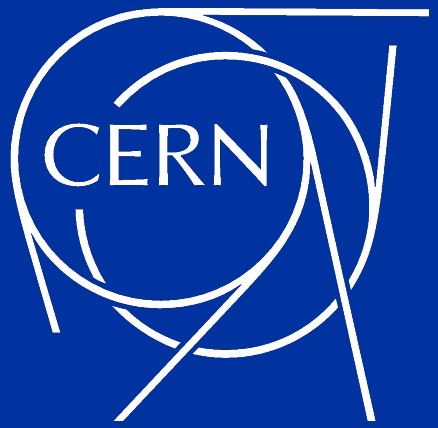


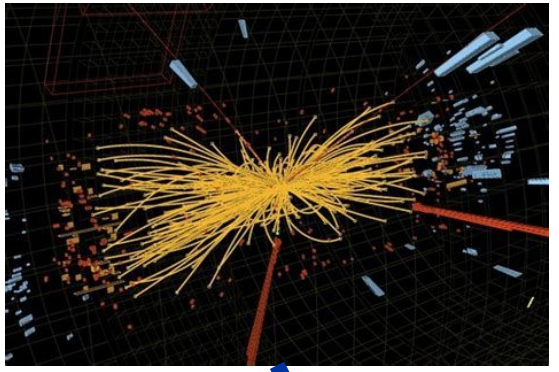
CERN CTA Service

writing LHC data to tape with opensource software on commodity hardware

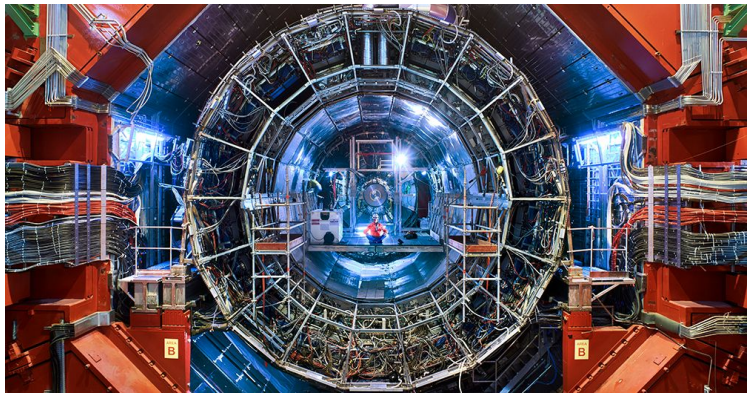
Julien Leduc
CERN Tape Archive Service Manager for the CTA team

