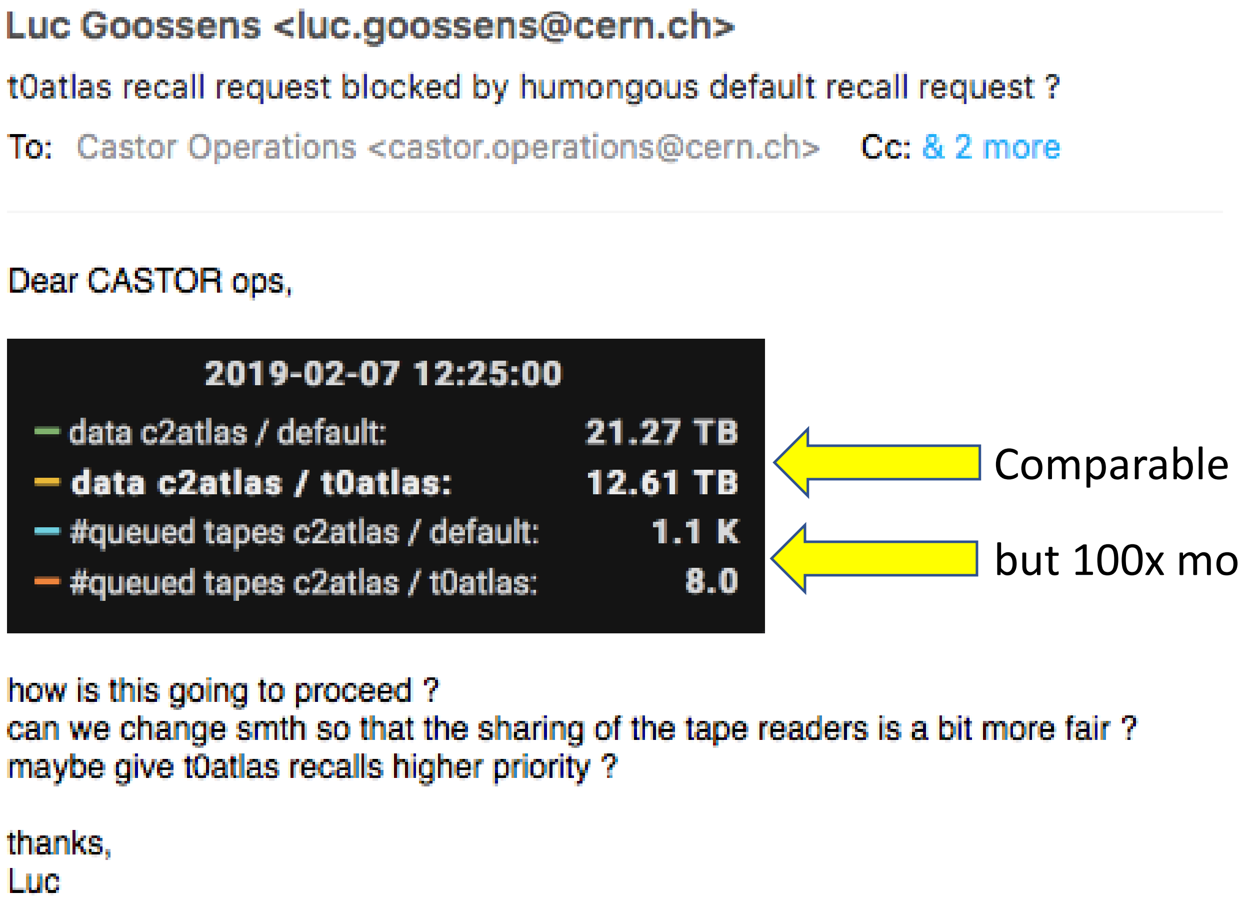
# ATLAS and CERN Tape

## Input from IT/ST

# Overview

- ATLAS 2018 recall analysis
- Discussion – fallout from last week
  1. More efficient / fairer queueing
  2. Recall priorities / policies
  3. Storage class selection
- CTA deployment plans for ATLAS

# ATLAS recall analysis at CERN, 2018



**Luc Goossens <luc.goossens@cern.ch>**

t0atlas recall request blocked by humongous default recall request ?

To: Castor Operations <castor.operations@cern.ch> Cc: & 2 more

Dear CASTOR ops,

**2019-02-07 12:25:00**

| | |
|---|---|
| 🟩 data c2atlas / default: | **21.27 TB** |
| 🟨 **data c2atlas / t0atlas:** | **12.61 TB** |
| 🟦 #queued tapes c2atlas / default: | **1.1 K** |
| 🟧 #queued tapes c2atlas / t0atlas: | **8.0** |

Comparable volumes…

but 100x more mounts on “grid”/default ☹

how is this going to proceed ?
can we change smth so that the sharing of the tape readers is a bit more fair ?
maybe give t0atlas recalls higher priority ?

