

Data access patterns analysis at PIC CMS Tier 1

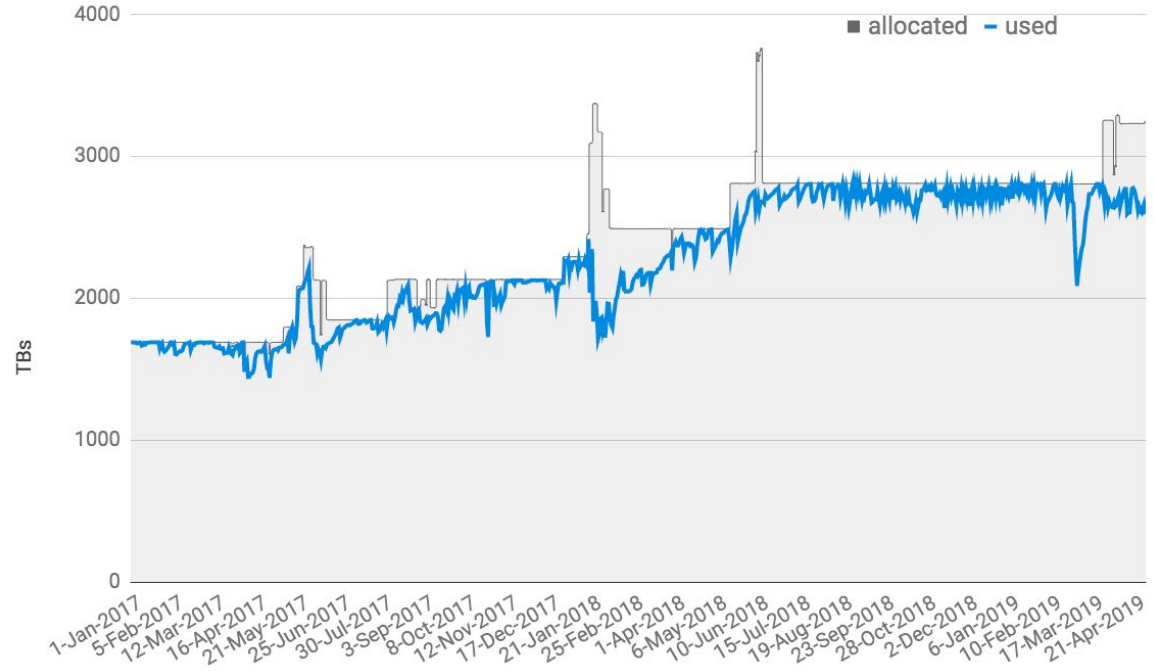
Carlos Pérez Dengra [PIC/CIEMAT]
PhD student at PIC

pre-GDB XCache - CERN 8th July 2019

Data access patterns and disk usage at PIC Tier-1 [CMS]



CMS T1-Disk (dCache-Prod)



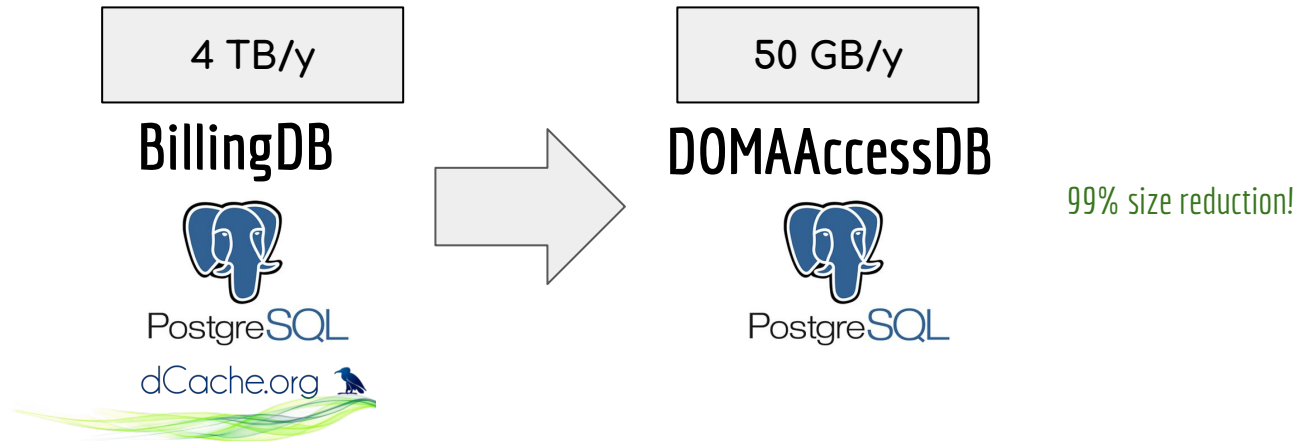
Dynamo manages around 93% of disk usage at PIC CMS Tier-1.

The rest (7%) is used for temporary files, tests, and buffer for the tape system.

Cleanup manager takes care of Dark data, which is centralized at the moment

Data access patterns and disk usage at PIC Tier-1 [CMS]

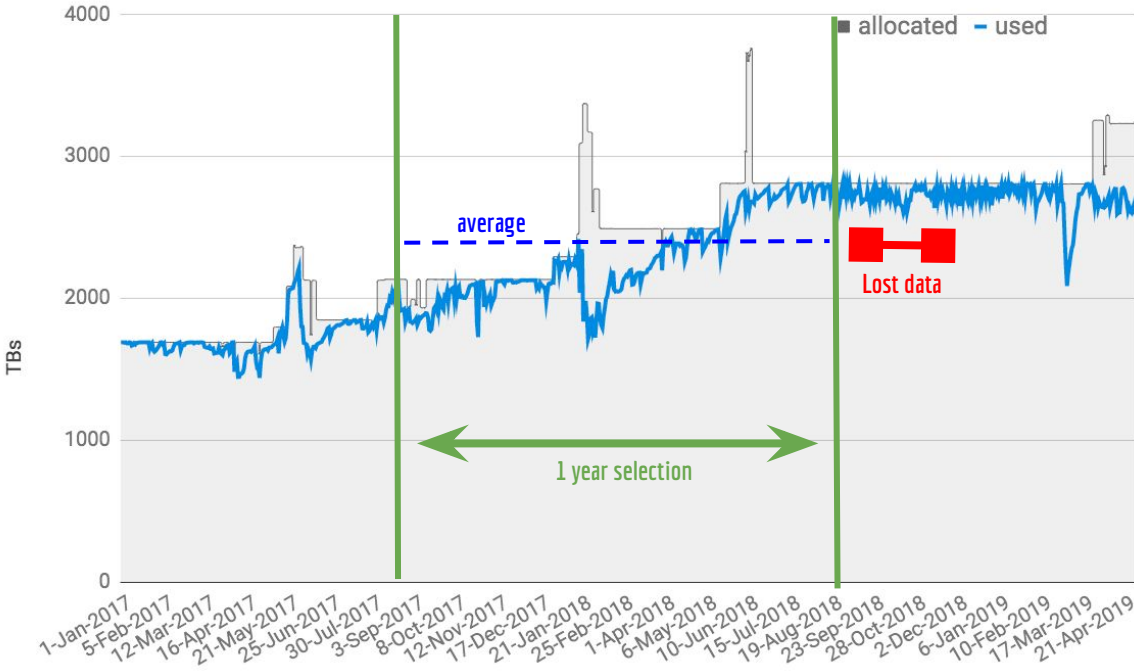
- PIC has a dCache system for storage management.
- All the accounting information of all experiments we provide resources at PIC (besides a CMS Tier-1 is also an ATLAS T-2 centre) is stored at the **billingDB**.
- In order to study the acces and disk usage by CMS, we have made a new and smaller Postgres BBDD allowing us to access all the relevant information in a straightforward way.