2026-06-10

CERN *oversimplified*

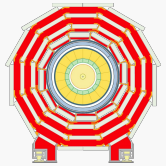


CERN Data Taking Flow



LHC Experiment Site

DAQ



CERN Data Centre

T0 Disk

EOS

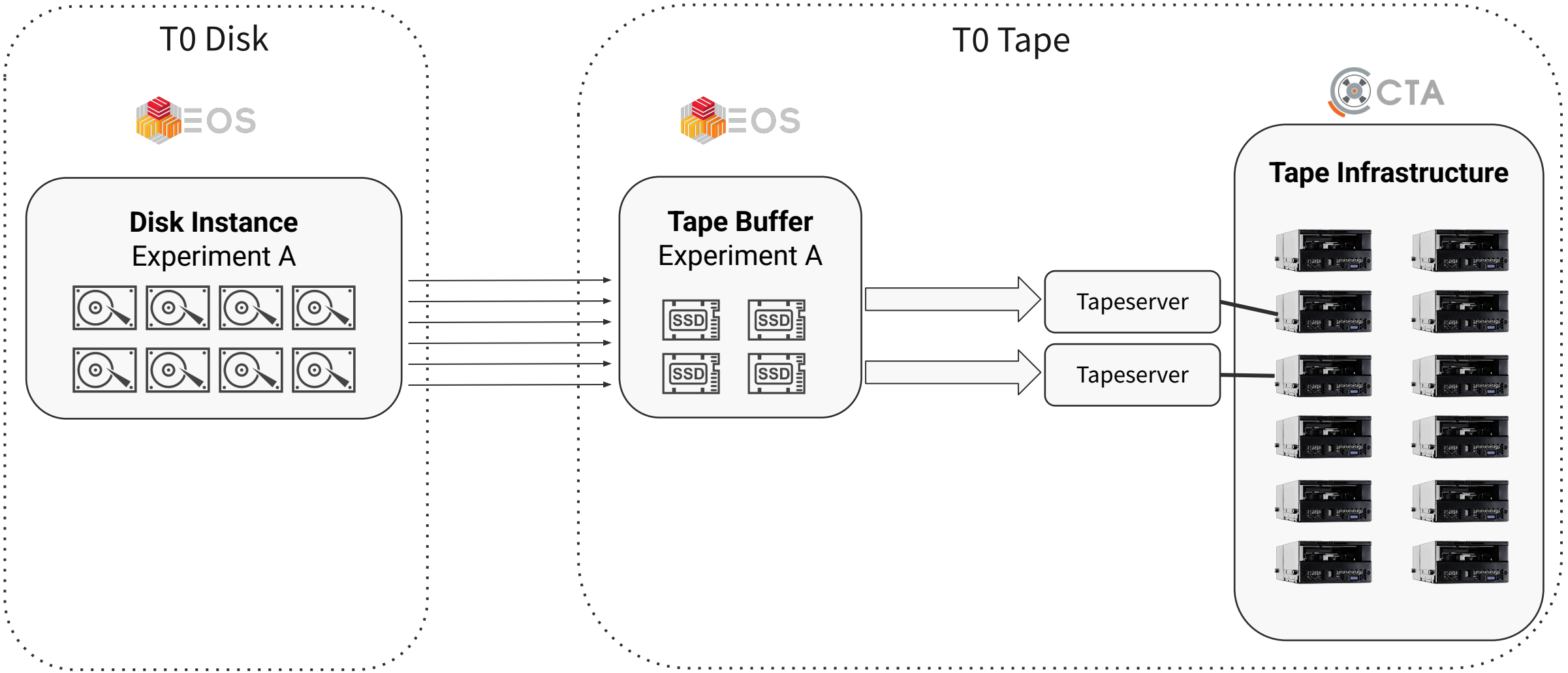
FTS
File Transfer Service

T0 Tape

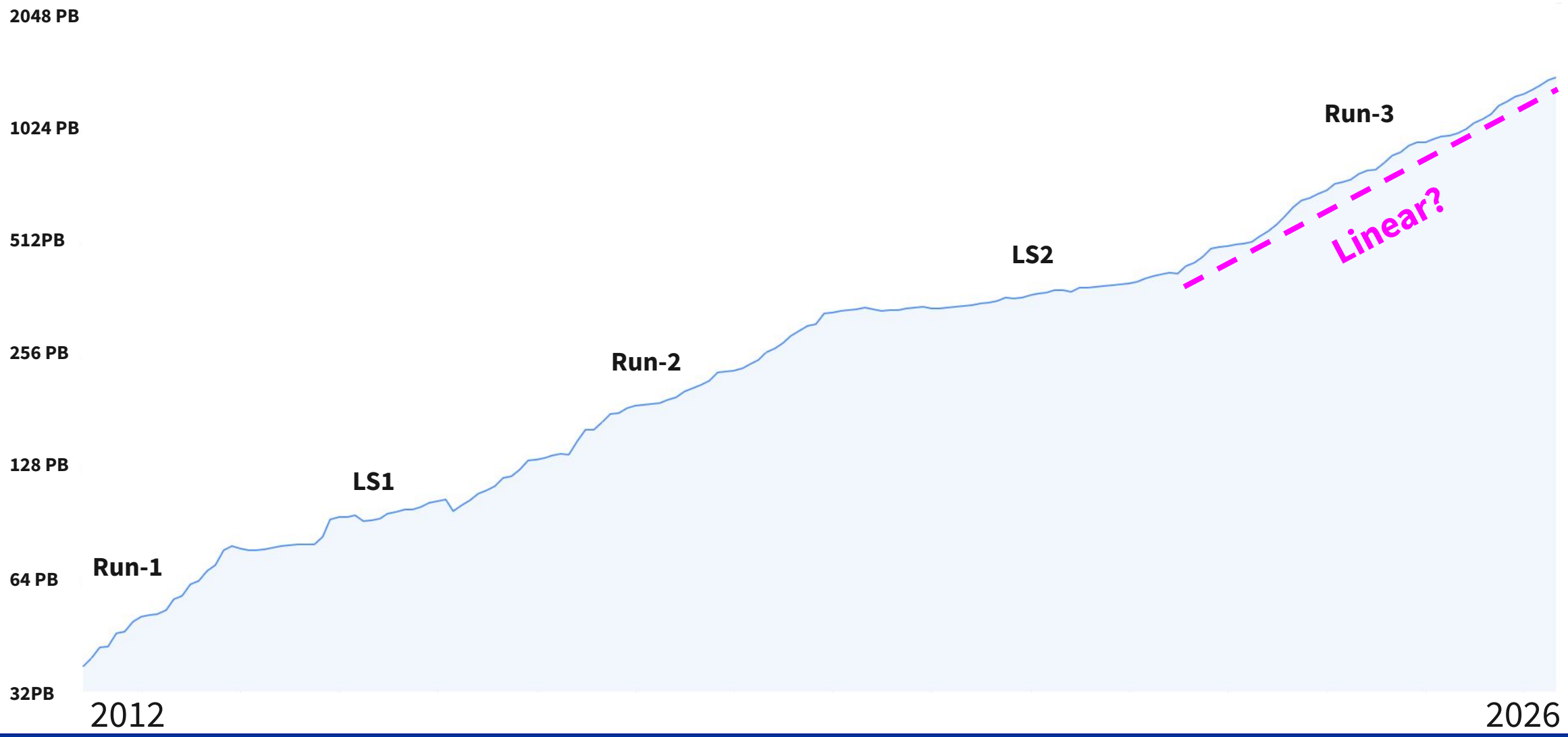
CTA

Experiment Data Management

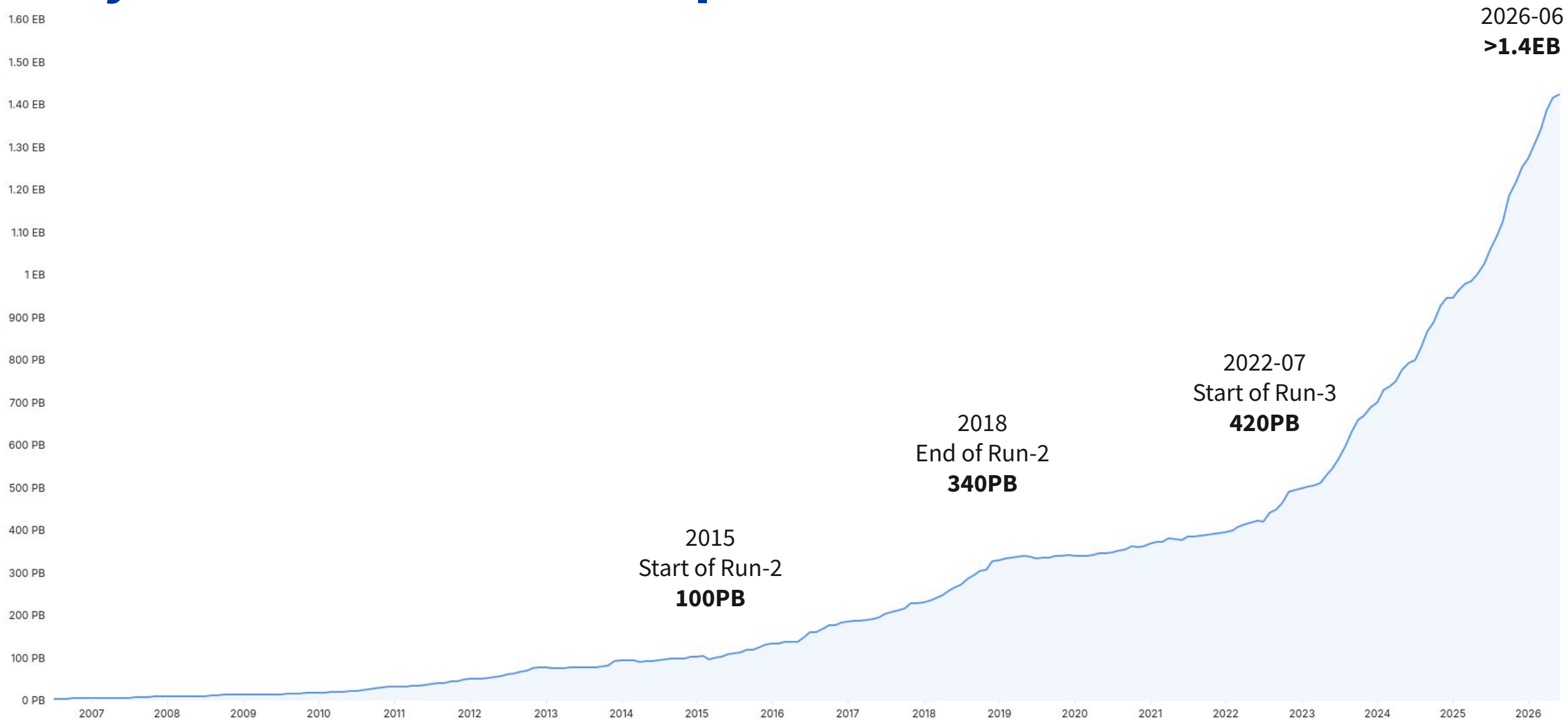
T0 Data Taking Storage details



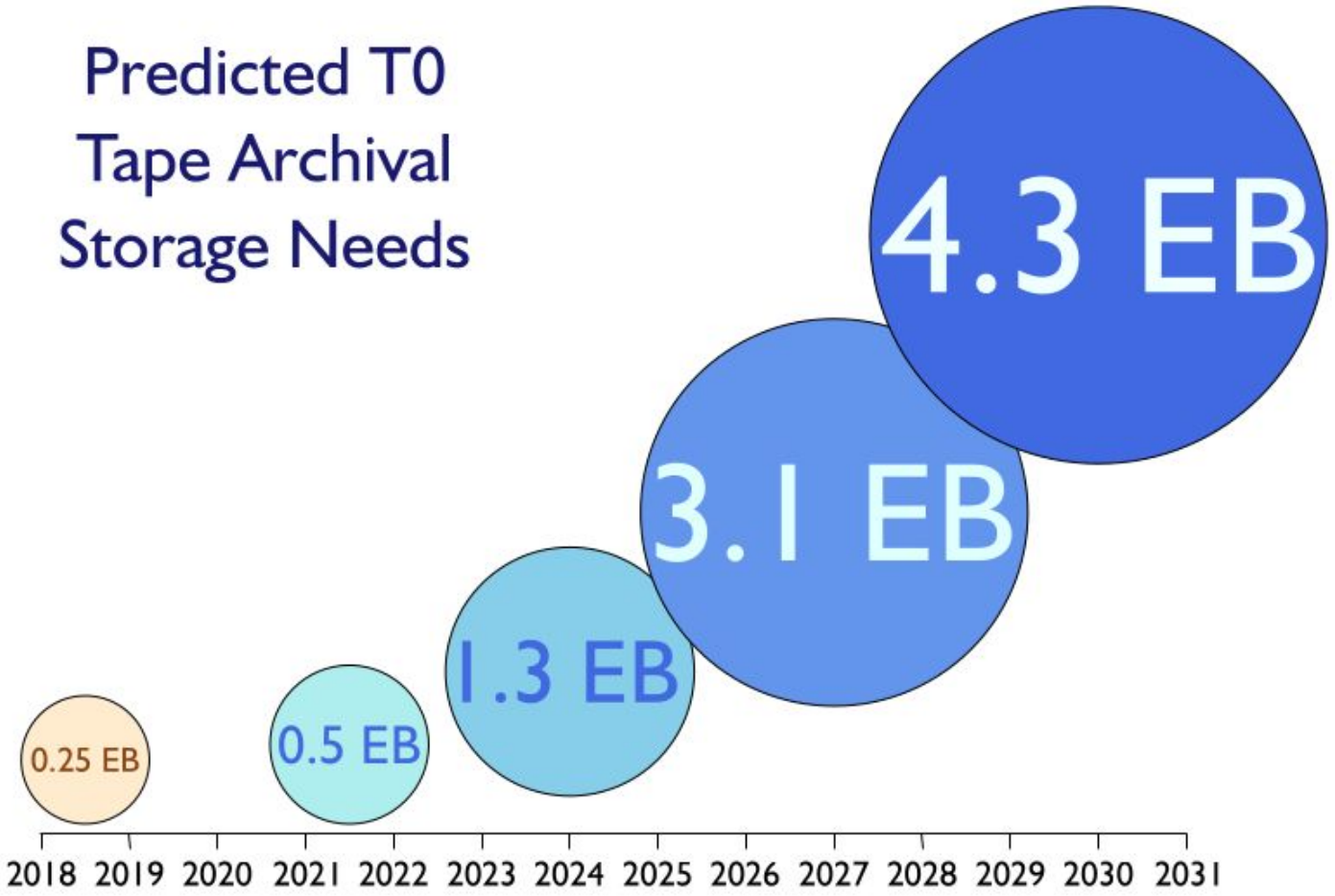
Physics data stored on tape



Physics data stored on tape



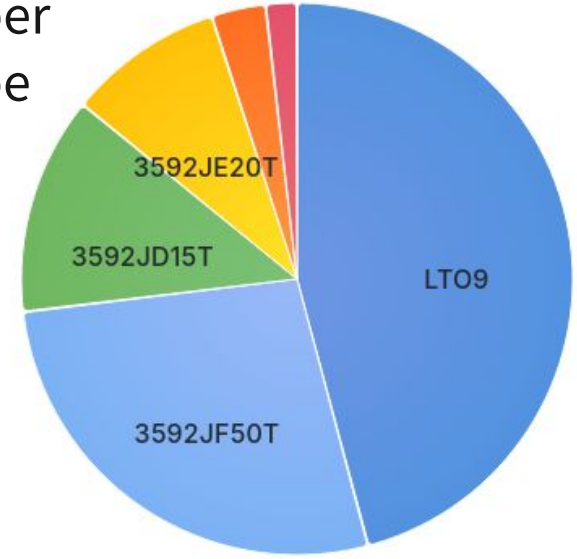
Data Management at CERN: toward HL-LHC



Tape infrastructure for CTA Service at CERN

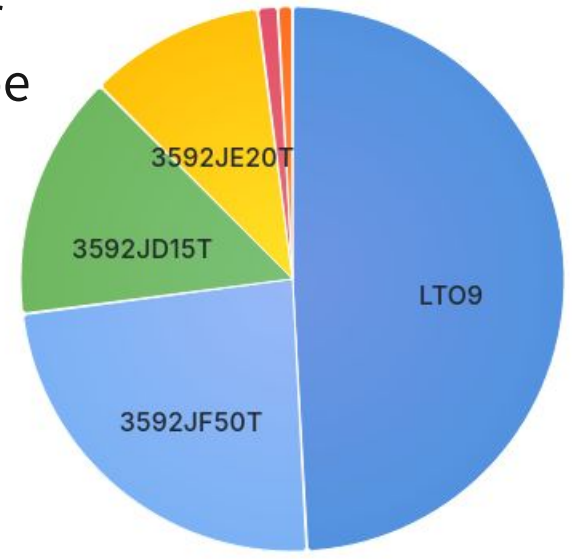
Provisioned capacity	1650 PB	
Libraries	4 x IBM TS4500	3 x Spectra Logic TFinity
Drives	46 x IBM TS1160 40 x IBM TS 1170	10 x LTO-8 88 x LTO-9

Capacity per MediaType



	Value
