

# ATLAS & Rucio

Special Topics? Or ... what the experiment is actually doing with Rucio

7th Rucio Community Workshop

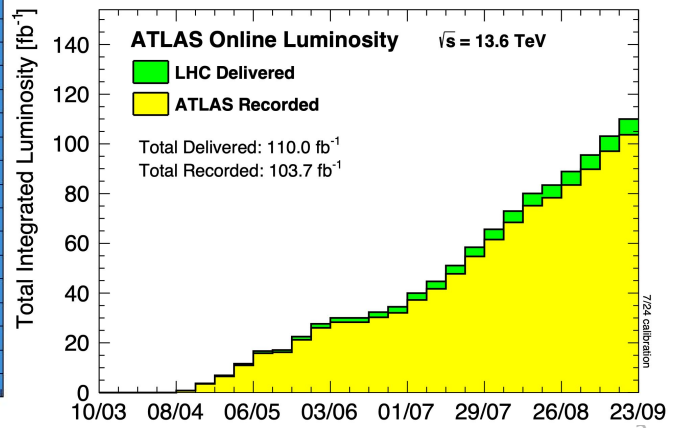
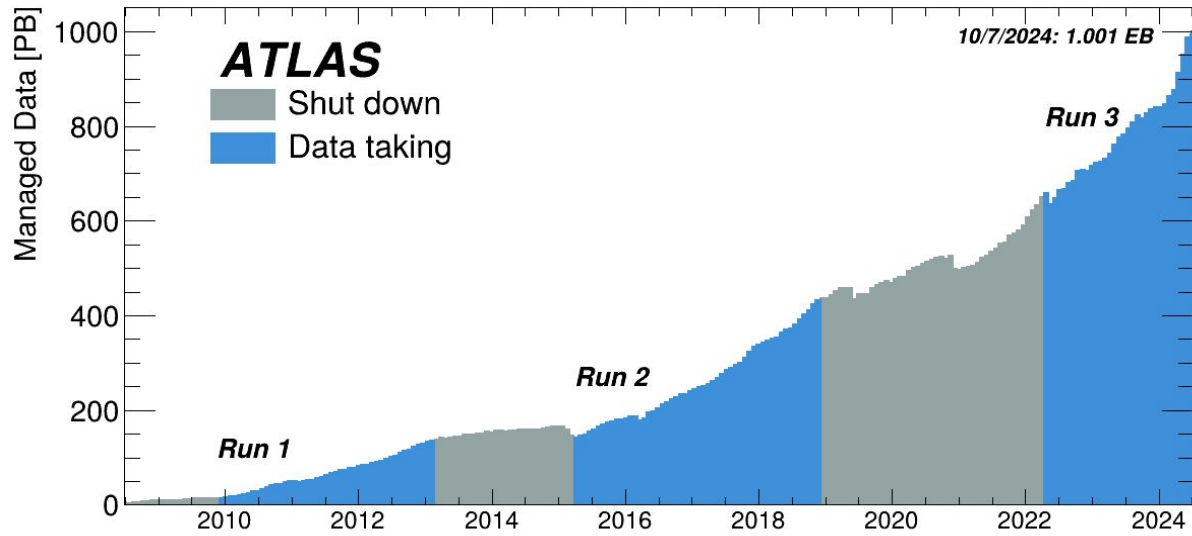
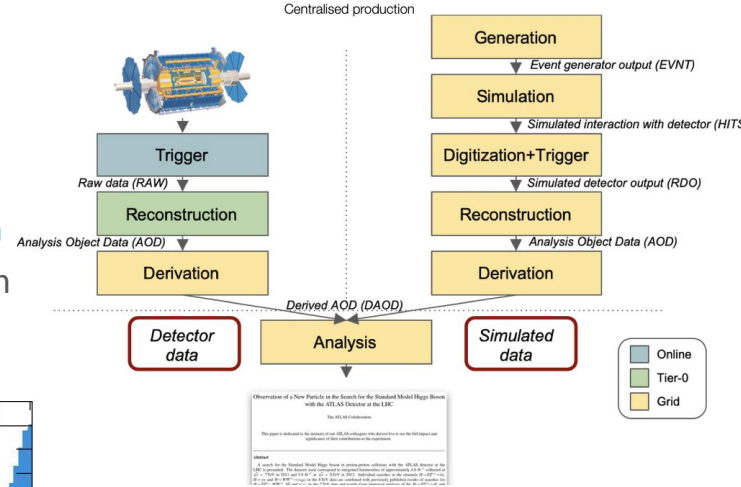
2024-10-01