Data access patterns and disk usage at PIC Tier-1 [CMS]



CMS T1-Disk (dCache-Prod)



Selection of a year to study

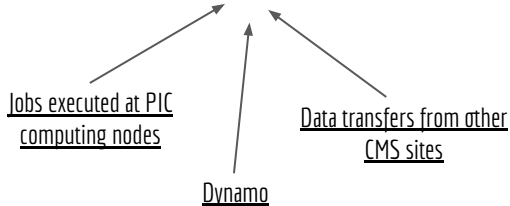
Sept-2017 → Sept-2018

Average disk usage ~2.3 PB

8.8 PB writes (10.5M files)

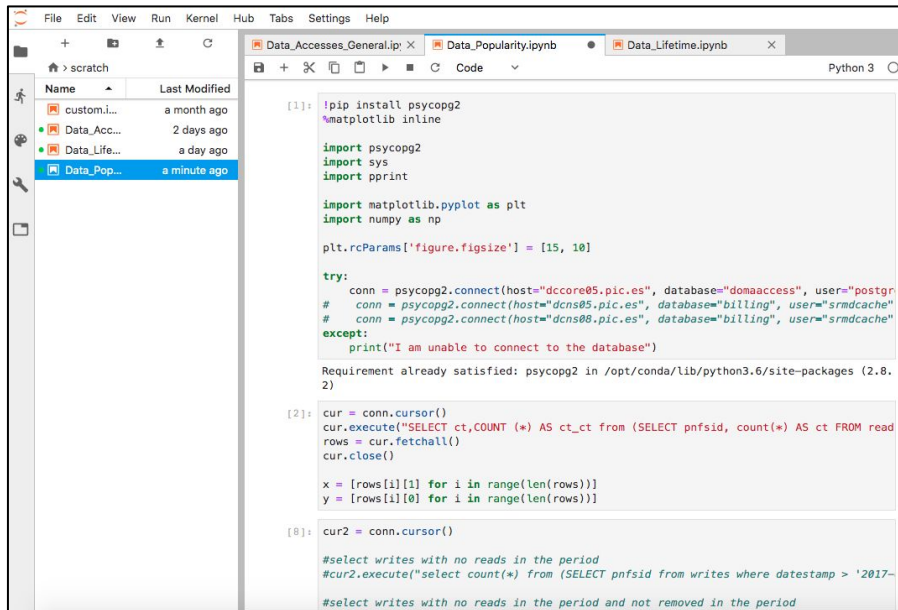
24.0 PB read (3.5M distinct files)

8.8 PB removed (11.0M files)



Data access patterns and disk usage at PIC Tier-1 [CMS]

- Data analysis and calculations are performed by a containerized JupyterLab instance through Kubernetes, with a Dask cluster available.



```
[1]: !pip install psycopg2
      %matplotlib inline

import psycopg2
import sys
import pprint

import matplotlib.pyplot as plt
import numpy as np

plt.rcParams['figure.figsize'] = [15, 10]

try:
    conn = psycopg2.connect(host="dcore05.pic.es", database="domaaccess", user="postgr
# conn = psycopg2.connect(host="dcns05.pic.es", database="billing", user="srmocache"
# conn = psycopg2.connect(host="dcns08.pic.es", database="billing", user="srmocache"
except:
    print("I am unable to connect to the database")

Requirement already satisfied: psycopg2 in /opt/conda/lib/python3.6/site-packages (2.8.2)

[2]: cur = conn.cursor()
cur.execute("SELECT ct,COUNT(*) AS ct_ct from (SELECT pnfsid, count(*) AS ct FROM read
rows = cur.fetchall()
cur.close()

x = [rows[i][1] for i in range(len(rows))]
y = [rows[i][0] for i in range(len(rows))]

[8]: cur2 = conn.cursor()

#select writes with no reads in the period
#cur2.execute("select count(*) from (SELECT pnfsid from writes where timestamp > '2017-
```



kubernetes



Data popularity (how many times are we accessing a file?)

