Transitioning CMS to Rucio Data Managment

Eric Vaandering[1],*

[1]Fermilab, Batavia, IL 60510, USA

Abstract. Following a thorough review in 2018, the CMS experiment at the CERN LHC decided to adopt Rucio as its new data management system. Rucio is emerging as a community software project and will replace an aging CMS-only system before the start-up of LHC Run 3 in 2021. Rucio was chosen after an evaluation determined that Rucio could meet the technical and scale needs of CMS. The data management system for CMS needs to manage the current data sample of approximately 200 PB of data with 1 PB of transfers per day. The data management system must also have a development path suitable for LHC Run 4 (2026) data rates, which are expected to be 50 times larger.

This contribution details the ongoing CMS adoption process as we replace our legacy system with Rucio, focusing on the challenges of integrating Rucio into an established set of experimental tools and procedures. This will include the migration of metadata, the construction of interfaces to the rest of the CMS computing tools, scale tests, operations, monitoring, and the plan to gradually turn over primary responsibility for the management of data to Rucio. A description of CMS user data management with Rucio will also be provided.

1 Introduction

The current CMS [1] data management system, consisting of PhEDEx [2] and Dynamo [3], is able to meet CMS's needs. However, the software is aging and was custom developed by CMS. Data management functionality is split between PhEDEx, which is responsible for ensuring that data is moved and placed where requested, and Dynamo, which makes higher level decisions about which data is needed, how many copies are needed, and which data can be safely deleted. In 2018, CMS identified a replacement of this crucial piece of software as a priority for CMS and a down-select review was held.

To replace PhEDEx, a new system needed to be capable of managing the CMS data set of about 100 PB on tape and 50 PB on disk spread across approximately 100 storage sites. The ability to transer over 2 PB per day, as the current system does, was also a requirement. Furthermore, the possibility to scale up by a factor of 50 by the mid-2020s when the HL-LHC (high luminosity LHC) [4] comes on line was also demanded. Figure 1 shows the files and bytes transferred during a typical month of CMS operations.

After a thorough review, CMS chose Rucio [5] to satisfy its data management needs for LHC Run 3 and beyond. Moreover, Rucio has a distinct advantage in being an emerging community project for data management for scientific communities as well as being simpler to operate: for example, PhEDEx requires an installation of the PhEDEx agent at every site;

[*] e-mail: ewv@fnal.gov

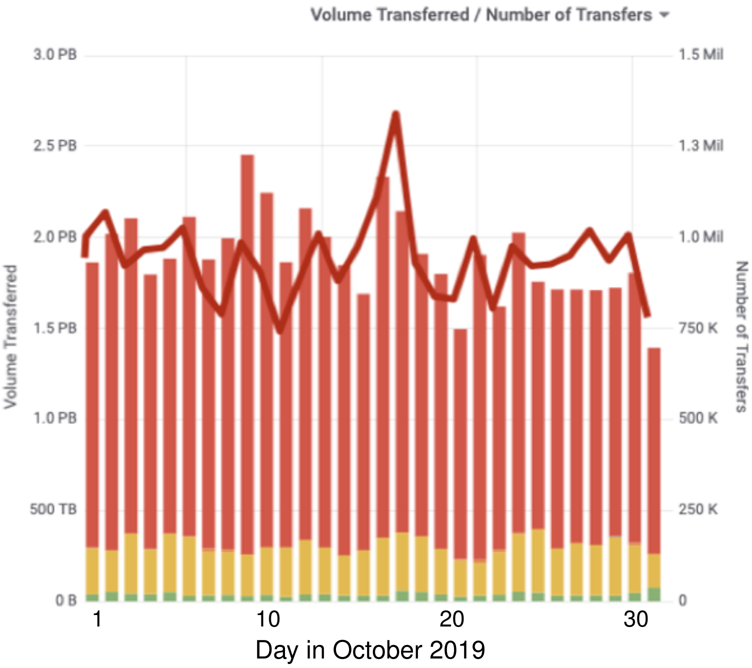

Figure 1. Current CMS data transfer requirements. This bar graph shows one month of data transfers for CMS, one bar per day. Red and yellow show the bandwidth used by PhEDEx (different versions) while green shows the bandwidth used for user transfers. The solid red line shows the number of files transferred per day, approximately one million. User transfers are a comparable number of files as PhEDEx, but much smaller average file size.

