including metadata should be an atomic operation. Thus it should not be possible that one part of the operation, i.e., registration of the data identifier or metadata, fails while the other is successful. Due to both requirements the design decision was made that the metadata will reside in the same relational and transactional database system as the logical data identifiers. This was mostly made possible due to recent advancements in Oracle [7], MySQL [8], and PostgreSQL [9] which allow the storage of arbitrary JSON datatypes and search operations within them. Thus the metadata associated to a data identifier is transformed into a JSON structure and stored to a JSON-type column within the relational database.

The development was done within Google Summer of Code 2018 [10] framework as part of the CERN & HSF project. APIs were created which allow a user to `get`, `add/update`, `delete` metadata for a specific data identifier. Also a command to `search` for data identifiers, matching a specific metadata filter within a scope was added. This new generic metadata feature is supported with Oracle 12c, PostgreSQL 9.3, and MySQL 5.7.8 and available with the Rucio 1.18.0 release.

3 Event level data management

The workflows of accessing data on the grid evolved during the last years. Today, particle physics production workflows access very specific physics events within files instead of entire files. This optimized and fine grained access of data will become even more prominent with LHC Run-3 and HL-LHC. This evolution is also expressed in recent developments with the ATLAS Event Service [11].

This trend raises the question when data objects within a file, such as events, become the common way to access scientific data, and if or how these objects should also have a dedicated representation within Rucio.

Tape Drives used	Data sample	Average Tape (re)mounts	Average Tape throughput	Stable Rucio throughput	Test Average throughput
31 LTO6/7 drives	200TB; 110k files; ~1.8GB/file	2.6 times	1~2.5GB/s	866MB/s	545MB/s (47TB/day)
8 T10KC/D drives	178TB; 92k files; ~1.9GB/file	>20 times	~400MB/s	300MB/s	286MB/s (25TB/day)
2 T10KD drives	125TB; 57K files; ~2.2GB/file	Majority tapes mounted once	277MB/s	300MB/s	255MB/s (22TB/day)
5~6 T10KD drives	90TB; 50k files; ~1.8GB/file	Some outliers (>40 times)	500MB/s	[2] 380MB/s	400MB/s (35TB/day)
11 LTO7 drives	186TB; 91k files; ~2GB/file	Very low (near 0) remounts	1.1GB/s	1GB/s	700MB/s (60TB/day)
[3]36 T10KD drives	204TB; 84k files; ~2.4GB/file	~5.33 times	2.2GB/s	3GB/s	2.1GB/s (180TB/day)
10 T10KD drives	195TB; 69k files; ~2.8GB/file	2.6~4.8 times	500~700MB/s	640MB/s	630MB/s (54TB/day)
10 T10KD drives	225TB; 106k files; ~2.1GB/file	n/a	1.6GB/s	2GB/s	1.6GB/s (138TB/day)

Table 1. Chart displaying the different performance characteristics of the measured tape systems.

There are essentially two options to realize this within Rucio. One is to expand the data identifiers (files, datasets, containers) to events and make sub-file-objects an official data identifier of Rucio. This is a major design change and would require changes throughout the full architecture stack. The advantage would be that users can operate on events similar to other data identifiers, thus add metadata to them, download them, or add replication rules. Next to the development effort the implications to the database layer would have to be evaluated, as storing sub-file information could result in one or two orders of magnitude more database objects.

The second option is to connect Rucio to an event catalog, such as the ATLAS Event Index [12]. This would allow users to use Rucio tools to download specific events, as Rucio resolves the events via the APIs of the Event Index and then accesses the specific byte offsets of the file on storage. The development effort for Rucio is significantly smaller for this, but besides downloading specific events, the possibilities of interacting with events are only very limited.

It is not clear yet which option will be implemented, as it is mostly dependent on available development power and development plans of related projects.

4 Increased usage of tapes and networks

Rucio is prepared for an increased usage of tape in the experiment and is supporting the wider ATLAS Distributed Computing (ADC) *tape carousel* activity. The goal of this first study within the tape carousel activity is to establish baseline measurements of current tape capacities, to know how much throughput we can expect from our current tape sites, stress-test to help optimize system settings, and identify bottlenecks. This study thus serves as the beginning of a decision process for the future developments and improvements required in Rucio with respect to hierarchical storage systems.

The actions of the iterative process are straightforward: run test, identify bottleneck, improve, repeat. The full test was run through Rucio, requesting the staging from tape to local disks at the sites. The data sample contained about 200 Terabytes of AOD datasets, with an average file size of 2GB. Replicas of the data that already existed on disk were manually cleaned beforehand.