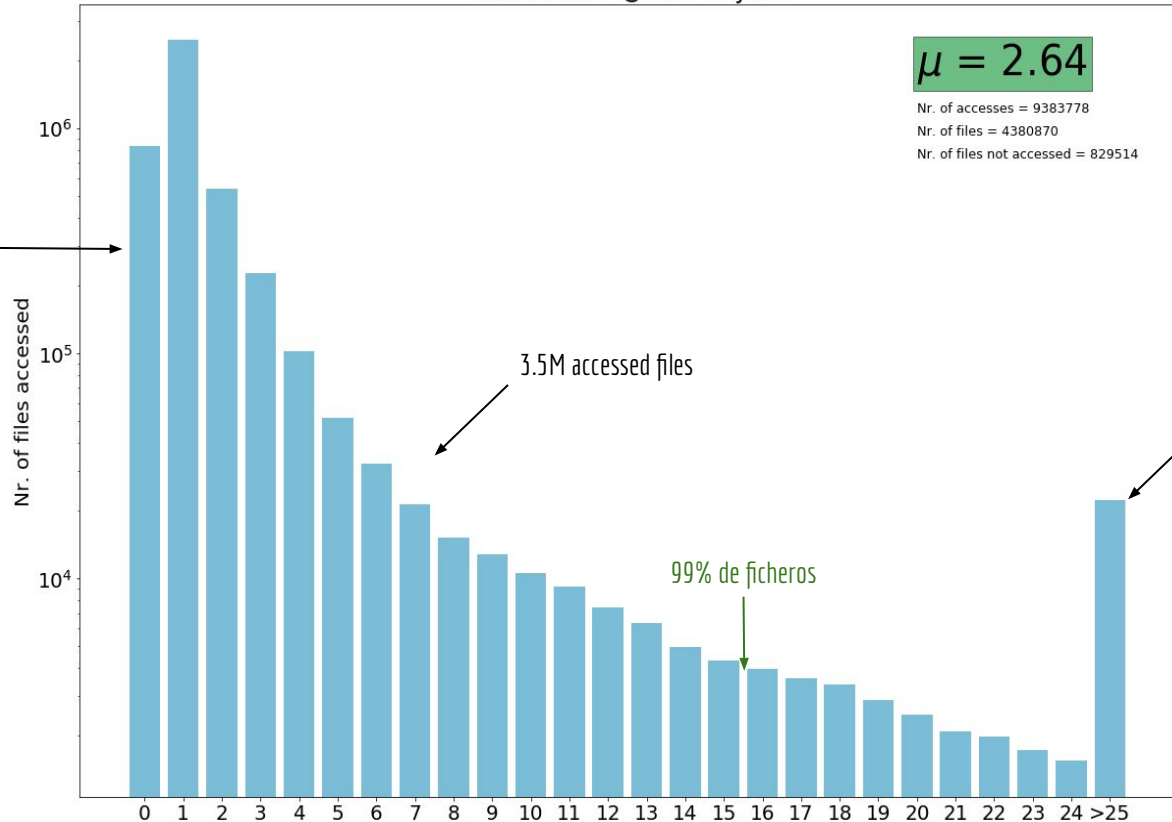
All Data/MC
Tiers

800k written and
unaccessed files

Early files:
temporal undeleted
test files; Dynamo
erros, etc.

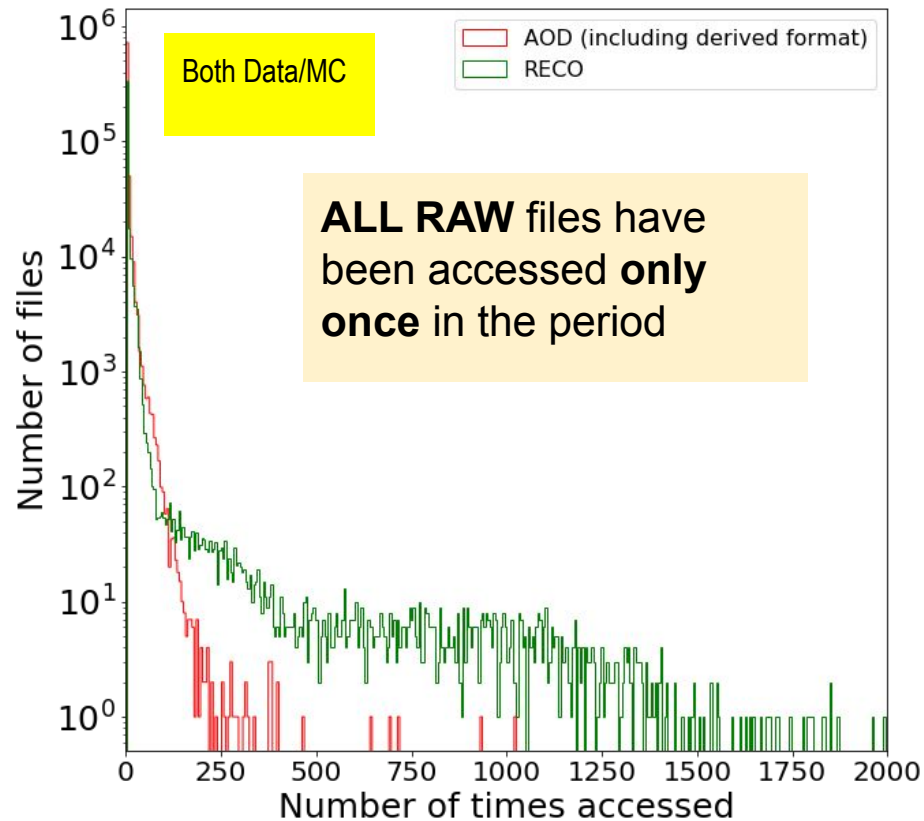
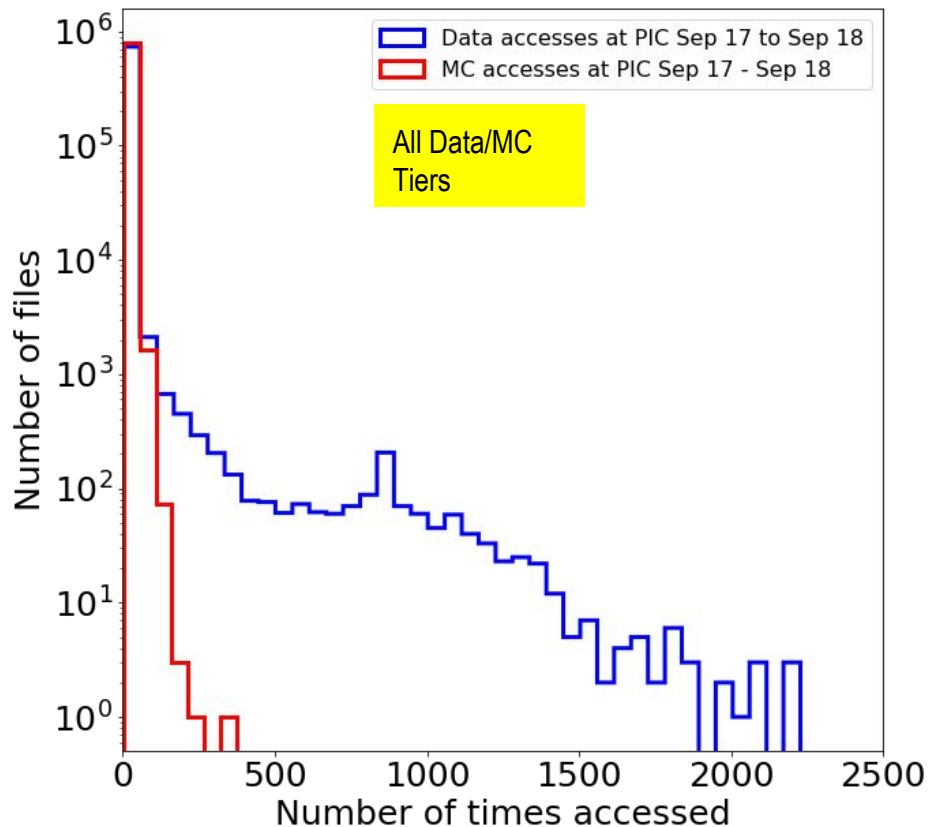
6.7M written files,
unaccessed and
deleted.
Non-considered
here...