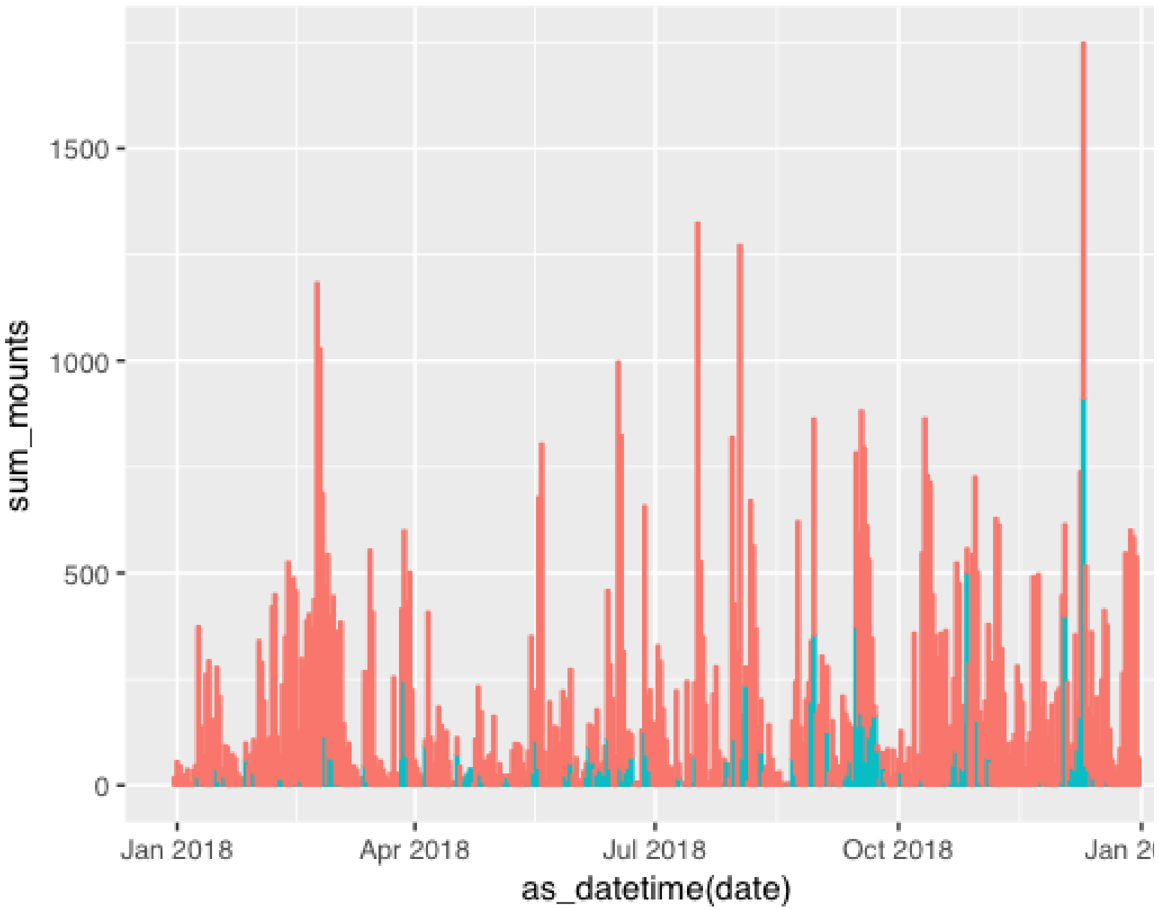
thanks,
Luc

# ATLAS recall analysis at CERN, 2018

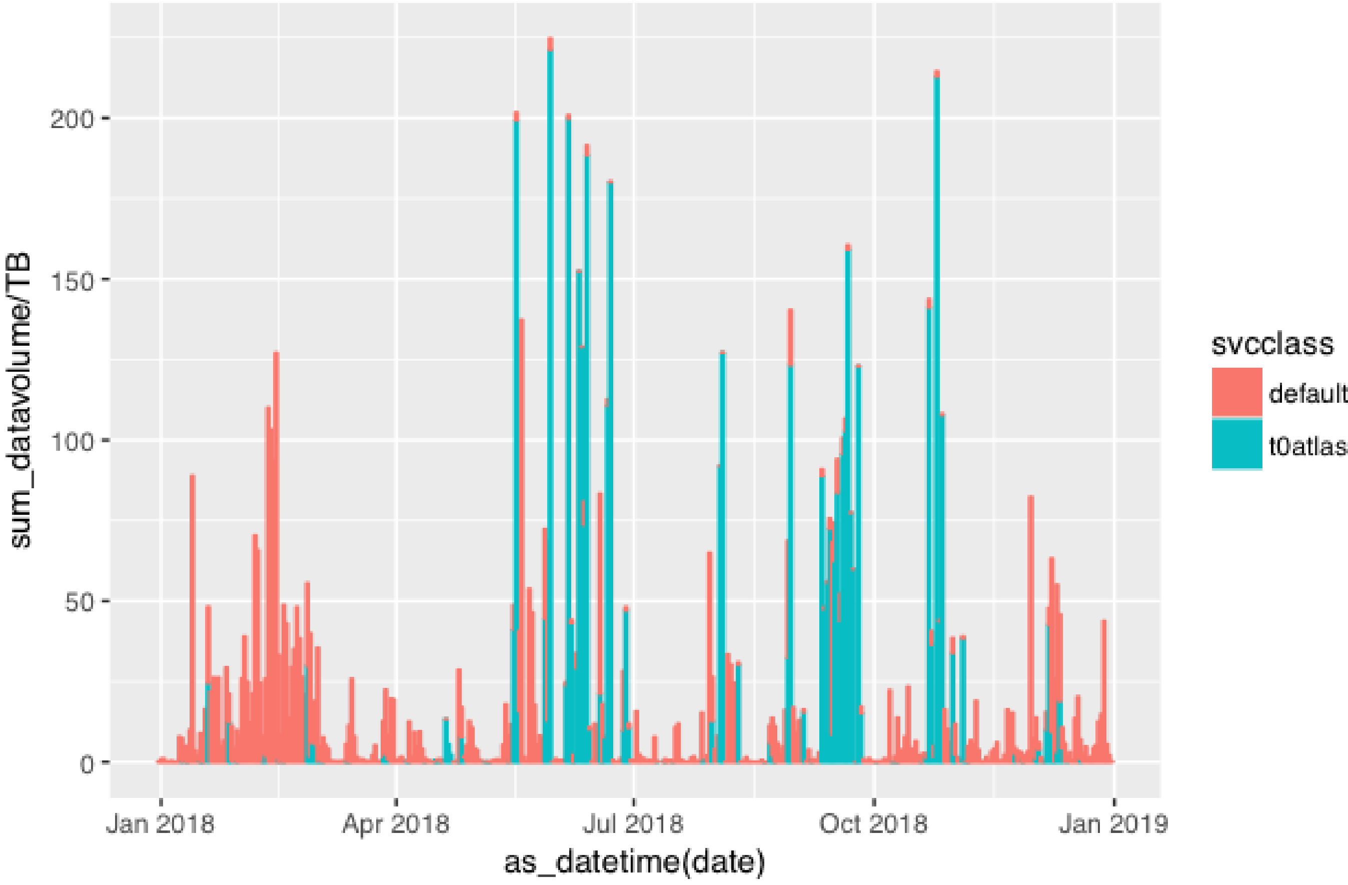
- ~4.2M file retrieves (7.9PB, 81K mounts) across two service classes (“default”, “t0atlas”)
- ATLAS recall efficiency greatly differs between default and t0atlas
  - Comparable data volume and files recalled (~55% t0atlas, 45% default)
  - “default” access required ~10x more library access via read mounts (7.7K vs 73K mounts)
  - Average per-mount volume and files per mount are an order of magnitude higher on t0atlas
  - Contiguous file reads: 30% on t0atlas, 18% for default
    - faster service speeds on t0atlas
  - “default” handles recall requests for files that have a significant dispersion in terms of creation date
    - service class handling very heterogeneous requests that are not organised by dataset
  - t0atlas tape retrieval is active on less days (1/3rd year) while default is continuously active

