FTS
File Transfer Service

FTS DC24 Retrospective

WLCG/DOMA DC24 Retrospective

Presenter: Steven Murray

Authors: Joao Pedro Lopes, Shubhangi Misra, Steven Murray, Mihai Patrascoiu and Luca Mascetti

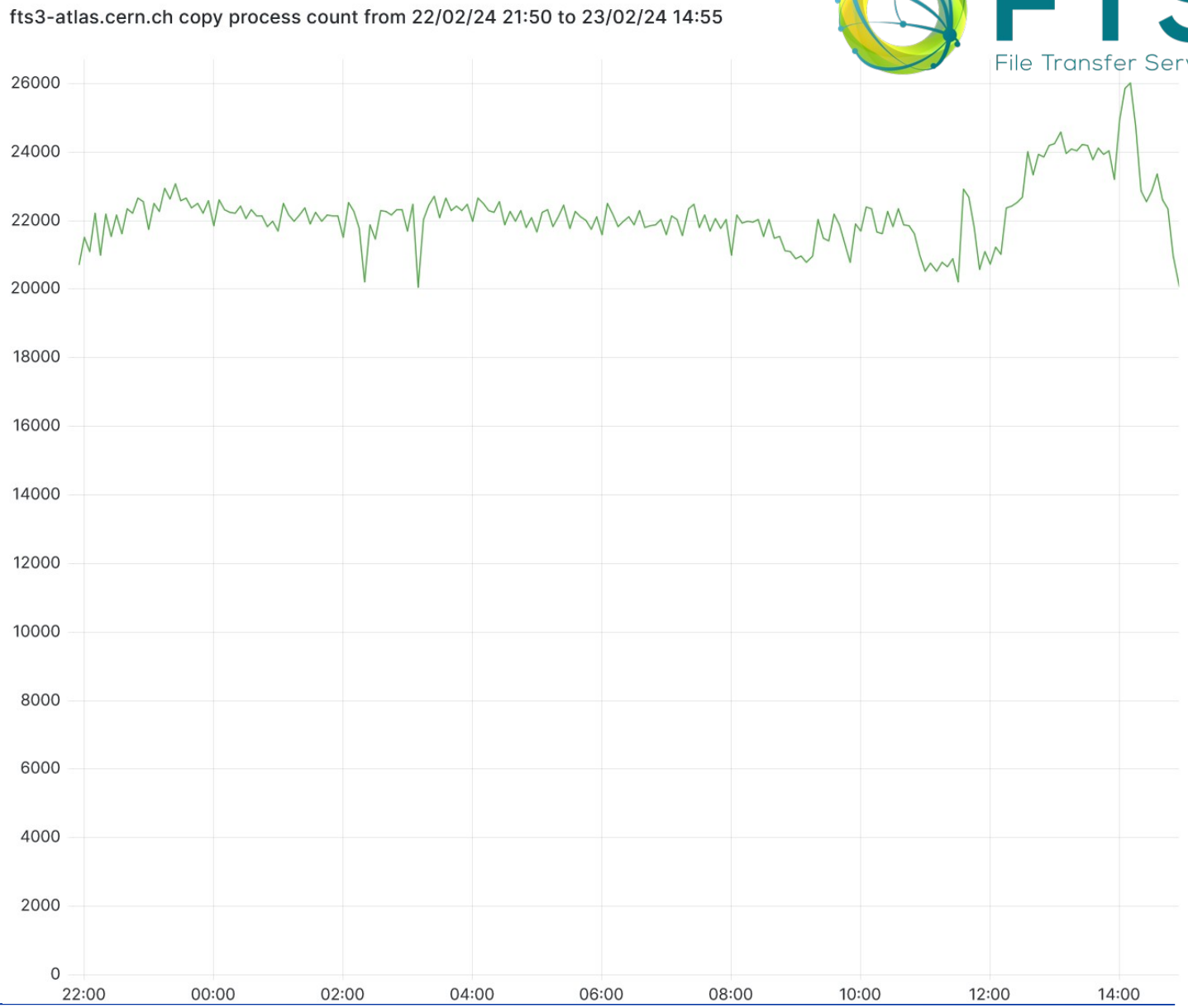
Wednesday 6th March 2024

General overview and impressions

- The data challenge was a success for FTS and its support of tokens
  - A little too much fire fighting behind the scenes
  - Defragmentation of the `fts3-atlas.cern.ch` DB was not completed
- FTS ran at double its normal “concurrent” transfer rate
  - A new FTS record
- The data challenge highlighted misconceptions about how to use FTS which ultimately resulted in not reaching the target data throughput of DC24 for 48 hours – yes this is positive!

Successes 1 of 4

- FTS went above and beyond its usual 10K concurrent transfers per instance
- **`fts3-atlas.cern.ch`** sustained over 20K transfers for 17 hours
- Many thanks to the database-on-demand team for quickly increasing the DB RAM of **`fts3-atlas.cern.ch`** from 80GB to 120GB