Mario.Lassnig@cern.ch



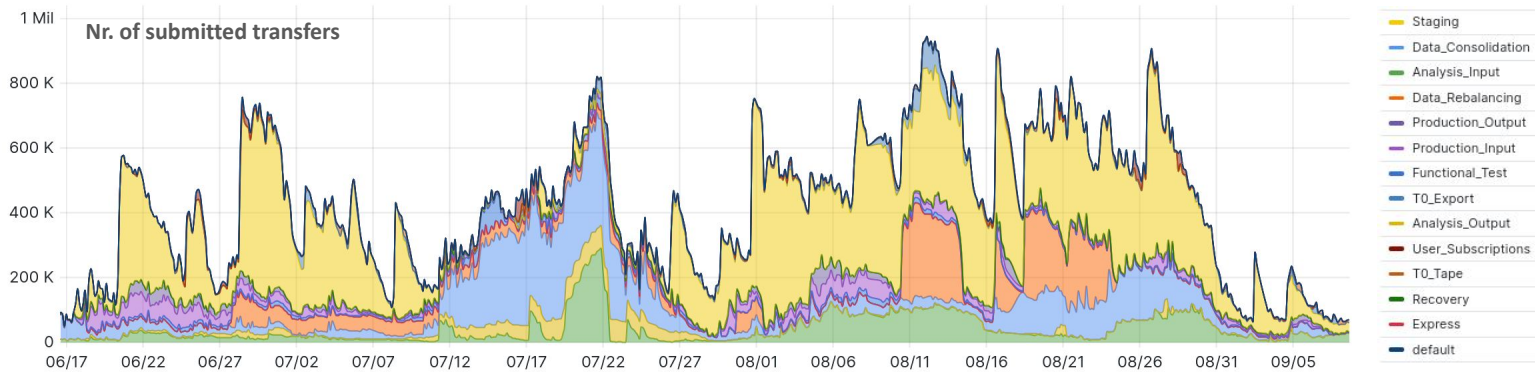
# A major milestone for ATLAS

- We have reached ONE EXABYTE of Rucio-managed data 🥳
  - Data taking in Run-3 has been extraordinary
  - LHC delivered more than full 2024 pp target of 110/fb<sup>-1</sup> already
  - Data-taking efficiency at 94.3% !
- Centralised production and user analysis is also running at full steam
  - Multiple concurrent campaigns of various intensity and duration
  - Physics Validation, Production, Reprocessing, Derivation



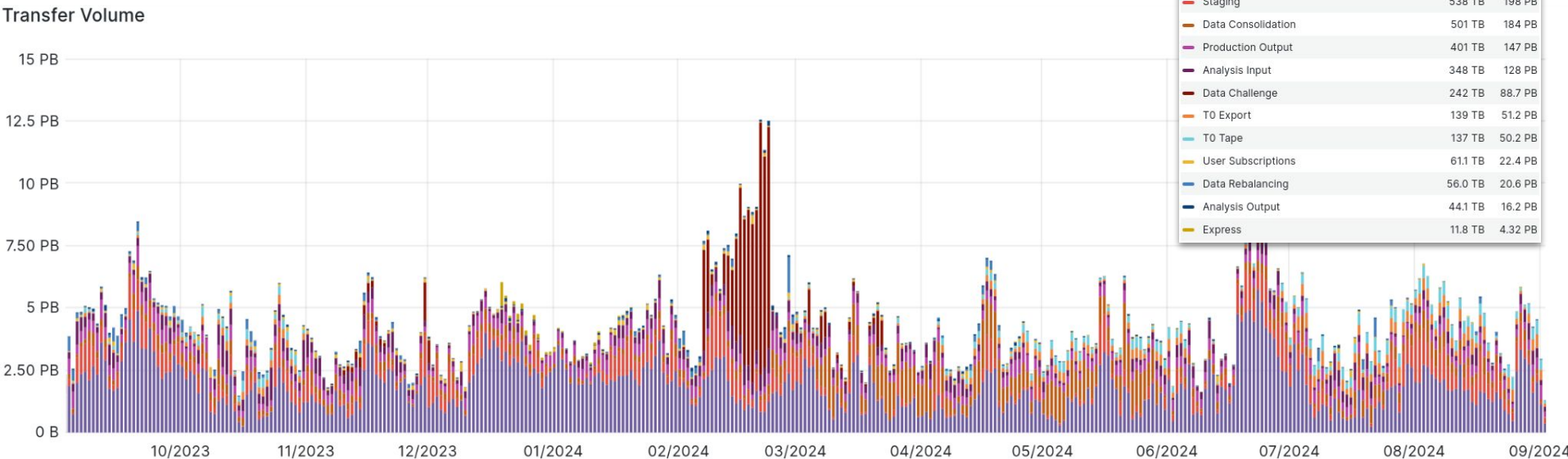
# Current ATLAS system scale

- **Namespace is growing, but thankfully only slowly**
  - **68 M containers**, cf. tasks, also serve as centralised production backend, with **only 8.9 M actually containing datasets**
  - **28 M datasets**, cf. job input/output, data transport, merging results, can get created and deleted quickly
  - **1.24 B files** 🍷 with **1.35 B replicas** 🐱 — Our replication factor is heavily skewed because reasons
- **System heartbeat**
  - Average **#files in transfer queue** typically corresponds to # starting jobs: bursty **~200 K / 40 Hz** to **1 M / 150 Hz**
  - Average **#files in deletion queue** typically corresponds to # finished jobs + cache behaviour + administrative tasks: **~500 K / 60 Hz**
- **ATLAS organisational entries**
  - **346 RSEs** with a long history of migrations ... 659 deleted RSEs!
  - **8'773 Accounts** kept fresh from the ATLAS VO administrative service
  - **9'075 Scopes** mostly one scope per account, with dedicated group and activity accounts
  - **20'761 Identities** recent changes on X.509 distinguished names led to a bit of inflation
  - **#files unavailable** is embarrassingly high but also includes scheduled but not yet transferred files: hovers at **~2 M** 😊



