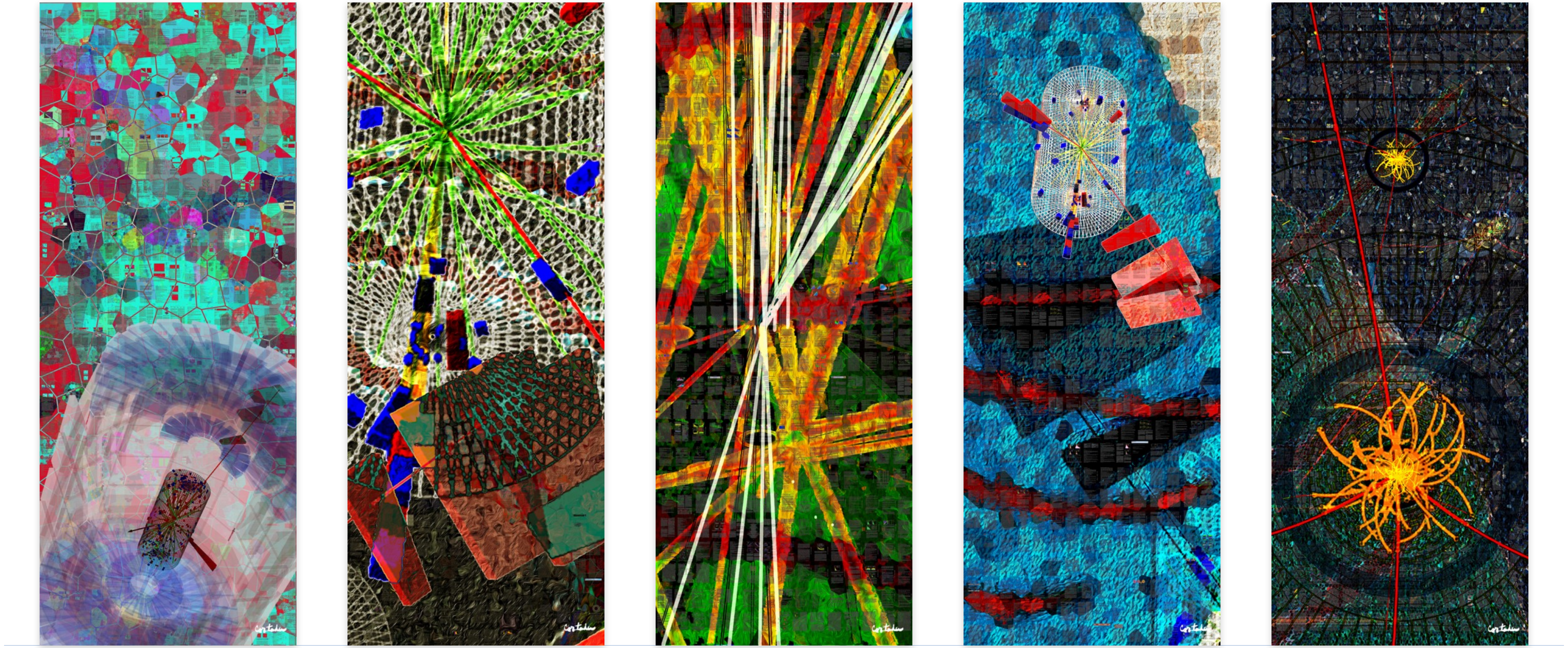
Rucio for CMS

D. Ciangottini, K. Ellis, F. Garzón, Y. Guo, C. Huang, R. Lopez,
I. Mandrichenko, D. Mielaikaite, S. Piperov, N. Smith, E. Vaandering



Overview

- CMS data management needs and data model
- Helm and kubernetes setup
- First steps of transition
- Large n-files & large dataset tests
- CTA and tape testing
- Other areas of work
- Suggestions and next steps