Rucio has no such requirement. Furthermore, Rucio can perform all of the functions of PhEDEx and most of the functions Dynamo provides. The CMS experiment will continue to use DBS [6], a custom physics metadata database, in conjunction with Rucio.

2 Rucio Transition Progress

A transition from PhEDEx to Rucio was begun in the fall of 2018 with a small team of about 8–10 people working part time (totaling about 3 full time equivalents).

2.1 Existing CMS Data Model

It is not feasible to change the existing CMS data model; fortunately the CMS data model is similar enough to the Rucio data model that it could be easily mapped onto the Rucio model.

Data in CMS is organized by file, block (a group of files), and dataset (a group of blocks). Each of these are many-to-one relationships. Rucio uses files, datasets, and containers (and containers of containers). Rucio allows each of these to be many-to-many relationships, *e.g.* a file may be in several datasets. For CMS, we chose to map blocks to datasets and datasets to containers, ignoring the possibility of many-to-many relationships.

Files have a logical file name (LFN) which is mapped to a physical file name (PFN) by means of a cascade of regular expressions per storage site. Rucio supports a pluggable LFN to PFN mapping. The CMS scheme was implemented using attributes of the Rucio storage element (RSE) and plugin code to derive the mapping.

2.2 CMS Rucio Infrastructure

As part of a collaborative effort between the core Rucio developers, ATLAS, and CMS, we are basing our infrastructure on Helm [7], Kubernetes [8], and Docker [9]. These are de facto standards in industry for providing services. All of the Rucio services are built into a single Kubernetes cluster which can be brought up from scratch in under an hour, including the provisioning of CERN Openstack resources to host the Kubernetes cluster.

The Kubernetes-based installation of Rucio relies on a number of third-party tools, which are also provisioned using Helm, such as nginx-ingress [10] and prometheus [11].

Helm allows the various flavors (production, integration, developers' tests) of the Rucio service to share a common setup, typically only with a few parameterized settings easily modified through configuration values. Third-party services configured in this way also typically require very few values changed from their defaults.

A major portion of our current Rucio setup is the syncing of metadata between PhEDEx and Rucio. While PhEDEx continues to operate, Rucio must shadow PhEDEx to have a consistent view of the CMS data. We use a special Docker container and Kubernetes pod which uses the APIs of both PhEDEx and Rucio to maintain this consistent view. A full synchronization round takes a few hours in "keep-up mode." The RSEs corresponding to PhEDEx endpoints are put into read-only mode so that PhEDEx and Rucio do not both try to write into the same physical namespace.

2.3 Rucio Scale Testing

As part of our testing of Rucio, and to gain operational familiarity with the product, we have performed two tests referred to as "Million File Tests". The smallest data format in CMS is the NanoAOD, which is targeted at final physics analysis and contains approximately 1 kB of information per event. The "Million File Test" aims to distribute multiple copies of this data with Rucio in a way that mimics the real data distribution of CMS.

The existing CMS sites were divided into four groups based on network interconnectivity: the Americas, Asia and Russia, and two groups within Europe. Additional RSEs were set up at each of these sites which were in read-write mode instead of read-only. Rucio then was instructed to place a copy of the entire NanoAOD format in each of these regions and two copies in one of the European regions. These copies were made with Rucio subscriptions, which generate rules based on dataset metadata.

In each of these tests, the total number of files distributed was 450 000 in 300 000 datasets with a total size of 320 TB. This redistribution of NanoAOD took less than three days. Figure 2 shows the number of files and data volume transferred during this test. While the total size of data moved was much smaller than a typical day of operations for CMS, this was intended as a test of moving large numbers of files. On that metric, the test was a success, having transferred the same number of files as PhEDEx during the same period of time. Both PhEDEx and Rucio use FTS [12] as a transfer tool which meets our requirements for total possible data volumes moved.

