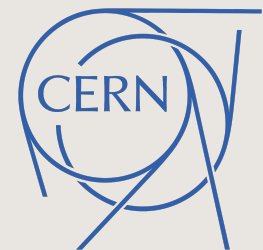


CMS community talk



FTS Workshop 9th Sep 2024
Panos Paparrigopoulos (CERN), [Katy Ellis \(RAL\)](#)



Science and
Technology
Facilities Council ¹



CMS Data Management Personnel

- A small team in operations
 - Including one “operator” in European timezone (CERN) and one in Americas timezone (FNAL)
 - Many are funded on short-term contracts 2-3 years
- A small team in development (Rucio)
 - Many are part-time, short-length contributors e.g. PhD students

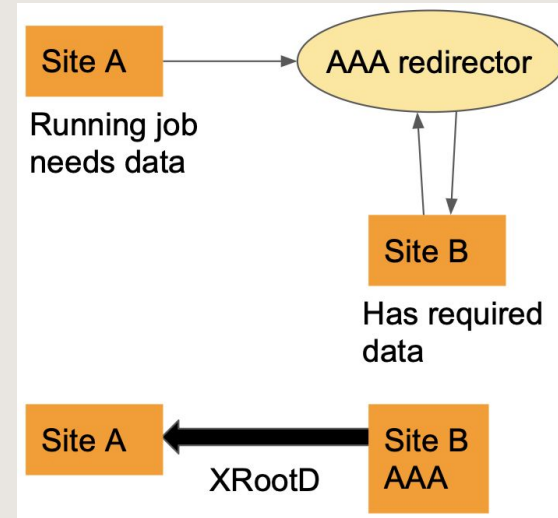
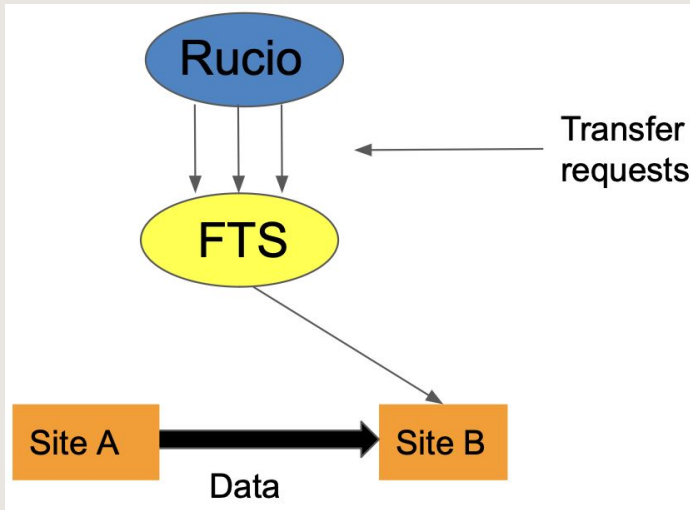


CMS Data Management

- CMS keeps ~250PB on disk and ~540PB on tape
 - Approx. increase of 50PB (disk) and 190PB (tape) since the last meeting!
 - Data-taking period 'Run 3'
 - Continuous MC (simulated data) production in parallel
- Data is TPC-copied by FTS (with or without Rucio) or streamed directly to jobs from local or remote storage systems using XRootD federation (AAA)

CMS Data Management

- CMS keeps ~250PB on disk and ~540PB on tape
 - Approx. increase of 50PB (disk) and 190PB (tape) since the last meeting!
 - Data-taking period 'Run 3'
 - Continuous MC (simulated data) production in parallel
- Data is TPC-copied by FTS (with or without Rucio) or streamed directly to jobs from local or remote storage systems using XRootD federation (AAA)