# Our transfers :: Volume

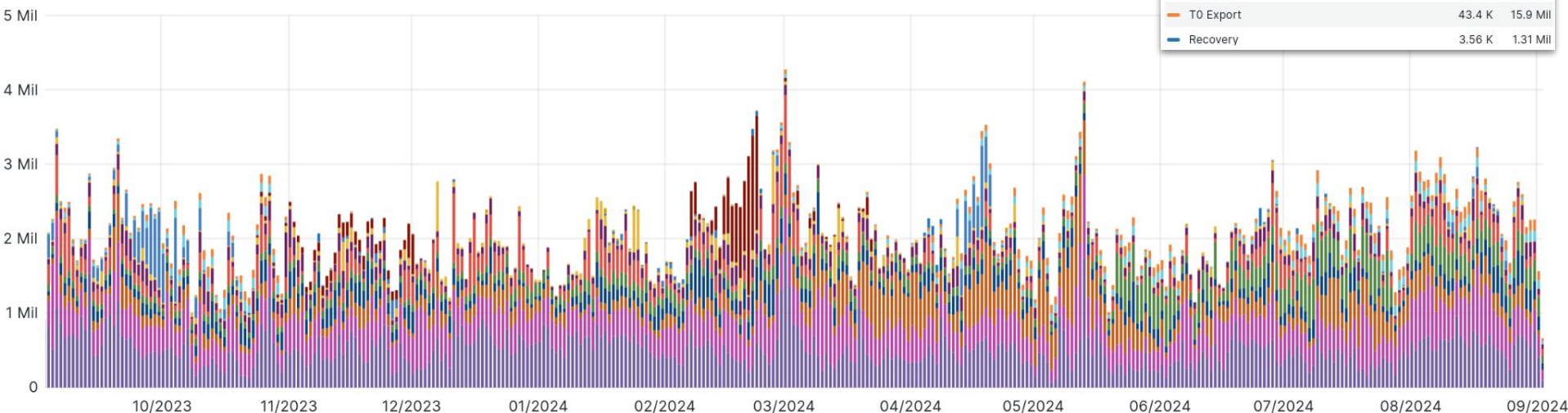
- More than half of our transfer volume due to production activity
  - We have to move this data
  - There is neither enough CPU nor storage for the textbook "just send the job to the data" case
- The remainder is a healthy mix of various experiment activities
  - Tier-0 Export, Consolidation of job outputs, rebalancing of data between sites, ...
  - Noticeable peak in February related to Data Challenge 2024



- Number of files transferred is a relevant metric for ATLAS
  - However, we need to have complete datasets at the destination
    - Doesn't help if we can move a lot of files, if they don't belong together
  - Average size of our datasets are in the order of 100 files
  - Typical input sizes for processing are tens of datasets
  - Usually 500-700'000 jobs concurrently in the system at any time

|                    | avg    | total    |
|--------------------|--------|----------|
| Production Input   | 514 K  | 189 Mil  |
| Production Output  | 418 K  | 154 Mil  |
| Data Consolidation | 262 K  | 96.2 Mil |
| Analysis Output    | 179 K  | 65.6 Mil |
| Functional Test    | 178 K  | 65.4 Mil |
| Staging            | 164 K  | 60.2 Mil |
| Analysis Input     | 129 K  | 47.2 Mil |
| User Subscriptions | 54.3 K | 19.9 Mil |
| Data Rebalancing   | 52.3 K | 19.2 Mil |
| Data Challenge     | 48.9 K | 18.0 Mil |
| T0 Tape            | 43.5 K | 16.0 Mil |
| T0 Export          | 43.4 K | 15.9 Mil |
| Recovery           | 3.56 K | 1.31 Mil |

Transfer Successes



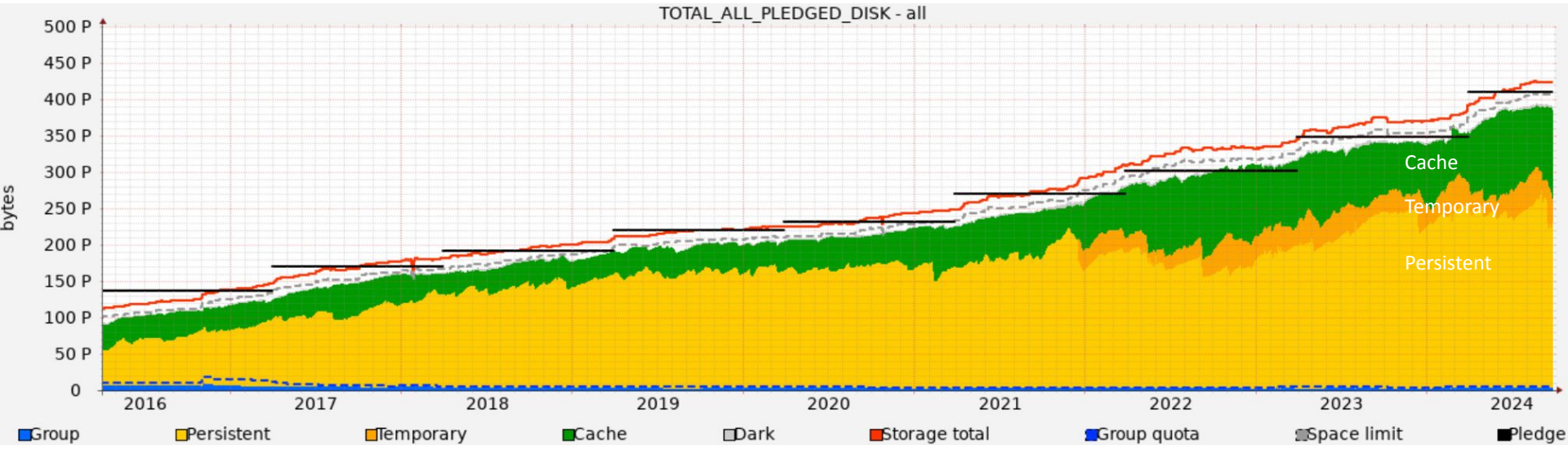


# Storage space evolution

- Storage is pledged per site to the experiment once per year
  - Typically (and hopefully deployed) in April
  - Pledges are set via CRIC, then storage reports actual values via SRR json
  - Split into various RSEs per site: DATADISK, DATATAPE, MCTAPE, LOCALGROUPDISK (unpledged), ...
- Various levels of data classification, three important ones
  - Cache** Files without any rule on them, can disappear at any time
  - Temporary** Rules that have a lifetime, e.g., ongoing physics campaigns
  - Persistent** Rules without lifetime, i.e., data we need to keep
- We have additional unpledged storage, e.g., from cloud R&D

```

storage-service:
  name: "AGLT2"
  id: "head01.aglt2.org"
  servicetype: "multidisk"
  implementation: "dCache"
  implementationversion: "9.2.7"
  qualitylevel: "production"
  latestupdate: 1727453701
  storage-capacity:
    online:
      totalsize: 16664434453774336
      usedsize: 11506780271051632
      reservedsize: 12056353649969656
  storage-endpoints:
    0: {}
  
