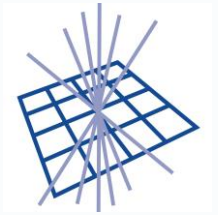
Science and Technology Facilities Council
Rutherford Appleton Laboratory



# LHCb status

GRIDPP 49

ALEXANDER ROGOVSKIY

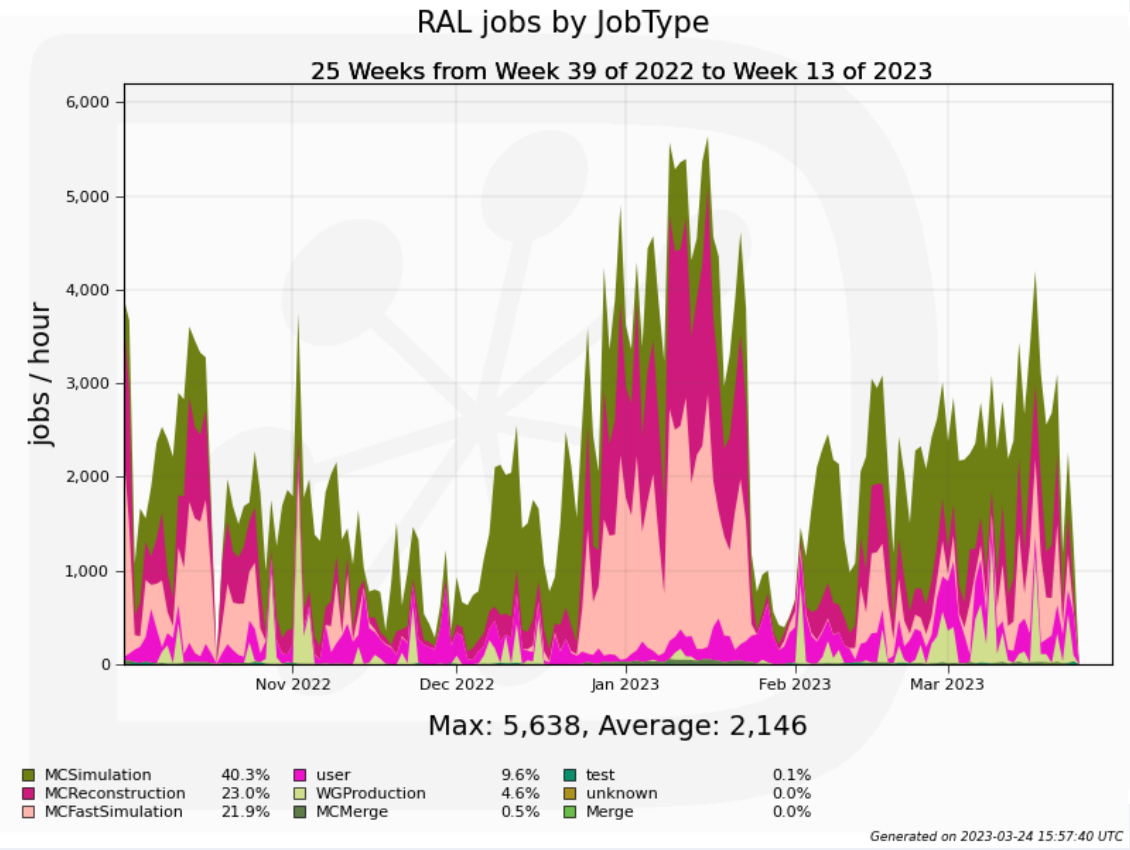
# Contents

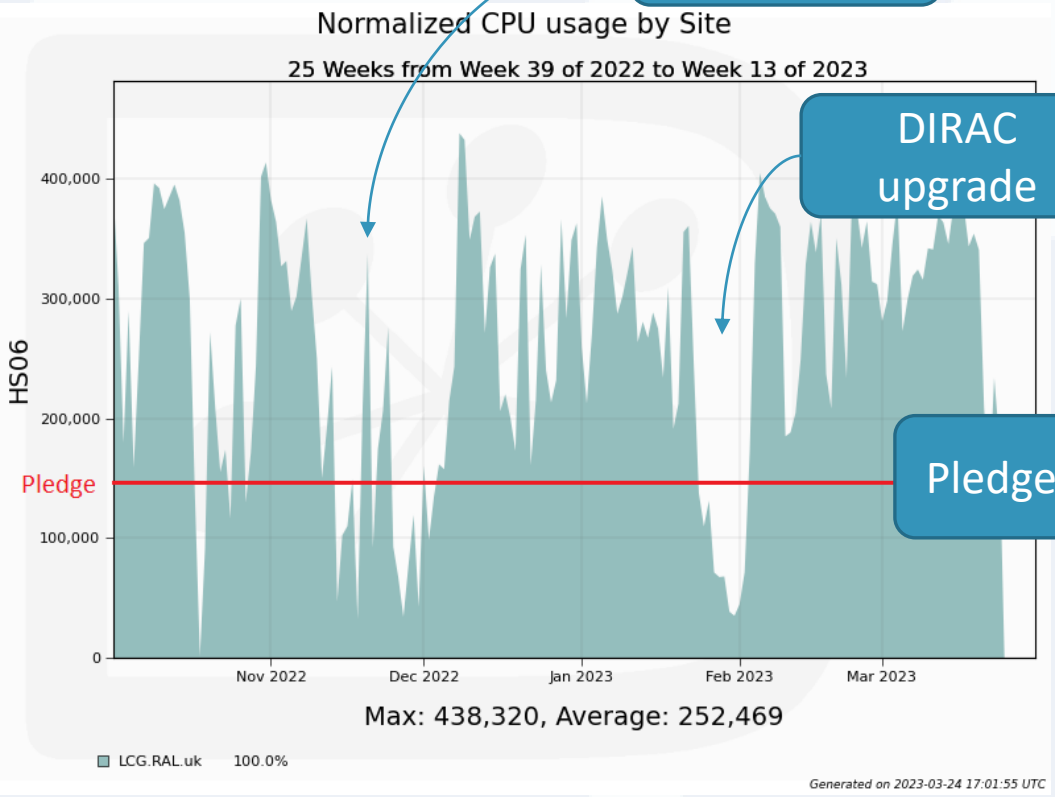
In this talk I'll try to cover the following topics:

1. LHCb T1 resource usage for the last half year.
2. LHCb activities.
3. RAL T1 problems and solutions.
4. LHCb T2 resource usage.
5. LHCb plans.

# LHCb Jobs at RAL T1

The majority of LHCb jobs at RAL for the last 6 months are Monte Carlo jobs.