Data Access @ PIC - 1 year

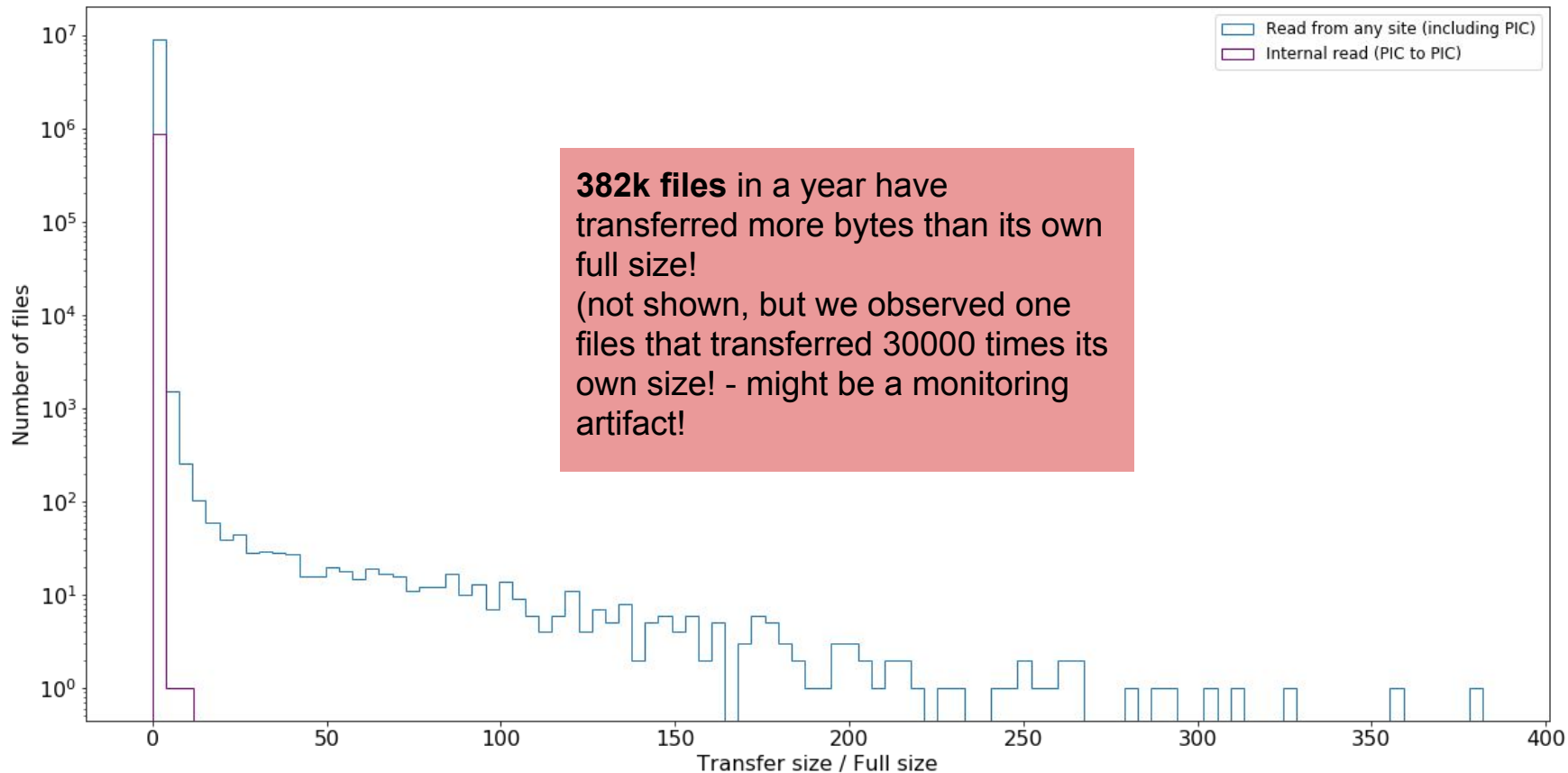


22k files
accessed more
than 25 times.

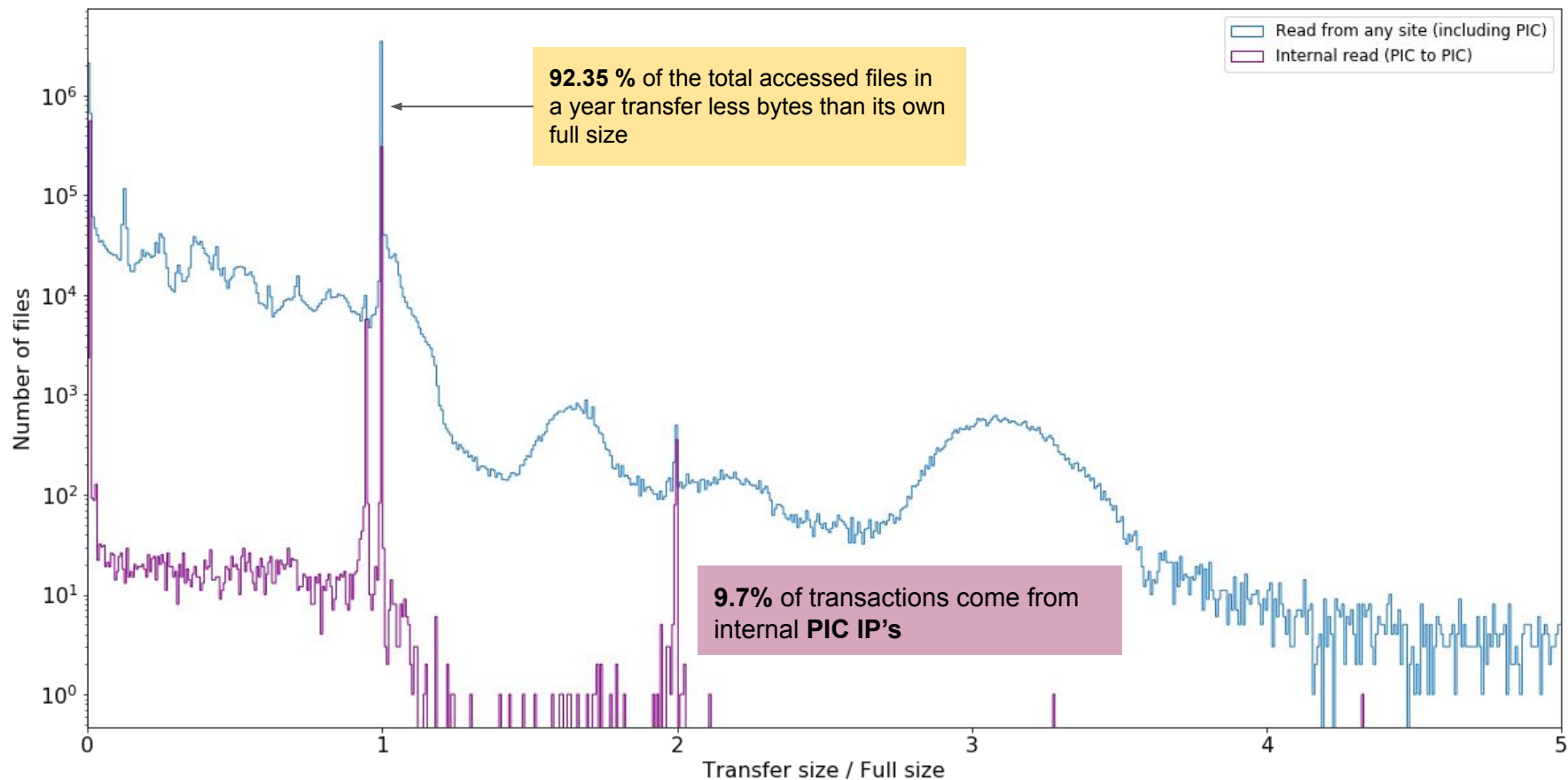
Data popularity (finer view and date tiers)