```



# Replication policies

- ATLAS follows strict replication policies for most of its centrally managed data
  - Discussed by a team of senior people 😞
  - Written down in human-readable form in a TWiki, to be approved by the collaboration ✓
  - Implemented with two mechanisms, **subscriptions** and **RPG (Replication Policy on the Grid)**
- Curious incident this year where an overlooked RPG configuration led to ~60PB of data not being copied to tape over the last 3 years
  - Every now and then some jobs were using these data, causing two effects: (1) they stayed on disk and (2) no one noticed a problem

**data24\_13p6TeV**

All RAW data and data produced by T0 has a copy on the T0 EOS buffer (CERN-PROD\_DATADISK) with a lifetime of 3 weeks.

- RAW
  - 1 copy CERN-PROD\_RAW created directly by T0
  - 1 copy T1 DATATAPE
- AOD
  - 1 copy CERN-PROD\_DERIVED created directly by T0
  - 1 copy T1 DATATAPE
  - 2 copies on DISK (90 days lifetime)
- DAOD (derivations, produced on the Grid)
  - Initial output in NUCLEUS DATADISK
  - 1 copy T2 non-nucleus
  - DAOD\_PHYS and DAOD\_PHYSLITE: 4 replicas on any sites which can run analysis
- DRAW
  - 1 copy DISK (6 month lifetime)
  - 1 copy T1 DATATAPE
  - 1 copy CERN-PROD\_DERIVED created directly by T0
- DESD
  - 1 copy DISK (3 week lifetime)
  - 1 copy T1 DATATAPE
  - 1 copy on CERN-PROD\_DERIVED created by T0
- DAOD (produced at T0 from DRAW)
  - 2 copies on DISK
- HIST
  - 1 copy CERN-PROD\_DATADISK
- Calibration data
  - data24\_calib.\*.calibration\_MuonAll.\*.RAW
    - 2 copies
      - one on each of the calibration T2s: AGLT2\_CALIBDISK, INFN-ROMA1\_CALIBDISK
  - data24\_13p6TeV.\*NTUP\_MUONCALIB.\*
    - One copy to AGLT2\_CALIBDISK

| Name                                       | Ok     |
|--|--------|
| *MC, EVNT                                  | 309316 |
| Backup from INFN-T1                        | 2609   |
| data DAOD to non-nucleus disk              | 115718 |
| DATA TAGS to CERN DATADISK                 | 265    |
| data15_13TeV TAGS p2685 to CERN DATADISK   | 92     |
| Enhanced bias for HLT                      | 149    |
| EVENT to 2 T1s                             | 31103  |
| EVENT to 2 T1s 1 year                      | 43     |
| Full matrix functional tests               | 271591 |
| group.phys-gener to CERN-PROD_PHYS-GENER   | 5374   |
| MC TAGS to CERN DATADISK                   | 714    |
| MC15 DAOD to T2 disk                       | 7850   |
| mc15_13TeV TAGS p2671 to CERN DATADISK     | 6      |
| mc15_13TeV TAGS to CERN DATADISK           | 101    |
| mc15_13TeV TAGS to CERN DATADISK           | 1      |
| MC16 NTUP_PILEUP datasets to CERN DATADISK | 272780 |
| non-T0 HIST to CERN disk                   | 1423   |
| PHYS-GENER_inputs                          | 65904  |
| sit*PAC archival to TAPE                   | 1      |
| T0 AOD to non-nucleus                      | 327    |
| T0 AOD to non-nucleus (Data Consolidation) | 186    |
| T0 AOD to nucleus                          | 528    |
| T0 AOD to nucleus (Data Consolidation)     | 2545   |
| T0 AOD to nucleus (T0 Export)              | 1545   |

```

{
  "scope": [
    "mc.*.*TeV"
  ],
  "datatype": [
    "DAOD.*"
  ],
  "transient": [
    "None",
    ""
  ],
  "prod_step": [
    "merge",
    "deriv"
  ],
  "did_type": [
    "DATASET"
  ],
  "split_rule": true
}

```

```

{
  "lifetime": 2592000,
  "copies": 1,
  "rse_expression": "(tier<3type=DATADISK&datapolicyanalysis=true&datapolicynucleus=false)",
  "activity": "Data Consolidation",
  "weight": "freespace"
}

```

Validation RDO, ESD, AOD and HIST 898 2

- ⚙️ MigrateArchiveData.conf
- ⚙️ MigrateArchiveMC.conf
- ⚙️ MigrateDataDAOD.conf
- ⚙️ MigrateMCDAAOD.conf
- ⚙️ MigrateSmallFilesToT2.conf
- ⚙️ MigrateToContainerMC.conf
- ⚙️ MigrateToContainerMC15Evt.conf
- ⚙️ MigrateToTapeData.conf
- ⚙️ MigrateToTapeDataAOD.conf
- ⚙️ MigrateToTapeMC.conf
- ⚙️ MigrateToTapeMCReconAOD.conf
- ⚙️ MigrateToTapeMCT01.conf
- ⚙️ MigrateToTapeMCT2.conf
- ⚙️ MigrateToTapeValid.conf
- 📖 README.md
- 📄 RPG.conf.template
- 📄 RPG.crontab
- 📄 RPG.py