Lack of Productions

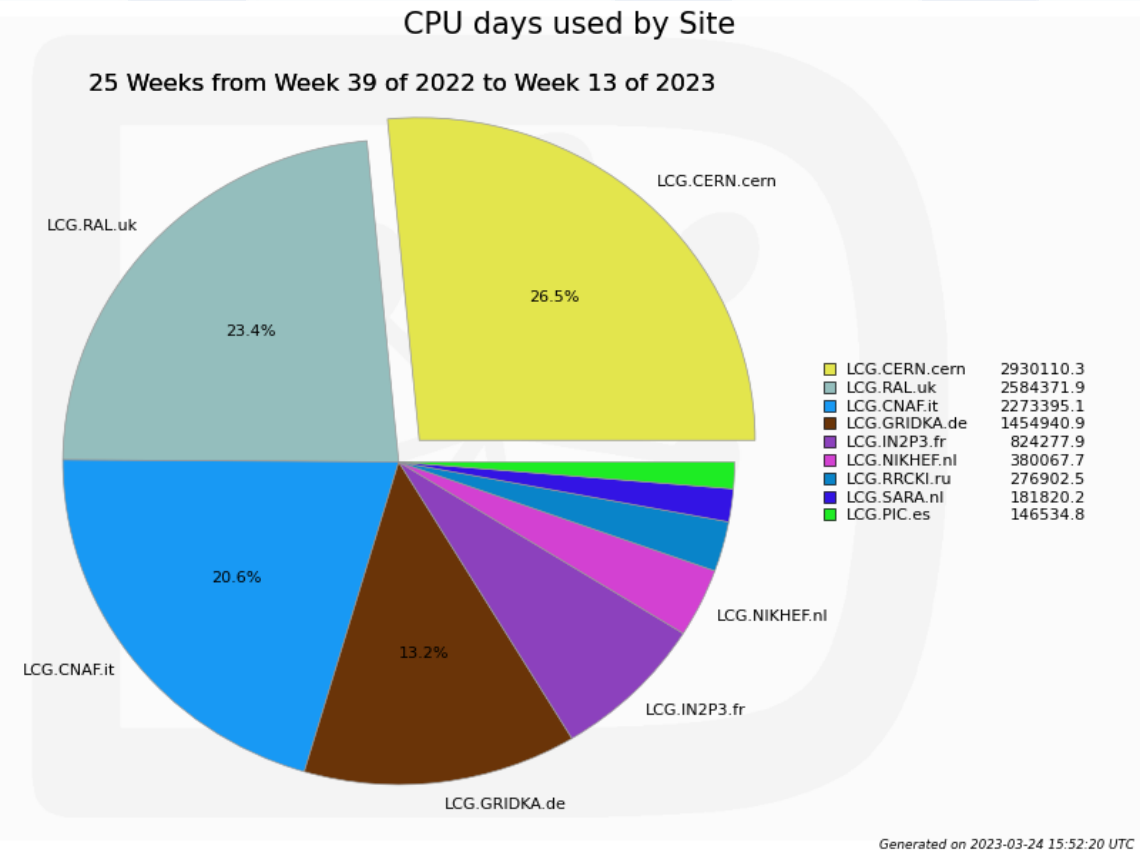
DIRAC upgrade

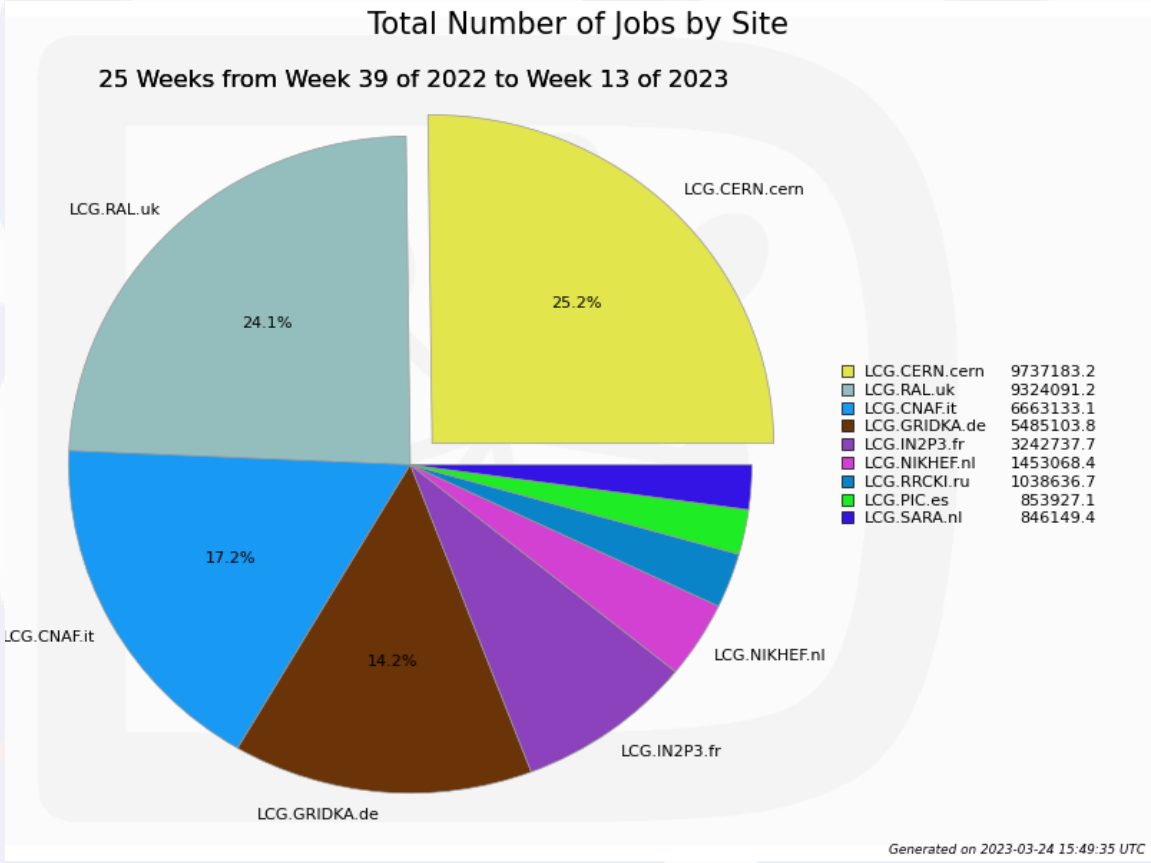
Pledge: 140k

# Comparisson

Everything is relative. How these results are compared with other T1s/CERN?

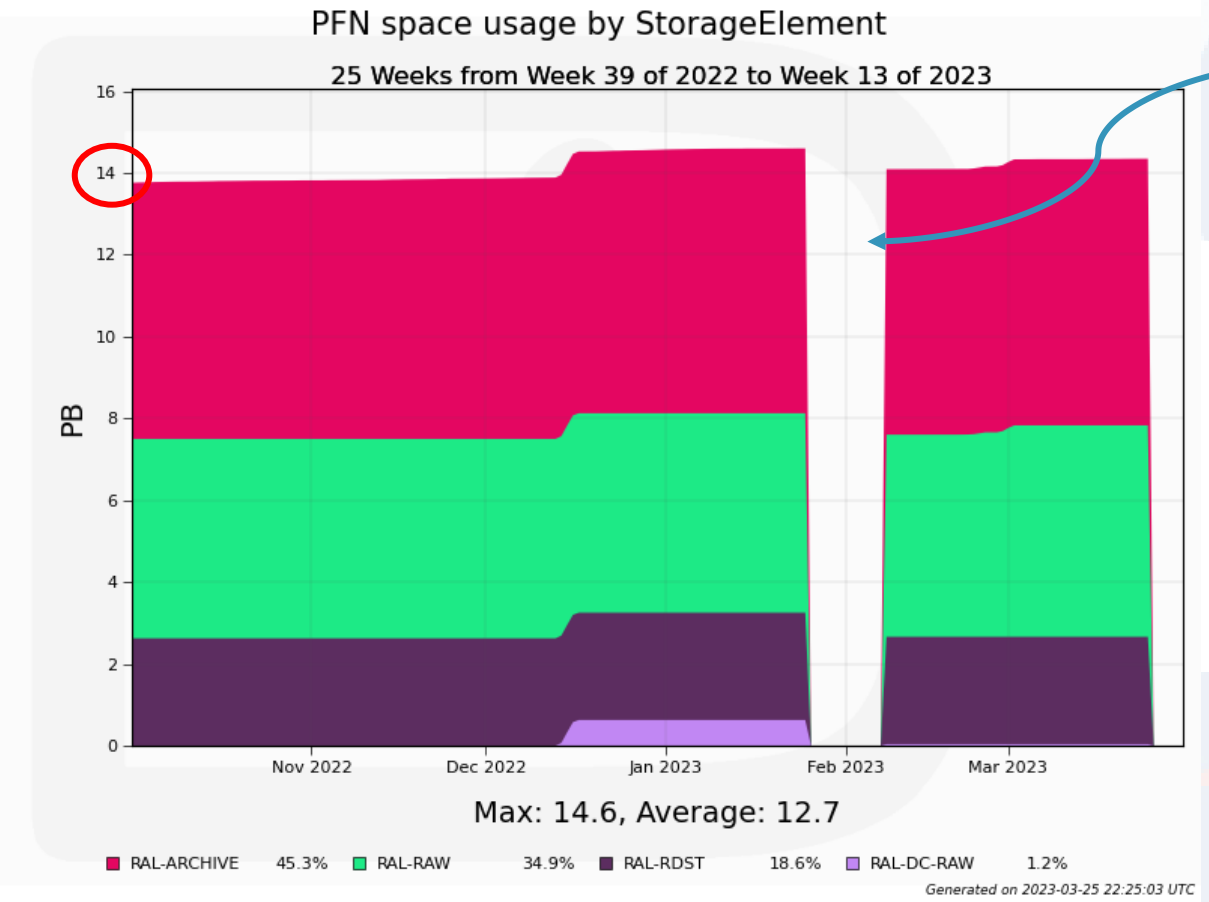
RAL T1 has provided for LHCb the most computational resources amont T1s, in terms of executed jobs and CPU time. Almost as much as CERN.



# Tape usage

Tape usage is significantly below the pledge. This is due to lack of data from the detector last year.




Major DIRAC upgrade

## Tape Space - TBytes

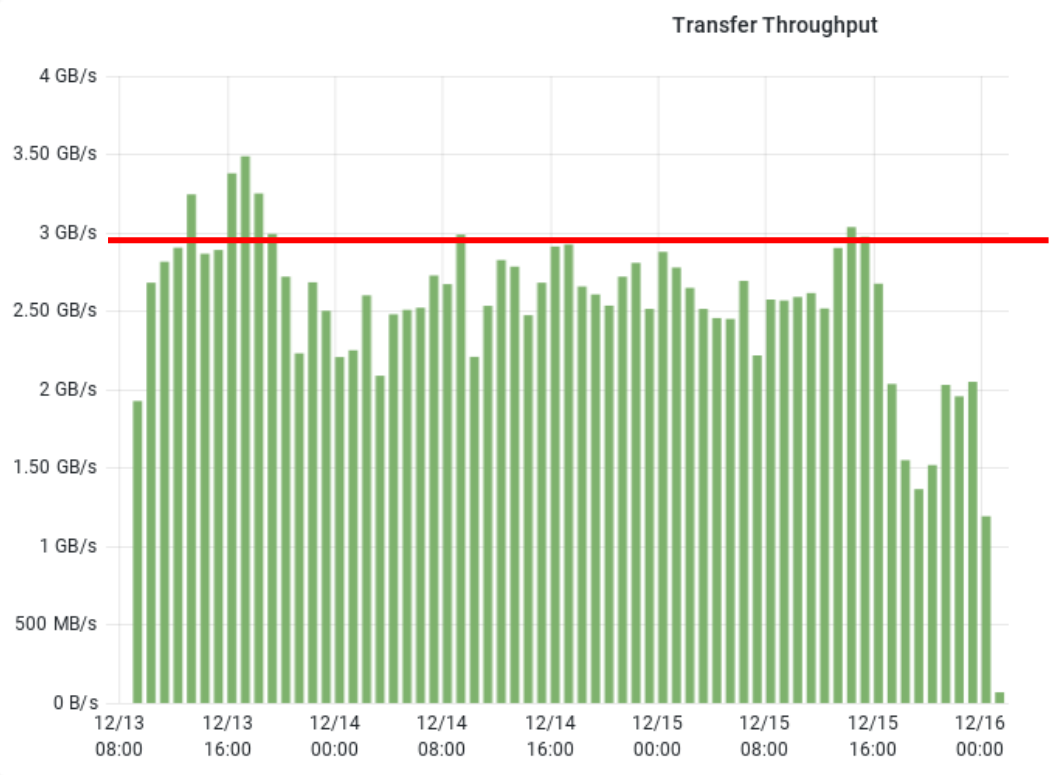
Search:

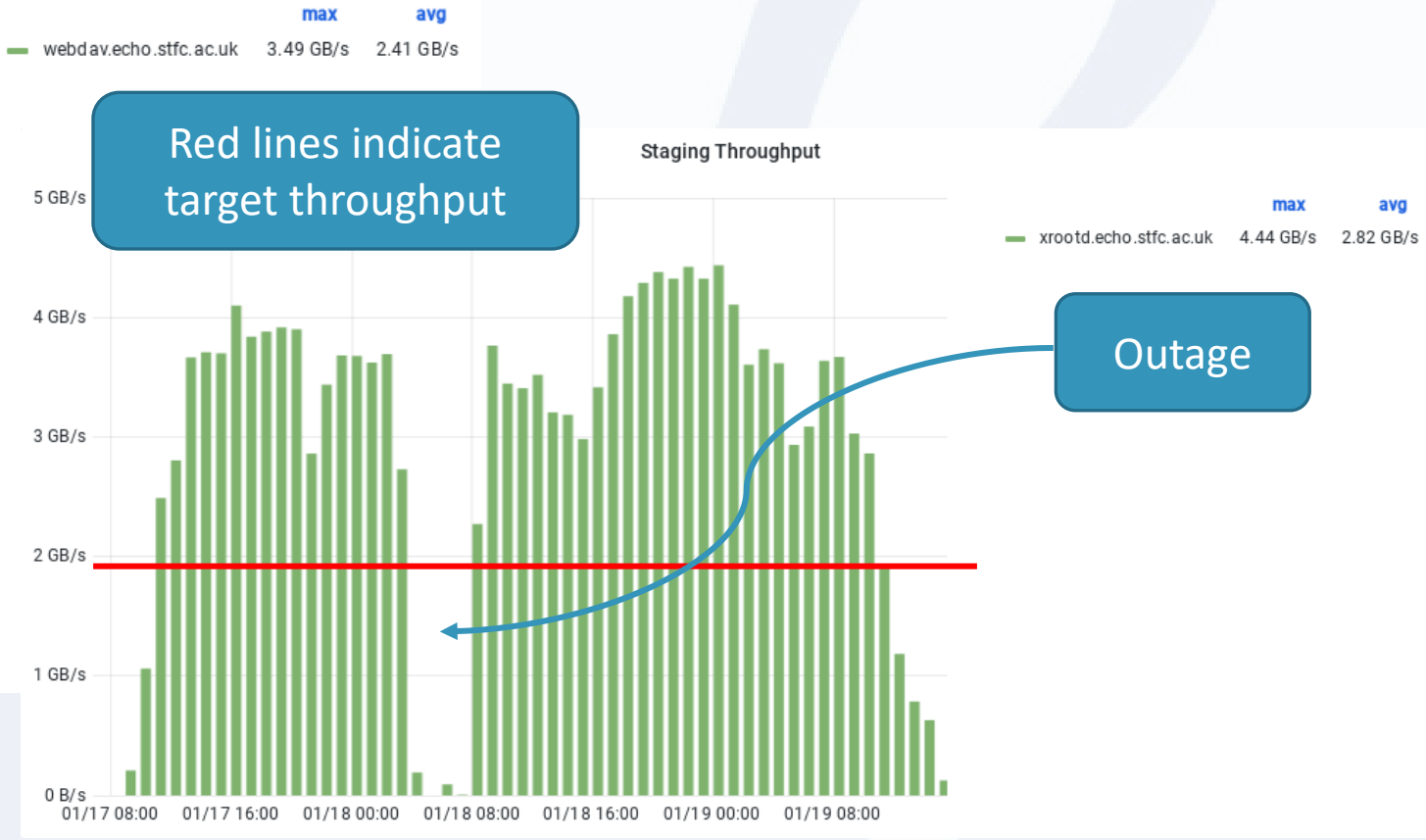
| Tape Space - TBytes | Oct 2022 | Nov 2022 | Dec 2022 | Jan 2023 | Feb 2023 | Total | % MoU |
|---|---|---|---|---|---|---|---|
| LHCB | 13,800 | 13,870 | 14,550 | 14,090 | 0 | 56,310 | |
| **Total** | **13,800** | **13,870** | **14,550** | **14,090** | **0** | **56,310** | **34%** |
| installed capacity | 0 | 0 | 0 | 0 | 0 | 0 | |
| MoU pledge | 32,776 | 32,776 | 32,776 | 32,776 | 32,776 | 163,880 | |