# Daily mounts and volume read

daily mounts, stacked

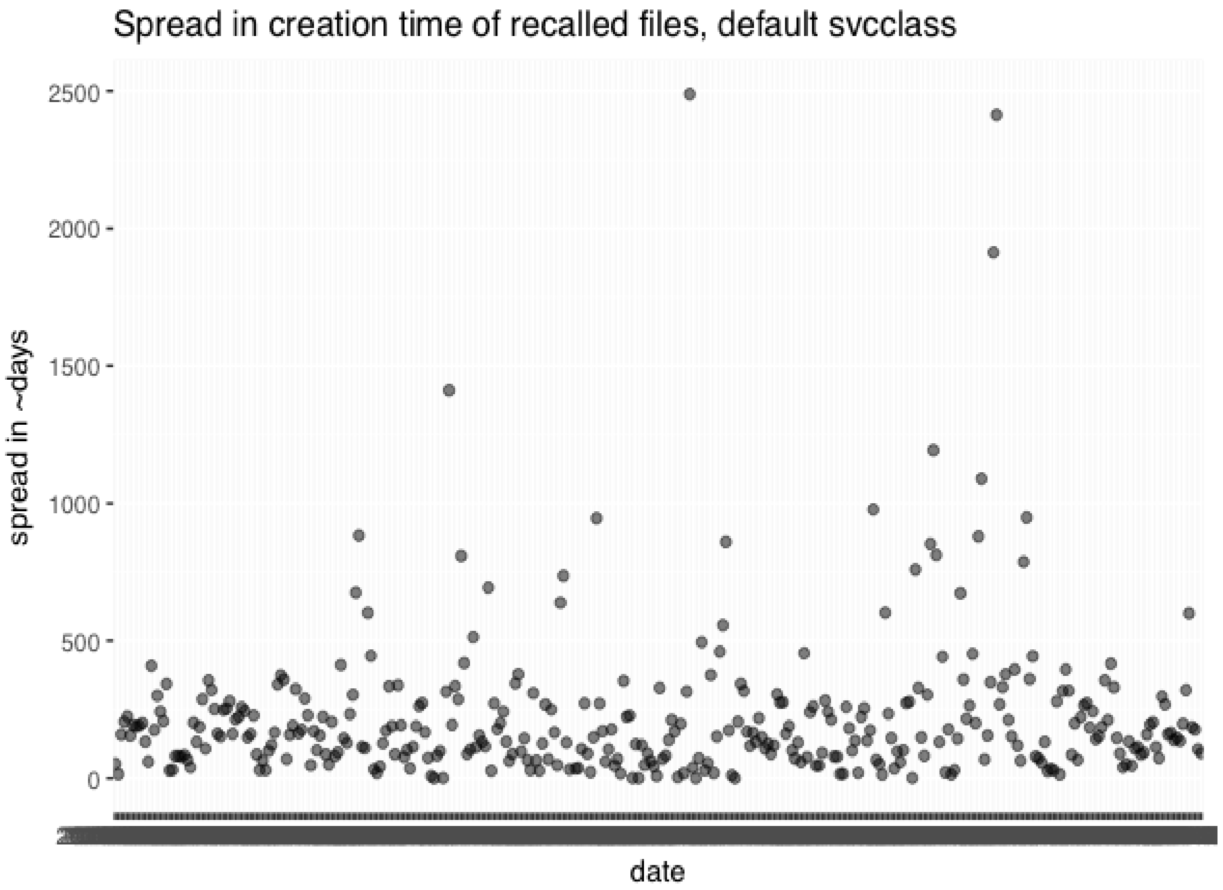


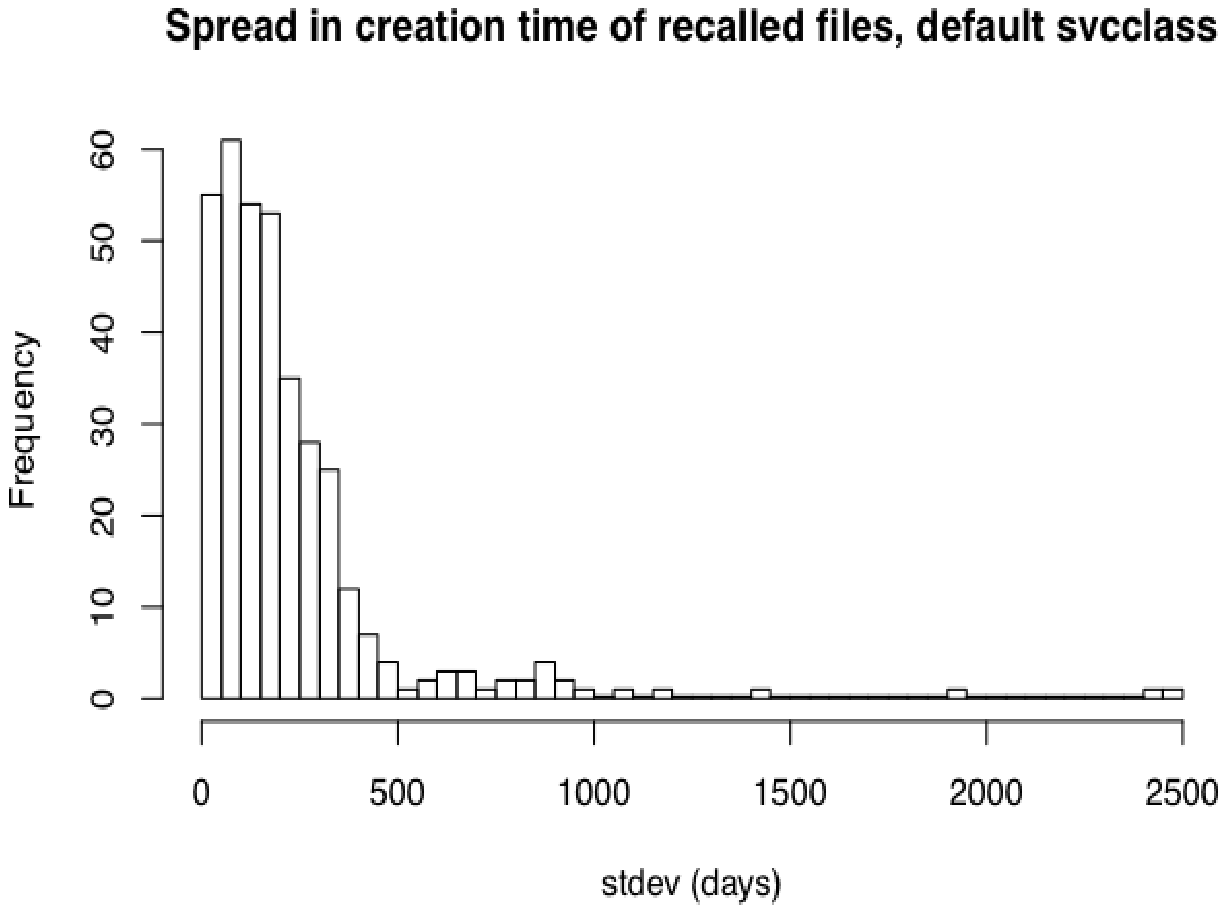
daily volume read, in TB, stacked