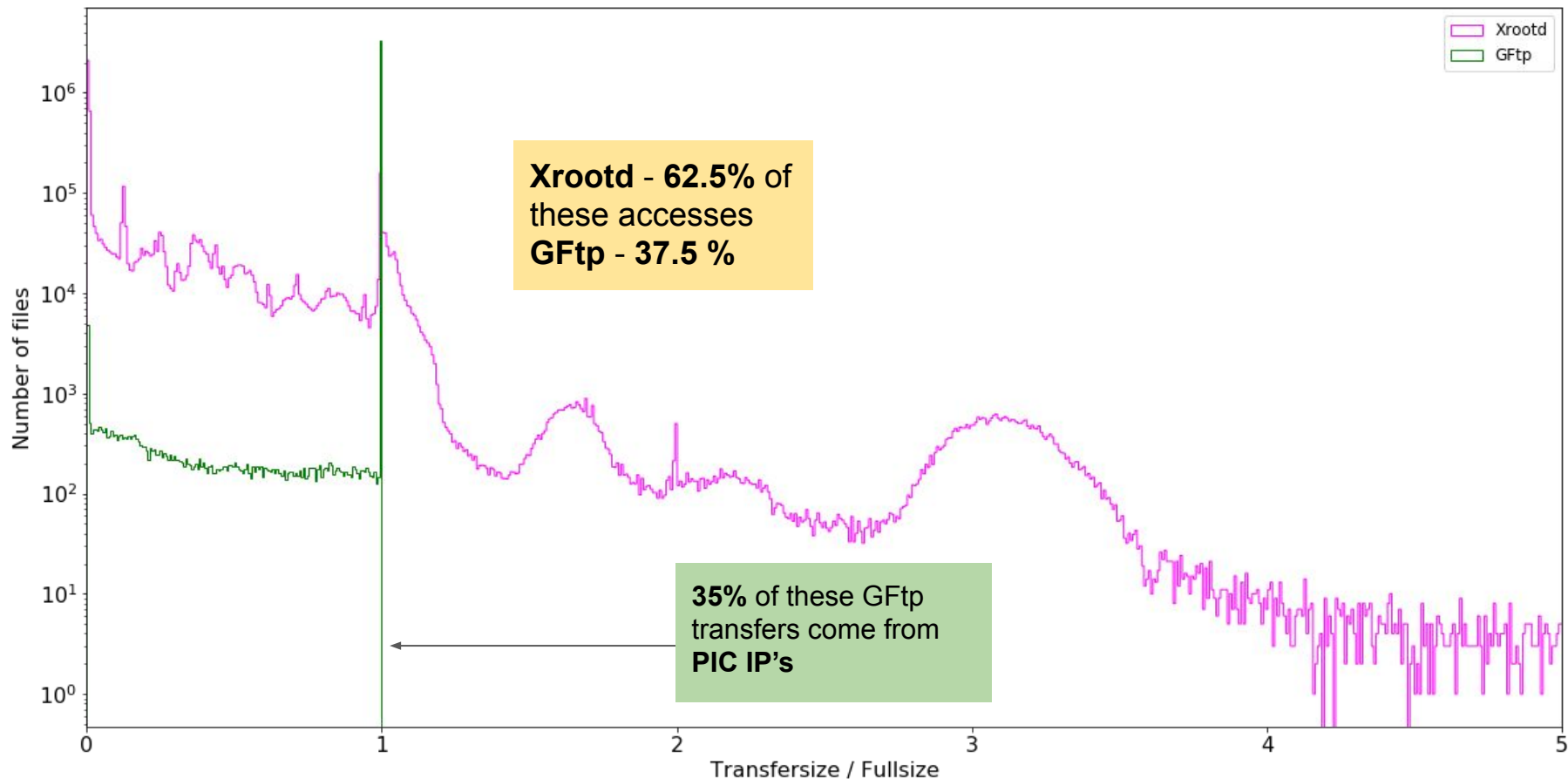
Fraction of the files accessed



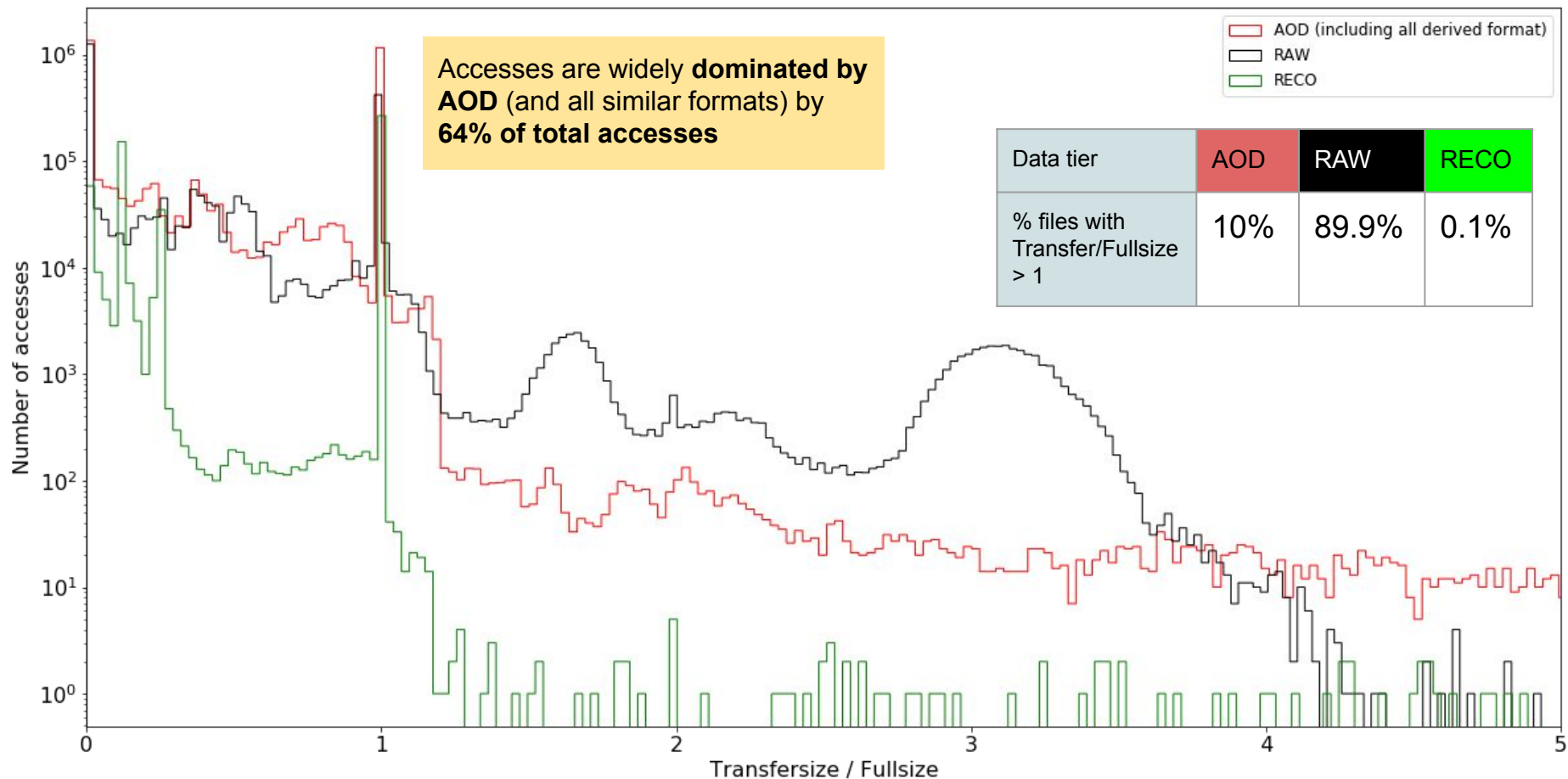