# Data challenge

We had one more data challenge for RAL T1 recently

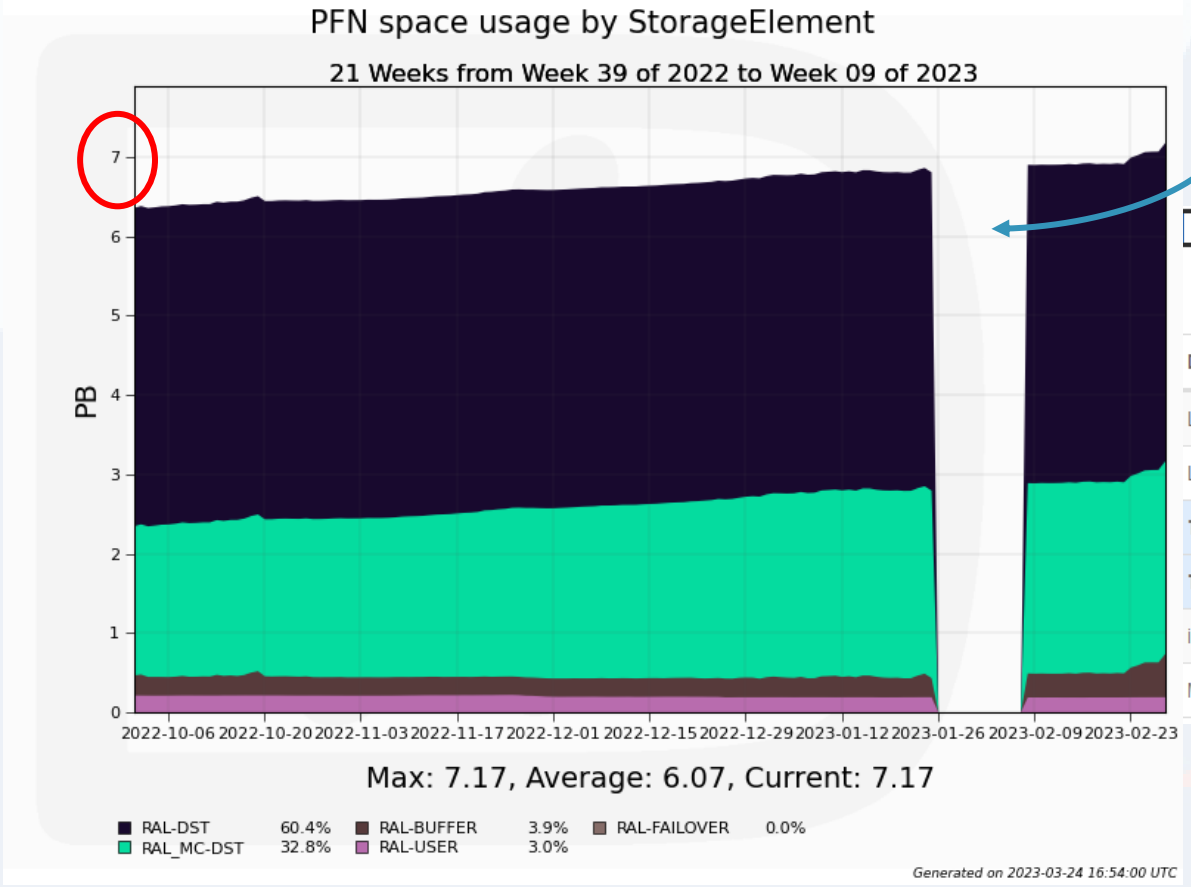
- During the writing part we almost achieved target throughput
- Reading/staging throughput was well above the target



# Disk usage

About 60% of allocated disk space is used. Interesting that there is a discrepancy between WLCG accounting and LHCb accounting




Major DIRAC upgrade

Disk Space - TBytes

Search:

| Disk Space - TBytes | Oct 2022 | Nov 2022 | Dec 2022 | Jan 2023 | Feb 2023 | Total | % MoU |
|---|---|---|---|---|---|---|---|
| LHCB allocated | 12,474 | 12,474 | 12,474 | 12,474 | 12,474 | 62,370 | |
| LHCB used | 7,359 | 7,476 | 7,608 | 8,003 | 7,943 | 38,388 | |
| **Total allocated** | **12,474** | **12,474** | **12,474** | **12,474** | **12,474** | **62,370** | **100%** |
| **Total used** | **7,359** | **7,476** | **7,608** | **8,003** | **7,943** | **38,388** | **62%** |
| installed capacity | 0 | 0 | 0 | 0 | 0 | 0 | |
| MoU pledge | 12,474 | 12,474 | 12,474 | 12,474 | 12,474 | 62,370 | |

# Problems: consistency checks

The last consistency check for LHCb storages was ages ago. Luckily, recently it was done.

- Antares
  - Several lost and dark files
- ECHO
  - Several hundreds of lost files
  - ~750k files or ~1PB of dark data
    - Some parts of this dark data originated from CASTOR -> ECHO migration
    - The origin of other parts is unknown (deletion problems?)
    - Waiting for a final confirmation from the LHCb Data Manager to proceed with the deletions

# Problems: long term issues

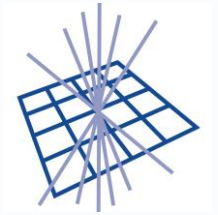
There are/were several long-lasting issues for LHCb:

1. Vector read (ticket).
2. Deletion problems (ticket, solved).
3. Slow stat calls (ticket).
4. Stub files (ticket), backup.

# Vector read issue

Vector read (aka readv) is an xrootd request that reads multiple chunks of file identified by offset and length.


These requests sometimes fail on ECHO, this causes job failures. This is a serious issue for LHCb since user jobs usually process a lot of files (up to 50). A single failure during the processing of the last file causes the whole job to fail, and results for the first successfully processed files are discarded.



# Vector read issue: problem

The error happens when readv operations takes too long

- There is a “stream timeout” in xrootd – if nothing is transferred in the data channel for the given amount of time, failure will be declared.
- It is possible to increase this timeout via environment variable.
- There were a lot of attempts to fix the issue this way, without any success.

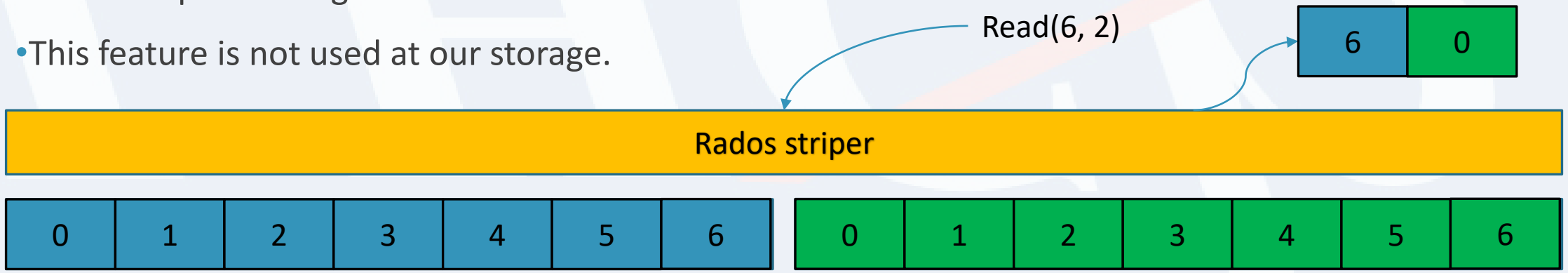
Example:

```
Error in <TNetXNGFile::ReadBuffers>: [ERROR] Socket timeout
```

# Vector read issue: how storage works

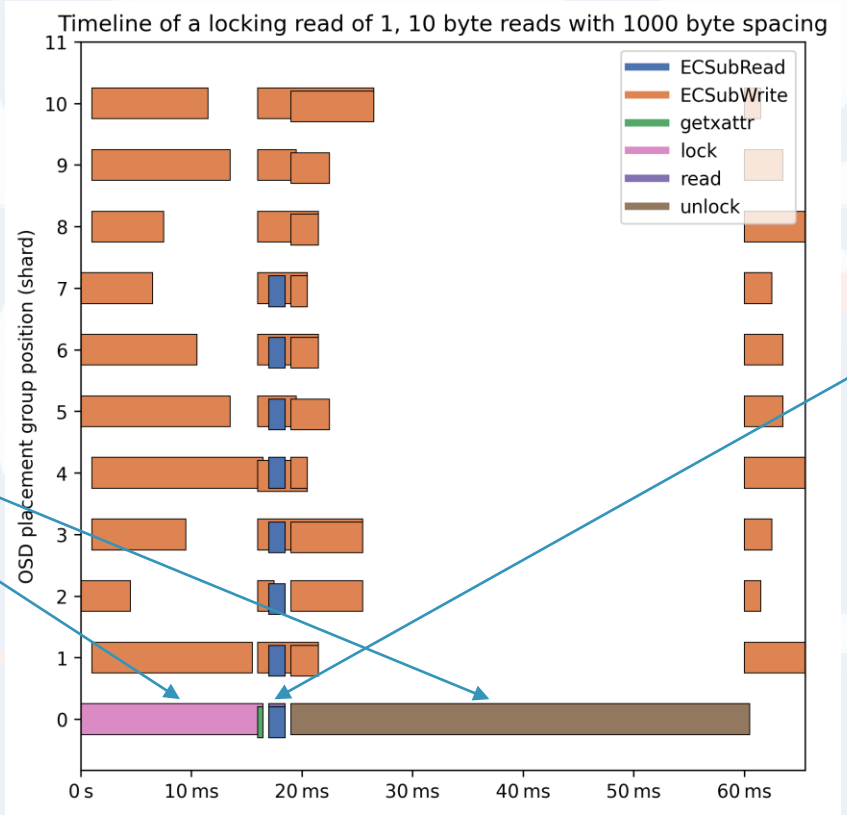
To understand the reasons for these slow reads, let's see how the files are stored on ECHO

- Data is stored in ceph
- Clients access data through gateway servers
- Every file is split into "blocks" (i.e. ceph objects) of size <= 64MiB.
- Rados striper library allow one to handle this transparently.
- Rados striper is designed to handle simultaneous reads and writes of the same file.
- This feature is not used at our storage.