# Daily creation time dispersion of recalled files (stdev, mean): default

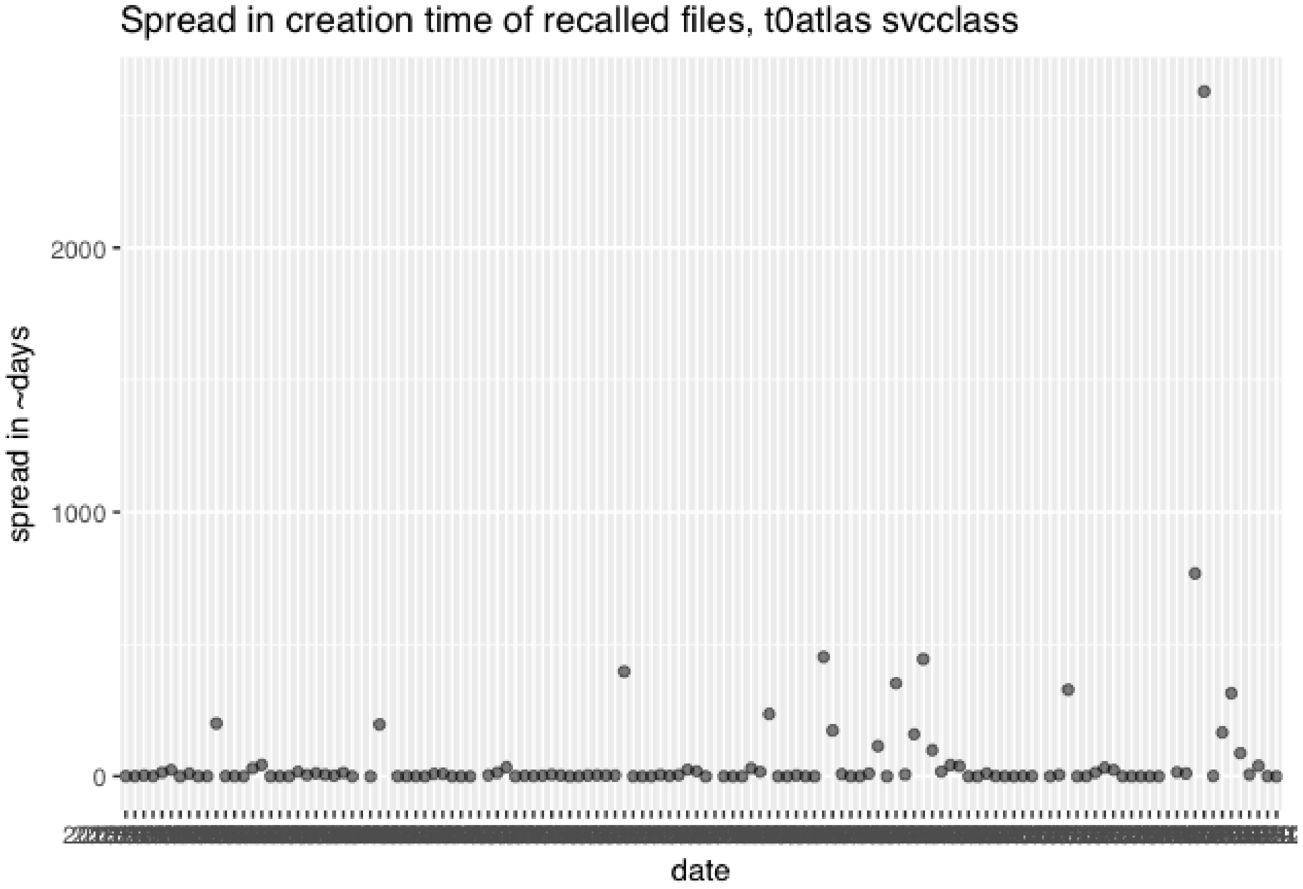
Mean number of different file creation dates on a given day: 49.6