LTO9	758 PB
3592JF50T	450 PB
3592JD15T	210 PB
3592JE20T	150 PB
LTO7M	51.4 PB
LTO8	28.9 PB

Data per MediaType



	Value
LTO9	704 PB
3592JF50T	340 PB
3592JD15T	208 PB
3592JE20T	148 PB
LTO8	16.0 PB
LTO7M	11.3 PB

Tape Infrastructure Impact

- Had to buy truckloads of tape media
- Avalanche of data
  - Did not allow us to dedicate drives to data migration
- Two additional libraries
  - Constant library footprint was a dream…

150PB of LTO9 ->

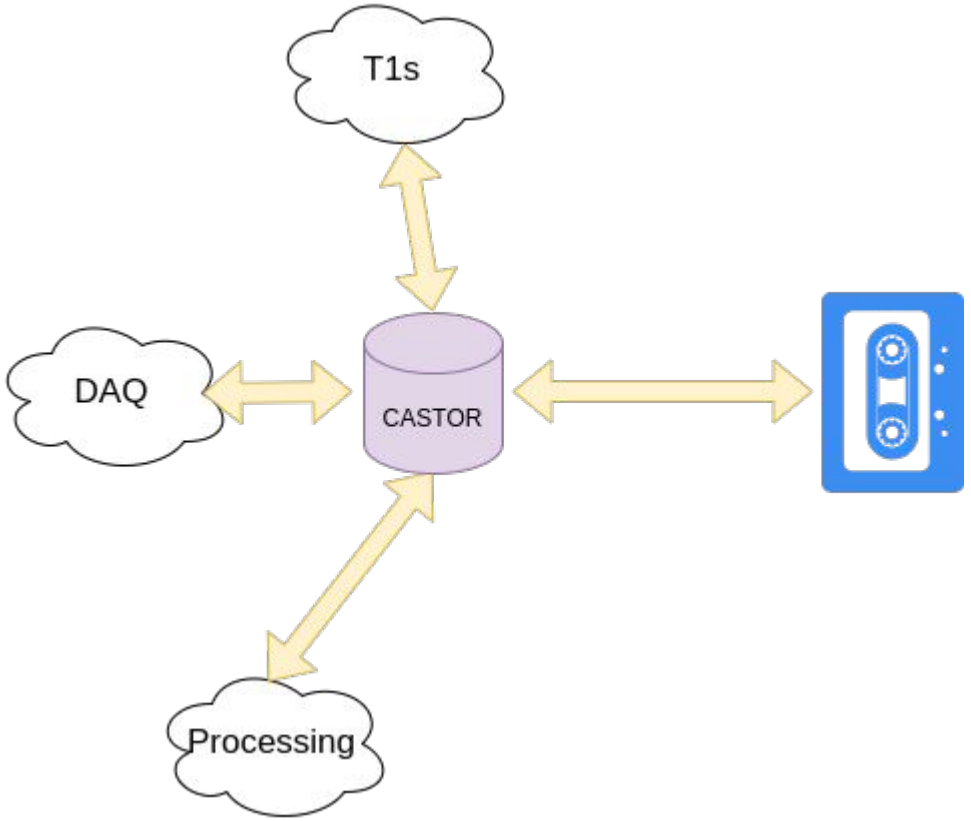
CERN Cold storage evolution

open source since day 1

LHC Run-1 (2009-2013): CASTOR (HSM)

CASTOR HSM

- **One CASTOR disk instance per experiment with its associated disk pools**
- **single namespace for all tapes and their file index**
  - **Manage all disk tape data movements**



LHC Run-2 (2015-2018): EOS + CASTOR (HSM)

EOS is experiment facing:

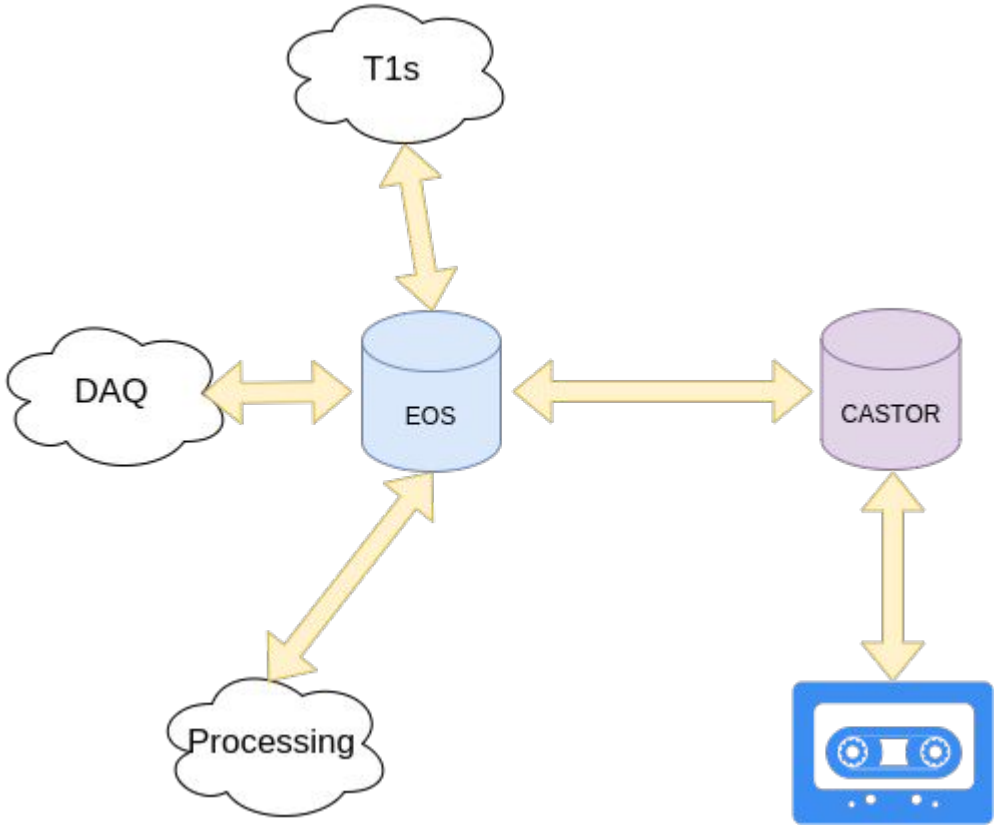
- **one Namespace per experiment**
  - **disk capacity: huge bandwidth as a by product**

CASTOR HSM

- **single namespace for all tapes and their file index**
- **requires enough bandwidth to feed tape drives**
  - **Consumes significant disk capacity as a by-product of required bandwidth**

As users had to use CASTOR disk capacity tape VS user disk activity was still getting in the way for efficient use of tapes.

Faster than disk write speed to tape requires in memory file buffering on the tape servers.



LHC Run-3 (2022-2026): EOS + CTA (tape buffer)

EOS is experiment facing:

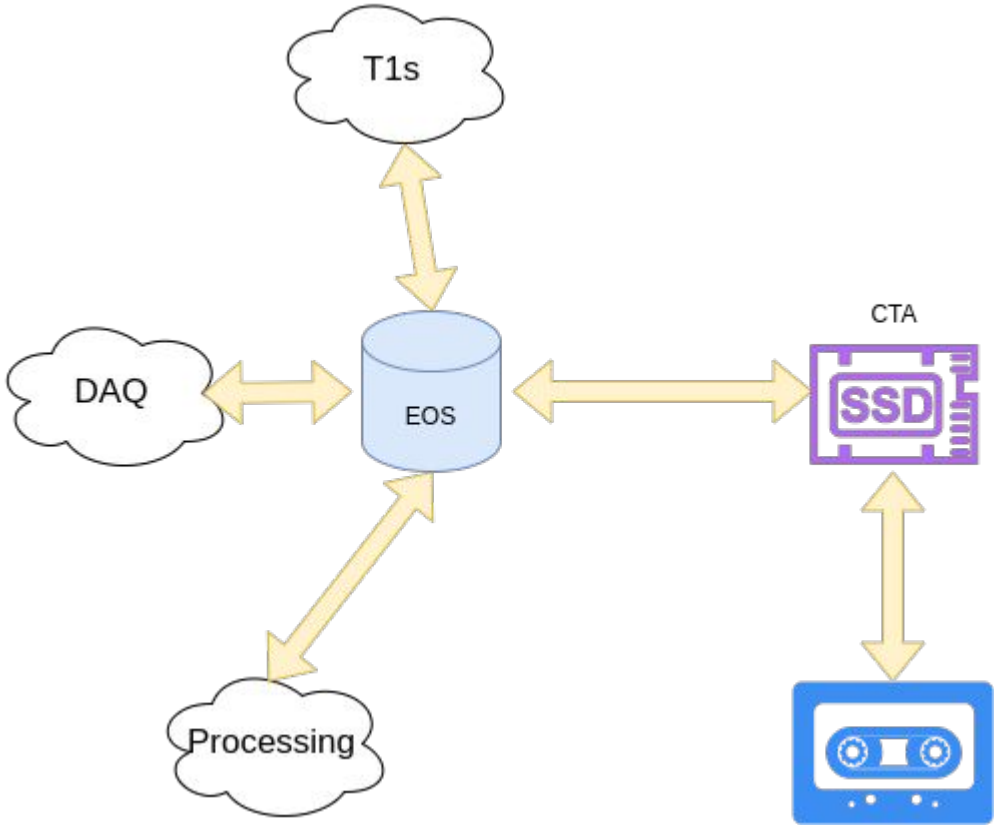
- **one Namespace per experiment**
  - **contains all disk capacity for an experiment: huge bandwidth as a by product**
  - **For example:**

CTA is a pure tape endpoint

- **Shared tape catalogue for all tapes and their file index**
- **Tape drives fed from SSD buffer outside of pledge**

Experiment larger files 10GB per file and drive increased throughput prevents any meaningful in memory buffering.

Transition toward Run4 requirements (o(100GB/s), o(100GB) perf file) required to move to this CTA architecture for Run3.



CASTOR to CTA dataflow migration

- **CTA is a pure tape system: DATA IS SAFE WHEN IT IS ON TAPE**
  - Compulsory for all DT workflows to use FTS CheckOnTape feature (or equivalent)
    - **supported by xrootd AND http**
- **Disk cache duty consolidated in the main EOS instance**
  - Separate disk and tape concerns
- Operating tape drives at full speed full time **efficiently requires a SSD based buffer: EOSCTA**
  - CTA cannot afford redundancy on SSDs
    - files corrupted/lost in the tape buffer are quickly marked as failed transfers by CheckOnTape
    - transfer must be retried from main EOS

What is CTA about?