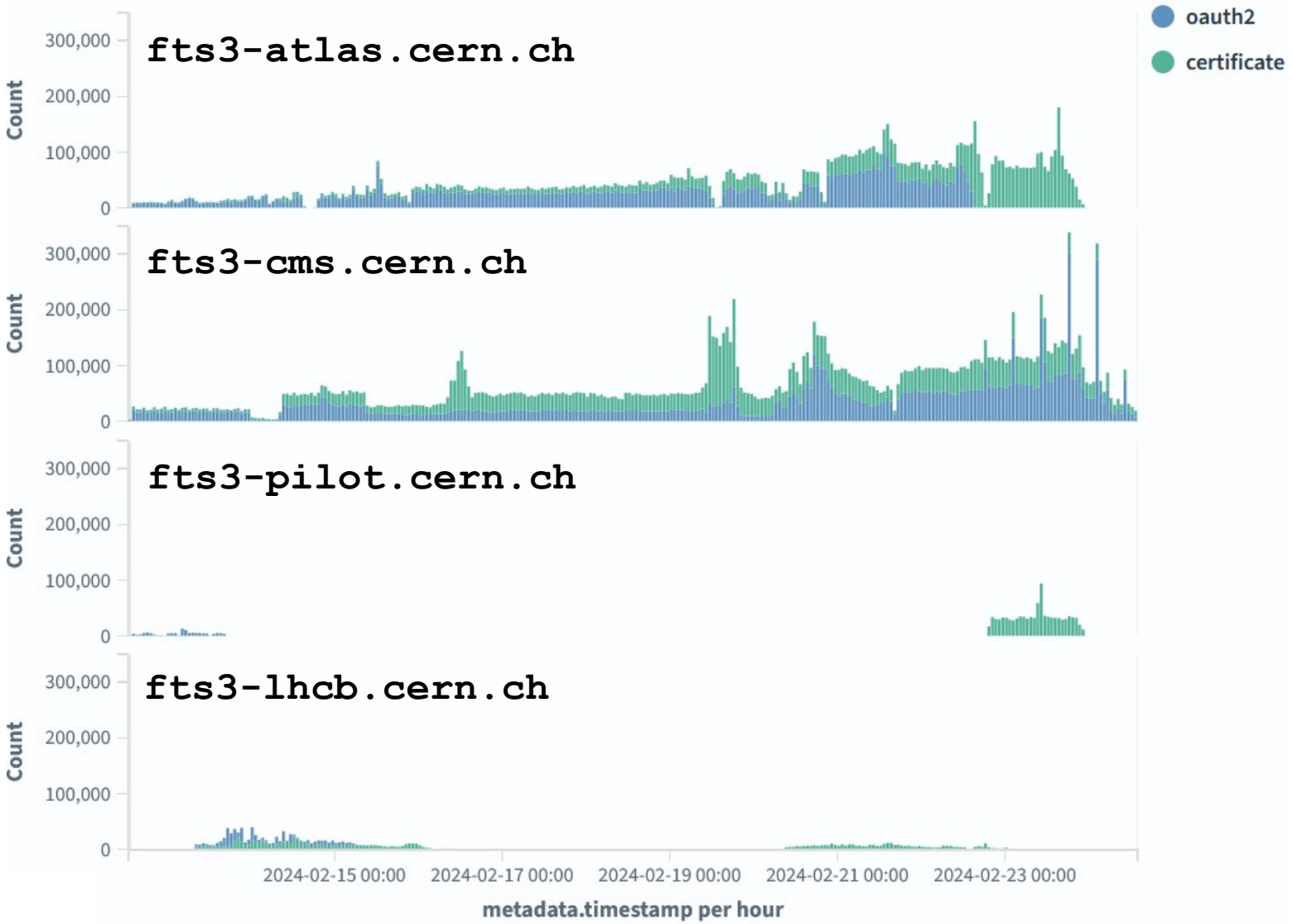


Successes 2 of 4

fts3-cms.cern.ch copy process count

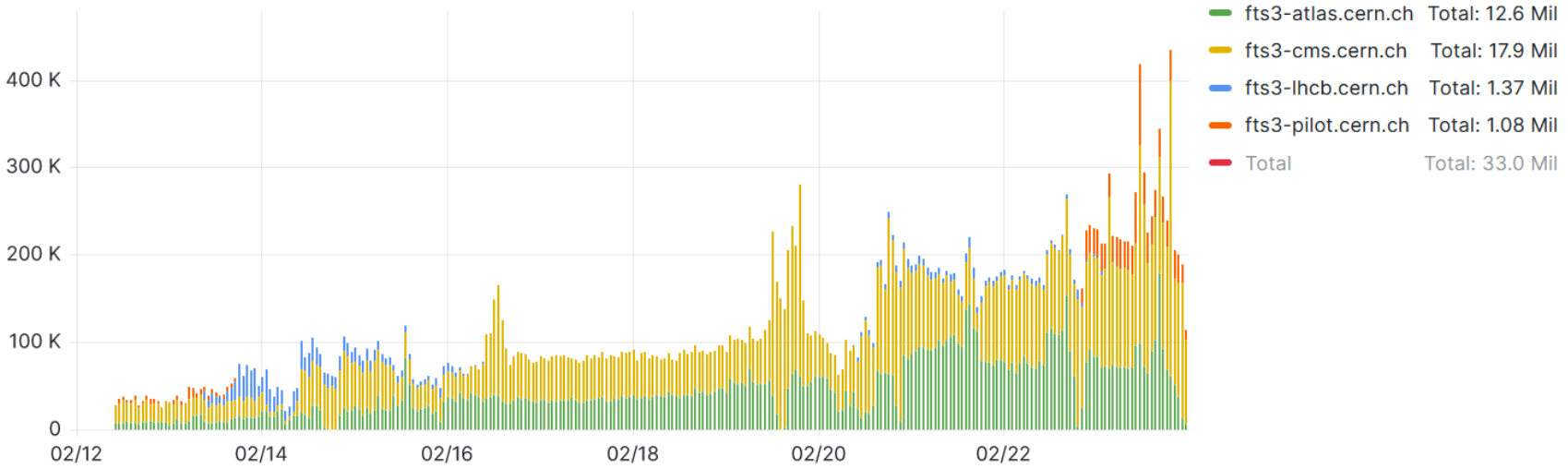
fts3-lhcb.cern.ch copy process count