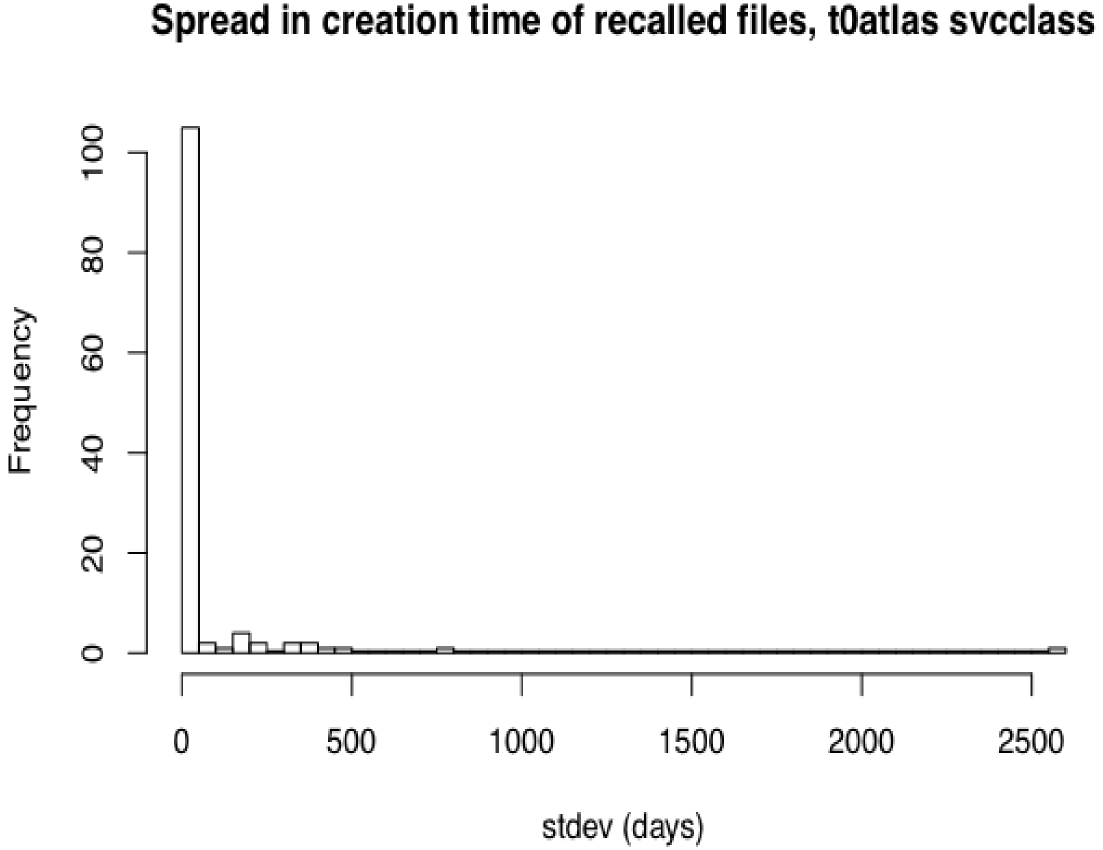




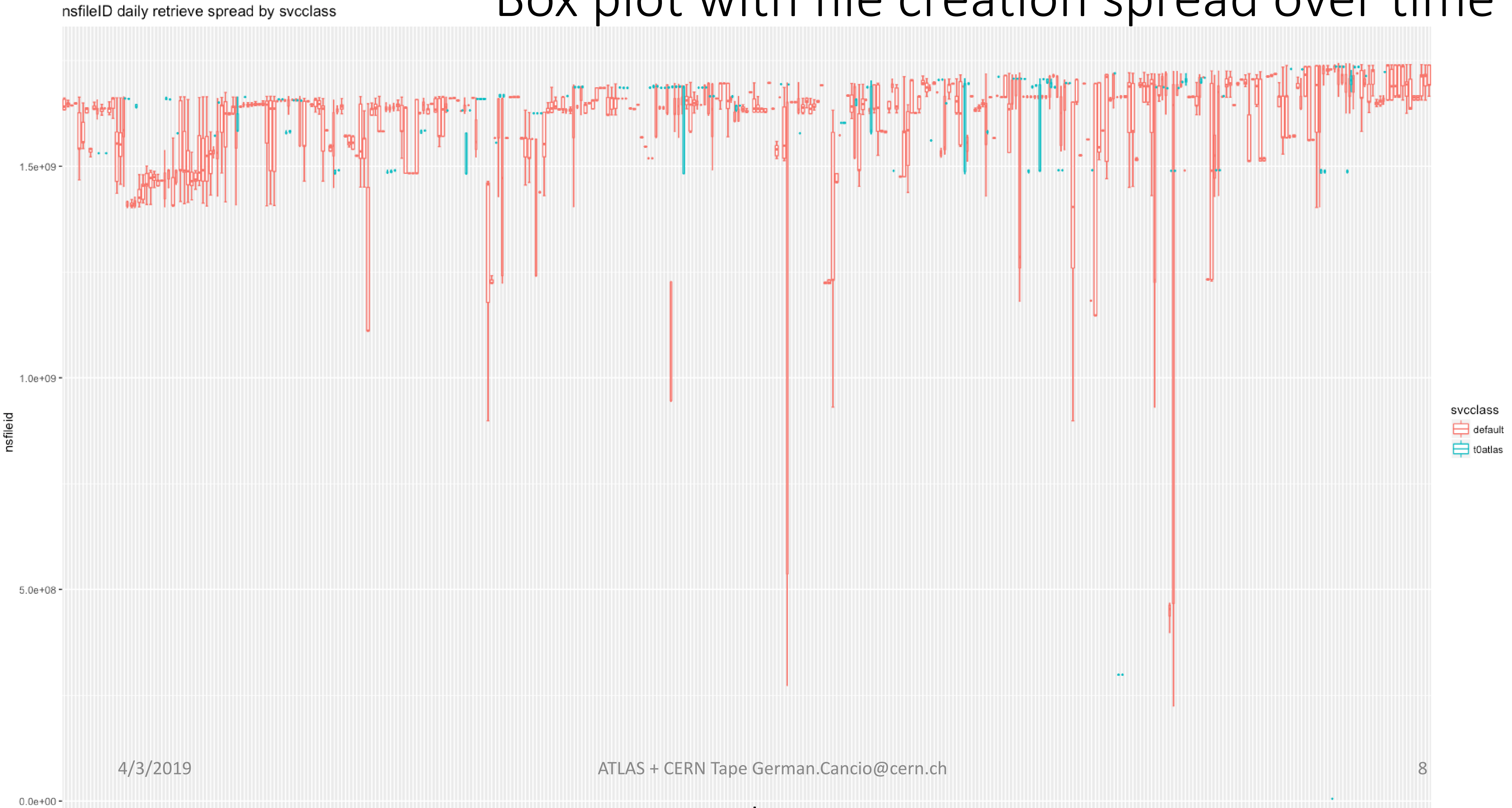
# Daily creation time dispersion of recalled files (stdev, mean): t0atlas