CMS Data Management Needs

- Current statistics on our data storage and movement
  - Stored on tape O(100 PB) and disk O(50 PB) at 50+ sites
  - Per day transfers ~1 PB, 1 M files (combined user, production)
- Numbers stay more or less constant for next 7-8 years, go up 50x in 2027 and beyond
- Primary data management is done by PhEDEx
  - Each site typically hosts a PhEDEx agent to manage its own data. Also manages local tape
    - ★ Requires non-trivial effort at each of our sites
  - Maintains a database of the desired states (blocks at sites) and issues FTS commands to achieve it
  - PhEDEx is aging and would not survive the HL-LHC era without major effort
- A higher layer, Dynamo, monitors popularity of data and, based on rules, makes subscriptions to dynamically distribute popular data, cleanup unpopular
- Separate physics meta-data catalog (DBS)

- July 2018 — Made a decision to adopt Rucio before Run3 starts

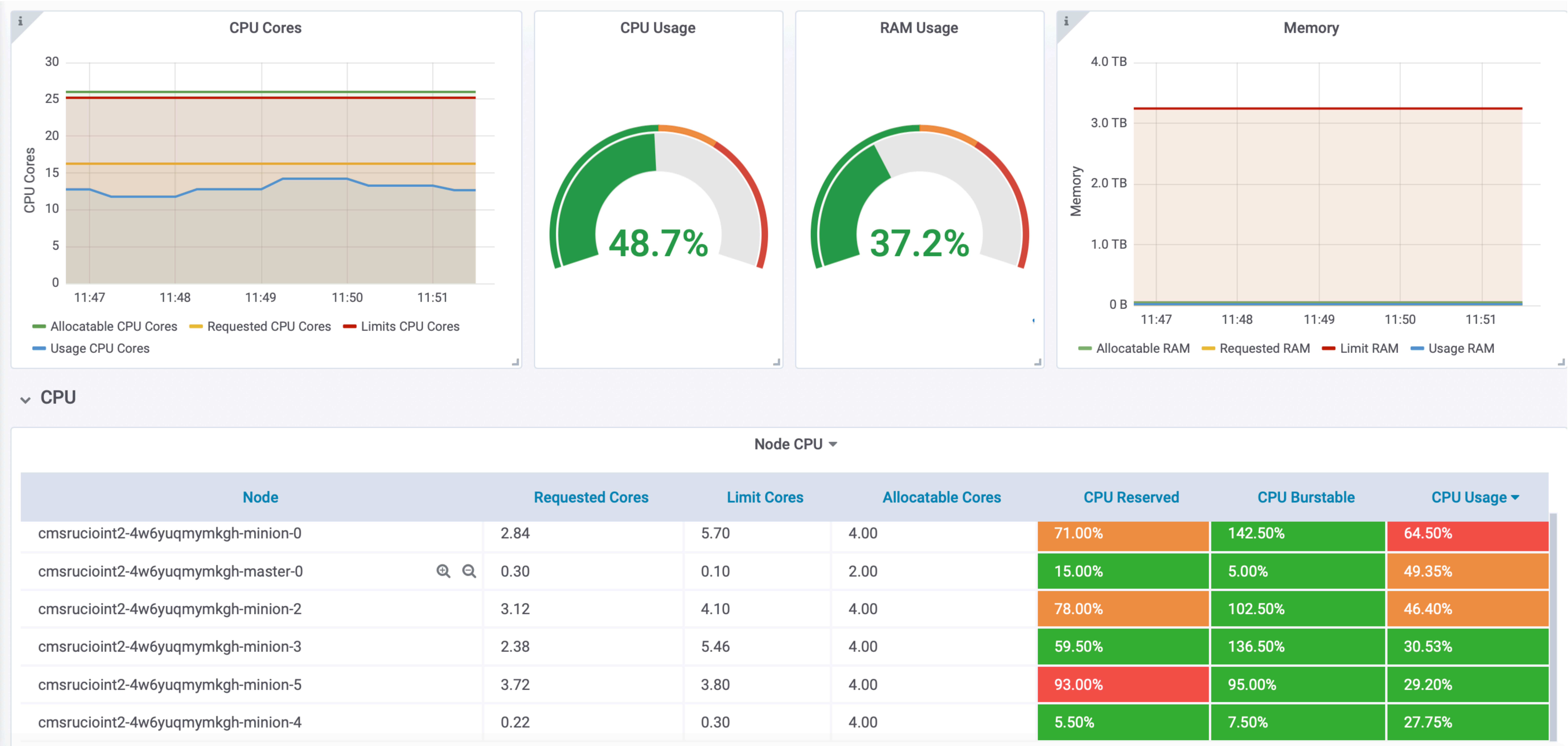
CMS vs. Rucio Data models

- CMS data stored in a three tiered structure:
  - Files - target size 4 GB
  - Blocks - usually about 100 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 no necessarily)
  - All many:one maps, not many:many (like rucio)
  - Not perfect but fits OK into Rucio model:
    - ★ CMS Dataset - Rucio Container
    - ★ CMS Block - Rucio Dataset
- CMS has a single namespace of data with different types of data in different places of this namespace
  - Use a (potentially) complicated map of LFN (logical) to PFN (physical) namespaces
  - We use Rucio's plugin and RSE attributes to implement this

CMS Rucio Server at CERN