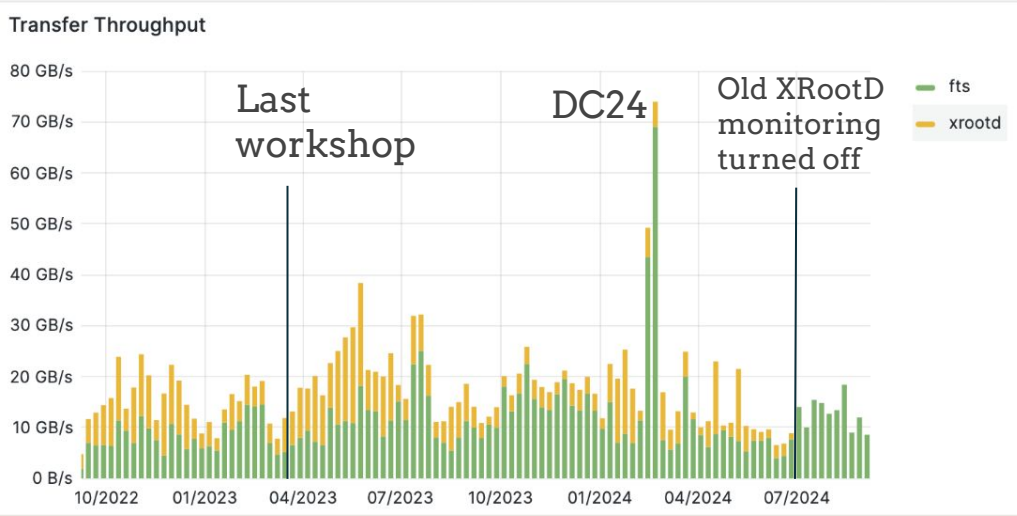


Notes on streaming

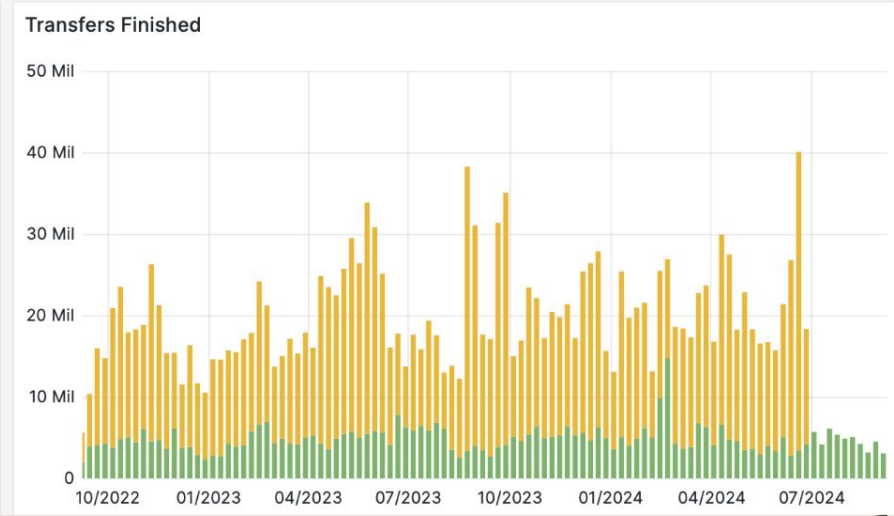
- Only parts of the data *required* by the job are transferred
 - Regardless of local or remote read
- When data is read remotely, CMS saves storage space at sites when the same files are read by jobs running at more than one site
- Particularly useful for reading 'premix' files (background MC events) which typically sit in huge datasets (500+TB) at CERN (Eurasia) and Fermilab (Americas)
- But is remote streaming as reliable as local streaming?

- Jobs read data using XRootD
 - Monitoring of these reads considered unreliable
 - New monitoring 'Shoveler' will take over
 - See my talk later this week!

CMS traffic in last 2 years



Volume transferred



Number of transfers



Data Challenge 2024

- Second of a series of four data movement challenges
 - Preparing for the significant increase in data volume (High-Luminosity LHC)
- In collaboration with all LHC experiments and others
- Target: 25% of estimated CMS usage in HL-LHC
- Targets were met, but only just!
- Helping to define future requirements

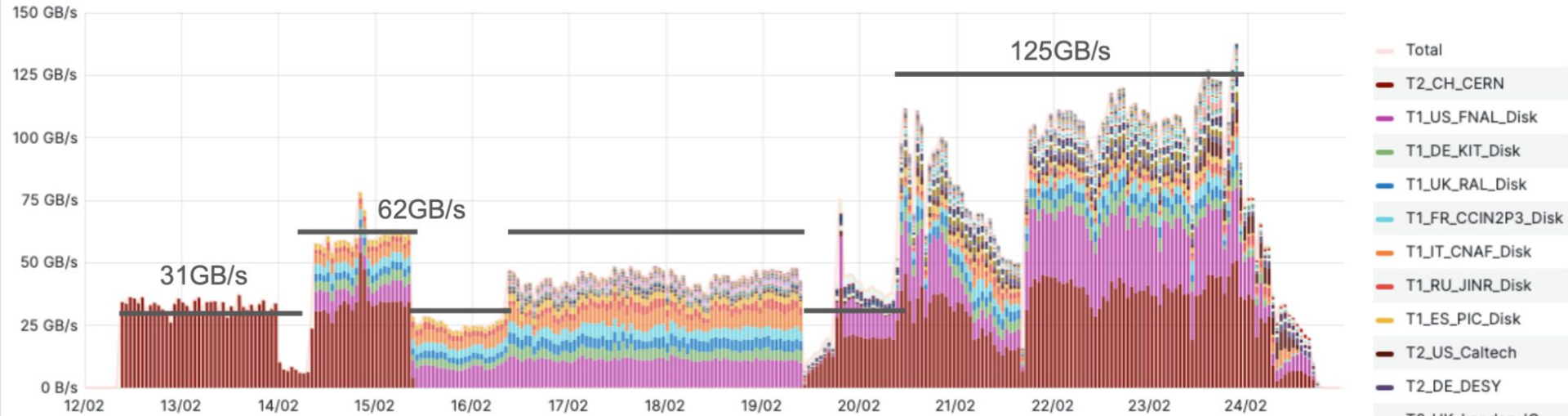
- The challenge pushed (and exceeded!) the current limits of FTS
 - In the case of CMS we were perhaps just ON the limit
 - CMS target rates were not as high as ATLAS