# Vector read issue: when read is not read

Rados striper is designed to handle simultaneous reads and writes (not useful for our storage), that's why every read operation actually involves writing of the lock. That's why reads are so slow. Many thanks to Tom Byrne for the plot and for the investigation!



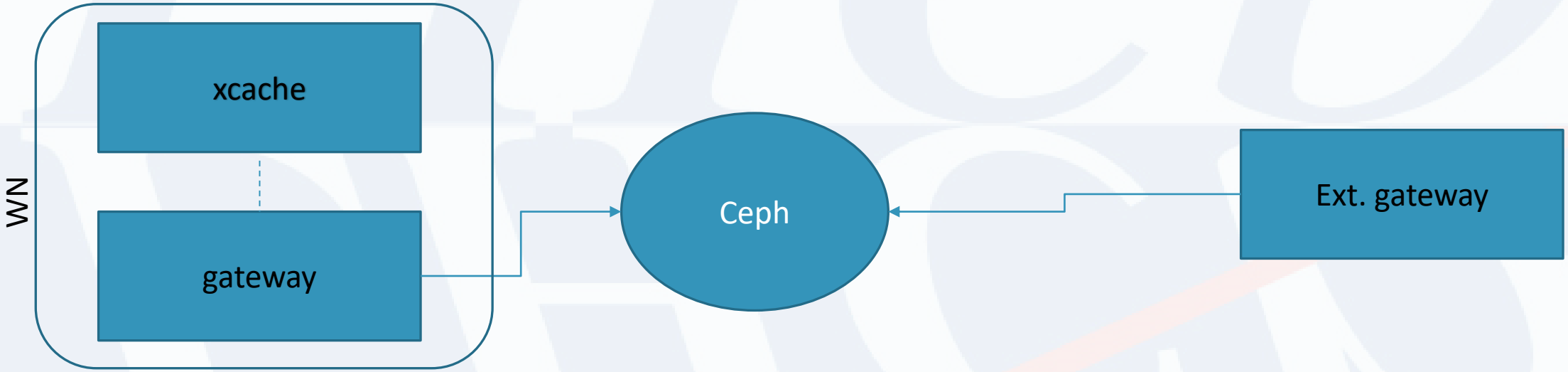

Lock and unlock Take a lot of time

The actual read is comparatively quick

And vector read requests are executed sequentially (i.e. requested chunks are read sequentially), which makes the problem even worse.

# Vector read issue: current configuration

To mitigate the problem, RAL's WNs have different configuration (from external gateways). There is an additional caching instance of xrootd, which tries to read data from ceph using huge blocks.




This mitigates the problem but does not resolve it completely (timeouts still happen sometimes). Furthermore, this configuration introduces additional errors.

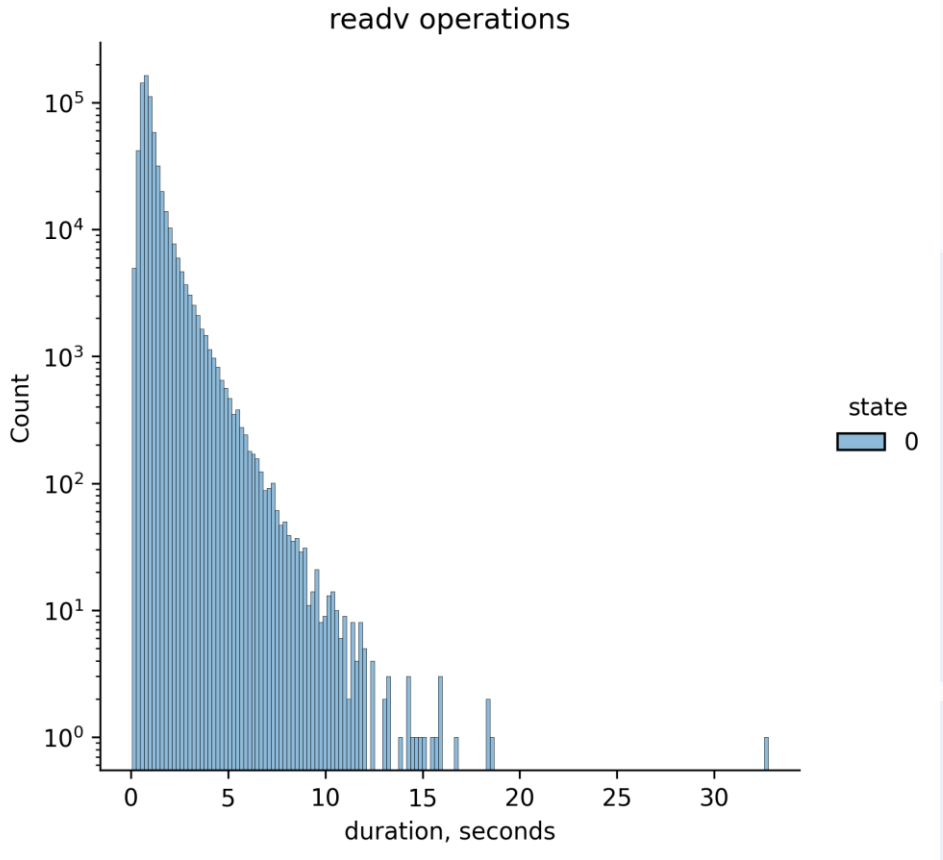
# Vector read issue: non-striper reads

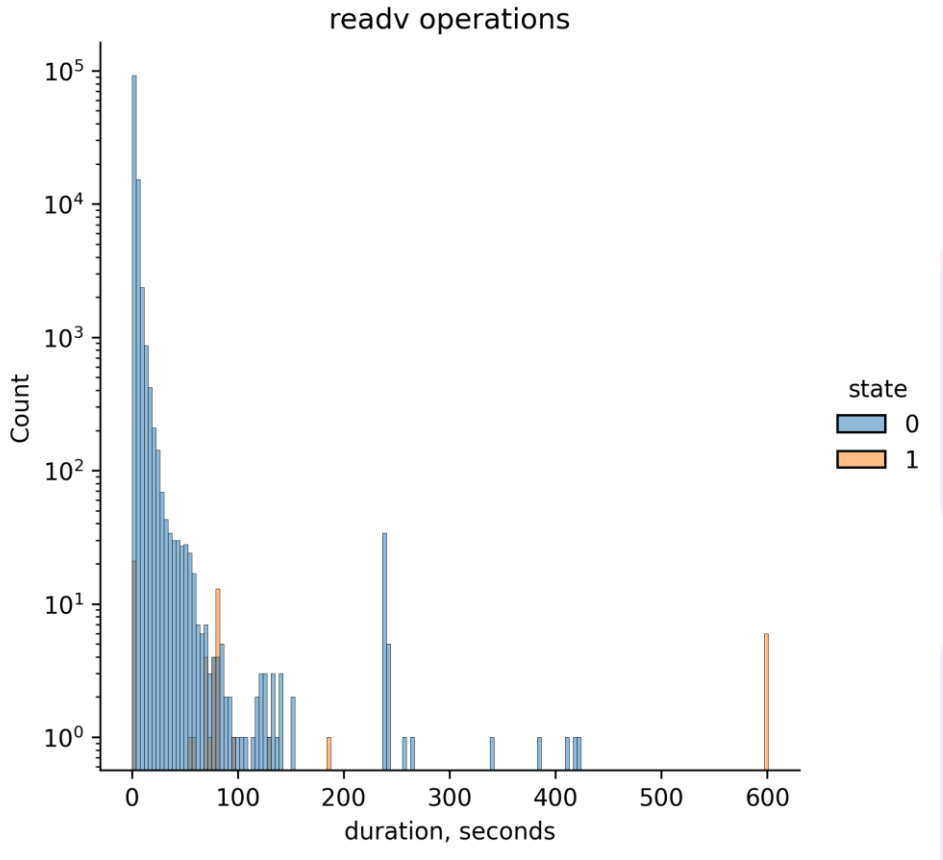
Can we remove this library for reads? Yes! This helps to improve the performance significantly

Ext. GW

| Test type | No striper | No changes |
|---|---|---|
| Time | 11m0.818s | 323m1.111s |