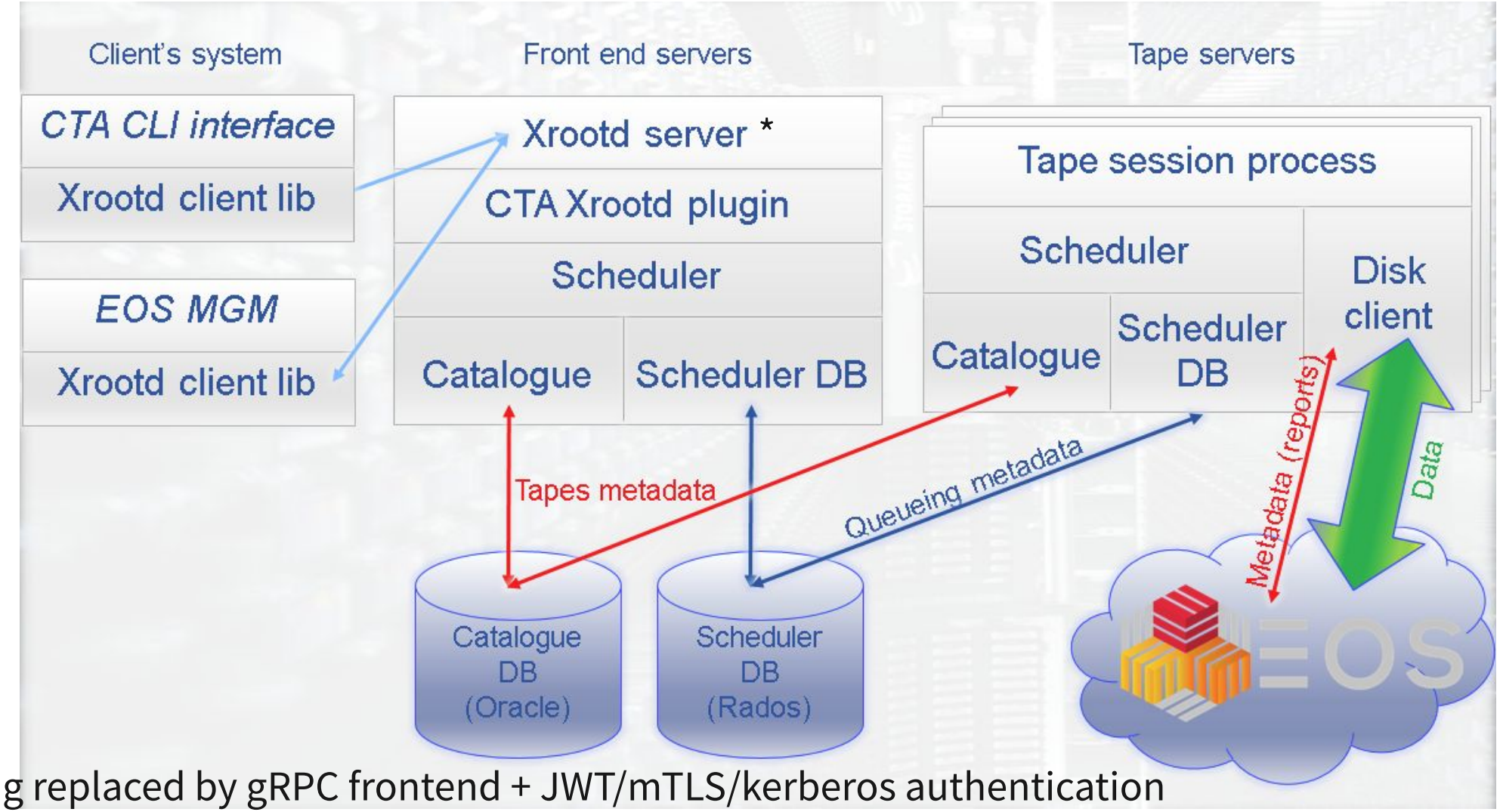
There is more than one flavor of CTA

CTA refers to the common tape backend

RESPONSIBLE TO QUEUE TAPE MOVEMENTS (ARCHIVE - aka write to tape, RETRIEVE - aka read from tape) AND SCHEDULE THESE MOVEMENTS

- **EOS + CTA - CERN, RAL**
  - EOS for the tape buffer in front of CTA
    - some instances with spinners for HSM reads
- **dCache + CTA - DESY, Fermilab**
  - dCache HSM for the tape cache in front of CTA
- **S3 + CTA (PoC)**
  - CERN future Backup Service

CTA architecture - *evolving*



* being replaced by gRPC frontend + JWT/mTLS/kerberos authentication

(EOS)CTA - CERN Tape Archive

Tape backend to EOS

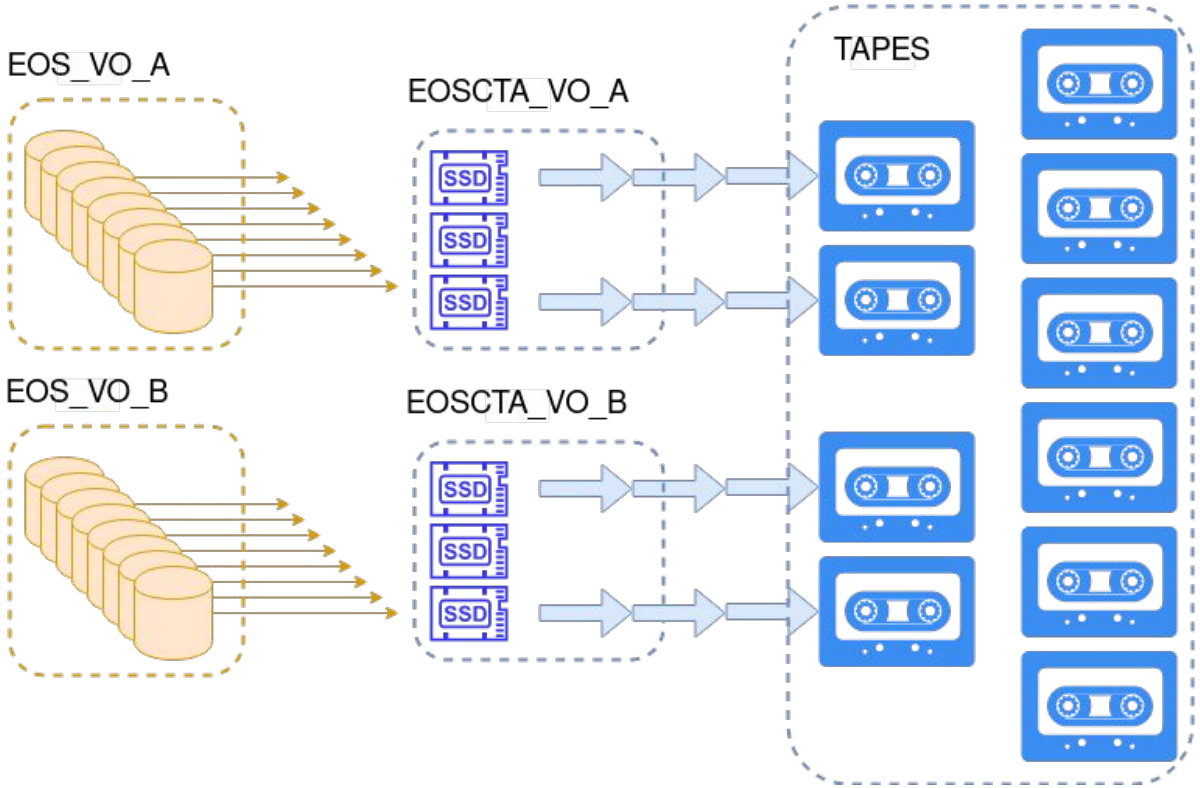
EOSCTA Run-3 tape buffer characteristics

EOSCTA tape buffer hardware:

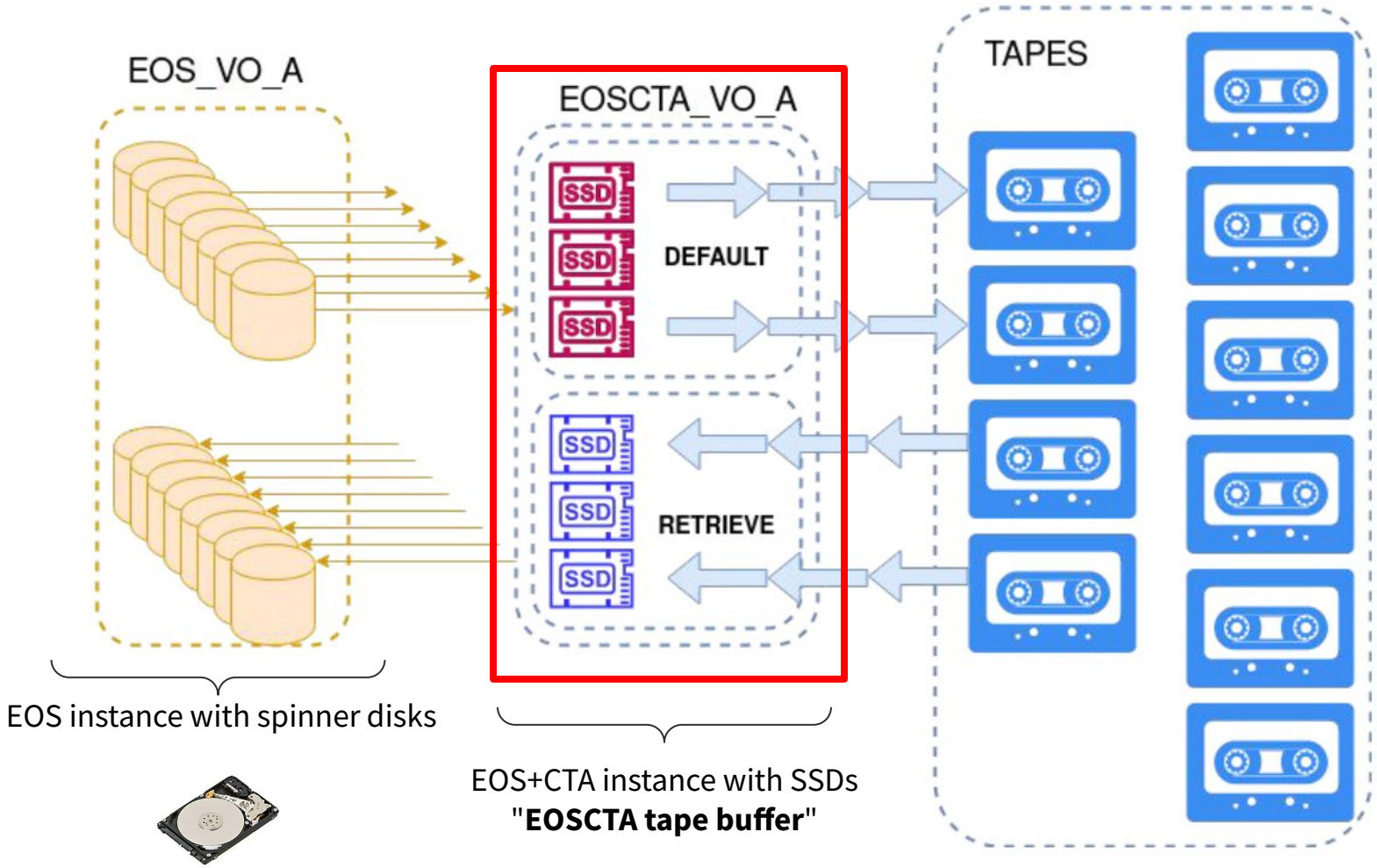
- 64 x hyper-converged servers
  - 16 x 2TB SSDs
  - 25Gb/s Ethernet
- 4:3 blocking factor connectivity to CERN CC router
  - 1.2Tb/s or 150GB/s of full duplex buffer bandwidth