The actual operation was then done in bulk mode, with three sites requesting a throttle of 2000 incoming staging requests. Two sites also had concurrent activities with production tape writing/reading and other experiment activity, but not at a significant level.

The tape frontend was quickly identified as the current bottleneck by limiting the number of incoming staging requests which were passed to the backend tape. This also limited the number of files which could be retrieved from tape disk buffer, and the number of files to transfer to the final destination. The requests which timed out were then rescheduled by Rucio, and eventually all requests were done. Figure 1 shows the summary of the individual tests.

Another problem that was identified was the size of the disk buffer on the tape pool servers, because dCache [13] reserves disk space before sending requests to backend tape and some sites may not have enough disk space in order to pass all requests to tape. The hardware solution is to increase this disk space to match the expected throughput from tape. Since we have not yet identified the full workflow for the tape carousel activity, this is unrealistic. The software solution is to loosen the reservation requirement or make it configurable and will be discussed with dCache developers for a stop-gap solution. Since we cannot expect sites to increase hardware an alternative can be applied from the Rucio side. Rucio has the capability to throttle transfer requests, and this could be extended to tape systems with knowledge about the buffer size. That way we can ensure that custom throttling can be applied per endpoint.

Some pool servers also got overloaded with thousands of requests, it is unclear yet why, but since dCache keeps track of status of all the staging requests submitted to the backend tape, the retrievals from tape, once they are ready, and eventually transfer them to the final destination, a capacity upgrade of the number of pool servers seems almost mandatory. A potential software improvement could be in the mechanism of tracing requests and retrieving them when ready only.

Another finding was the distribution of requests among pool servers, where requests got dispatched to dCache pool servers that have more space. Eventually requests got stuck on low performing servers, although they have more space. Manual intervention was needed to redistribute the requests to other servers. There's a clear software improvement possible in the way requests can be dispatched among pool servers and automatically recovered when stuck.

For writing to tape, good throughput was seen from sites who repack or organise before writing to tape. This was the major reason for performance difference between sites with similar system settings. Since this requires writing in the way one wants to read later, it is not straightforward, especially if there are multiple workflows to support. Even though tape systems provide such file families and most sites use it, a grouping by experiment dataset seems more convenient and efficient. For full tape reading, near zero remounts were observed with sites that are already doing that. Finally, ADC is also working on bigger file size with a target of 10 Gigabytes as this will greatly increase tape writing throughput.

Next to tapes, another avenue for R&D is the network being used by Rucio. Especially the orchestration of transfers across wide area networks has received little attention yet due to the overabundance of network capacity. We foresee a major increase of network use therefore several activities have started to evaluate the use of novel network features. At the time of writing we have recently finished the first preliminary test of *bandwidth load balancing*.

The goal of the test was to overload the LHCOPN [14] links between CERN and SARA-NIKHEF with fake Rucio file transfers, and then use manual router loadbalancing to offload traffic from LHCOPN to LHCONE to increase the overall available bandwidth. We want to verify that the offload is effective and thus it is possible to use spare bandwidth when the primary links are congested. Eventually, the goal is that this router loadbalancing will be triggered by Rucio automatically.

We identified a data sample of 30 Terabytes, containing 4500 files, and estimated roughly 4 hours transfer time. Shortly after the transfer start the two links were fully used at 10Gbps each, however after half an hour the throughput went down to 60 percent without having activated the additional bandwidth. It was quickly discovered that the source storage system was badly configured in their loadbalancing algorithm which caused hotspots on the disk pools. After the intervention, the throughput increased again to 80 percent usage.

Unfortunately it was not possible to saturate the links, therefore the activation of the additional bandwidth did not show any increase of overall transport throughput, since the primary link could not be overloaded. The destination storage did not exhibit any abnormal characteristics in the meantime. Another unexpected fact was that most of the traffic was on IPv6 and not IPv4, which required some fixing of the routing tables configuration on the CERN outgoing routers.

Eventually the test was aborted and will be repeated in the future with greatly scaled out source storage to ensure link overload. Having this alternate path mechanism automatically triggered through scheduling decisions in Rucio promises to be extremely beneficial for large flows which are typical for experiment data export.

5 New authentication mechanisms

Across the WLCG the primary method of authentication and authorisation are X.509 certificates and their derivative proxies. While X.509 certificates are an industry standard they come with administrative and technological processes, such as certification authorities, which can be inhibiting to new experiments, smaller data centres, as well as sometimes the software itself. Keeping certificate authentication and authorisation synchronised across an experiment is also a source of major configuration and operational troubles. Within Rucio, the X.509 layer is one of the primary supported method next to username/passwords, Kerberos, and SSH public key exchange. To support a wider community of experiments new ways of authentication are in development, most importantly capability-based tokens and OAuth2/OpenID workflows. Both are expected to go into user testing during 2019.