3 Future of the Rucio Transition for CMS

We expect to transition to Rucio during 2020 in preparation for LHC Run3 in 2021. We will transition less-used data as a first step, taking advantage of some of the Rucio features which PhEDEx does not have.

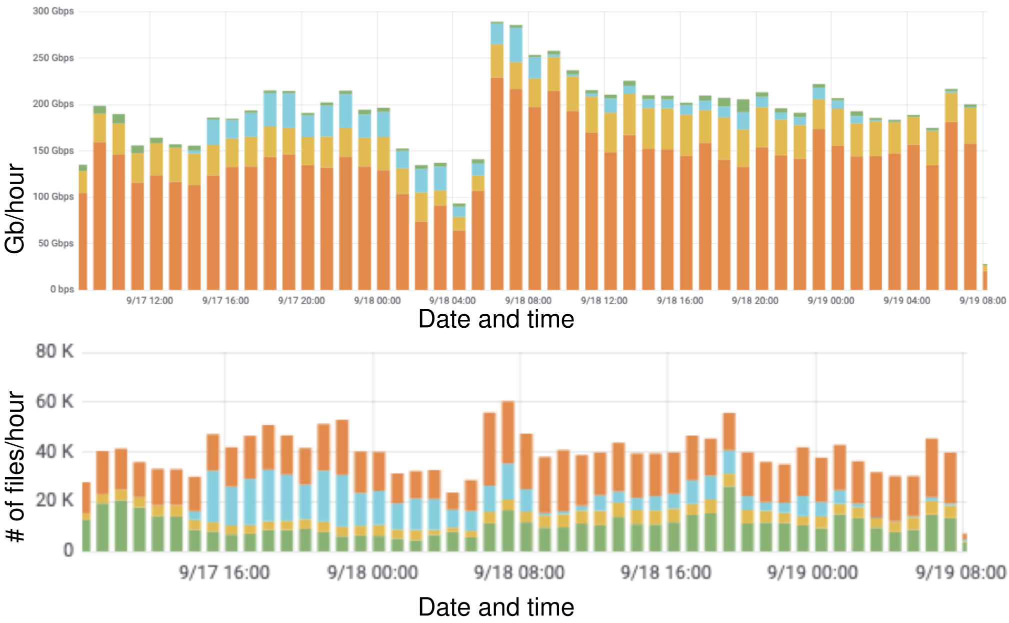

Figure 2. The top plot shows the volume of data transferred with PhEDEx (orange and yellow), Rucio (blue), and by users (green) during the "Million File Test." The lower plot shows the number of files transferred in the same categories during the same time period. The number of files transferred by Rucio and PhEDEx during the test was comparable.

3.1 NanoAOD Transition

As a logical follow on to the "Million File Test" in which we tested the full-scale distribution of the NanoAOD format, we plan to fully transition the management of NanoAOD to Rucio as soon as other developments, described below, are completed.

The NanoAOD datasets are an ideal candidate for this transition for two reason. First, they are easily reproduced from data on disk if they are lost. Second, they are not read by subsequent processing steps and only read by users.

The first step of this plan has already been tested as part of the earlier tests, namely the development of Rucio subscriptions and rules to widely distribute this data tier. Because PhEDEx will no longer have an accurate picture of where NanoAOD has been distributed by Rucio, we must either have our production system inject NanoAOD metadata into Rucio directly or synchronize the initial injection of NanoAOD into Rucio and then remove the metadata from PhEDEx.

With the timely synchronization of all other data from PhEDEx to Rucio described above, all the CMS software tools can move to Rucio as their source for data location information.

3.2 Modifying Existing Software

The CMS experiment has a large suite of software which has been built up around our current data management system. Each of these pieces of software must be modified to interact with Rucio rather than PhEDEx.