WN test



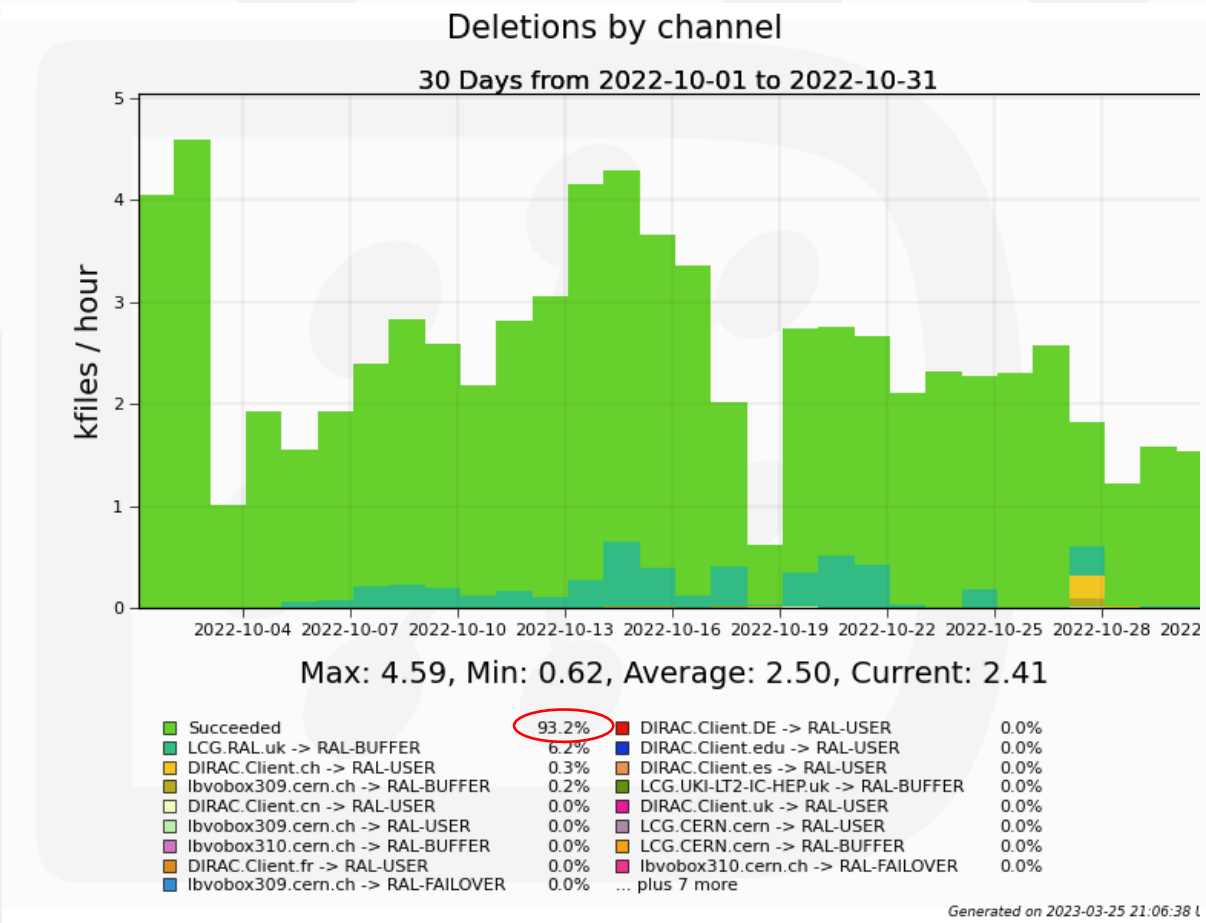
# Vector read issue: plans

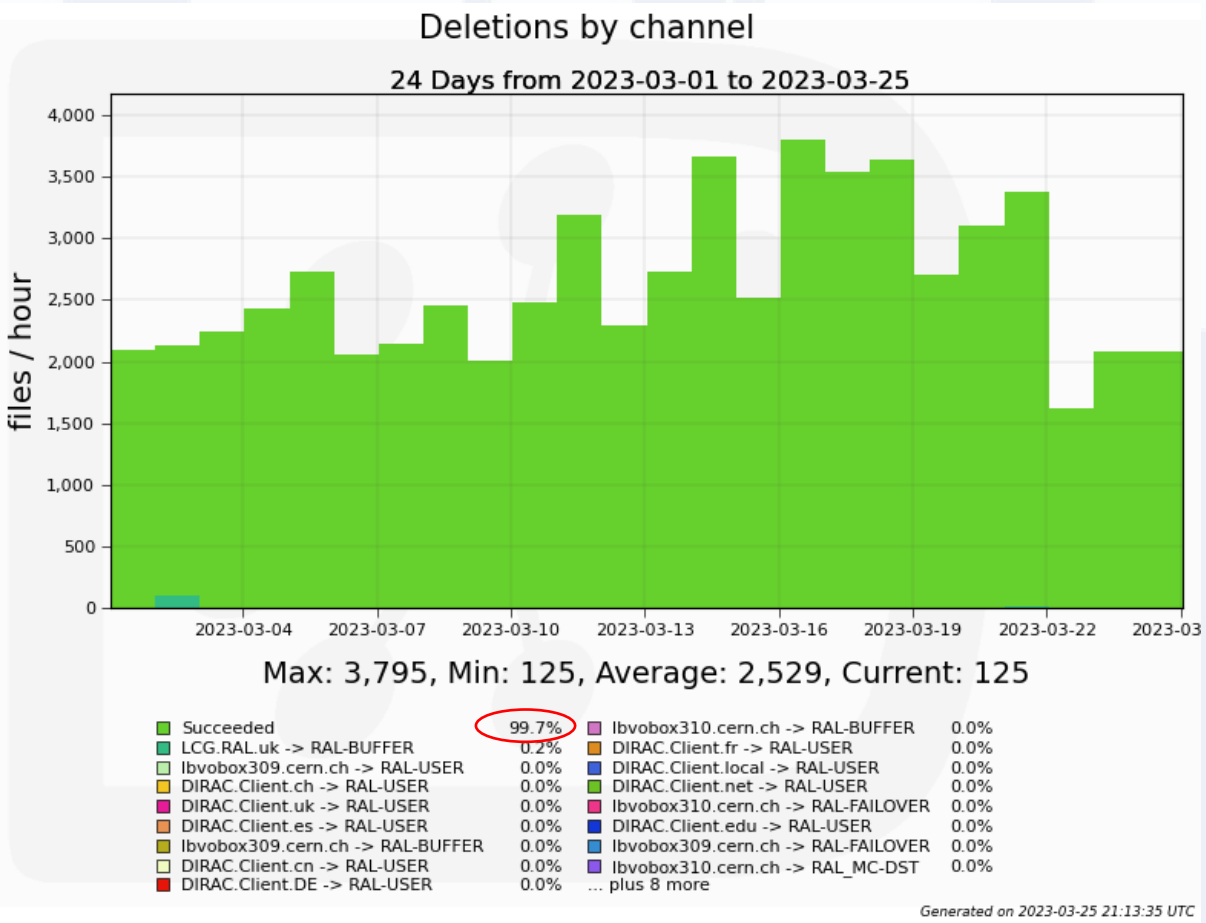
- There is a patch that removes rados striper for reads.
- The patch was applied on several test WNs ~1.5 months ago, since then there were no job failures due to vector read issues.
- Though the number of user jobs is not very high -- ~1500.
- Test with other VOs (waiting for CMS).
- Code Review is ongoing, feel free to comment!
- Large-scale test is planned once it is finished.
- Many thanks to all involved: Alastair Dewhurst, James Walder, Jyothish Thomas, Raja Nandakumar, Robert Currie, Steven Simpson, Tom Birkett, Tom Byrne et al.

# Deletion problems

For a long time failure rate of the deletion operations on ECHO was high.

The issue was solved by xrootd development and by tuning timeouts on the client side.



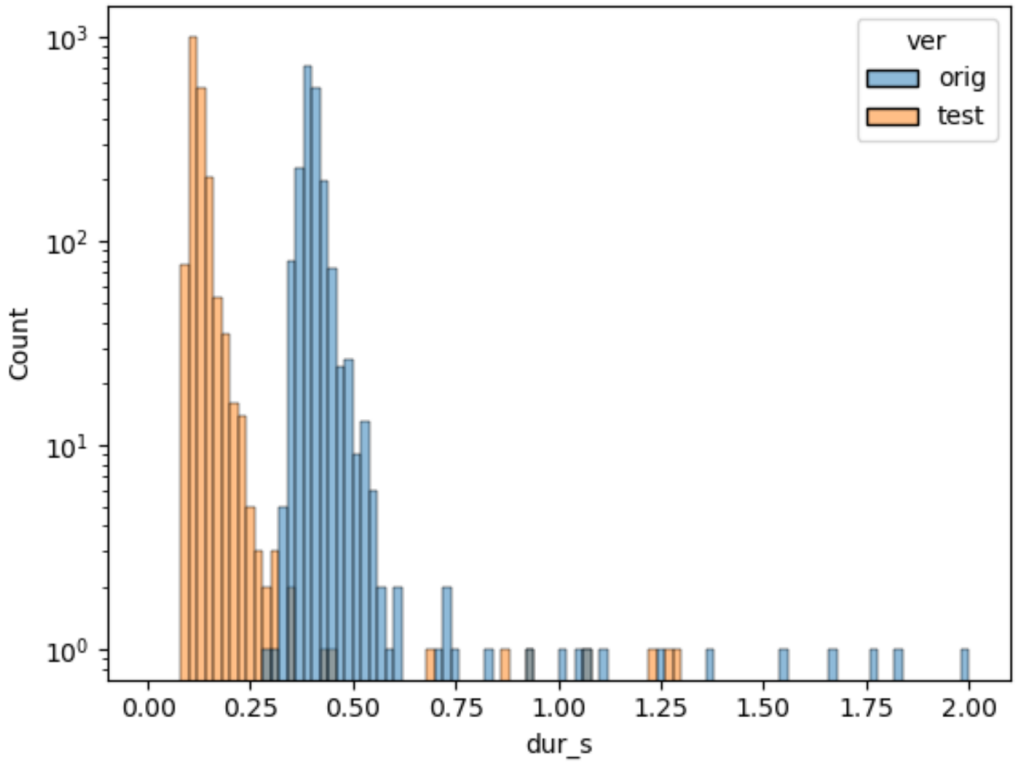
# Slow “stat” calls

There is a ticket with a complaint about “Slow stats” at RAL’s ECHO.

It turned out that it is not the stats that are slow, but the checksum requests (different request in terms of xrootd).

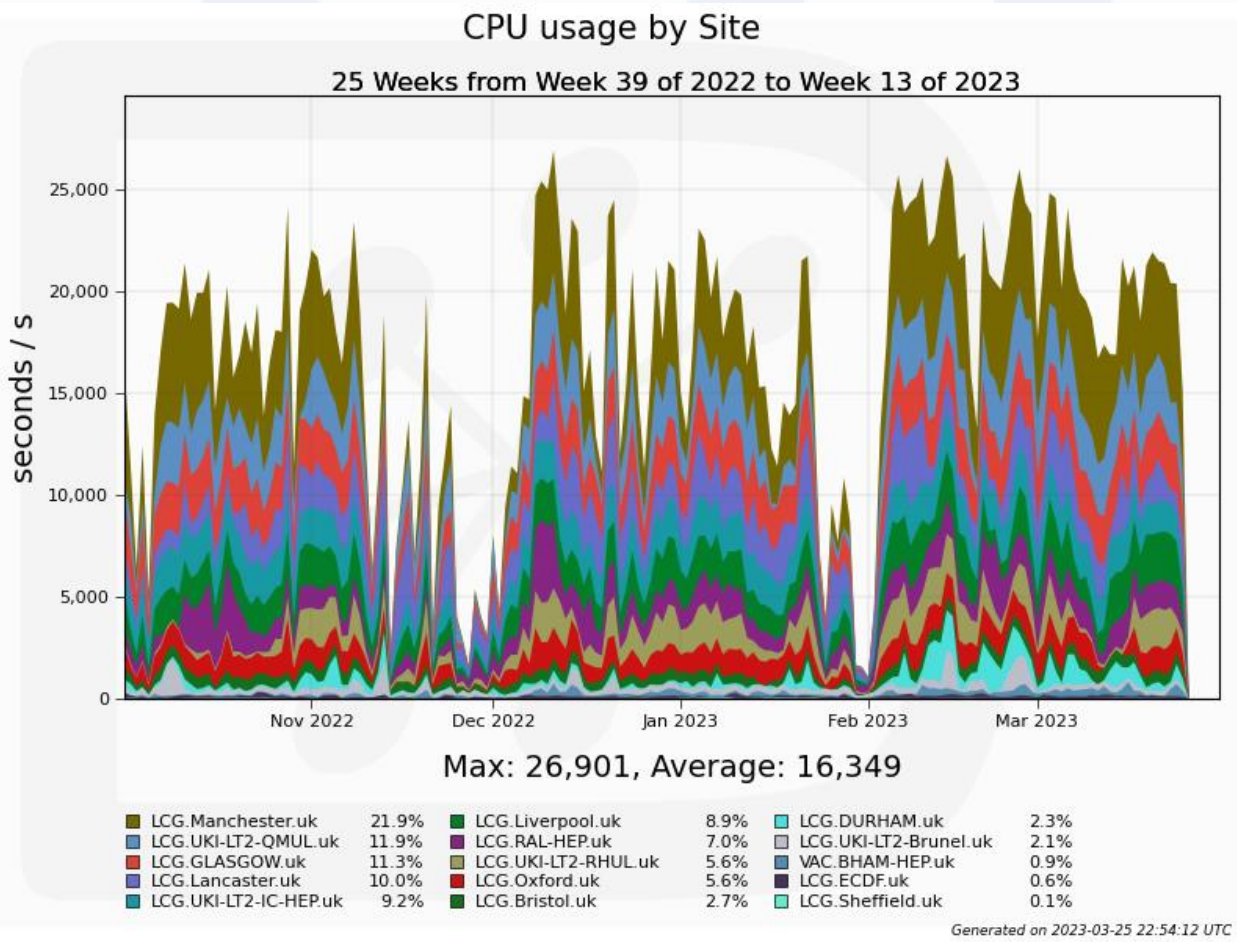
Xrootd development is ongoing (thanks to James W)