- Based on Docker, Kubernetes (k8s), Helm, OpenStack, CERN Oracle
  - Very collaborative effort with ATLAS
  - Helm enables minimal config changes for CMS
  - Zero to operating cluster is ~30 minutes (tested regularly)
  - Effort in CMS to get other web-facing services on k8s and OpenStack
    - ★ Some differences but lots of shared knowledge, e.g. interface to CERN monitoring layer
- Allows us to have production and testbed on a shared set of resources
  - Developer, testbed, production instances all will be identical except for scale
    - ★ Integration is on production hardware
- Rucio server and all rucio daemons are operating in k8s
  - Liveness checks now give automatic restart, possibility for load detection with automatic scale-out/in
  - Added monitoring, logging, proxy renewal, synchronization — fed back to official Helm charts as appropriate
  - All Cron Jobs also running and managed by kubernetes (no special servers)

Kube-eagle monitoring + Grafana



Node	Requested Cores	Limit Cores	Allocatable Cores	CPU Reserved	CPU Burstable	CPU Usage ▾
cmsrucioint2-4w6yuqmymkgh-minion-0	2.84	5.70	4.00	71.00%	142.50%	64.50%
cmsrucioint2-4w6yuqmymkgh-master-0	0.30	0.10	2.00	15.00%	5.00%	49.35%
cmsrucioint2-4w6yuqmymkgh-minion-2	3.12	4.10	4.00	78.00%	102.50%	46.40%
cmsrucioint2-4w6yuqmymkgh-minion-3	2.38	5.46	4.00	59.50%	136.50%	30.53%
cmsrucioint2-4w6yuqmymkgh-minion-5	3.72	3.80	4.00	93.00%	95.00%	29.20%
cmsrucioint2-4w6yuqmymkgh-minion-4	0.22	0.30	4.00	5.50%	7.50%	27.75%

NanoAOD transition plan

- NanoAOD is CMS’s smallest data format: Few kB/event. 100TB for all Runs, versions