# Keeping storage under control

- Lifetime model

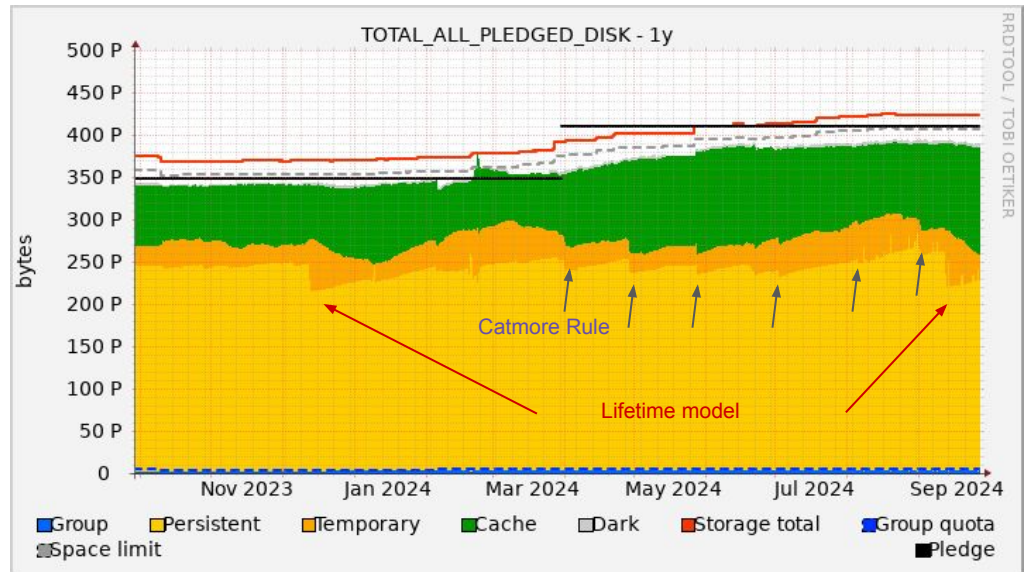
- Periodic deletion of old and unused data
  - Applied up to four times per year, with deletions spread over a two week period
  - Deletion policies are based on projects and data types
  - List of affected datasets is published to the collaboration
  - Users can then submit exceptions, e.g., for analyses in their review periods, typically 6 or 12 months
- Rucio keeps track via traces and Kronos daemon updates dataset's *updated\_at* field
- "Triangle"-like shape
  - "Yellow" to "Orange" immediately
  - Force deletion of replicas



James Catmore,  
U Oslo

- *Catmore Rule* application

- Disk datasets with at least one complete replica on tape that have not been accessed in the latest 6 months are unlocked
  - Can then be deleted if and when there is shortage of free space
- Typically run once per month
- "Yellow" to "green" immediately

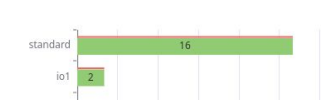




# Our Rucio deployment

- GitOps driven on CERN OpenStack with Kubernetes / helm / flux
- Private repositories on gitlab.cern.ch
- 3 production, 1 integration cluster
  - Now also have a small ARM allocation, will add 2nd integration cluster on ARM
- Most of this is very custom and not generally applicable, but come and talk to us, most likely we've encountered your problem before 😊

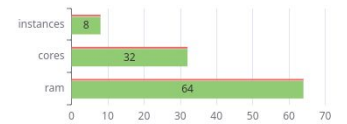
Number of Volumes



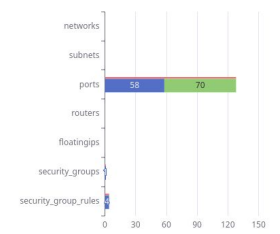
## Compute



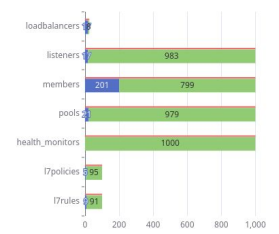
Compute



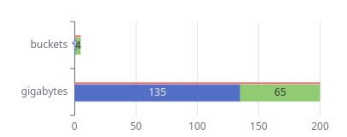
## Network



## Load Balancer



## Object Storage



### 4) Allow IP-in-IP traffic from our load balancers

Verify the node list in [felixconfigurations-patch.yaml](#), then configure calico to accept IP-in-IP encapsulation

```
kubectl patch felixconfigurations default --type merge --patch-file felixconfigurations-patch.yaml
```

### 10) Add new servers to load balancers:

Check that load balancers were created correctly. The same IPs should be found in the service external ip and on openstack side:

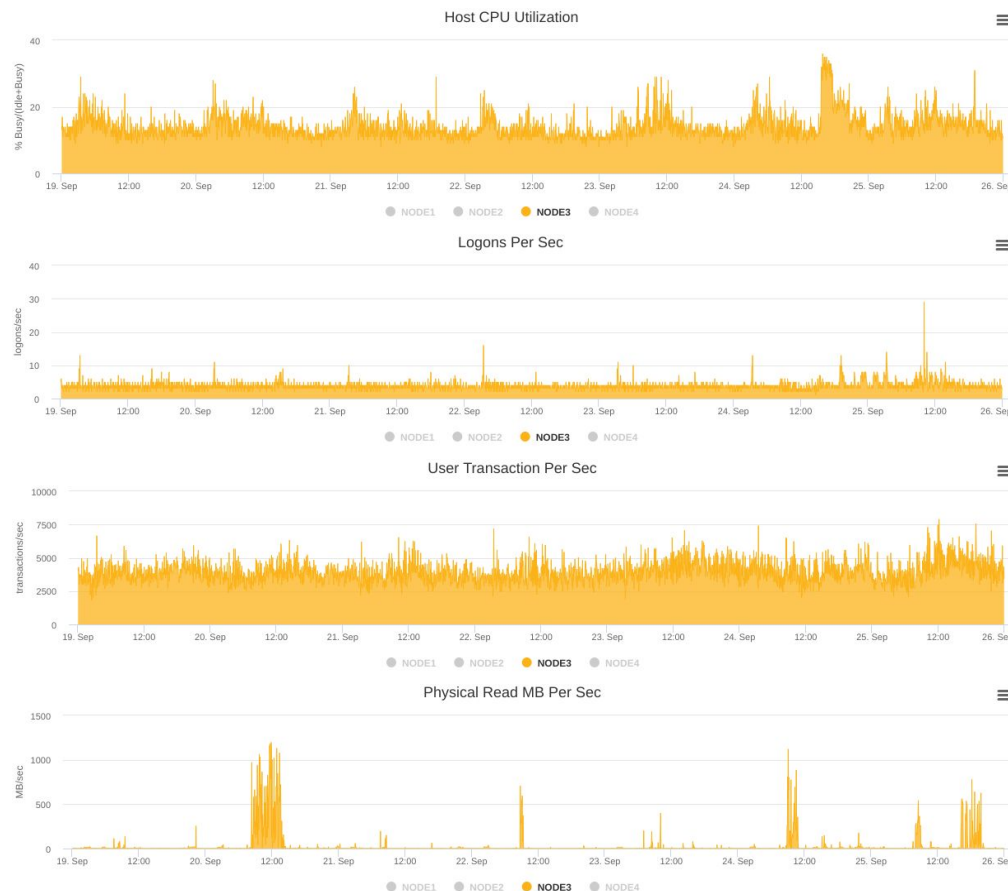
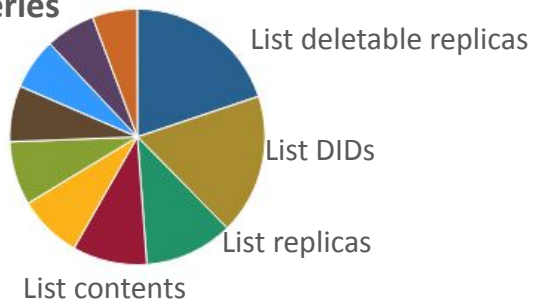
```
kubectl --namespace rucio get services
openstack loadbalancer list | grep $(openstack coe cluster show atlas-rucio-prod-99 -f json | jq -r '.UUID')
```

| Name                           | Last commit  |
|--------------------------------|--|
| clusters                       | add prod-01 flux config                                    |
| README.md                      | add instructions about updating the LBAAS load balancers   |
| create_cluster_template.py     | Force IPv6 support   |
| delete_lb.sh                   | update cluster creation documentation and scripts          |
| felixconfigurations-patch.yaml | add auth nodes to felixconfiguration patch                 |
| generate_flux_config.sh        | use different rucio customize entry point for each cluster |
| get_cluster_config.py          | select last cluster config by default                      |
| main.tf                        | use different addresses for purelb subnets                 |
| main.tf.auto.tfvars.json       | use different addresses for purelb subnets                 |
| post_setup_configure.sh        | Merge branch 'remove_thanos' into 'master'                 |
| renew_certificates.sh          | Rename CERN AuthZ app                                      |
| renew_fs_proxy.sh              | add simple script to trigger fs renewal jobs               |
| values-purelb.yaml             | use different addresses for purelb subnets                 |

| Name                | Last commit   |
|---------------------|---|
| ..                  |   |
| atlas-rucio-int-01  | use 1.25 config as default  |
| atlas-rucio-int-02  | fix cluster in which integration runs                             |
| atlas-rucio-prod-01 | fts-cron: remove test alma9 fts proxy renewal cron job from prod1 |
| atlas-rucio-prod-02 | use 1.25 config as default  |
| atlas-rucio-prod-03 | release-32.6.0.post1 PROD   |
| base                | Update secrets/longproxy  |
| integration         | weblint: bump to 35.3.0   |
| production          | Disable PROD Reaper on Tier-1 DATATAPEs and MCTAPEs               |

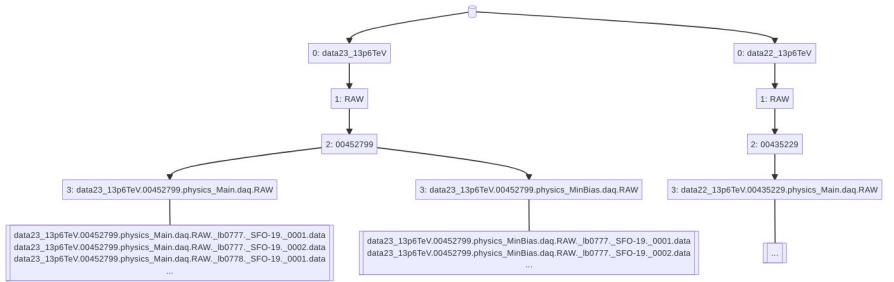
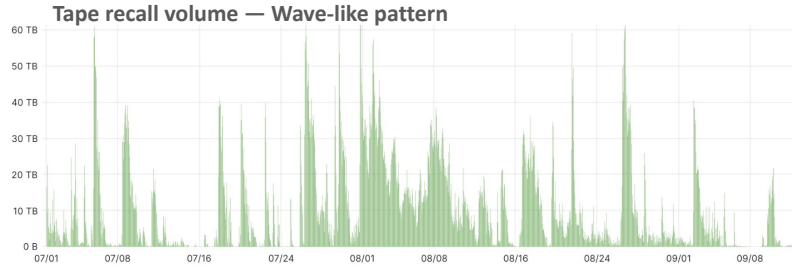
- Hosted on Oracle 19c
  - Long Term Support until 2027
  - 24/7 piquet service from CERN IT
  - CPU-based licence
  - ADCR Node 3 dedicated to Rucio
- Database interaction
  - Growth roughly 500GB / year
    - 80% of that goes to our \*\_history tables
  - Stabilised CPU level
    - Daemon sessions configured to minimise logons
  - Transactions are remarkably stable
    - Sometimes there's read spikes, but very rarely
    - No obvious impact on daily usage
  - Nevertheless, some improvements are needed
    - SQLAlchemy, cx\_oracle need updates
    - Transaction handling is *old school*