EOSCTA tape buffer properties:

- Conservative setup *evolved*
  - tape buffer separated from tape infrastructure
  - up to 8 hours of buffer to tape at 10GB/s
- Move files to/from tape
- Not part of the pledge: **not available for physics jobs**
- Files are *evicted* as soon as they are safely archived on tape
  - or copied on “Big EOS” for retrieves
- Efficiency first
  - **Cannot afford redundancy**
- Early failure notification for retries



EOS + CTA architecture @ CERN



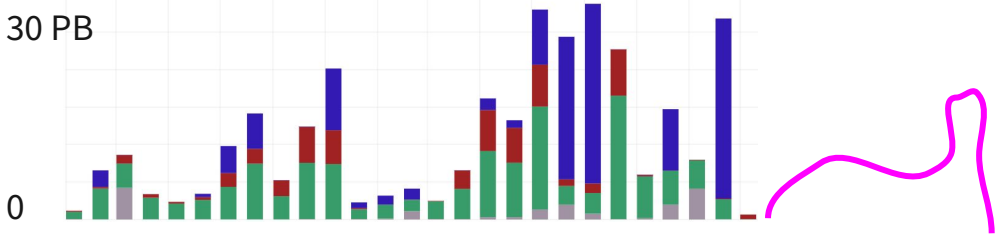
Archive/Staging bandwidth allocation

Standard LHC eoscta instance: 10GB/s archival SLA for CTA T0

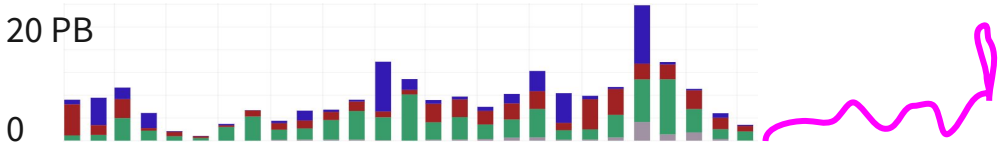
- **Archive boost needed during data taking, tape flushing, Heavy Ion run**
- **Staging boost during Year End Technical Stop (YETS) Heavy Ion data duplication to T1s/T2s**
- **Change eoscta bandwidth allocation accordingly**

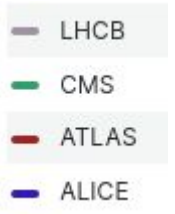
Run3 LHC planning

Archived volume per month



Staged volume per month



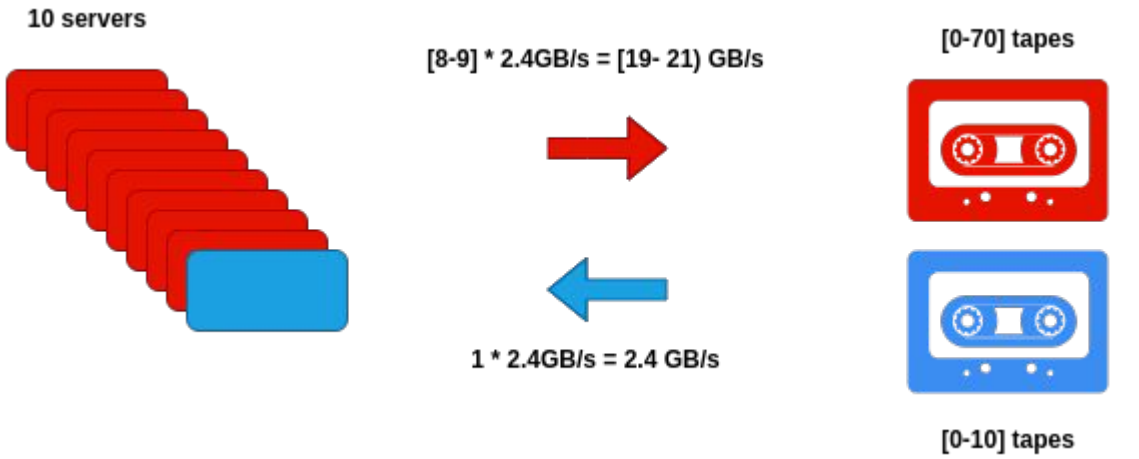


AI/quantum computing generated interpolation

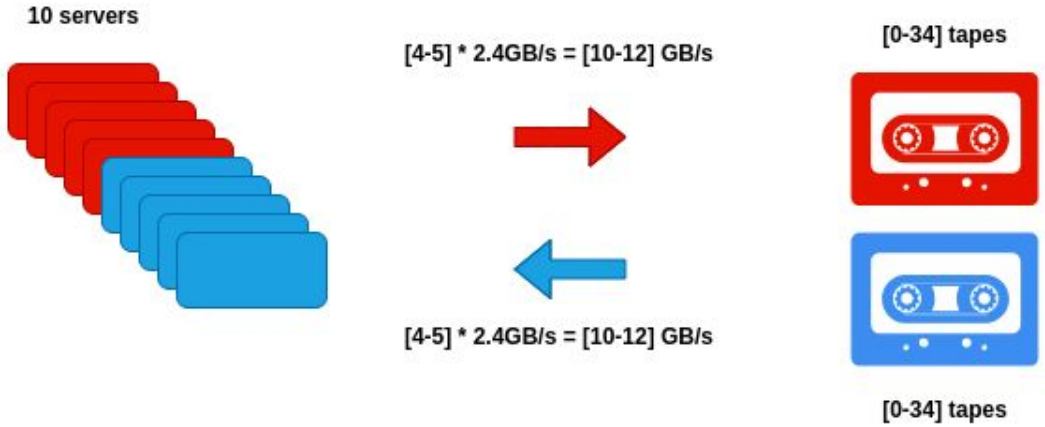
Archive/Retrieve bandwidth allocation

Standard LHC eoscta instance: 10GB/s archival SLA for CTA T0

- **10 SSD servers**
- **Archive boost during data taking, tape flushing, Heavy Ion run**
- **Staging boost during Year End Technical Stop (YETS) HI data duplication to T1s/T2s**
- **Configure CTA ALICE VO writemaxdrives, readmaxdrives accordingly**
- **Measure bandwidth to/from tape buffer AND INDIVIDUAL TAPE DRIVE EFFICIENCY**



Data taking configuration



YETS configuration

Lessons Learned: Buffer Infrastructure

- Dedicated SSD buffer in front of tapes is the way to go
- Split instances to separate use cases
  - Separate buffers per LHC instance
  - Inside buffer separate SSD servers for read/write
- Don’t underestimate buffer infrastructure operations complexity

Dimension tape buffer

From synthetic benchmarks to production

Synthetic benchmark of one machine

Make sure there is no internal bottleneck: SSDs, HBA, PCI,...