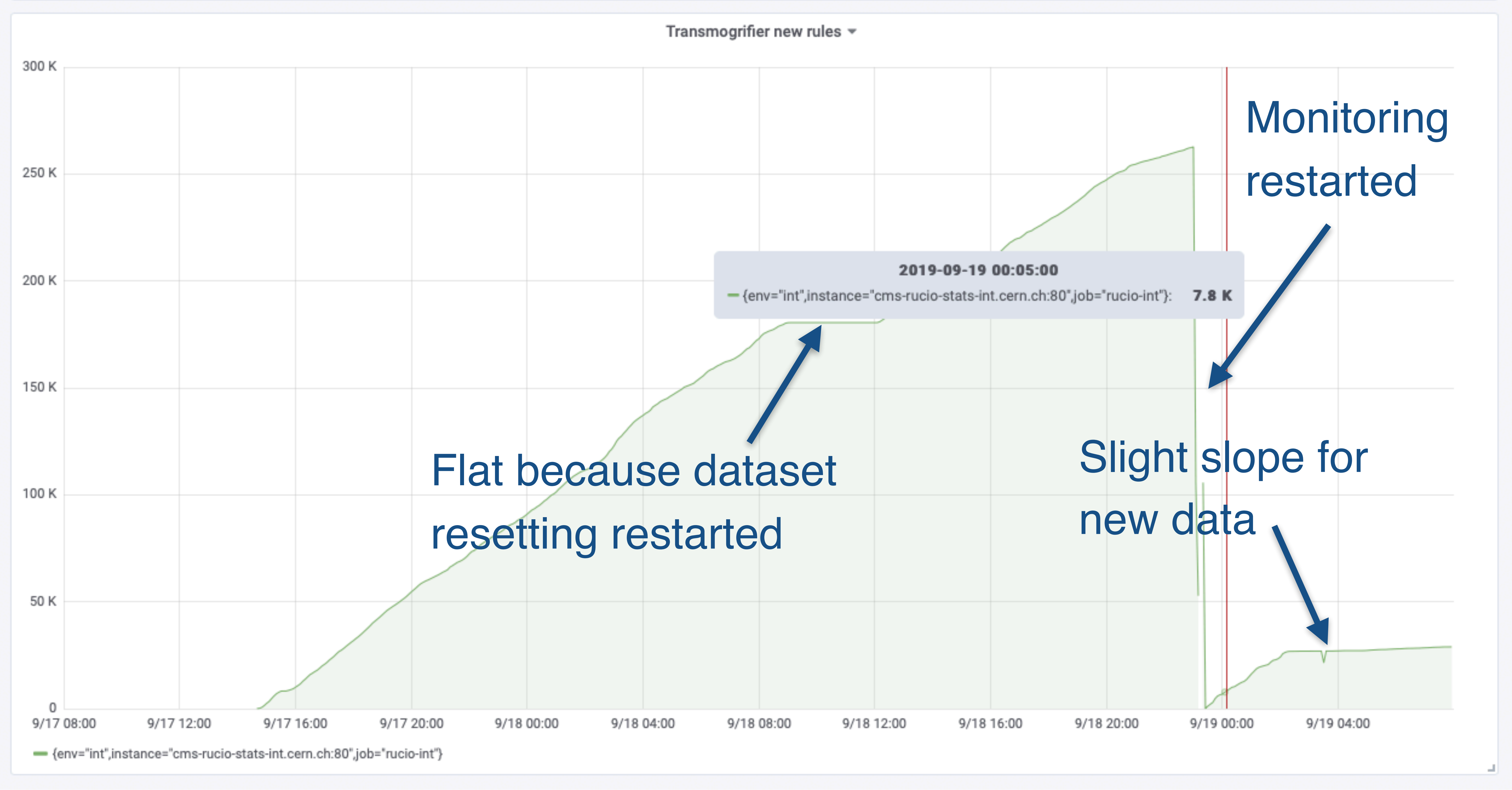
- Goal: transition all management of NanoAOD to Rucio as a test case.
  - Good candidate; not read in production
- Step 1: Sync all data on NanoAOD from PhEDEx to Rucio
- Step 2: Develop Rucio subscriptions and rules to distribute NanoAOD to test space
  - Done as a “million file test.” Not used in production: dedicated test name space
- Step 3: Publish NanoAOD directly into Rucio, Rucio as the full data location store
  - Sync non-NanoAOD data from PhEDEx; all tools (DAS, CRAB, WMAgent) will lookup in Rucio
  - Rucio distributes NanoAOD with subscriptions and/or rules
  - Dynamo and PhEDEx no longer manage NanoAOD

- Currently preparing for this last step

Million File Test

- Did this test twice on two different Rucio instances
- Make a total of 5 copies of all 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 did a cleanup campaign of the first test
- We did this with Rucio subscriptions: Generate placement rules based on dataset metadata
  - Subscriptions are still generating rules as new blocks/datasets are added to Rucio by production
- Workflow:
  - Transmogrifier scans datasets, creates rules
  - Rule engine demands new replicas (minimal to satisfy rules)
  - Conveyor submits transfer requests to FTS