We use a suite of software called Unified [13] to orchestrate data workflow management and to perform some of the data management needed for running workflows. We have begun a process to move the data management aspects of this software into the core of the workflow system. After this is completed (but still interacting with the PhEDEx-based data management software), the newly refactored software can be more easily switched to use Rucio. Currently, the new system that places workflow inputs is in pre-production testing. The portion of the system which does data placement and data locking for workflow outputs will be ported and tested this year.

To run workflows, CMS uses its own software suite named WMAgent [14]. A version of WMAgent which uses Rucio for data location is under testing, and the ability to inject data into Rucio is expected to be ready soon.

End users in CMS use DAS [15], the data aggregation service, to find data and get a coherent view of metadata from PhEDEx, DBS, and other metadata services. Rucio has been added as another data source for DAS and DAS is able to automatically check for inconsistencies between Rucio, DBS, and PhEDEx.

3.3 User Data Management

Rucio will be able to provide full data management to CMS users for the first time. Currently, CMS users can use the CRAB [16] software to produce their own datasets, and CRAB moves them to their final destination using FTS.

However, users are not able to delete datasets easily nor can they move them from site to site or duplicate them. A first version of CRAB that is able to inject data into Rucio and move it to a site where the user has quota is now available. Subsequently, the full suite of tools in Rucio can be used to manage the resulting dataset.

Since user data is written with the user's credentials and Rucio data is all owned by CMS on the storage level, we needed to develop a solution to change ownership. We accomplished this by using a non-deterministic RSE in Rucio in which both the LFN and PFN are specified. Rucio transfers data from the non-deterministic RSE to the normal, deterministic, RSE with CMS LFN to PFN mapping described above.

4 Conclusion

Following the selection of Rucio as the next data management system for CMS, the CMS Rucio team has worked towards incorporating Rucio into the CMS software ecosystem and testing Rucio at the scale needed for Run 3 operations. Rucio has met these challenges and will be deployed as the next CMS data management solution in 2020.

References

[1] CMS Collaboration, JINST **3** S08004 (2008).

[2] J. Rehn, *et al.* "PhEDEx high-throughput data transfer management system", Proc. CHEP06, Computing in High Energy Physics, Mumbai, India, (2006).

[3] Y. Iiyama, C. Paus, M. Goncharov, "Dynamo — The dynamic data management system for the distributed CMS computing system', Proc. CHEP16, Computing in High Energy Physics, San Francisco, USA, (2016).

[4] G. Apollinari, I. Béjar Alonso, O. Brüning, M. Lamont and L. Rossi, doi:10.5170/CERN-2015-005

[5] M. Barisits, T. Beermann, F. Berghaus, *et al.* Comput. Softw. Big Sci. **3**, 11 (2019). https://doi.org/10.1007/s41781-019-0026-3

[6] M. Giffels, *et al.* J. Phys.: Conf. Ser. **513** 042022 (2014).

[7] Helm homepage. https://helm.sh

[8] Kubernetes homepage. https://kubernetes.io

[9] Docker homepage. https://docker.com

[10] Nginx ingress homepage. https://www.nginx.com/products/nginx/kubernetes-ingress-controller/

[11] Prometheus homepage. https://prometheus.io

[12] A. A. Ayllon, M. Salichos, M. K. Simon, O. Keeble. J. Phys.: Conf. Ser. **513** 032081 (2014). https://doi.org/10.1088/1742-6596/513/3/032081

[13] J. Vlimant. J. Phys.: Conf. Ser. **898** 032081 (2017). https://doi.org/10.1088

[14] S. Wakefield *et al.* Journal of Physics: Conference Series CHEP 2012. (2012).

[15] V. Kuznetsov, D. Evans, S. Metson "The CMS Data Aggregation System" ICCS 2010, Procedia Computer Science **1**, Issue 1, 1529-1537; https://doi.org/10.1016/j.procs.2010.04.172

[16] M. Mascheroni, *et al.* J. Phys.: Conf. Ser. **664** 062038 (2015).