## Top queries



# Data Carousel and tape interactions

- We cannot recall all data for large campaigns at once
  - Production system requests dataset from tape via a single rule
  - Rucio schedules the recalls and then reports rule progress via message queue
  - Production system can then promptly processes a sliding window of data
  - Only a small fraction of inputs are pinned on disk at any time
  
- Archival metadata
  - Efficient tape reading depends on smart writing
  - Scheduling, collocation, and optional hints
  - Rucio policy package so we can define archive metadata
  - For tape destinations this metadata is passed via Conveyor to FTS to the storage
  - Site admins can then tailor their tape flush



```
archive_metadata = {
  "scheduling_hints": {
    "archive_priority": "100" # highest priority
  },
  "collocation_hints": {
    "0": "data23_13p6TeV", # project
    "1": "RAW", # datatype
    "2": "00452799", # runnumber
    "3": "data23_13p6TeV.00452799.physics_Main.daq.RAW", # dataset
  },
  "optional_hints": {
    "activity": "T0 Tape", # Tier-0/DAQ
    "3": { # dataset level
      "length": "19123", # total number of files at specified level
      "bytes": "80020799318456" # total size of files at specified level
    }
  }
}
```

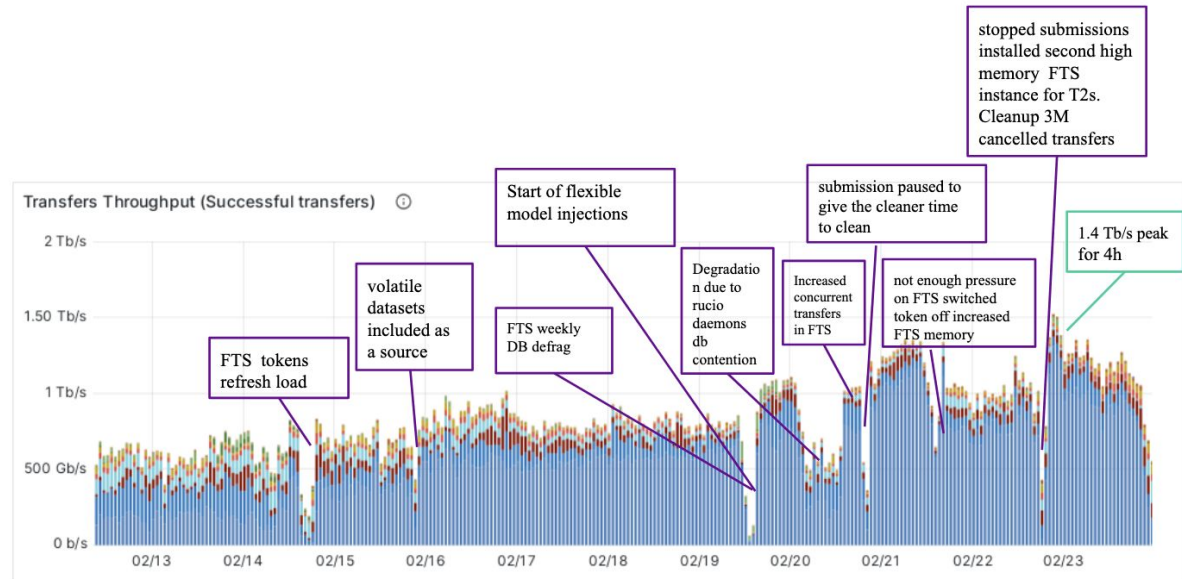
# Data Challenge 2024

- WLCG Data Challenges demonstrate readiness for HL-LHC data needs
  - Increasing volume/rates, increasing complexity, new technologies
  - A data challenge roughly every two years until HL-LHC startup

|                       |                               |
|-----------------------|-------------------------------|
| Attempted Transfers ⓘ | Successful Transfers (%) ⓘ    |
| 44.8 Mil              | 68.33%                        |
| Failed Transfers ⓘ    | Successful Transfers (vol.) ⓘ |
| 14.2 Mil              | 107.96 PB                     |

- In a nutshell

- 107 PB moved in ~12 days
  - Avg 0.82 Tb/s
  - max 1.4 Tb/s for ~4h
- None of the bottlenecks were due to the network
- FTS and Rucio central services affected the transfers much more
- Storage at sites also affected the rates either due to hardware, bugs, or tuning