Rule creation during and after test



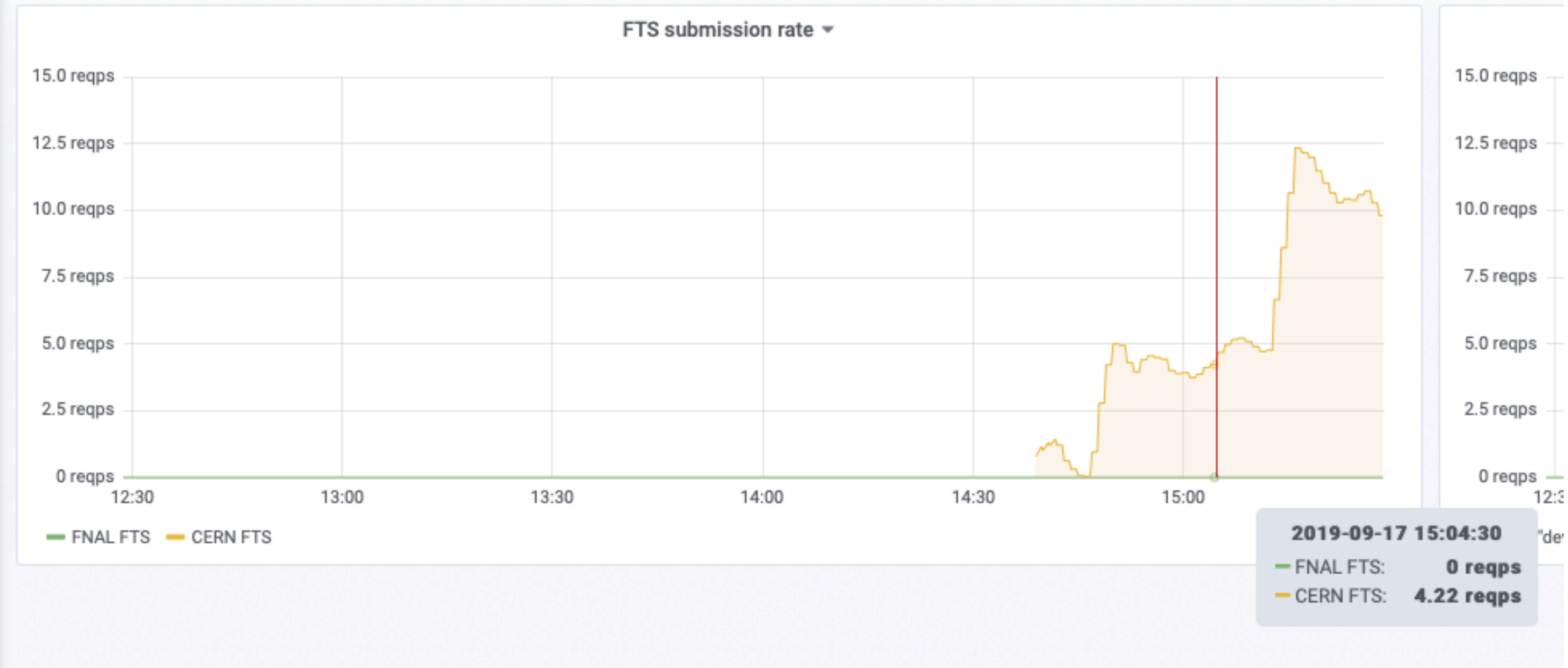
Monitoring restarted

Flat because dataset resetting restarted

Slight slope for new data

- Transmogrifier updates about 10k datasets/hour

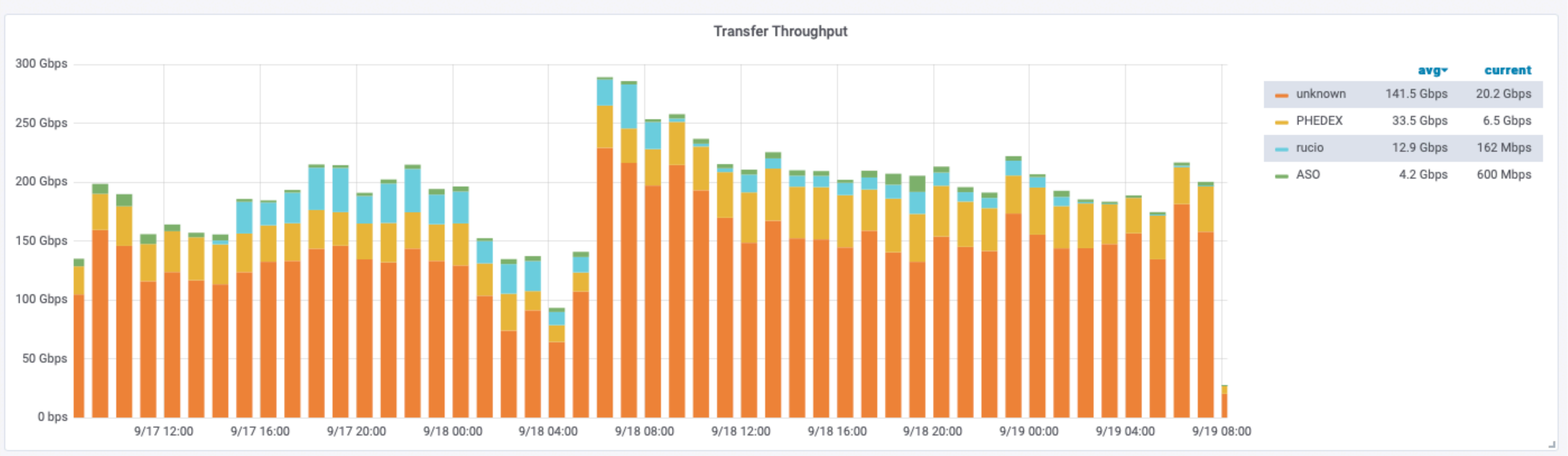
FTS submissions and scale up



- Submission rate to FTS at 5 Hz. One line change to bring on another submitter, momentarily doubled to 10 Hz, then kept up