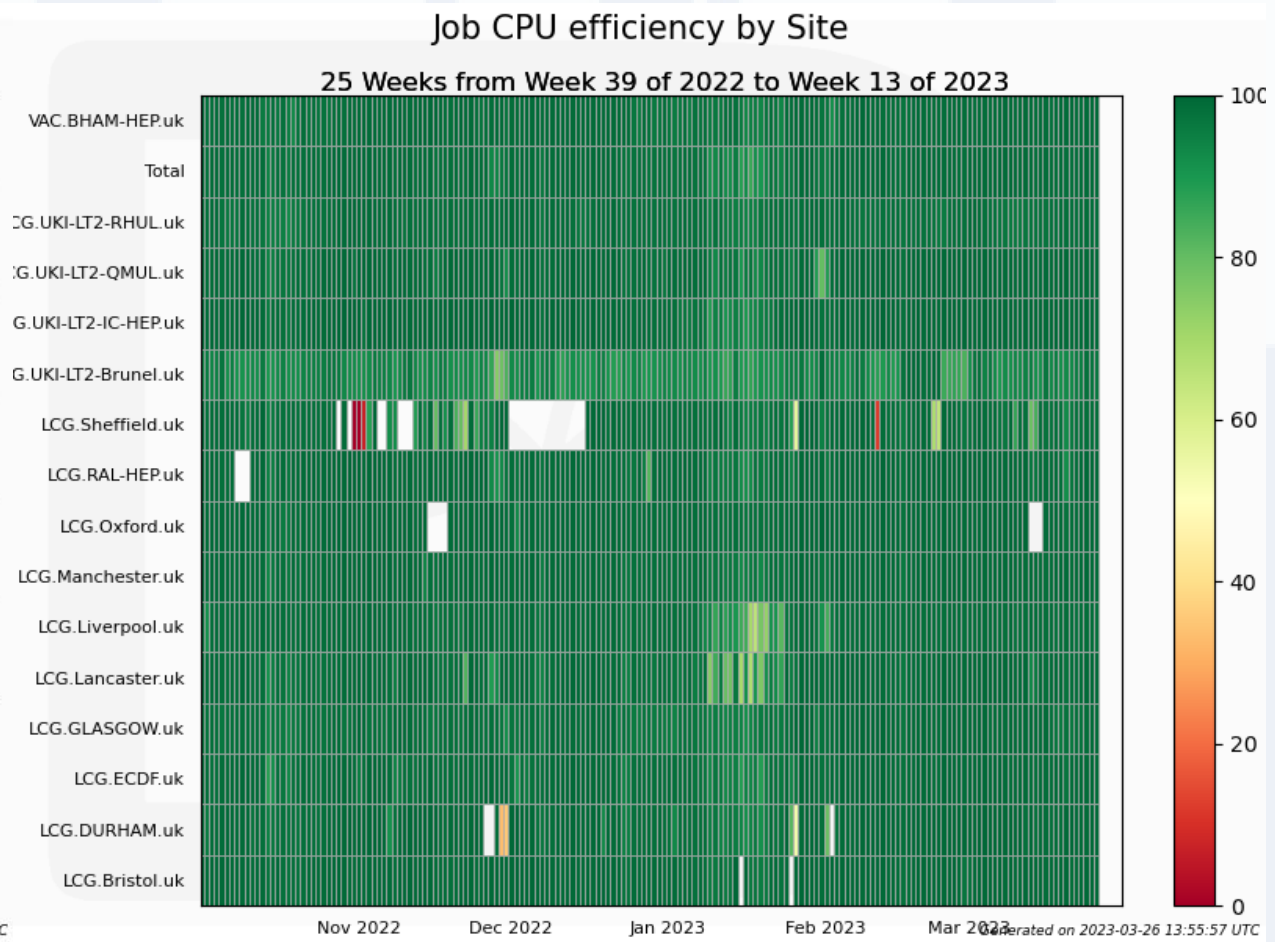
- There is a patch that is being tested right now



# Tier-2 statistics

Tier-2 jobs efficiency also looks OK

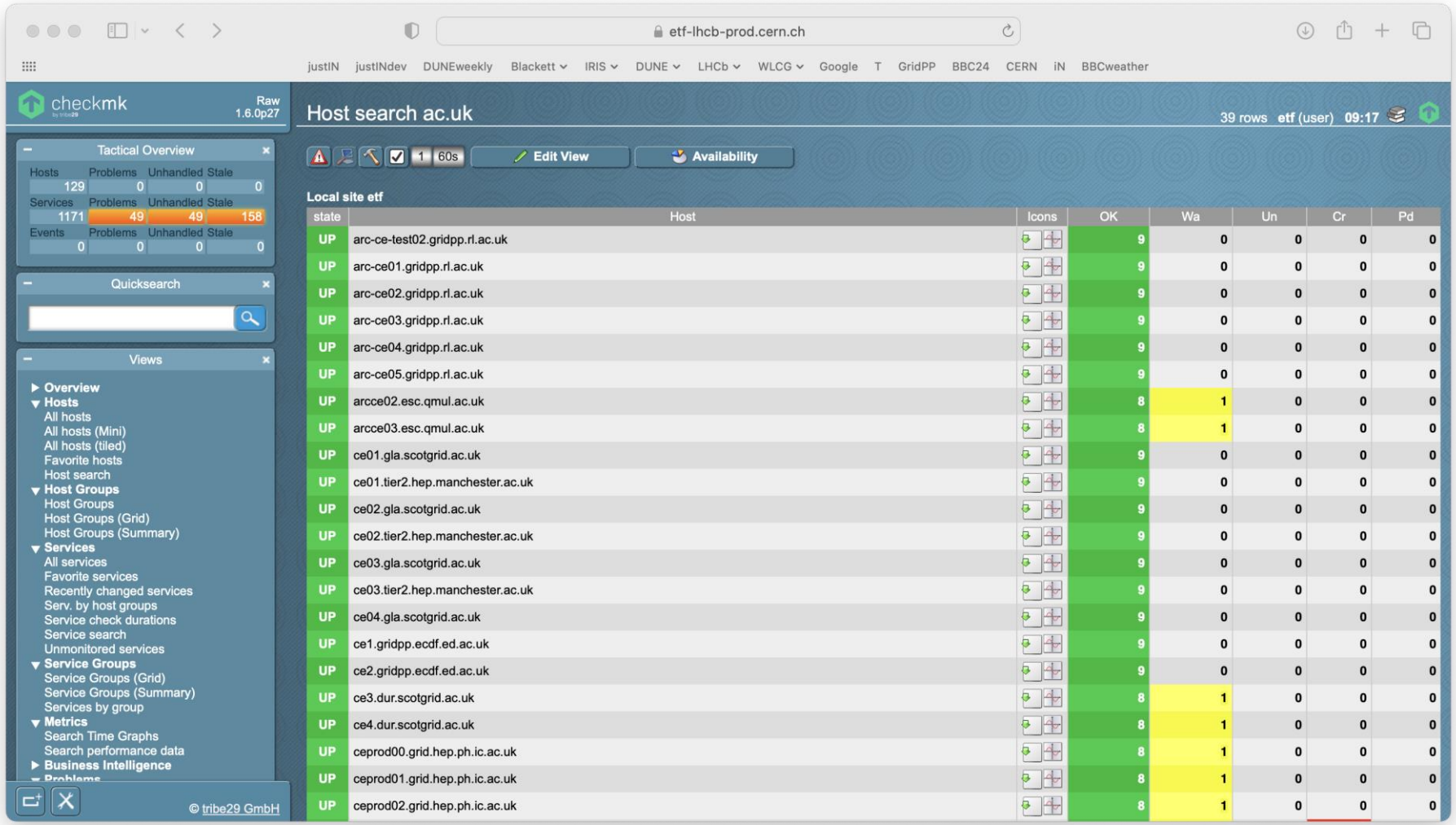


# Ticket summary

| Site | Tickets opened since Oct 2022 | Open tickets |
|---|---|---|
| RAL-T1 | 14 | 4 |
| UKI-LT2-Brunel | 3 | 0 |
| UKI-LT2-QMUL | 4 | 0 |
| UKI-LT2-RHUL | 2 | 0 |
| UKI-NORTHGRID-LANCS-HEP | 3 | 0 |
| UKI-NORTHGRID-LIV-HEP | 2 | 1 |
| UKI-NORTHGRID-MAN-HEP | 2 | 0 |
| UKI-NORTHGRID-SHEF-HEP | 2 | 0 |
| UKI-SCOTGRID-DURHAM | 1 | 0 |
| UKI-SCOTGRID-ECDF | 3 | 0 |
| UKI-SOUTHGRID-BRIS-HEP | 3 | 0 |
| UKI-SOUTHGRID-RALPP | 1 | 0 |
| All | 40 | 5 |

# LHCb ETF tests .ac.uk

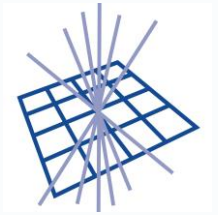


# LHCb ETF tests status

- Tests of functionality of services at sites
  - In most cases by submitting jobs to CE
  - Test CE API, certificates, cvmfs, memory on worker node …
- Tests are simple shell scripts
  - Easy for admins and other GridPP staff to look at
  - Outputs of tests are visible to site admins
- Manchester took on ETF maintenance again, at start of March
- Imminent release of updated suite of tests
  - Including new storage tests
  - Need special handling for subset of WebDAV RAL Echo has
  - Especially important for GridPP/LHCb since UK has so many of the LHCb Tier2s with storage

# LHCb ETF tests developments

- Plan is to use ETF tests for advance testing of new DIRAC requirements / API support
  - ETF tests are trivial shell scripts
    - easy to change
  - DIRAC is a large monolithic system which LHCb relies on for production and user work
    - harder to change and riskier to deploy updates
- Site admins can see ETF test outputs and the test source code
  - Provides a concrete target in advance of deployment in DIRAC
- Key changes
  - Storage technology changes - eg move away from DPM
  - Token support for storages - one goal of DC24 in early 2024