Fraction of the files accessed (re-scaled)



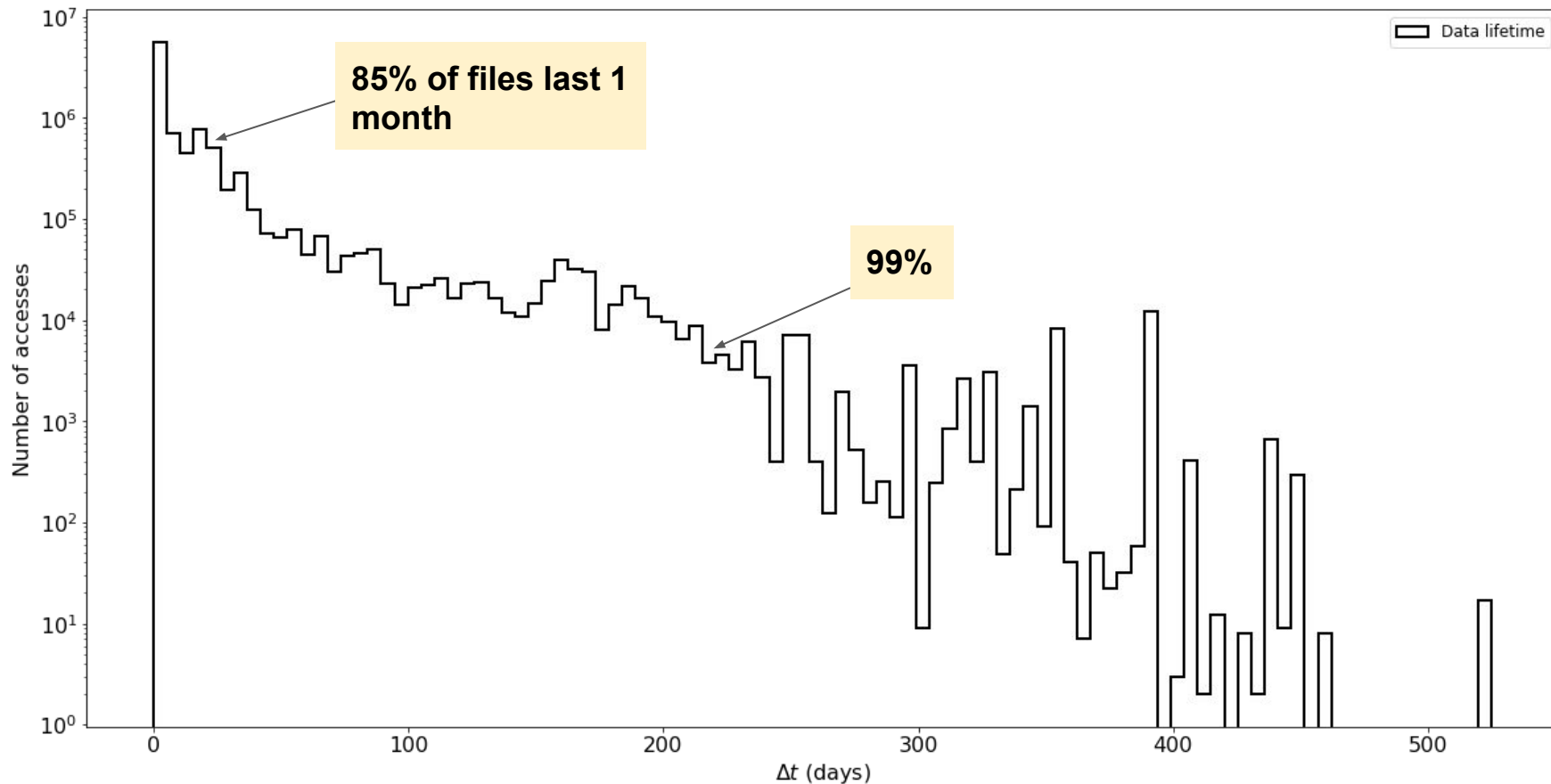
Fraction of the files accessed (protocols)