Bandwidth by hour during tests

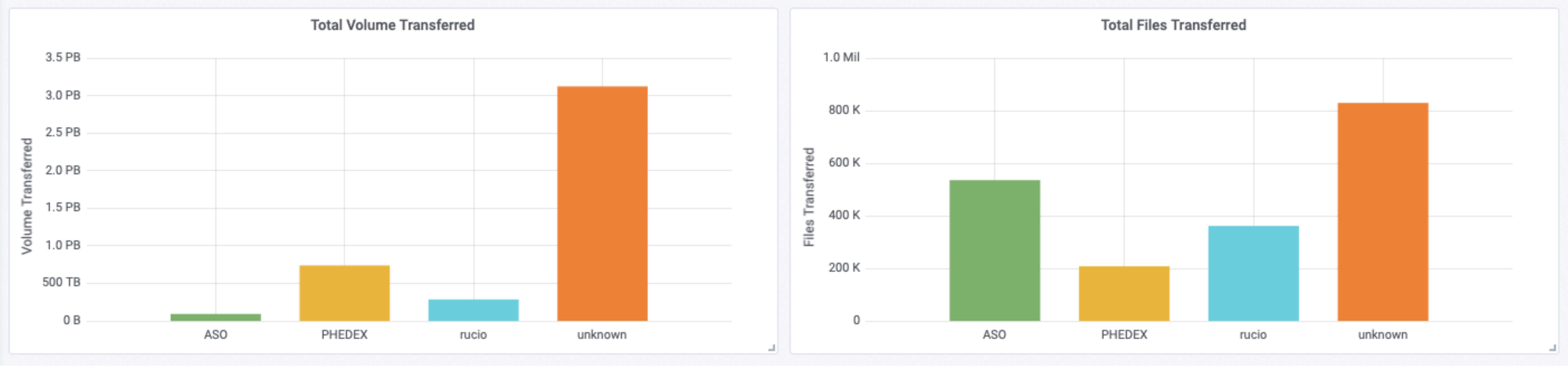
- Rucio (cyan) throughput is clearly visible during test period



	avg▾	current
unknown	141.5 Gbps	20.2 Gbps
PHEDEX	33.5 Gbps	6.5 Gbps
rucio	12.9 Gbps	162 Mbps
ASO	4.2 Gbps	600 Mbps

Million file test within CMS

- Volume is low as expected since NANOAOD files are small (as are user files for ASO)