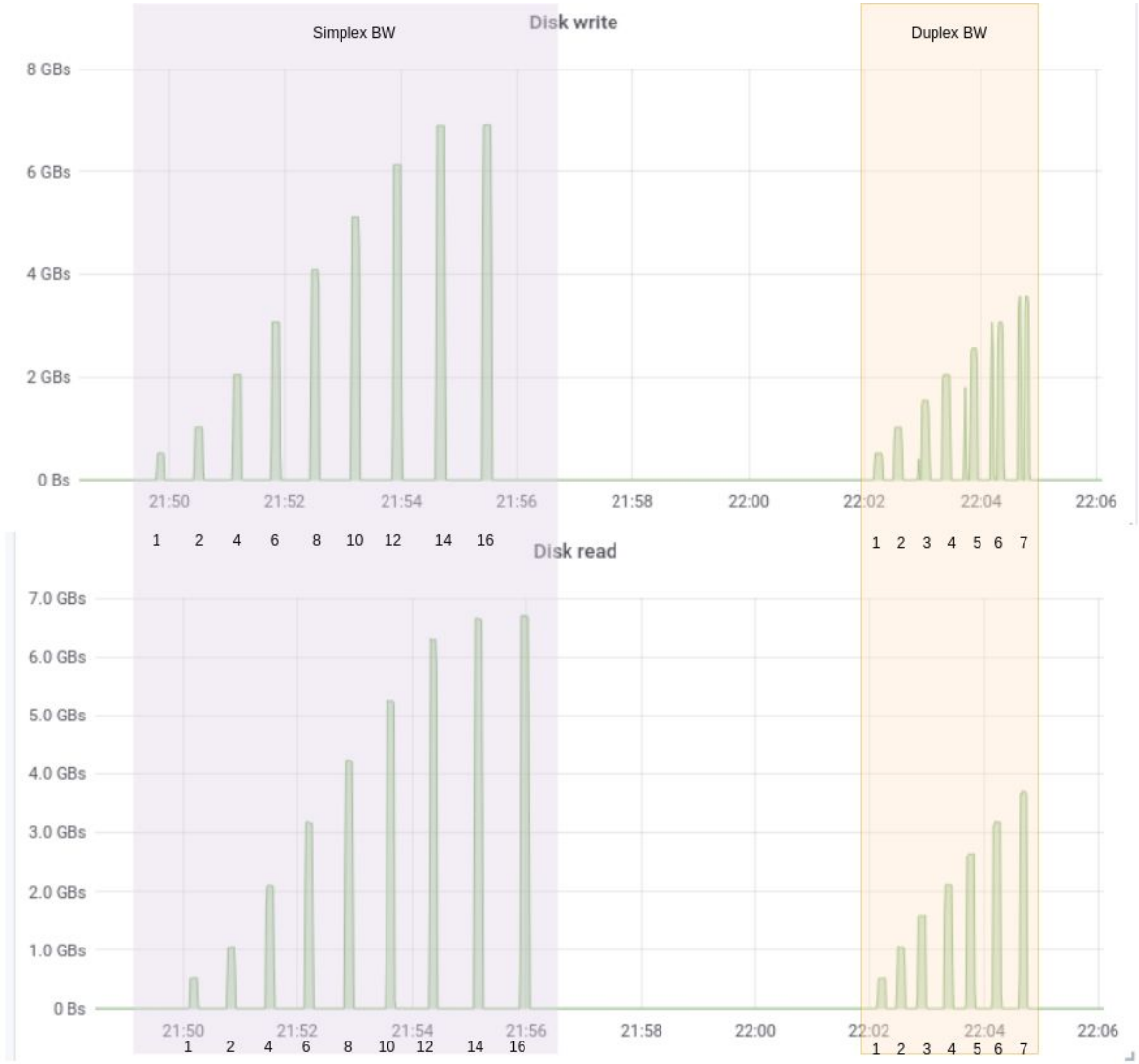
Buffer servers bought in 2018

- **Received 2018-11**
- **…aka before COVID**

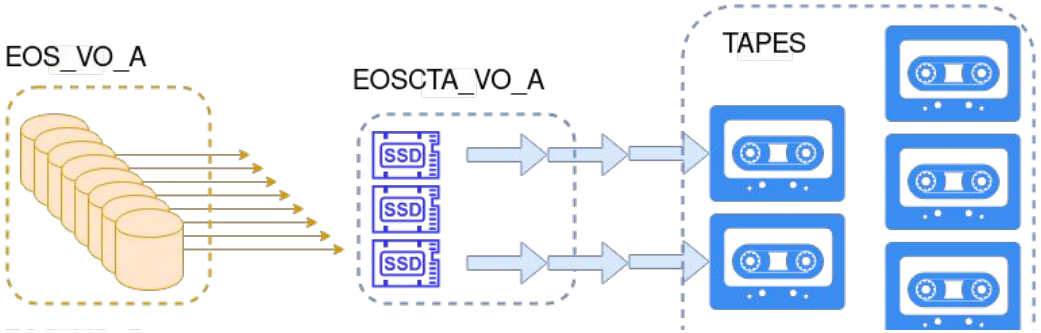
Tests using dd

- **streams of large files**
- **writes: sync**
- **reads: clear caches**

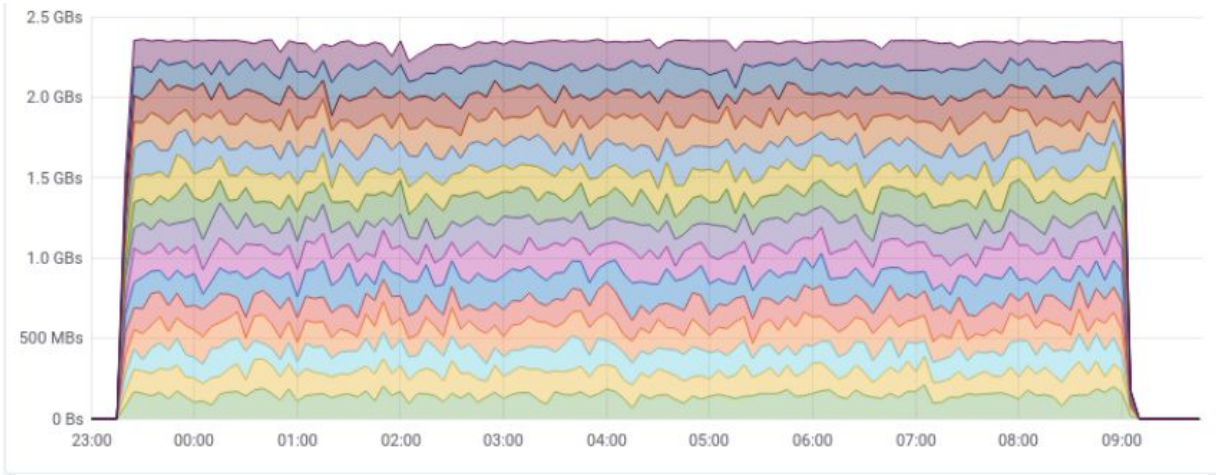
HBA offers >7GB/s of internal throughput >> 5.5GB/s network BW on one 25Gb/s port