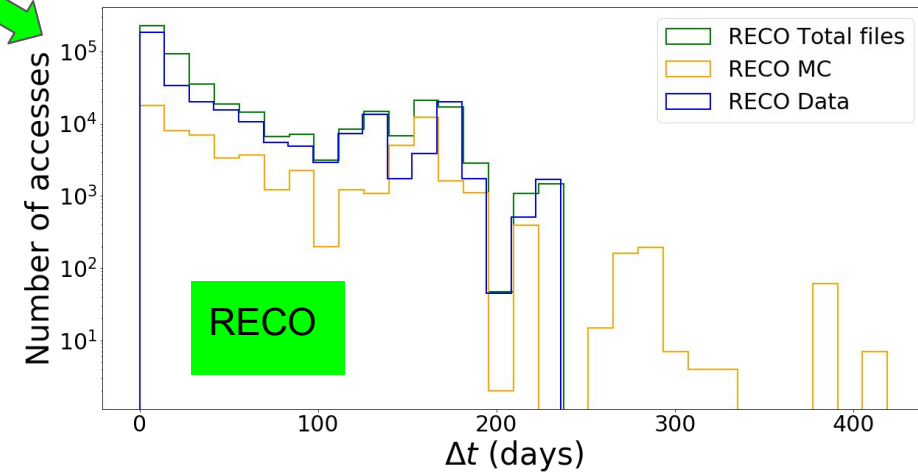
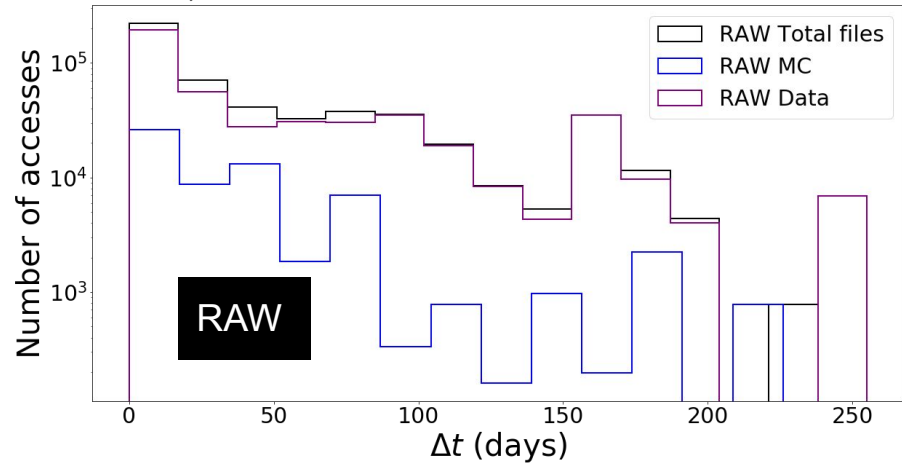
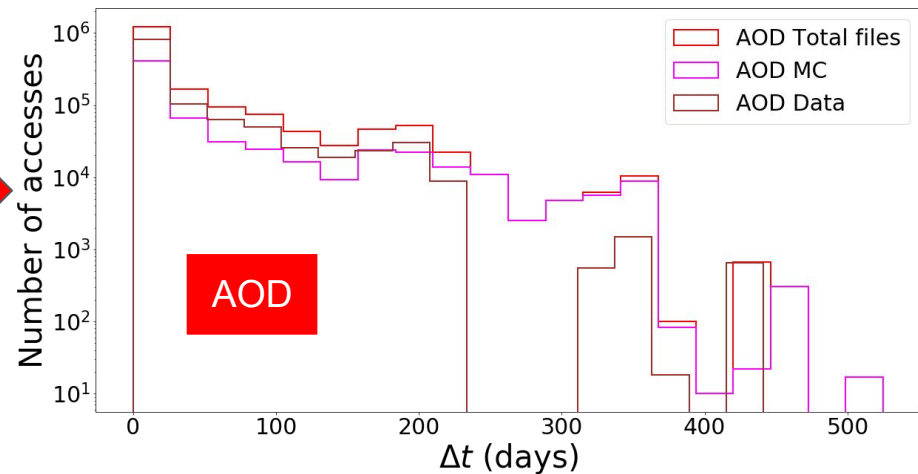
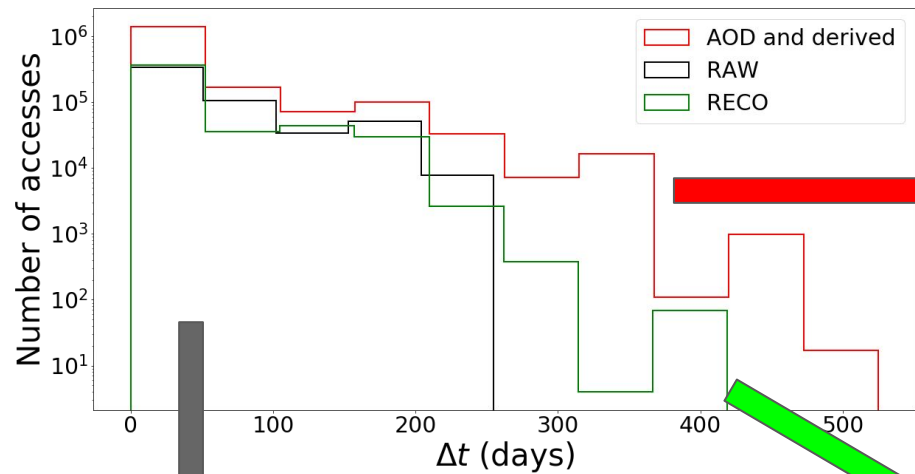


Fraction of the files accessed (data tier)