- CMS learnt a lot about how to configure FTS to increase throughput
- FTS was not quick to throttle for failing sites due to huge number of requests (Optimizer)
- A lot of progress with tokens, see next slide



Data Challenge 2024

Transfer Throughput



1 2 3 4 5 6 7 8 9 10 11 12





FTS support for tokens

- During DC24, CMS enabled tokens at around half of its sites, successfully authenticating with them for approximately 50% of the total throughput
 - This was made possible by the FTS developers, who deployed test implementations and applied several hotfixes during the challenge.
- After DC24, it was decided to roll back to certificates while FTS, Rucio, and other stakeholders determine how token-based transfers should be properly implemented.
- CMS will be eager to re-enable tokens once all relevant parties have agreed on a token authentication implementation and hopes this will come very soon.

➡ Rahul will present on this on Wednesday



Needed improvements - Optimizer

- "The optimizer" is the FTS component that manages the system's throughput to sites based on error rates.
- Recently, when CMS needed to push a large amount of data to tape sites, one of them couldn't handle the rate and went down.
 - We observed that the FTS optimizer took a very long time to throttle transfers to the site effectively.
 - In the meantime, the Rucio throttler was implemented to reduce rates if needed, but it would be ideal if FTS could make decisions faster and automatically adjust the rate within the configured limits.



Tape REST API

- Benefits?
- In production at 5/7 CMS tape sites
 - IN2P3, KIT, RAL, CNAF, JINR
- In test/commissioning at the remaining 2 sites
 - FNAL, PIC (using davs)



File exists error - Description

- Transfers to tape fail when a file already exists on disk buffer, at the destination.
- CMS Rucio is configured not to overwrite existing files on tape during retries.
- Failed transfers are retried repeatedly, as FTS cannot overwrite the existing file, leading to a cycle of failures
- Continuous errors escalate to support teams, requiring manual intervention to resolve, increasing the operational burden.
- With close collaboration between the CMS DM-ops and the FTS team a solution was proposed last year and was implemented in July 2024.



File exists error - Solution

- A dedicated flag was introduced in FTS to handle the "file exists" error by enabling conditional overwrite:
 - If the destination file is only on the disk buffer and not on tape, FTS will delete it and retry the transfer.
 - If the file already exists on tape, FTS will fail the transfer with a distinct error message.
- Rucio development was needed to include this new flag in its submission for transfers to tape-enabled endpoints.
- All the development has concluded this summer and the feature will be deployed soon in production***



File exists error - Solution

- A dedicated flag was introduced in FTS to handle the "file exists" error by enabling ***
 - The solution was deployed in August 2024 and enabled for CERN Tape.
 - After deployment, we realized that destination file report were missing for some transfers and Mihai deployed a prompt fix.
 - Currently, the feature is running as expected.
 - We noticed 2 new error categories after this improvement:
 - BAD_ADDRESS: Tape endpoint doesn't report the locality of the files, thus FTS doesn't know whether to overwrite or not, thus failing. Such files are observed to have 0 length.
 - Corrupt files on tape: As per our policy, we don't overwrite them. We don't know why this happens at the moment. Under investigation
- All the development has concluded this summer and the feature will be deployed soon in production***



Needed improvements - Configuration

- FTS can be configured to apply limits either to a storage service or to a link between two of them.
- These configurations are mainly left untouched; currently, only the CMS DM or the FTS team can change them, and they do so occasionally.
 - During DC24, it was realized that many of these configurations were misconfigured.
- Site admins are the ones who know the limits of their sites:
 - It would be beneficial if they could configure FTS for their sites.
 - Even though the current FTS doesn't allow this level of authorization granularity, a solution could be envisioned based on the existing CMS siteconf repositories in GitLab.
 - Site admins could edit FTS configurations in the same place where they configure their RSEs, and these configs could then be propagated to FTS through the API.

Error message improvements





- Since migrating to WebDAV, the error messages received via HTTP have become less helpful.
- It has become considerably harder to identify system problems when a variety of different issues can be obscured by a generic “403 - permission denied”
- We understand this might not be an FTS issue per se, but we believe this is the right forum to raise such concerns, as these issues, rather than simplifying, increase the manual operational effort required for debugging.