# Summary

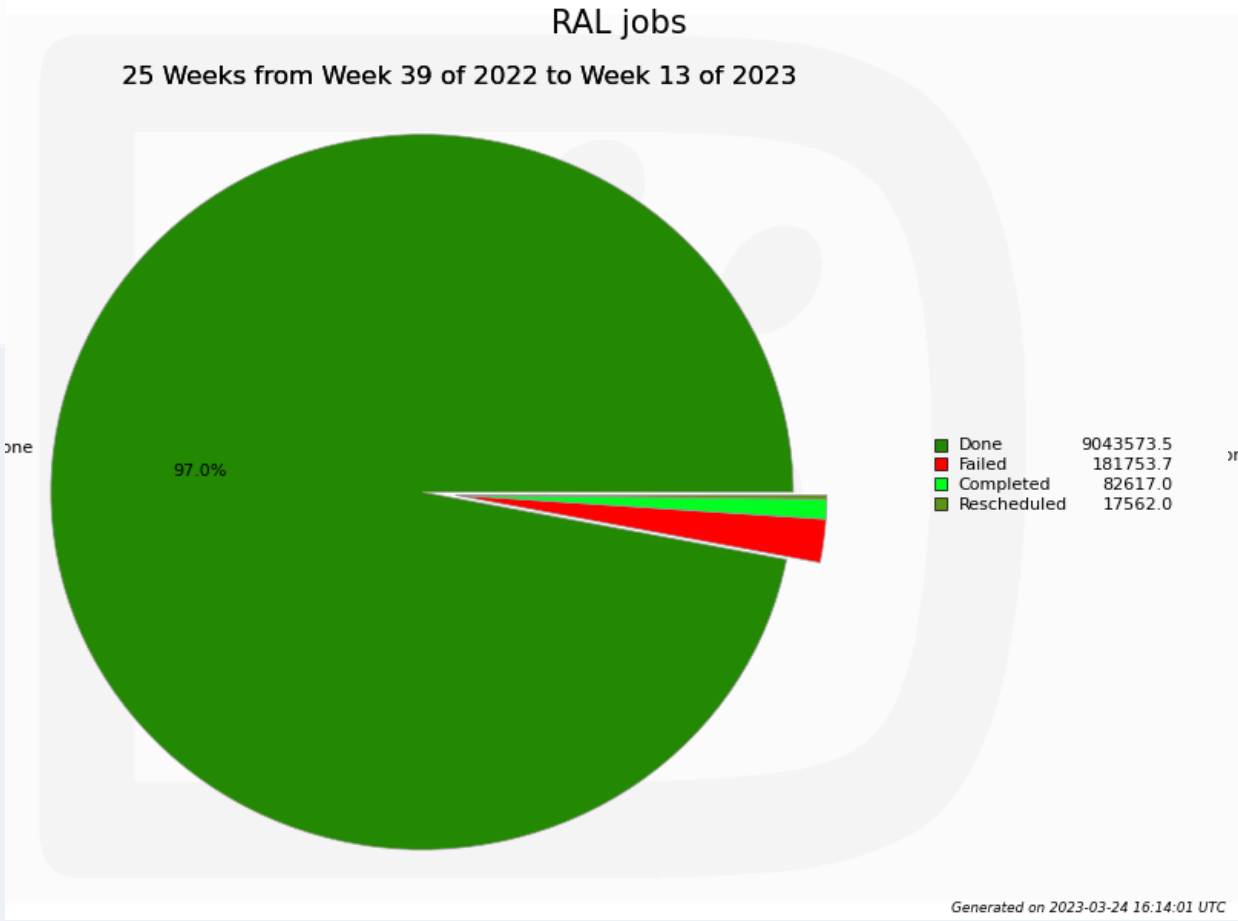
- UK sites provided a lot of resources to LHCb during the last 6 months.
- In general operations were smooth.
- There are some long-lasting issues, their solutions are progressing.
  - xrootd-related issues, require xrootd development

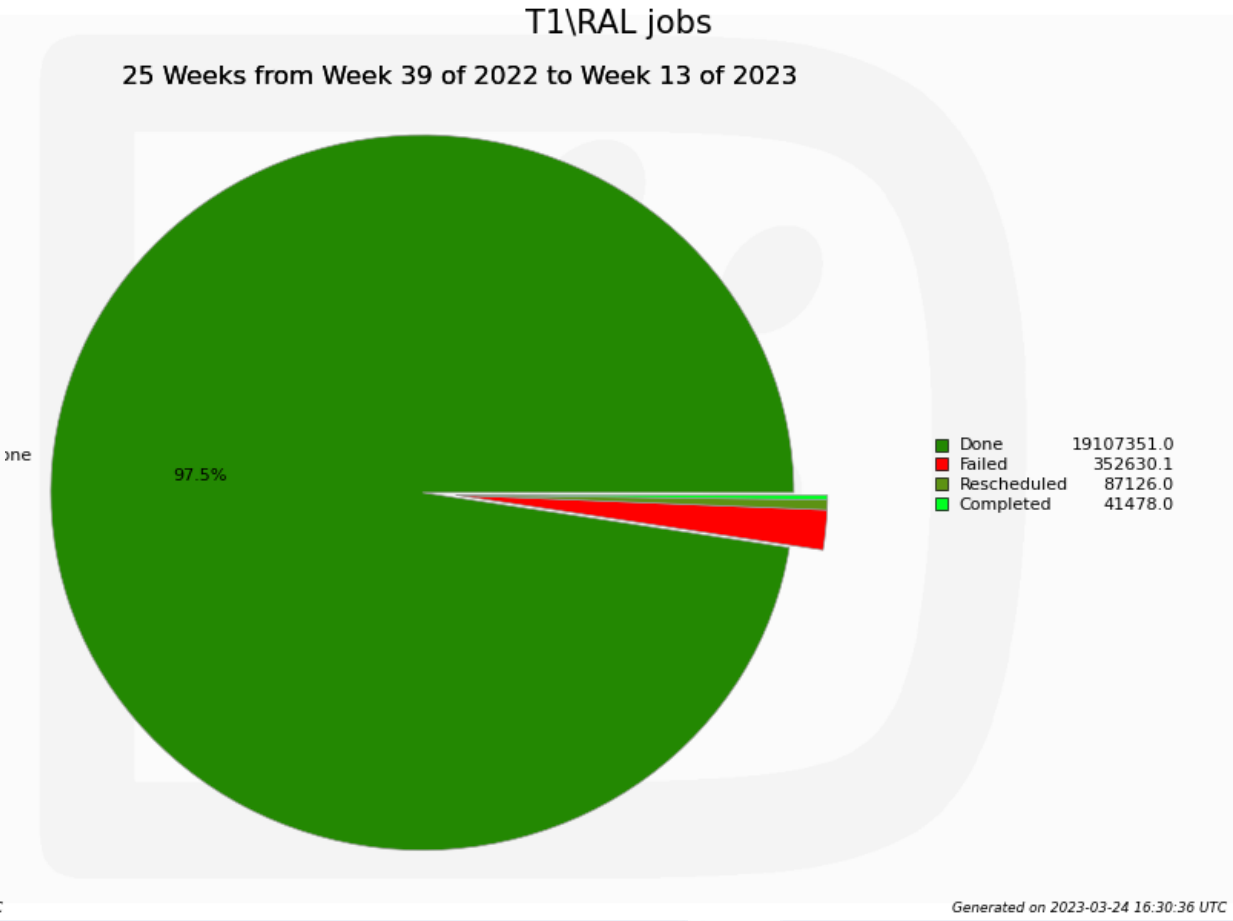
# Backup: Stub files

- Sometimes a failed upload to ECHO results in a corrupted file: some garbage data (i.e. some ceph objects) are left on the storage.
- This prevents further uploads of the file.
- To solve the problem these garbage objects have to be removed manually using low level rados operations (only accessible by admins).
- Previously such files were identified via GGUS tickets from the VO.
- Hopefully, this will change soon -- tools that will allow one to detect such files on the server side (i.e. perform internal consistency checks) are being developed (almost finished).