Successes 3 of 4

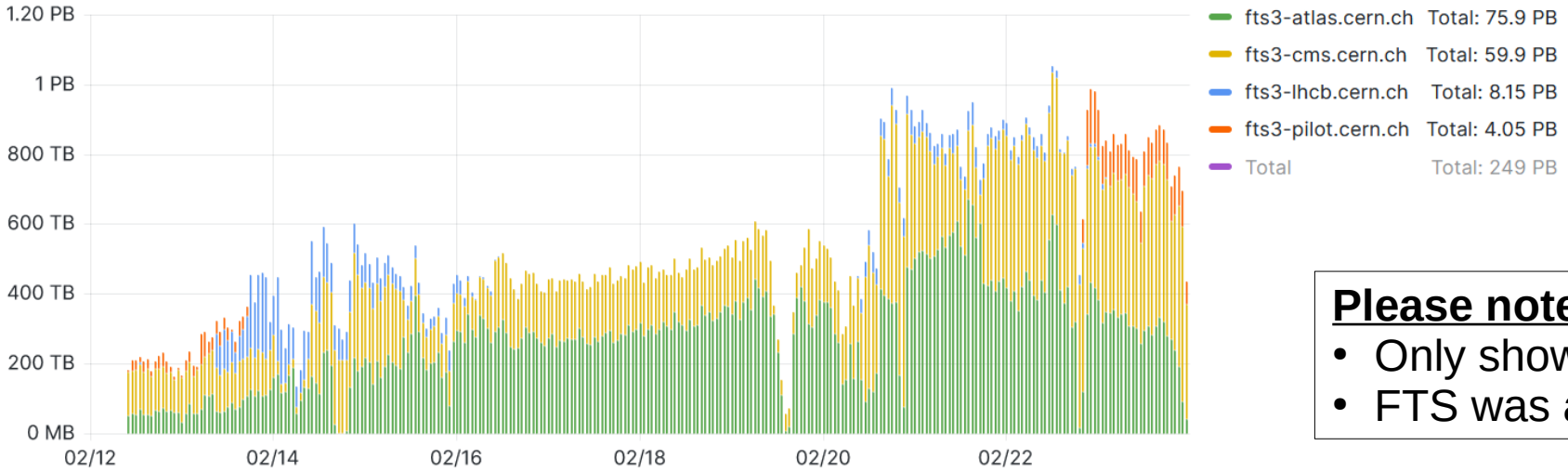


Successes 4 of 4

DC24 file transfers per FTS instance per hour



DC24 data volume transfered per hour