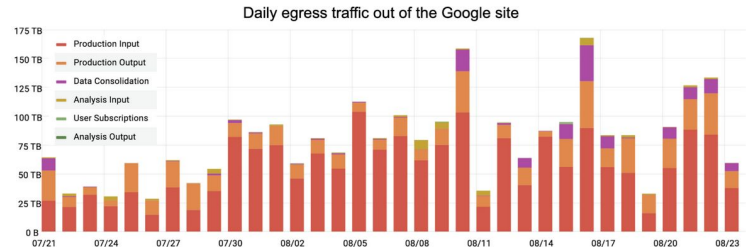
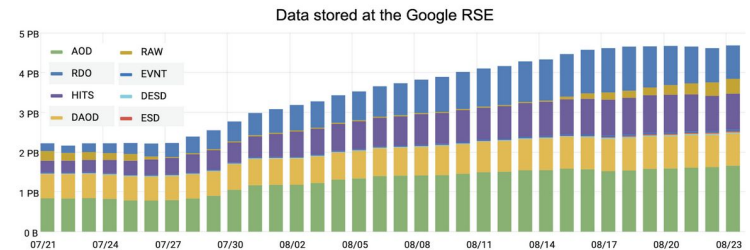
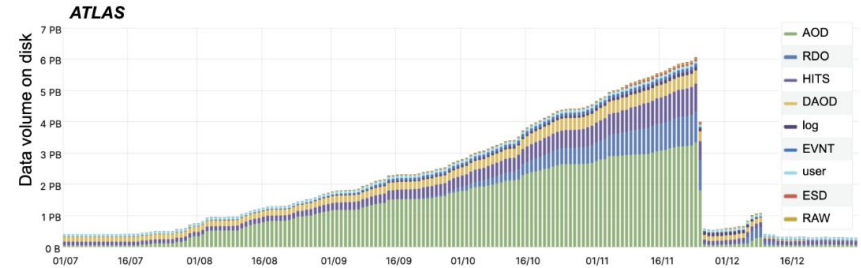
- Challenge design to push pushed the whole system
  - Used production Rucio & FTS infrastructure
    - Data Challenge traffic backfilling
  - `dc_inject.py`
    - Parsed unique lists of files from Rucio, manually created with `sqlplus` 🐱
    - Create rules with carefully selected rule parameters
- Scale
  - Number of sites
    - 9 Tier-1s and 57 Tier-2s
  - Injections every 15 minutes on ~1200 links
    - ~2000 links in total if we include production
    - Pushed FTS really hard to orchestrate
  - Short datasets lifetime **1h** -> **2h** -> **3h** to keep the space free
    - Pushed the deletions rates up
    - Pushed Rucio to maintain a balance between submissions and deletions
    - 3h interval caused some sites to run out of available space
- We had to repeat the RAW data Tier-0 Export to Tier-1s after the Data Challenge



- SEAL
  - cf. [Matt's](#) & [Mayank's](#) talks
  - 10 PB of storage provided for free for cloud R&D / integration with ATLAS [**Ongoing**]
- Amazon
  - R&D project through **UFresno** ✓
- Google Cloud
  - **Phase 1** Initial R&D ✓
  - **Phase 2** Evaluate the Total Cost of Ownership of employing a commercial cloud site at scale ✓
  - **Phase 3** Run ATLAS site pledge through GCS [**Ongoing at UTA**]

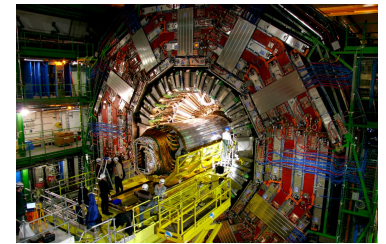
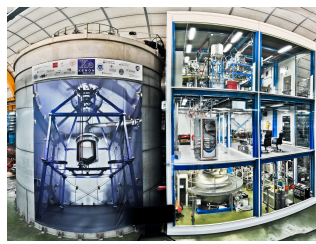
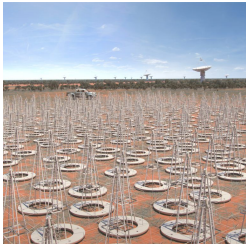
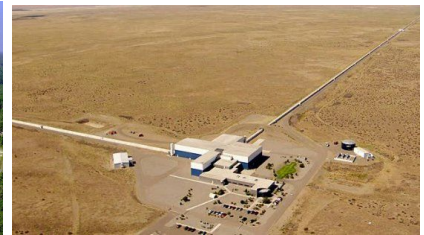
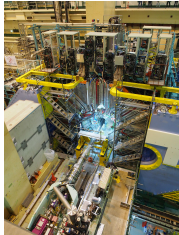
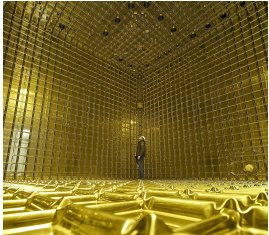
# Google Phase 2 data management observations

- Flat subscription agreement contract negotiated
  - Read everything about this in the [article](#)
    - Or go to CHEP'24 for the plenary :-)
- RSE configuration and Google
  - We started to accumulate significant data
    - Mostly AODs but also HITS, RDOs.. up to 6PB
    - Resulting egress up to 300 TB / day
  - Re-establish control in two ways, but no longer "grid-like"
    - Data greedily deleted
    - Far distance of the Google RSE to all other ATLAS sites
    - Reduced egress by 95%
- In the last few months we scaled up again
  - Grid-like behaviour again for our sites in DE
  - Slight increase in DIDs without rules
  - Plateaued as expected related to available CPU in DE



# Multi-experiment data management

- Shared use of the global research infrastructures have become the norm with sciences at the scale of HL-LHC, DUNE, or SKA
  - Competing requests on a **limited set of storage and network**
  - Many data centres are already supporting **multiple experiments**
  - **Compute** seems well-covered, but **data** was always missing a **common solution** for our **shared challenges**
- Ensure **more efficient use of the available data resources**
  - **Allocate storage and network based on science needs**, not based on administrative domains
  - **Orchestrate dataflow policies across experiments**
  - Dynamically support compute workflows with **adaptive data allocations**
  - **Unify monitoring**, reporting and analytics to data centres and administration
  - Potential for **shared operations across experiments**
- **Allows more efficient use of the available resources while giving the sciences tangible schedules**
  - My dream is still to have Rucio instances across experiments **interact with each other**



- Rucio is working great for ATLAS
  - Our one-stop-shop for all our data needs
  - Thanks to the dedication of a great team
  - We are happy and grateful to be part of this community!
- Rucio is a fundamental technology for ATLAS
  - Allows us to do what we need to do
  - Supports us to explore new possibilities in data management
- The ATLAS data needs are increasing and evolving
  - Even more complex data flows to support our physics use cases
  - Throughput and file rates are ever increasing
  - And as usual, ATLAS has some crazy R&D projects to keep things interesting
- ATLAS will continue to contribute to and support Rucio!