From last workshop (Mar 2023)

Future improvements

- Long gap between FTS complete and Rucio OK
 - There has been a fix made for this - CMS Rucio should pick it up in next upgrade 
- Using different Rucio 'activities' for tape recall priority 
 - Could all FTS transfers be better prioritised?
- Better grouping of file transfers 
 - Although things to try on the CMS-Rucio side
- Better handling of 'destination file exists' 
- **Request:**
 - CMS would like to understand better large (>20GB) file transfer failures
 - Could 100GB file transfers be feasible by Run-4?
 - Would it be possible for FTS to resume a failed transfer rather than start from scratch?





Long-term

FTS webpage improvements

Source	Destination	V0	Submitted	Active	Staging	S.Active	Archivin	Finished	Failed	Cancel	Rate (last 1h)	Thr.
+ davs://webdav.ifca.es	davs://eoscms.cern.ch	cms	4	-	-	-	-	80	-	-	100.00 %	0.08 MiB/s

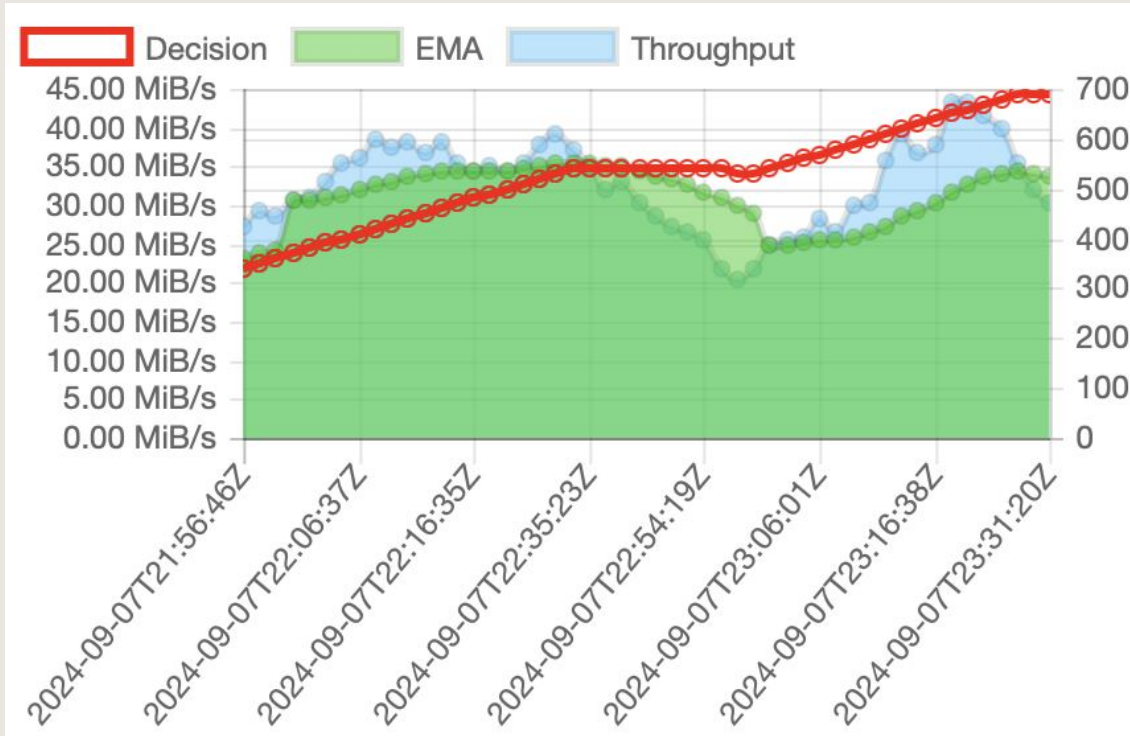
Link configuration

Parameters per link. If only source or only destination is specified, it applies to any transfer from/to that storage.

	Symbolic name	Source	Destination	Streams	Min Actives	Max Actives	Optimizer Mode	TCP buffer size	Disable delegation
 	*	*	*	0	10	1000	3	0	No
 	*-davs://cmsdc	*	davs://cmsdca	0	200	600	3	0	No



FTS webpage improvements



https://fts3-cms.cern.ch:8449/fts3/ftsmon/#/optimizer/detailed?source=davs:%2F%2Feoscms.cern.ch&destination=davs:%2F%2Fcmsdata.phys.cmu.edu&time_window=24



Summary

- CMS are grateful for the continued collaboration with the FTS team
- FTS is easily able to cope with current demands from CMS
 - FTS production usage not expected to rise until Run 4
 - In most recent data challenge, FTS allowed CMS to hit target rates
 - The next data challenge will be the hardest yet!