Mean number of different file creation dates on a given day: 12.2

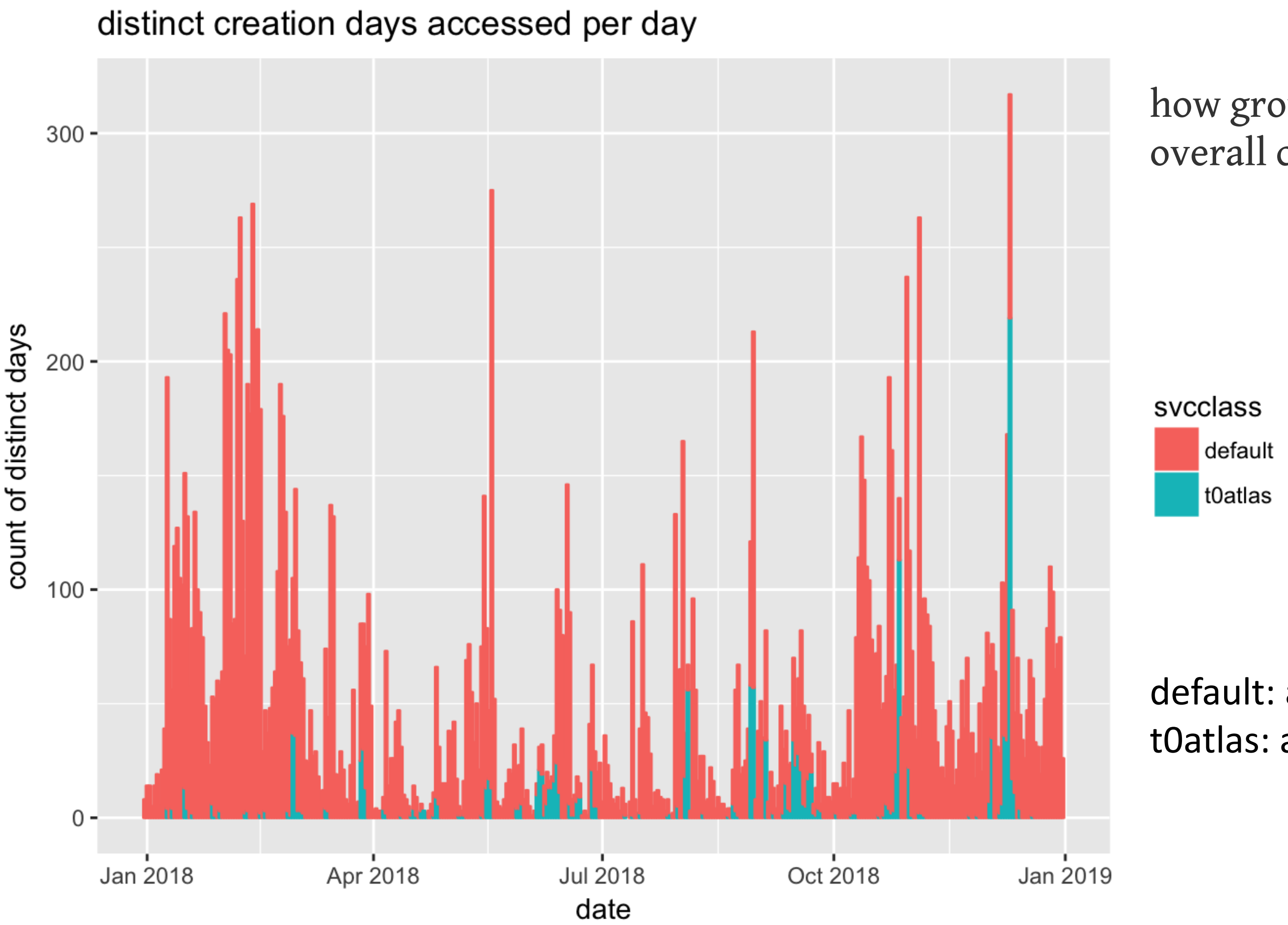




# Box plot with file creation spread over time



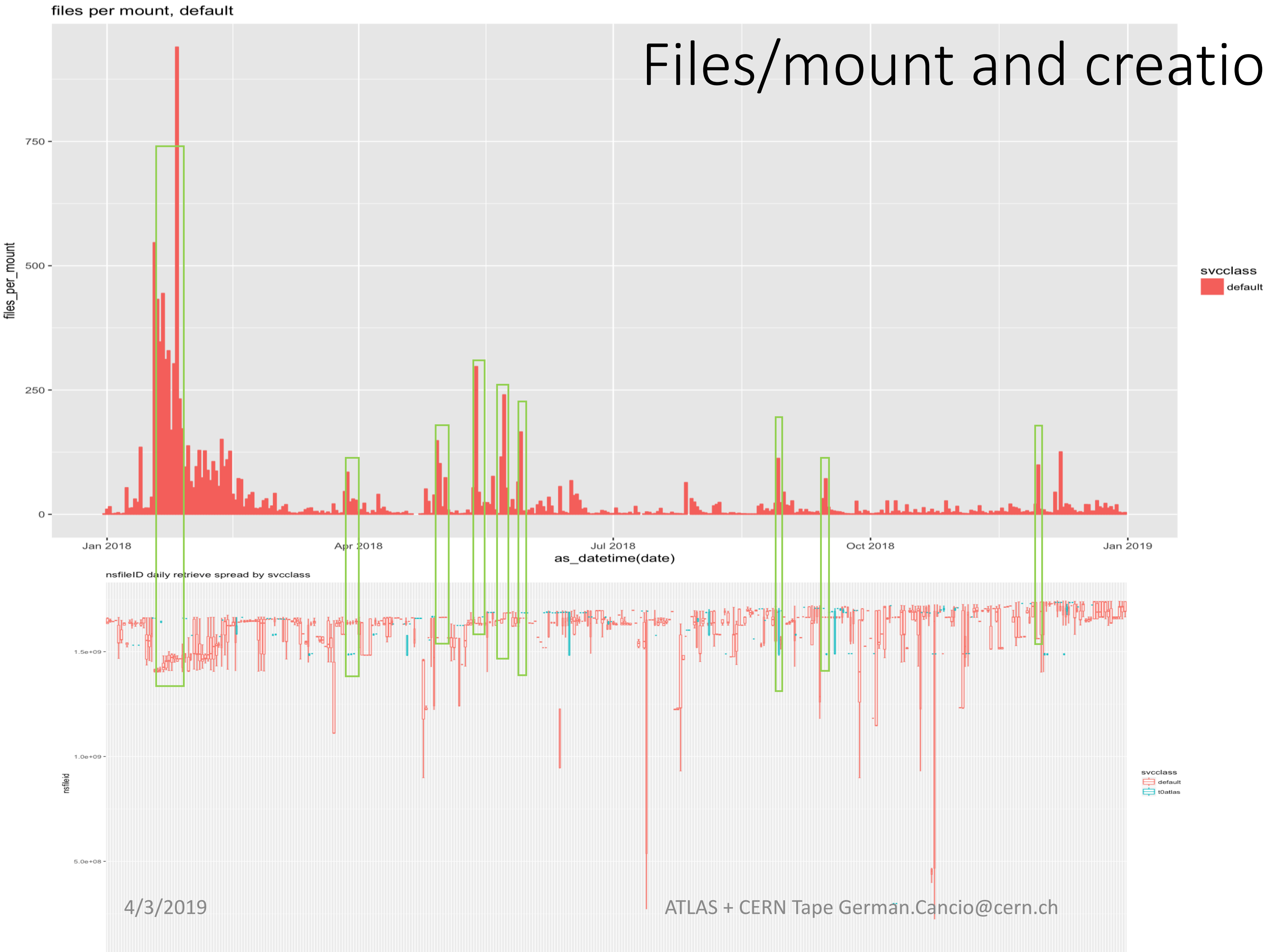
# distinct creation time days processed within a day



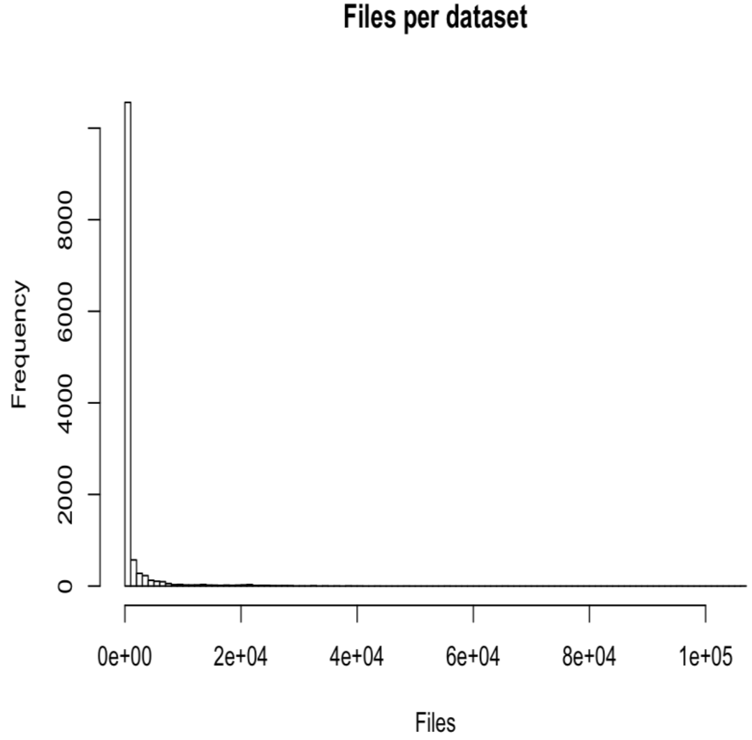
how grouped are the files, in addition to the overall creation time spread?

default: average of 50 different creation days
t0atlas: average of 12 different creation days