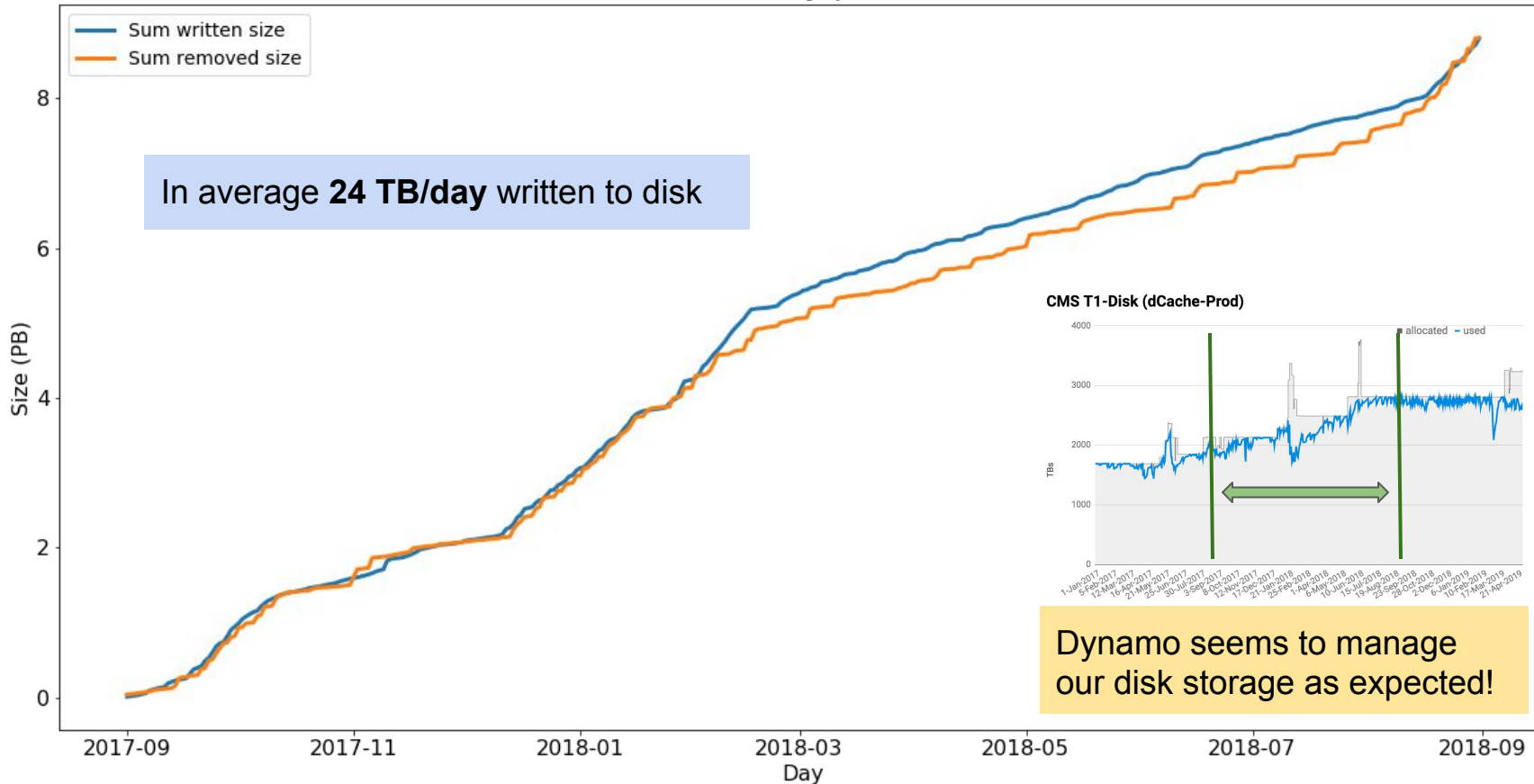
File lifetime (time interval since a file is written and removed from disk)





Write/Remove rate

Cumulative disk size during a year at PIC CMS Tier 1

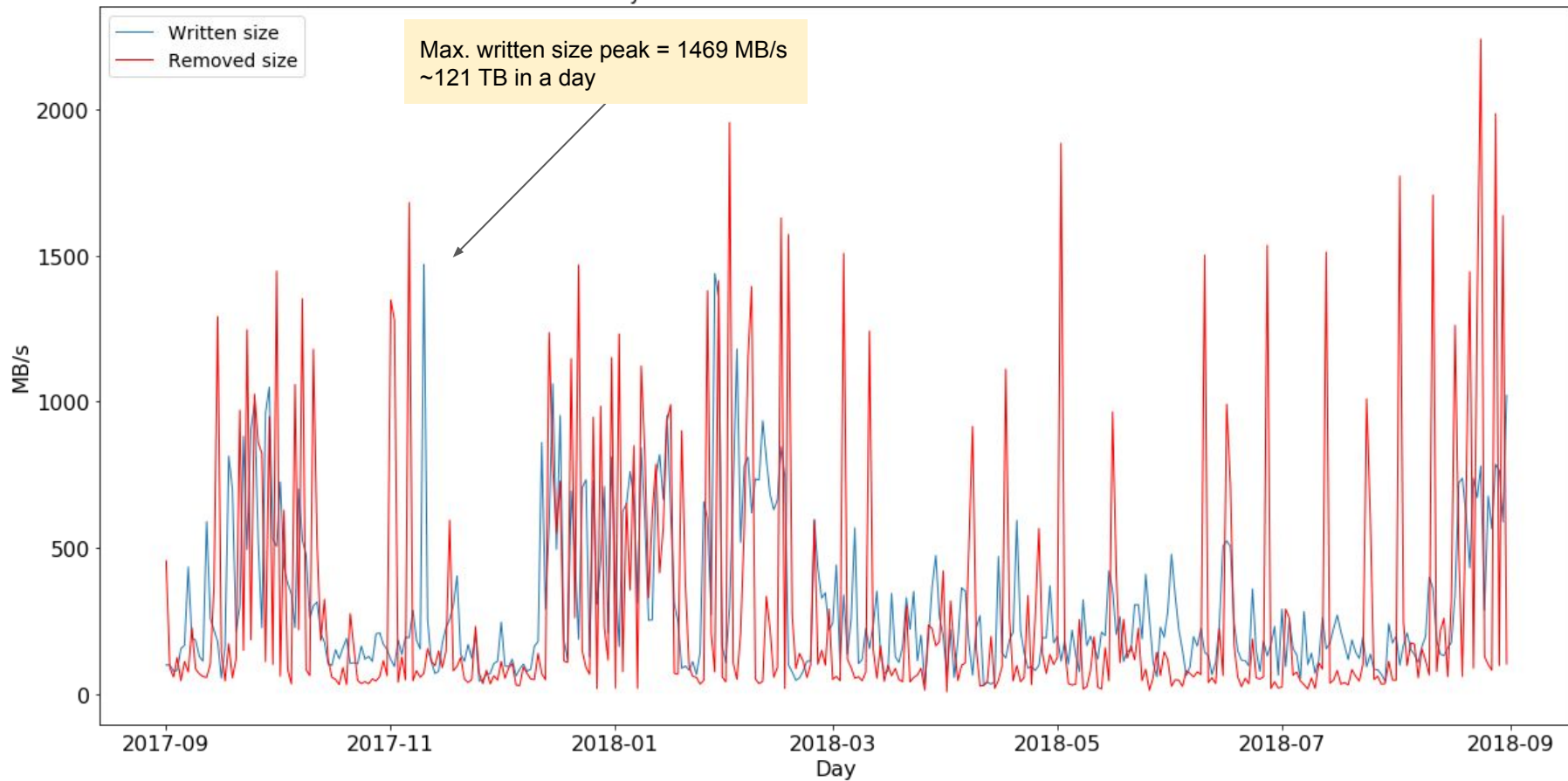


In average **24 TB/day** written to disk

Dynamo seems to manage our disk storage as expected!

Write/Remove rate

Daily written and removed files at PIC CMS Tier 1



Next actions:

- Preliminary results have been done and we are **extending it**
- We are working to study in detail the **file accesses in time**: from creation to first access, all of the accesses intervals, and last access to removal **[in progress]**
- **Simulation** for data likely to be **cached, based in real data**
- We are making the same data analysis by deploying the same setup at **CIEMAT Tier-2 (dCache user as well)**

Sites with dCache could do so?

- Further, we plan to use **Spark's** platform at **CERN IT Hadoop cluster** in order to **understand remote access** to input files for jobs executed at **PIC**.

Could we eventually 'cache' this data as well?

Thank you for your attention