



# Transitioning CMS to Rucio Data Management

Eric Vaandering - for the CMS Rucio team

CHEP 2019 — Adelaide, Australia

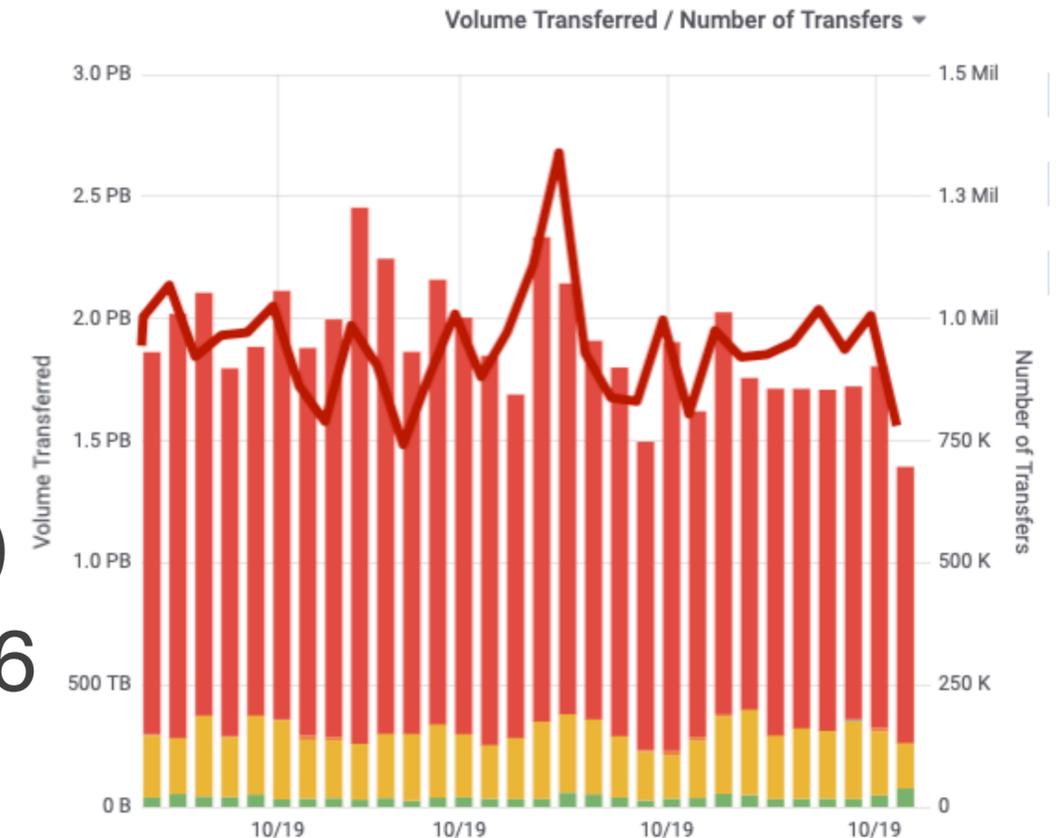
4 November 2019

# Overview

- CMS data management needs and situation
  - Review process and decision to adopt Rucio
- CMS Rucio infrastructure
- Plans for transition
- Current status and scale tests
- [User data management]

# Current CMS Data Management

- Data on tape O(100 PB) and disk O(50 PB)
  - 8 sites with tape, O(100) with managed disk
  - Production file size O(1 GB), user file size O(100 MB)
- Per day transfers ~2 PB, 1 M files (user & production)
- Similar scale for next 6-7 years, increases 50x in 2026
- Current data management is done by PhEDEx
  - Each site hosts a PhEDEx agent to manage its own data including tape
    - Requires non-trivial effort at each of our sites
  - Maintains a database of the desired state (files at sites); issues FTS commands to achieve it
  - PhEDEx is aging and we realize its lifetime is limited
- A higher layer, Dynamo makes PhEDEx requests to dynamically distribute and clean up data
- Separate physics meta-data catalog (DBS)