# Files/mount and creation time dispersion



# ATLAS datasets



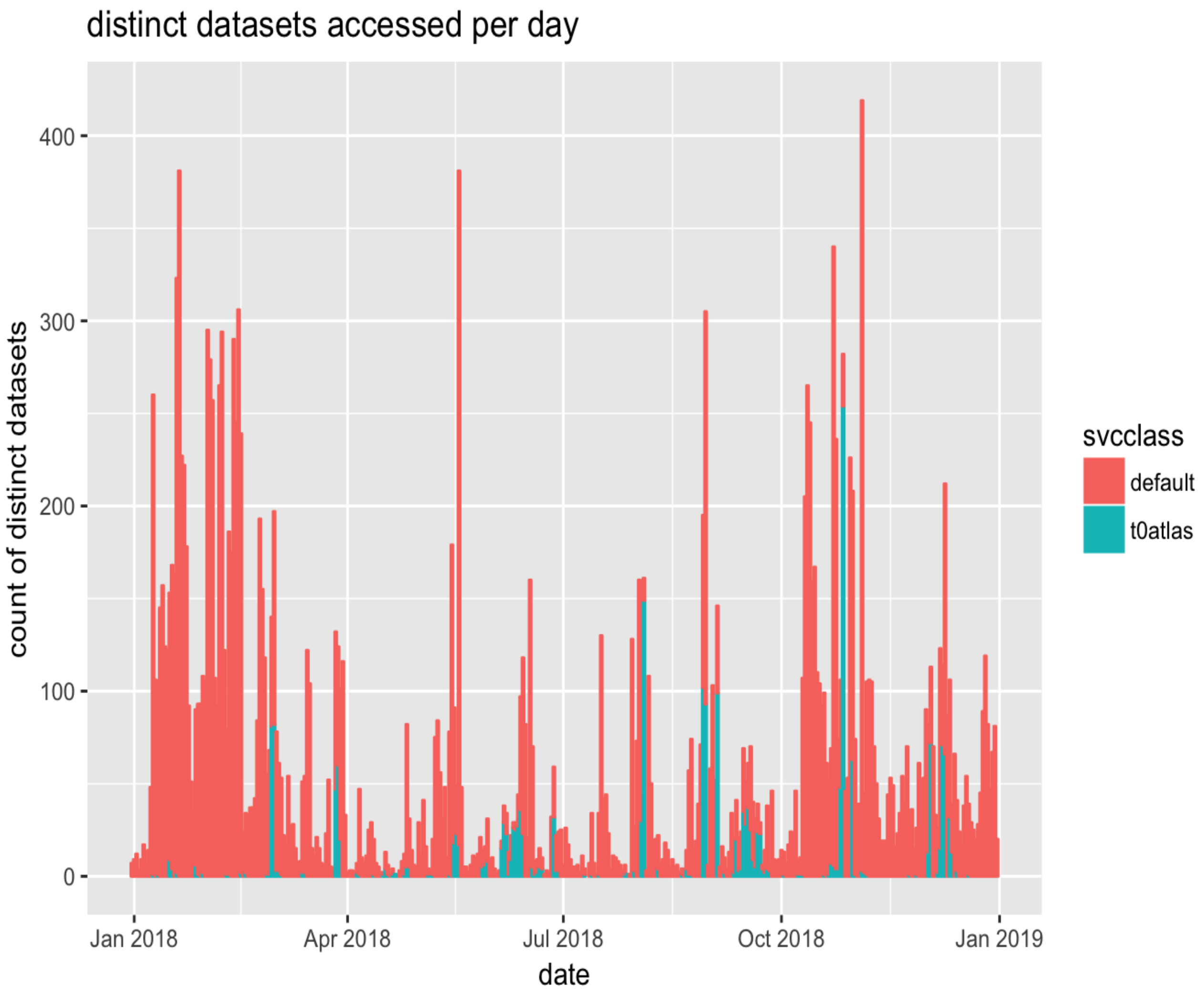
- Total number of datasets in CASTOR: 12461 (99.6 % RUCIO)
- Total number of files across all datasets: 14.3 million
- Total volume across all datasets: 30 PB
- Average number of files per dataset: 1145; median: 55
  - The mean and median differ substantially: There are many datasets with just 1-2 files (27.6%) and there is a tail of a few, long datasets
  - Average volume per dataset: 2.5 TB; median: 0.03 TB.
- Average spread in time (first to last creation time): 2.5 days, median: ~2 hours
  - If a file has been re-created (e.g. re-import from T1 following a file loss), the spread will become artificially large

# Dataset access patterns (2018)

- default: 9482 datasets accessed in 73.8K mounts, fraction of volume read per dataset: **13%** (3.7PB)
- t0atlas: 3214 datasets accessed in 7.6K mounts, fraction of volume read per dataset: **102%** (4.1PB)

Average mounts per dataset:

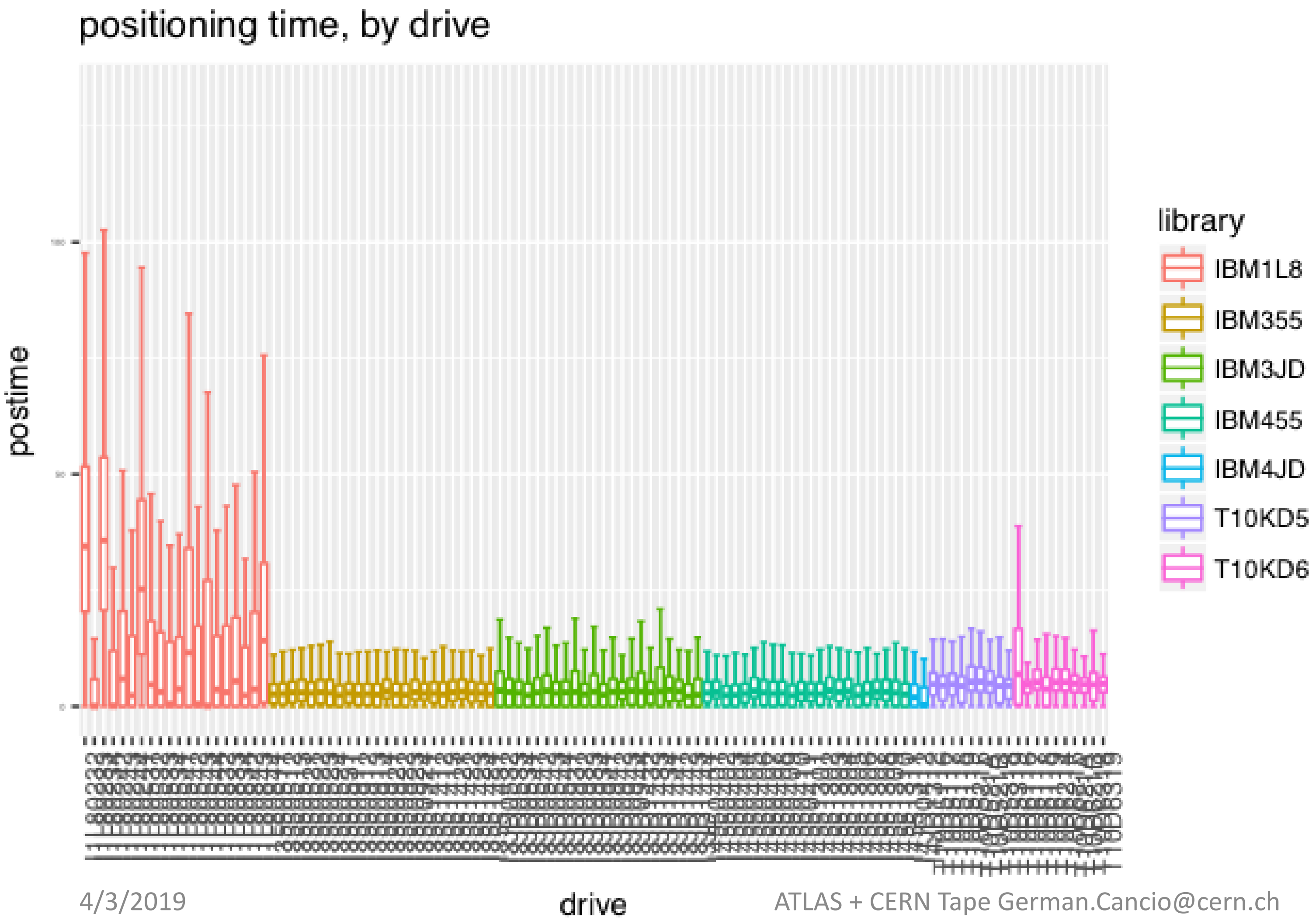
- default: 7.8 mounts / dataset
- t0atlas: 2.4 mounts / dataset