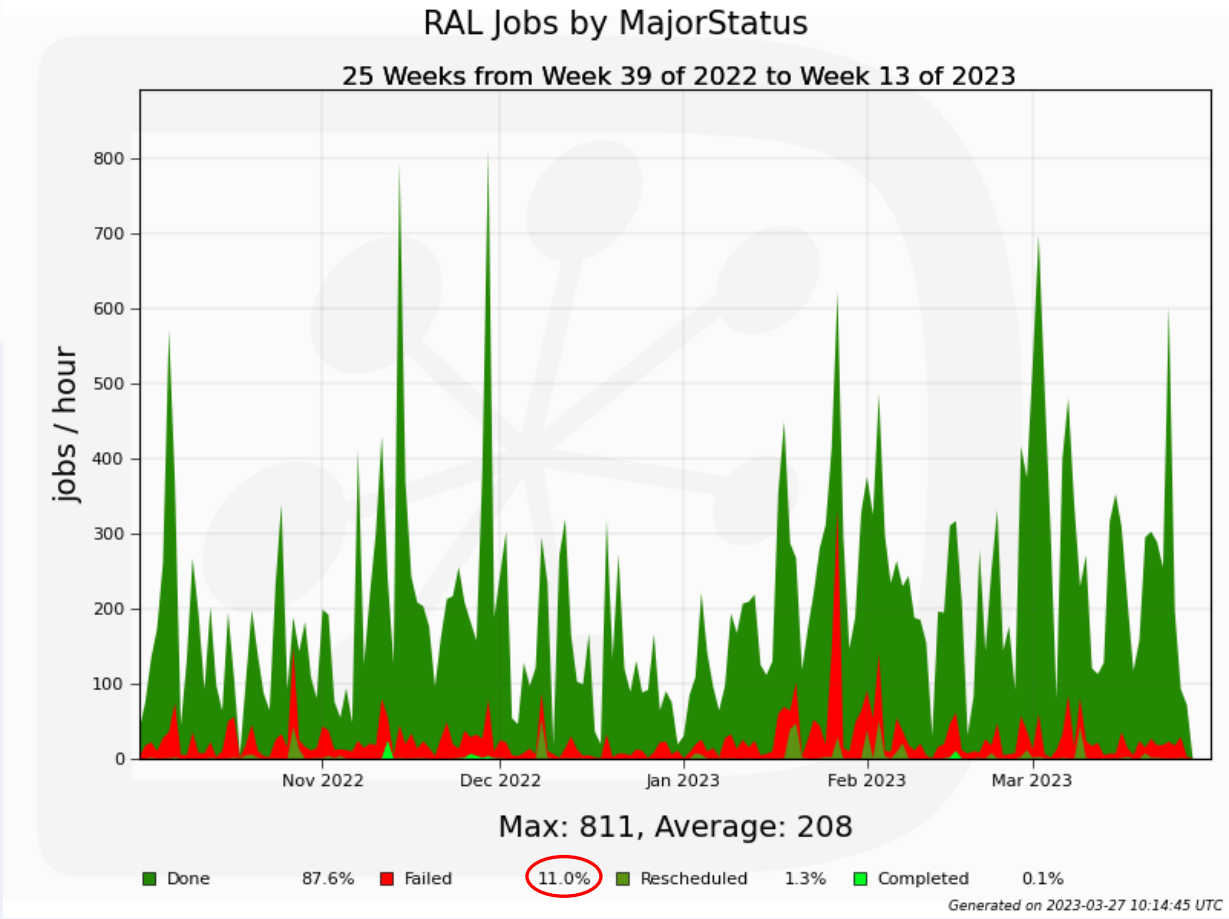
# Backup: RAL T1 jobs success rate

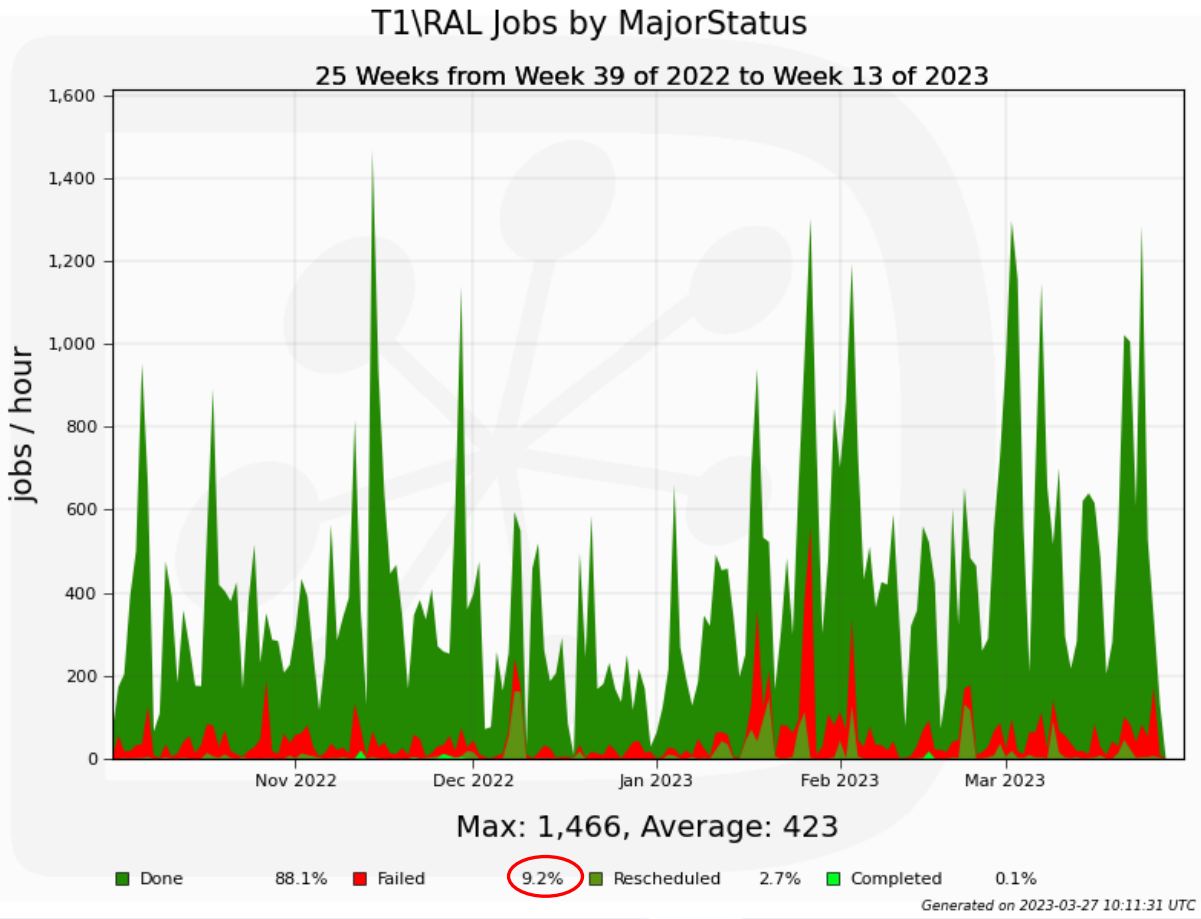
What about the success rate of RAL's jobs? Very high for MC jobs and acceptable for user jobs.



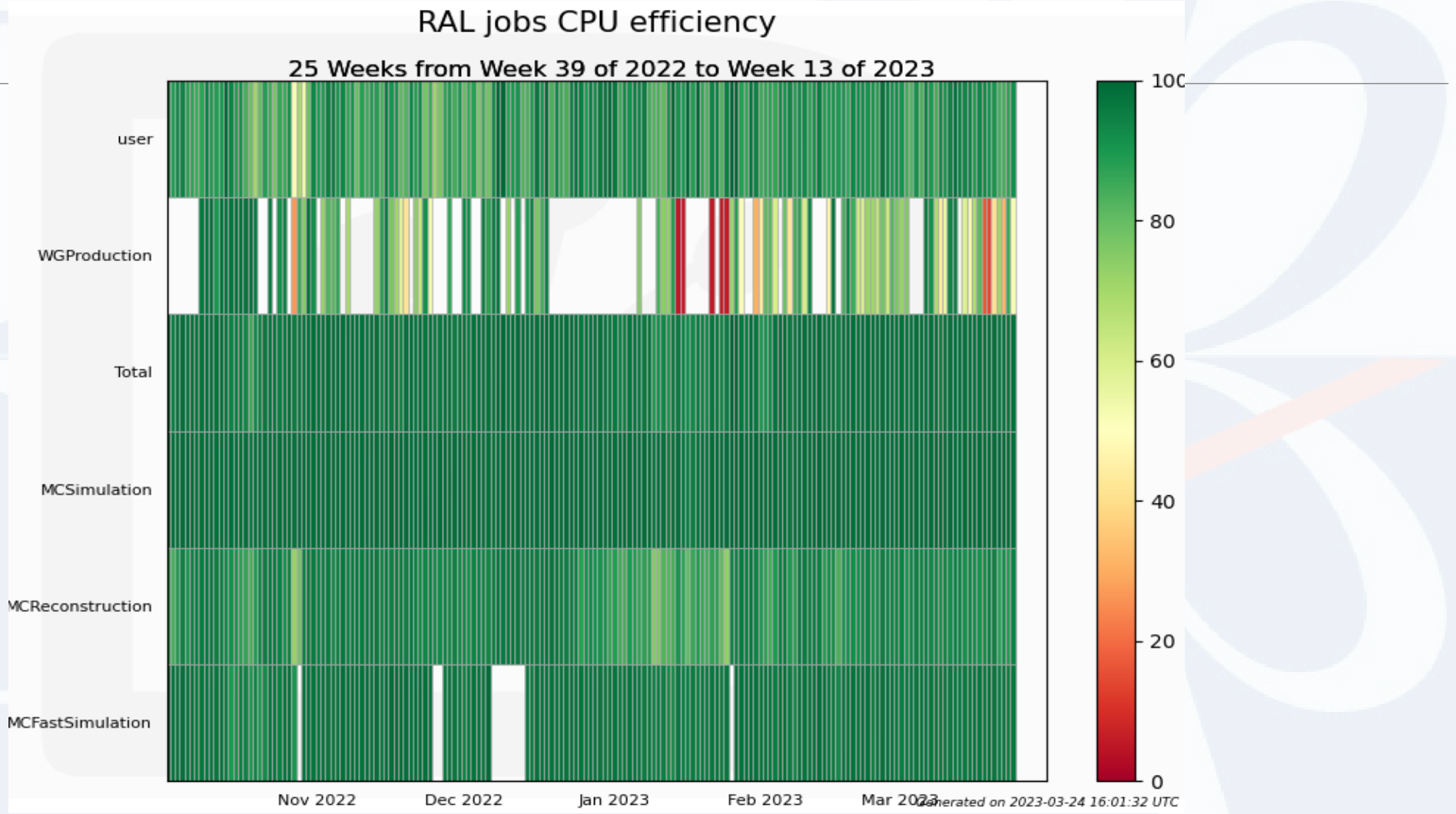
# Backup: RAL T1 user jobs success rate



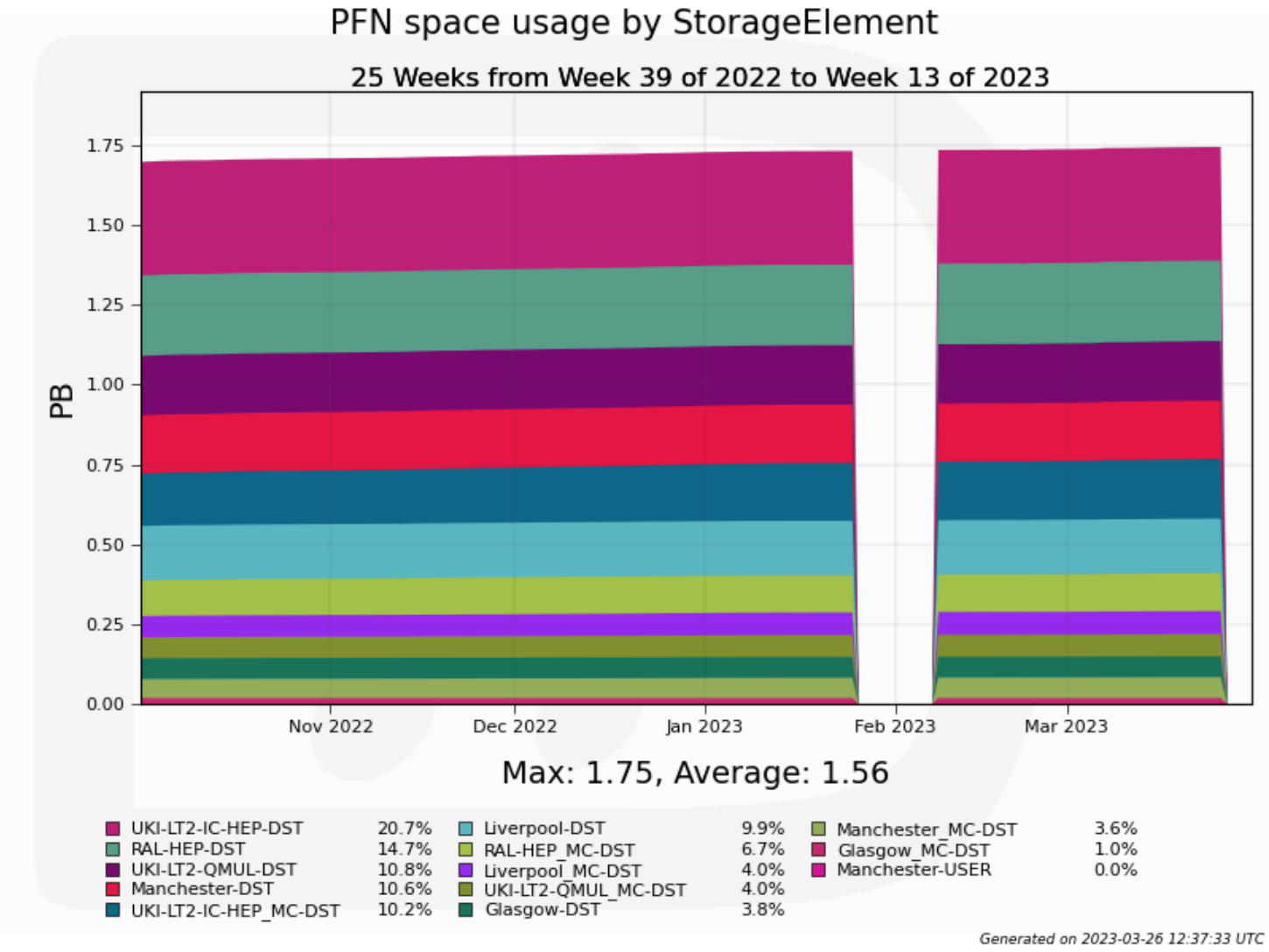
# Backup: RAL T1 Job efficiency

Efficiency of MC jobs is good, user and WG Production jobs have lower efficiency (not surprising).



# Tier-2 statistics

Limited data operations during the last 6 months



# Backup: xcache-gateway error

Presence of xcache v5.3.3 causes additional errors sometimes. This is due to inconsistency of read vector length between xcache and the gateway. Furthermore, xcache obscures error message and reports it as "File name too long"