Macaroons [21] are similar to HTTP cookies with several added benefits: while they can be delegated like cookies, they allow cryptographically-secure attenuation to restrict capabilities, and most importantly can be verified by a global generic verifier. Macaroons can thus be seen as the authorisation equivalent of symmetric cryptography. Alternatively, SciTokens [15] take the opposite approach, where instead of restricting from a full credential, claims are added to the bearer token giving additional rights. Verification follows the asymmetric cryptography approach, where the verification keys are publicly available.

The OpenID authorisation workflow built on top of OAuth2 is de-facto required for seamless integration with other federated infrastructures such as EduGAIN, and also serves as the distribution mechanism for the Macaroons and SciTokens. Rucio would create the necessary bearer token of the appropriate type depending on the request of the client. The tokens can then be handed off to FTS [22], which would use the restrictions or claims to authorise access to storage. Along the same lines, the tokens can be distributed to users who can then interact directly with the storage. Rucio could thus take the role of what is the certification authority

in X.509 environments, saving a full layer of authorisation and authentication and reducing individual node management complexity.

6 Deployment and open source software development

Rucio is available on PyPi [16] in three different packages: clients, webUI and a general package including server and daemons. On Docker Hub [18] we provide different containers for server, daemons, clients, and webUI which are heavily used in different deployments. The supported Python versions for the clients are 2.6, 2.7 and 3, while the server package is Python 2.7 compatible. Full-Stack Python 3 compatibility is planned for early 2019.

Current deployment of Rucio for ATLAS relies on Puppet to deploy Rucio Python packages and configurations on a multitude of different virtual machines. A large number of machines is used to provide redundancy and to separate different workflows, e.g., use different servers for the WMS and users. This has proven to work well in the ATLAS use case but it also results in a low utilization of the VMs and a complicated deployment for new experiments. For this reason a Kubernetes [17] deployment is currently evaluated. Kubernetes is an orchestration system for automated deployment, scaling and management for containerized application. For Rucio two separate containers for the server and daemons components are available. The containers are automatically build with every new release and openly available on Docker Hub [18]. They are providing an easy way to run Rucio in a controlled environment without the need to manually install Python packages with possible dependency issues. On top of that Kubernetes simplifies the deployment and scaling of those containers. Kubernetes usually runs on a set of nodes where one is the master and the rest are minions. The minions run the actual containers and the master is used to evenly distribute the work on the minions. The operator has to define a set of different Kubernetes resources to run the Rucio servers and daemons and to also make the servers available for outside access. Large parts of the resource setup can be reused and here is where Helm [19] comes in. It is a package manager for Kubernetes similar to apt on Debian or yum on Red Hat. Packages for the servers and the daemons are available on Github [20]. These charts (packages) can be installed with one simple command and the operator only has to provide a configuration file including information like the DB connection string, server/daemons configurations and the number of containers to run per service.

As mentioned before, this type of deployment is currently evaluated. There is a test cluster for ATLAS available that currently runs two integration servers and one integration daemon (Rule evaluator). Furthermore, a complete new instance of Rucio has been setup only using Kubernetes/Helm for the WLCG Data Organization Management Access (DOMA) third party copy (TPC) working group to initiate a constant flow of TPC transfers between test endpoints. This cluster runs in a stable manner providing valuable operational experience that will be useful later for production instances.

The Rucio project is organized as an open-source development project, under Apache 2.0 license, with contributions coming from 30 different contributors, with 19 contributors from the ATLAS collaboration. The establishment of the project as a community project, within the high-energy physics domain and beyond, was culminated with the hosting of the first Rucio Community Workshop in March 2018. Over 90 participants coming from 16 different communities participated in the workshop and presented their use cases and data management needs. The development is conducted as a formal software engineering process with the objective to produce performant, well-documented, scalable, and sustainable code. The Rucio development team conducts weekly development meetings with issues and development plans well documented on Github [23]. There is human code review of all pull requests and automatic unit testing with more than 400 unit tests with different Python version against

different database backends. We plan to foster this community approach by organizing yearly community workshops and fully focusing on the open-source development process of the software.