# LTO positioning times

LTO positioning is significantly worse than on enterprise tape

Will be addressed at CERN with CTA ("CERN RAO"), but will affect retrieval from other ATLAS sites



# Discussion - Tape Queueing (1)

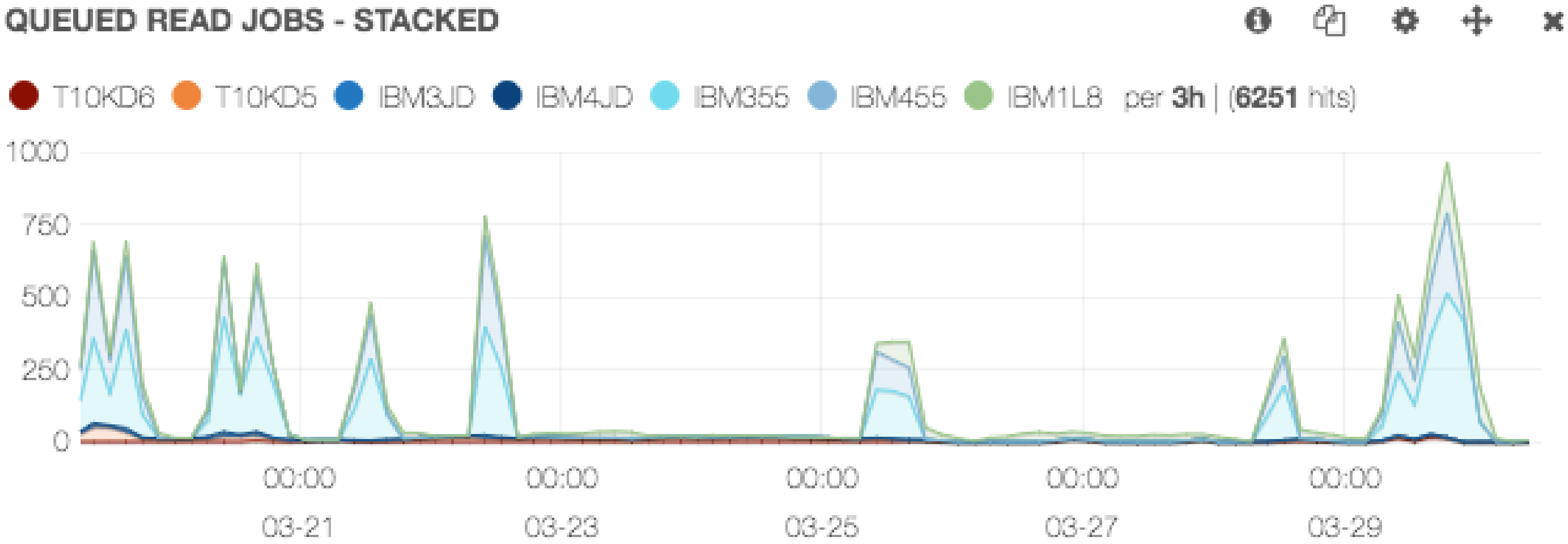
Today, in CASTOR, a two-level queueing is used: Tape read requests queue up a) first within the stager and b) then on VDQM.

a) the stager sorts requests, and holds back submitting tape mounts to VDQM to not exceed a given number of parallel mounts

- But it doesn't distinguish whether submitted jobs are running or just queued (e.g. due to busy library) – tarpit effect
- Workaround: Jobs are (manually) released to VDQM, sometimes creating "avalanches"

b) VDQM tape queues processed (per library) in FIFO order across all stagers.

- First come, first serve, VO's have all same priority.
- If stager of VO A submits 10K tape mounts for a given library before VO B submits a single one, VO B will have to wait.
- Consequence of "split-brain" queueing mechanism between CASTOR stagers and VDQM



# Discussion - Tape Queueing (2)

In CTA, queueing is handled uniformly

- Tape mount requests are inserted in a queue, picked up by tape servers
- Decision on what tape to be mounted next is done by each tape server, which knows exactly the availability of its library/drive
- Configurable policies
  - write: max concurrent drives (as function of data throughput), by storage class
  - read: priority, max concurrent drives, policy (volume_first|FIFO)
- Drive allocation across VO's is fairshare-based
  - VO B will get its mount through without having to wait for 10K mounts of VO A to complete!

# Discussion - Recall priorities / policies via FTS

- CASTOR mount policy does reorder tape requests taking into account volume
    - A tape with few files to be recalled may fall behind others in the queue with more files
- ATLAS requests:
    - Allow to retrieve tapes in FIFO - ensure that datasets can be retrieved completely before reprocessing
    - Allow to set priorities for different workloads (eg. T0 jobs >> grid jobs)
- CTA/FTS team proposal:
    - Mount policy to be (optionally) provided via FTS
    - `FTS.prepare("/atlas/mydataset/myfile.RAW?eos.CTAMountPolicy="FIFO") (or job metadata attribute)`
    - Defaults can be set
    - Priority to be honoured (already provided by ATLAS to FTS via bringOnline job API)
    - Open questions: How to protect from abuse / priority inflation? Will there be any non-FTS access?

# Discussion - Storage class selection via FTS

- CASTOR allows for setting a “file class” per directory, which is inherited further down
  - this requires using a CASTOR specific command (`nschclass`) with specific privileges
- ATLAS requests:
  - allow setting “storage class” via FTS
  - allow setting “storage class” on a per-file level
- CTA/FTS team proposal:
  - Storage class to be (optionally) provided via FTS using URL (and/or job) parameters
  - `FTS.destURL(“/atlas/mydataset/myotherfile.AOD?eos.CTAStorageClass=“ATLAS18_AOD”)`
  - Per-directory defaults can be set
  - Storage classes (and their mapping to tape pools) are pre-defined
  - Changing storage class of a file is possible, but takes only effect after repack

# CTA plans for ATLAS

- Step 1. Production-quality endpoint for T0 archiving
  - In parallel to CASTOR
  - Revamped eosctaatlaspps.cern.ch instance (Julien)
    - SSD-based disk servers, total capacity: ~20 TB, throughput: 40Gb/s (to grow to ~80 TB and 60 Gb/s by EOY 2019, then to ~250 TB and nominal rates (7.5GB/s?) by start of Run-3
  - Initially, 20 tape drives (IBM Enterprise, LTO) shared across EOSCTA instances (grow to ~200 by start of Run-3)
  - Target: ASAP
- Step 2. CASTOR->CTA switchover for ATLAS
  - Working on migration tools and workflow (Michael)
    - Will involve switching CASTORATLAS to R/O beforehand and pre-import of metadata from CASTOR
    - Switch off CASTORATLAS after CTA fully commissioned
  - Setting up operational environment, monitoring (Julien)
  - Data Challenges?
  - FTS enhancements (file on tape check)
  - Target: Q3/Q4 2019