# CMS Selection Process

- Performed an evaluation of Rucio from early 2018 through summer 2018
- July 2018 — CMS conducted a down-select review at which Rucio was chosen
  - Cited an existing product with a development plan for the future
  - Obviously collaboration within LHC/HEP was a big plus
- Began a methodical transition to Rucio in Fall 2018

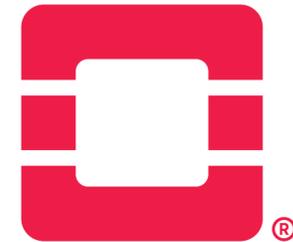
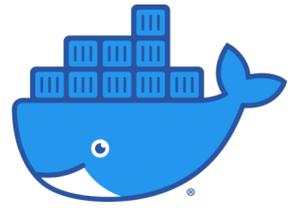


# CMS vs. Rucio Data models

- CMS data stored in a three tiered structure:
  - Files - target size 4 GB
  - Blocks - usually about 500 files, designed to be a unit that can be stored and transferred at one site
  - Dataset - some number of blocks, has a physics meaning (often stored all at a site, but not necessarily)
  - All many:one maps, not many:many (like rucio)
  - Not perfect but fits OK into Rucio model:
    - CMS Block → Rucio Dataset
    - CMS Dataset → Rucio Container
- CMS has a single namespace of data with data types organized by directory
  - Use a map of LFN (logical) to PFN (physical) namespaces
  - We used Rucio's plugin and RSE attributes to implement this

# CMS Rucio Infrastructure

- Based on Docker, Kubernetes (k8s), Helm, OpenStack, Oracle
  - Helm enables minimal config changes for CMS
- Zero to operating cluster is ~30 minutes
  - Upgrades are nearly instantaneous
- All components of Rucio and supporting code are operating in k8s
  - Monitoring, logging, proxy renewal, synchronization also in k8s Pods or CronJobs
    - Many 3rd party pieces built with helm as well
    - ATLAS also moving towards k8s/helm — collaborative effort
  - Liveness checks give automatic restart
- Allows us to have production and testbed on a shared set of resources
  - Developers' environment is smaller version of central cluster(s)



# Transition: Syncing data between PhEDEx & Rucio

- Critical part of transition is moving all the metadata over
- Sync scripts keep replica info for a site in sync between PhEDEx and Rucio
  - Define these Rucio endpoint mirrors as read-only (no transfers in, no deletions)
  - Can be used as a source for additional replicas. During the transition period we will also have a [Site]\_Test endpoint which writes to a test part of the physical namespace
- Currently uses REST APIs of Rucio and PhEDEx. May switch to direct database interface

# Transition plan starting with NanoAOD

- NanoAOD is the smallest CMS data tier:  $O(1 \text{ kB})/\text{event}$
- Goal: transition all management of NanoAOD to Rucio as a test case
  - Good candidate; not read in production, easily reproduced
- Step 1: Sync all data on NanoAOD from PhEDEx to Rucio
- Step 2: Develop Rucio subscriptions and rules to distribute NanoAOD to test space
  - Current size is 1M files/430 TB (250k/100 TB unique) in CMS
  - Qualification test detailed on next slides
- Step 3: Publish NanoAOD directly into Rucio; use Rucio as the full data location store
  - Rucio distributes NanoAOD with subscriptions and/or rules
  - PhEDEx distributes all other Tiers of data
  - Sync non-NanoAOD data from PhEDEx to Rucio.
  - All CMS software will lookup data in Rucio
  