7 Summary & Conclusion

Managing large volumes of scientific data is a major challenge for experiments. Rucio provides a well established solution for these problems and has a demonstrated track-record of stability, scalability, usability, and performance. In this article we present the evolution of Rucio as an open-source data management system for LHC Run-3 and beyond ATLAS. The current development plans focus on the features needed for LHC Run-3. However, due to the recent interest of research experiments within the HEP community and beyond, plans also include features relevant for these new communities. Generic metadata support is one of these features mostly relevant for new communities. Previously, Rucio only supported a limited amount of metadata, but with this new development, based on JSON columns in state of the art relational databases, Rucio fully supports generic metadata within the transactional system. Event level data management is a planned development for LHC Run-3 and HL-LHC. The design plan foresees two options of including event information into Rucio and therefore making events directly addressable within Rucio. A first study for the increased usage of tapes and networks in the *tape carousel* activity have shown great promise of more widespread use of tapes. Rucio is fortunately already prepared and can serve as the source for efficient tape orchestration through site distribution and dataset organisation. The first network routing tests also showed promise even though we were not able to saturate the links due to source storage overload. These tests will be repeated in the near future. The current industry standard to authenticate and authorize users to the data management system and to storage are X.509 certificates. To support a wider community of experiments, new ways of authentication are in development, most importantly capability-based tokens and OAuth2/OpenID workflows. Both are expected to go into user testing during 2019. Rucio is available on the Python Package Index and as containers on Docker Hub. Recently Kubernetes and Helm deployments are being evaluated with the objective of supplying a turn-key deployment for new experiments via Helm. Rucio is organised as an open-source project, under Apache 2.0 license, and recently fully embarked on the community path by hosting the first Rucio Community Workshop. The development team is now adapting to this community efforts to open up the development process and integrate use cases and developments beneficial to other experiments.

References

- [1] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, Journal of Instrumentation **3**, S08003 (2008)
- [2] L Evans and P Bryant (editors), *LHC Machine*, JINST **3** S08001 (2008)
- [3] European Organization for Nuclear Research (CERN), <http://cern.home> [accessed 2018-11-27]
- [4] Rucio – Scientific Data Management, <https://rucio.cern.ch> [accessed 2018-11-27]
- [5] M Aguilar, G Alberti, B Alpat et al, *First Result from the Alpha Magnetic Spectrometer on the International Space Station*, Physical Review Letters **110**, 141102 (2013)
- [6] E Aprile, J Aalbers, F Agostini et al, *Dark Matter Search Results from a One Tonne x Year Exposure of XENON1T*, arXiv preprint arXiv **1805**, 12562 (2018)

- [7] Oracle Database Technologies, <https://www.oracle.com/database/technologies/index.html> [accessed 2018-11-27]
- [8] MySQL, <https://www.mysql.com/> [accessed 2018-11-27]
- [9] PostgreSQL, <https://www.postgresql.org/> [accessed 2018-11-27]
- [10] Google Summer of Code, <https://summerofcode.withgoogle.com/> [accessed 2018-11-27]
- [11] P Calafiura, K De, W Guan, T Maeno, P Nilsson, D Oleynik, S Panitkin, V Tsulaia, P Van Gemmeren, T Wenaus, *PanDA: Distributed Production and Distributed Analysis System for ATLAS*, Journal of Physics: Conference Series **664**, 062065 (2015)
- [12] D Barberis, S Zárate, J Cranshaw, A Favareto, A Casaní, E Gallas and others, *The ATLAS EventIndex: architecture, design choices, deployment and first operation experience*, Journal of Physics: Conference Series **664**, 042003 (2015)
- [13] dcache, <https://www.dcache.org/> [accessed 2018-11-27]
- [14] E Martelli, S Stancu, *LHCOPN and LHCONE: Status and Future Evolution*, Journal of Physics: Conference Series **664**, 052025 (2015)
- [15] A Withers, B Bockelman, D Weitzel, D Brown, J Gaynor, J Basney, T Tannenbaum, Z Miller, *SciTokens: Capability-based secure access to remote scientific data*, arXiv preprint arXiv **1807**, 04728 (2018)
- [16] Python Package Index, <https://pypi.org/> [accessed 2018-11-27]
- [17] Kubernetes, <https://kubernetes.io> [accessed 2018-11-27]
- [18] Rucio Docker Containers on Docker Hub, <https://hub.docker.com/r/rucio/> [accessed 2018-11-27]
- [19] Helm - Kubernetes Package Manager, <https://helm.sh/> [accessed 2018-11-27]
- [20] Rucio Helm Repository, <https://github.com/rucio/helm-charts> [accessed 2018-11-27]
- [21] A Birgisson, J Politz, Gibbs, Erlingsson, Ulfar, Taly, Ankur, Vrable, Michael, M Lentzner, *Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud* (NDSS, 2014)
- [22] A Ayllon, M Salichos, M Simon et al, *Integrating network and transfer metrics to optimize transfer efficiency and experiment workflows*, Journal of Physics Conference Series **513**, 032081 (2014)
- [23] Git Hub, <https://www.github.com> [accessed 2018-11-27]