Please note

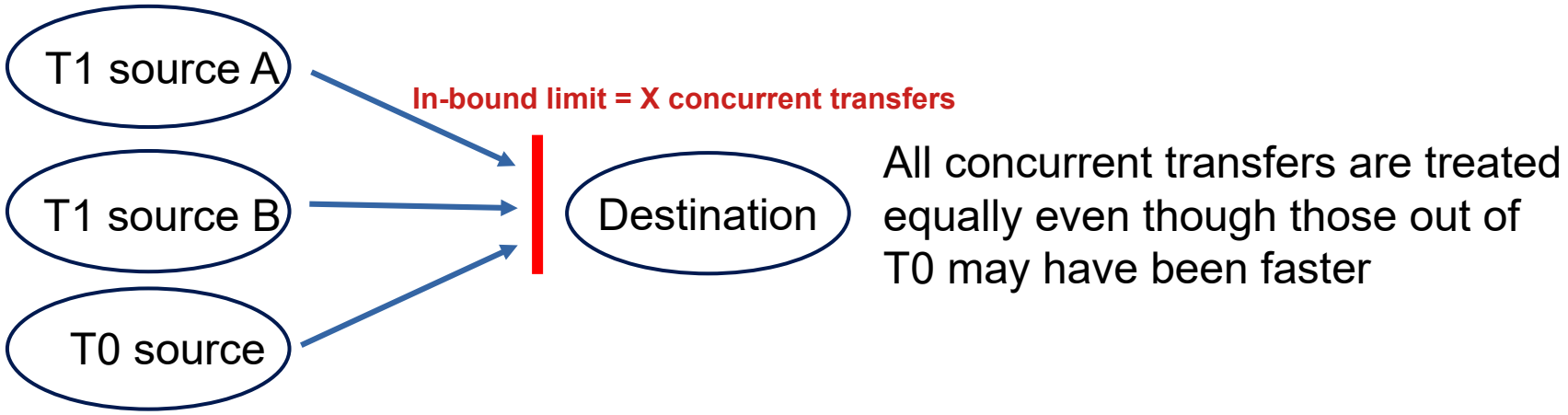
- Only showing the “Data Challenge” activity
- FTS was also running production transfers