- Other data tiers can be migrated by repeating step 3

# Million File Test

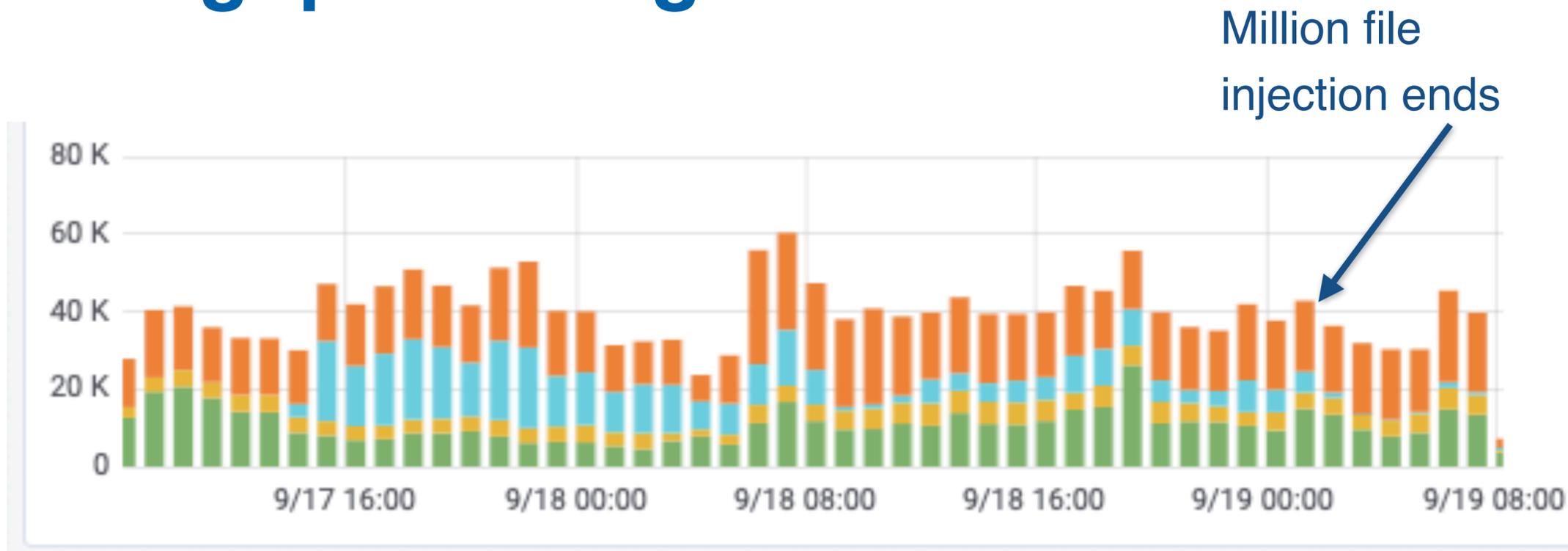
- Make a total of 5 copies of NanoAOD
  - 1 copy in Americas, Asia/Russia, and 1/2 of Europe. 2 copies in other 1/2 of Europe
  - Regions were defined by bandwidth between sites
- Total stats replicated were 450k files 299k datasets. Total size 320 TB
  - Also tested a cleanup campaign of placed files
- Used Rucio subscriptions: Generate placement rules based on dataset metadata
  - We had to re-tag all the relevant datasets as “New” to trigger the subscription
  - Subscriptions are still generating rules as new datasets are added to Rucio by sync
- Workflow:
  - Rucio daemon scans “new” datasets, creates rules
  - Rule engine demands new replicas (minimal to satisfy rules)
  - Submitter daemon makes transfer requests to FTS