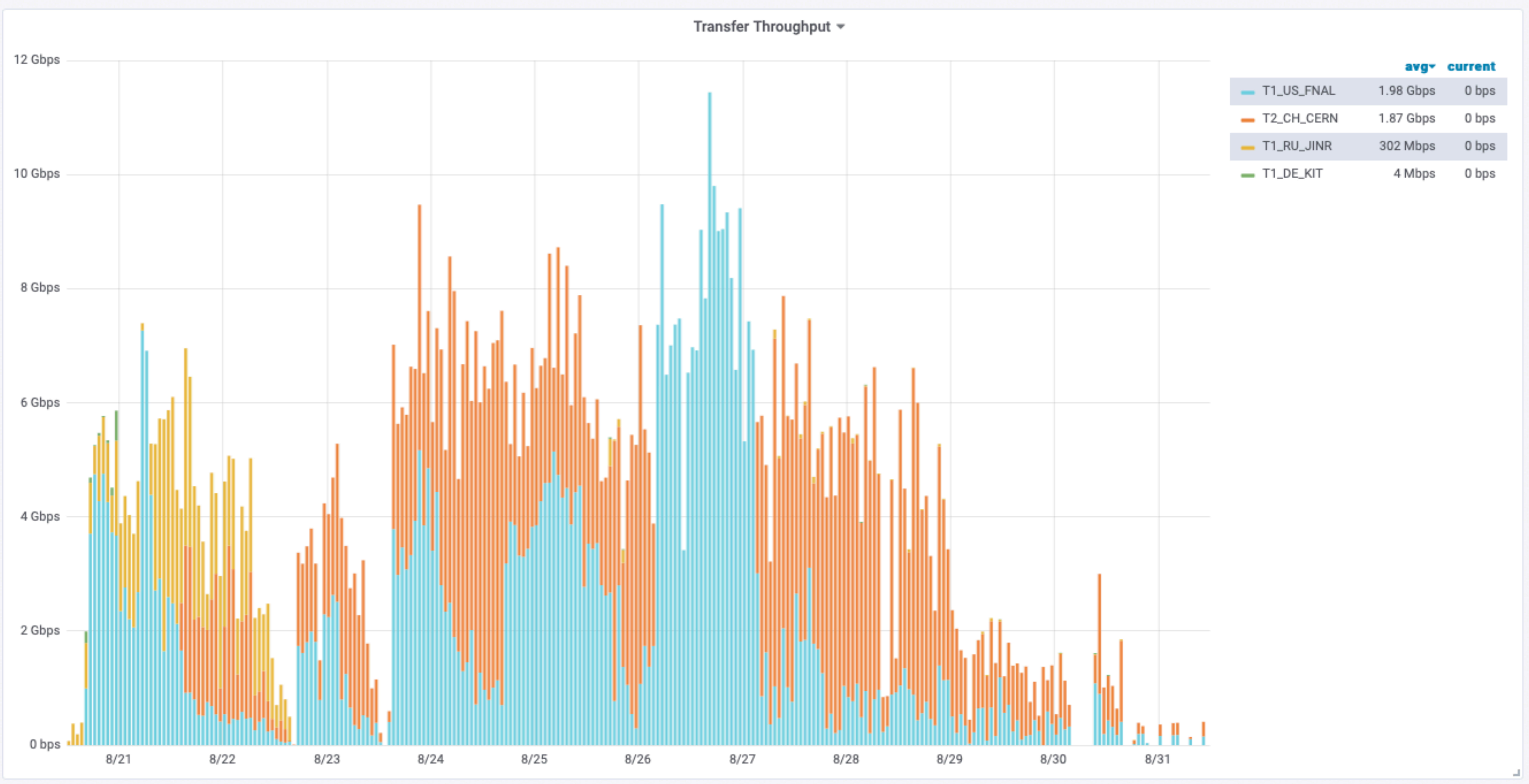
Rucio as part of production

- A couple of non-traditional sites where we can’t/prefer not to set up PhEDEx endpoints
- Currently NERSC and Spark cluster at Vanderbilt University
- Placing data to be used by production
- Especially at NERSC, large file sizes. Latest “test” peaked at >10 Gb/s from several sources

- Combined with small file tests, convinces us our setup can transfer at the scale needed for CMS