Pain

- Incorrectly used tokens are **NOT** secure:
  - Tokens were and **WILL** be leaked (not by FTS)
  - FTS filter added just before DC24
- Too much time spent “discovering” tokens, e.g. no agreed FTS configuration within IAM
  - Single-use refresh-tokens were discovered on the fly - thankfully fixed by an IAM configuration change
  - 10 hour tokens were refreshed into 1 hour tokens - thankfully fixed via an IAM configuration change
    - Is it correct for this to be a fixed-configuration rather than token-driven (same-in same-out)?
- FTS had to deal with “hard” token tests on the fly:
  - We replaced token refreshing cron-jobs with daemons to prevent overlapping jobs when IAM was slow
  - We separated “heavy” house keeping tasks for tokens from their refresh logic to reduce DB load
- FTS did not know its limits:
  - DC24 helped understand them but FTS has no concept of back pressure
  - Massively slow optimizer runs – 3 hours!
  - FTS team had to migrate the `fts3-pilot.cern.ch` database from a 20GB of RAM database to a 120GB one

More pain - which should be a gain

- The main reason for not being able to sustain the DC24 target for 48 hours was…
  - FTS manages concurrent data transfers per link and **NOT** throughput
  - FTS treats all links with the same activity with equal priority
- FTS saturated all of its configured destination endpoints
- FTS CANNOT reach maximum throughput for the following configuration:



Future work and investigations

- FTS will continue to carry out token tests at the request of experiments
- Short-term:
  - FTS will continue to work with “relaxed” but “risky” modify-tokens
  - FTS will decouple the parallelism of the token refresh protocol from the DB
  - FTS will add a back pressure mechanism – RUCIO kindly offered to switch on their FTS back pressure
- Long-term:
  - FTS would welcome one modify-token per file transfer
    - Reduces the blast radius of leaked tokens
    - Avoids complicated protocols to hand out modify-tokens sparingly
    - Avoids future complications for tape transfers and their associated clean up logic
  - Improve performance of the optimiser
  - Allow the optimiser to be switched off
  - FTS will provide a better way to show the saturation of destination storage-endpoints
- Very long-term:
  - Tape – disk must be finished first
  - New FTS scheduler – priorities between links
- First FTS release with token support will be in Spring

Food for thought

- We need a single “token” responsible for both development and deployment
- Can single-shot refresh-tokens be banned from the WLCG token lifecycle?
- Can dynamic IAM-client registration be banned to reduce the attack surface?
- Should FTS automatically refresh access-tokens?
  - Why can’t fresh tokens be pushed into FTS like X509 proxy certificates are today?
- Can we agree on how to put the VO in tokens?
  - FTS had to be modified to map tokens to VOs
  - VO values must be the same for tokens and certificates
- We learnt from ATLAS that not all tokens are equal – what optimisations can be made?
  - Read and create tokens can have wide scopes and long durations
  - Modify tokens should have narrow scopes and preferably short durations
- We learnt from CMS that they use the same file paths on all storage endpoints:
  - Can we all stop using the `https://wlcg.cern.ch/jwt/v1/any` wildcard for audiences?
  - Tokens must contain storage endpoint names
- Can IAM have a “reset button” or “DB purge script” to forget “one token per file” tests?
- Can all storages ensure they have integrated themselves with the `dteam` token provider?