# Rule creation during and after test



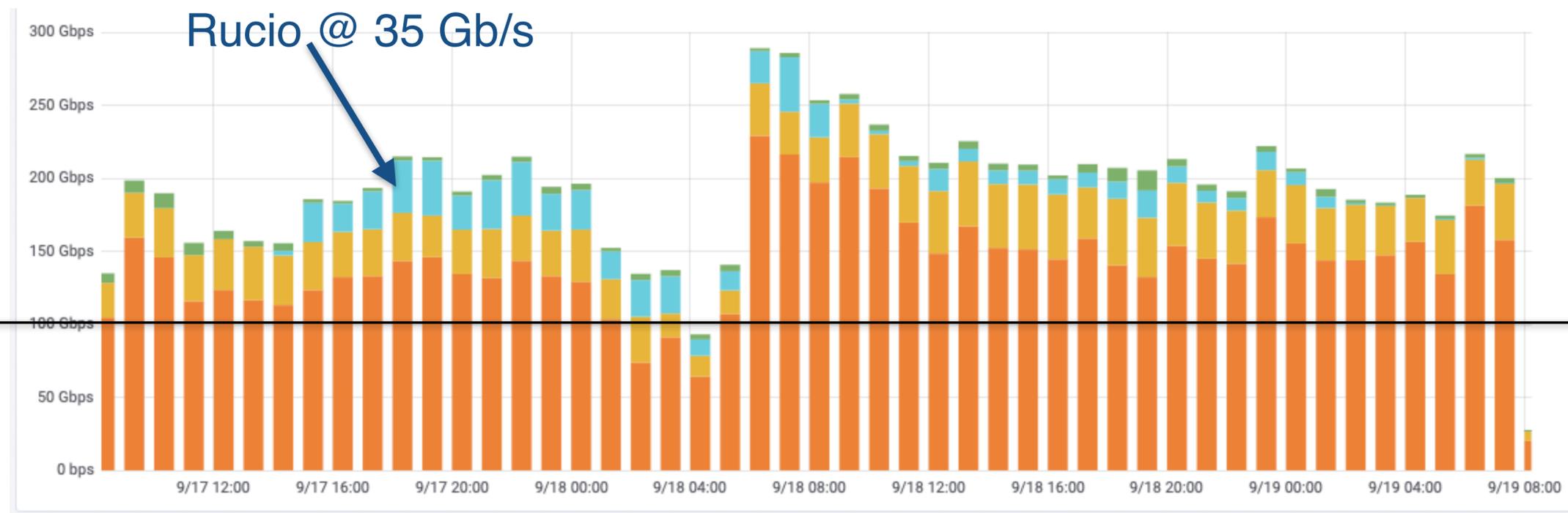
# Rucio throughput during test

Files Transferred



PhEDEx  
PhEDEx  
Rucio  
Users

Bandwidth  
100 Gb/s



# Existing software needs to interface with Rucio

- Orchestrator system for workflow management
  - In the process of rewriting DM aspects of this for existing DM tools, adopting to Rucio is simple
  - Input data placement is in pre-production testing
  - Output data placement and data locking to be done
- Adapting workflow management system (WMAgent)
  - A test agent which uses Rucio for data discovery is being tested
  - Next step, development underway, is to publish data into Rucio
    - Expect first tests this year
- Data aggregation service can already use Rucio as a source

# Current CMS User Data Management

- No user data management; movement and metadata only
- All done with AsyncStageOut (ASO) which is a thin layer on top of FTS
- Produced at Site A, moved to Site B (user has a relationship with Site B)
  - User has a storage quota at Site B as part of site pledge to CMS
- Can never be moved to Site C and have that reflected in DBS
- Some user data is completely unmanaged
  
- Some choices driven by tools, are reevaluating as we adopt Rucio

# User Data Management with Rucio

- First version of this is now available
  - Users will be given quota at a geographically associated site
  - Make dataset rule (for final state stageout) on task submission
  - Register output files and attach to dataset on job completion
- Final data owned by CMS, initially data staged to temp area owned by user
  - Job runs with user credentials
  - Solution: Temporary endpoint is “non-deterministic” which maintains PFN explicitly
- Roll out for user data management is orthogonal with production.
  - Bar to replace ASO is lower.

# Conclusions

- Rucio is a good match for CMS data management needs
- Transitioning an operating experiment at this scale is a non-trivial process
- CMS has tested Rucio near the LHC Run 3 scale needed
  
- Expect significant benefits from shared code base and operational experience with ATLAS — [Operational Intelligence paper in Track 3](#)
  
- The final transition will occur in 2020
  
- We expect to collaborate on future development which enable Rucio to function in the HL-LHC era