Rate to NERSC

- Just a fraction of the total CMS rate

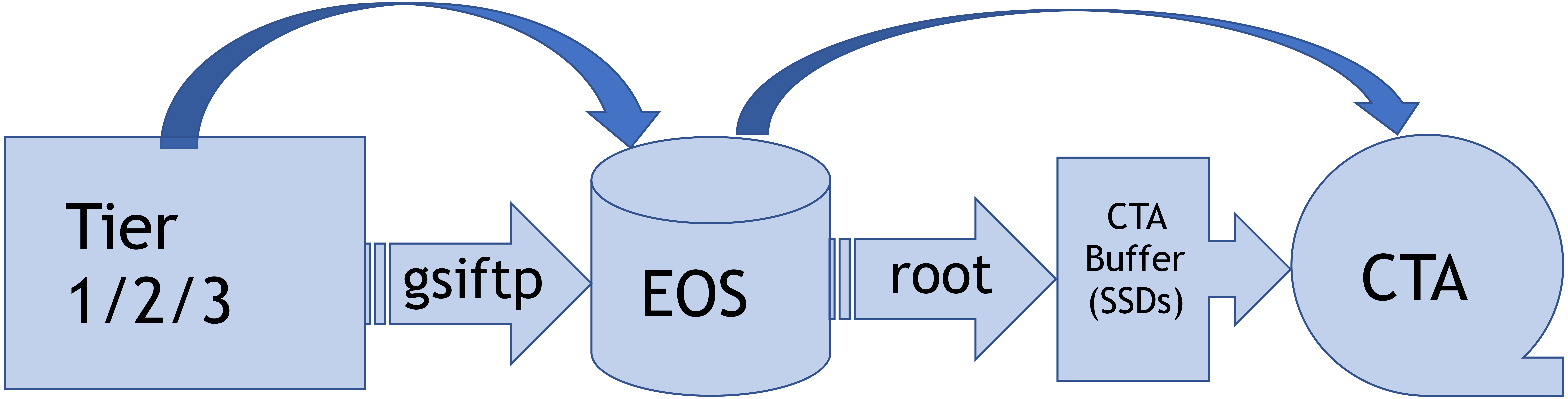


	avg	current
T1_US_FNAL	1.98 Gbps	0 bps
T2_CH_CERN	1.87 Gbps	0 bps
T1_RU_JINR	302 Mbps	0 bps
T1_DE_KIT	4 Mbps	0 bps

Rucio with CTA (CERN Tape)

- CTA is the new Tape Service at CERN (and soon at RAL)
- Small scale tests of CTA successful
- Large scale tests still coming
- Need to put multihop into production — automating manual process to bridge connectivity

Katy Ellis
RAL



“Multi-hop”

Consistency checking for CMS

- CMS has an existing consistency checking with our existing system using xrootd for remote listing

- Need to replicate this with Rucio to deal with two problems:
  - Data which is supposed to be at a site, but is not — missing data
  - Data which is at a site and is not supposed to be (any more) — dark data

- CMS work plan — ongoing
  - Use XRootD for creating Site Reports remotely — adapt existing mechanism to CERN infrastructure and Rucio input expectations
  - Dump Rucio DB reports via Sqoop
  - Adapt to Auditor format; Use Auditor for the comparison
  - Adapt Auditor code to handle native CMS LFN/PFN paths.

- Would like to do these comparisons weekly and on k8s cluster

Suggested areas for improvement

- Monitoring and messaging
  - Aware of a move from statsd to prometheus
  - More probes runnable by default? Remove ATLAS specific probes. Database choice may be an issue
  - Would be helpful to have options to easily plug into existing monitoring infrastructure
    - ★ Differences between CERN-ATLAS, CERN-CMS, Fermilab, presumably others
  - Messaging is similar. Interest by CMS in NATS, a high-performance messaging queue
  - Messaging server in kubernetes setup for simple installations?
    - ★ Already in docker compose?
- Auditor setup
  - Seems to be a big lift and not well documented
  - Perhaps a low performance version not involving external dumps could be supplied as a starting point
  - Hopefully CMS contributions help with getting information from site. May need further generalization.
- Helm and kubernetes are a big step forward
  - Need to make sure this is useful outside of CERN
  - Code customization can be done with experiment specific containers based on rucio/containers
    - ★ Will pip install rucio-cms be even easier?

Next steps

- Implement first steps of a real transition — using NanoAOD
- Gain additional operational experience
- Complete adaptation of external CMS code
- Sort out network issues with k8s identified at CERN
  - Or move production servers off to dedicated VMs
- Document

- Have set out a number of use cases to track these dependencies

- Expect to transition fully to Rucio this year