Saturate one machine

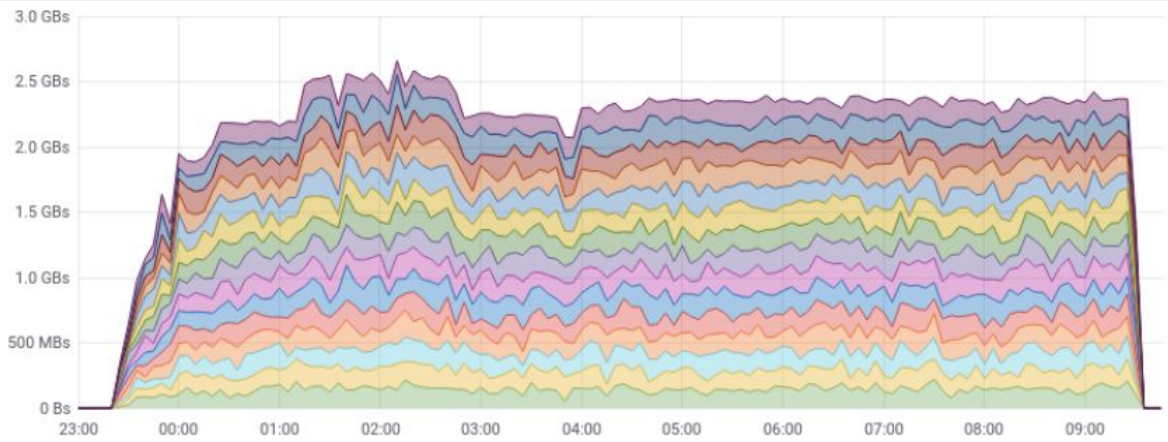


SSD writes



Progressively exercise the full stack on critical use cases

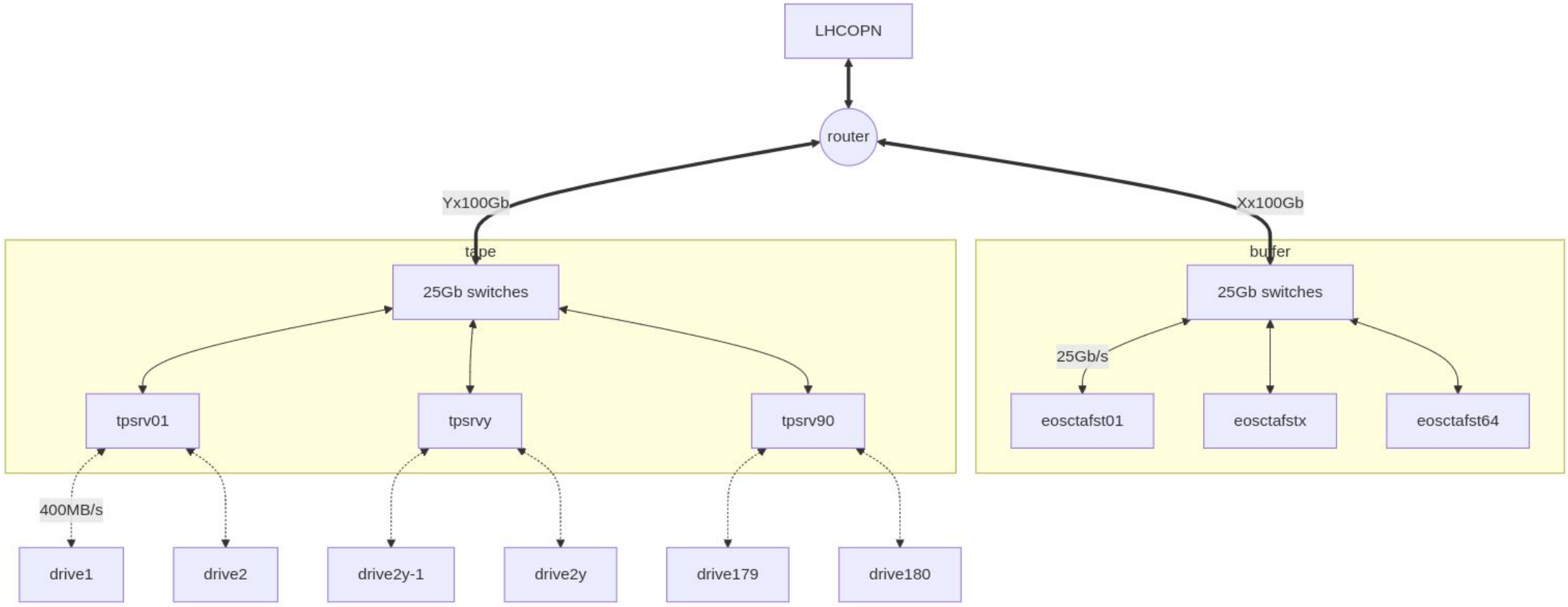
SSD reads



- **transfer files to 1 machine**
  - measure per SSD write speed for incoming transfers
  - measure per SSD read speed to transfers to tapes
  - measure per tape speed
- **MOVE TO NEXT SCALE TEST**
  - KPI is per tape write speed

tape writes

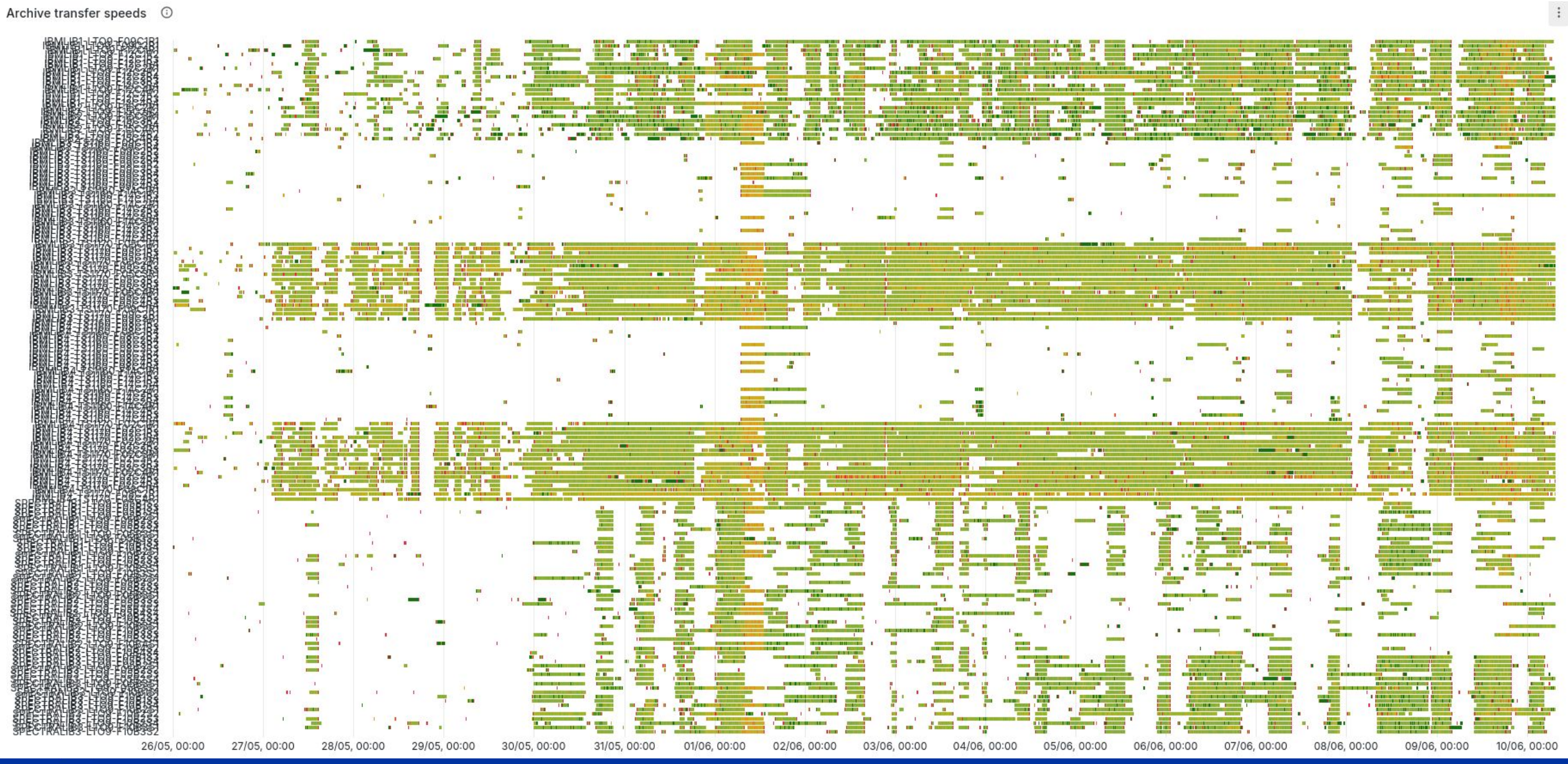
Scale to nominal performance



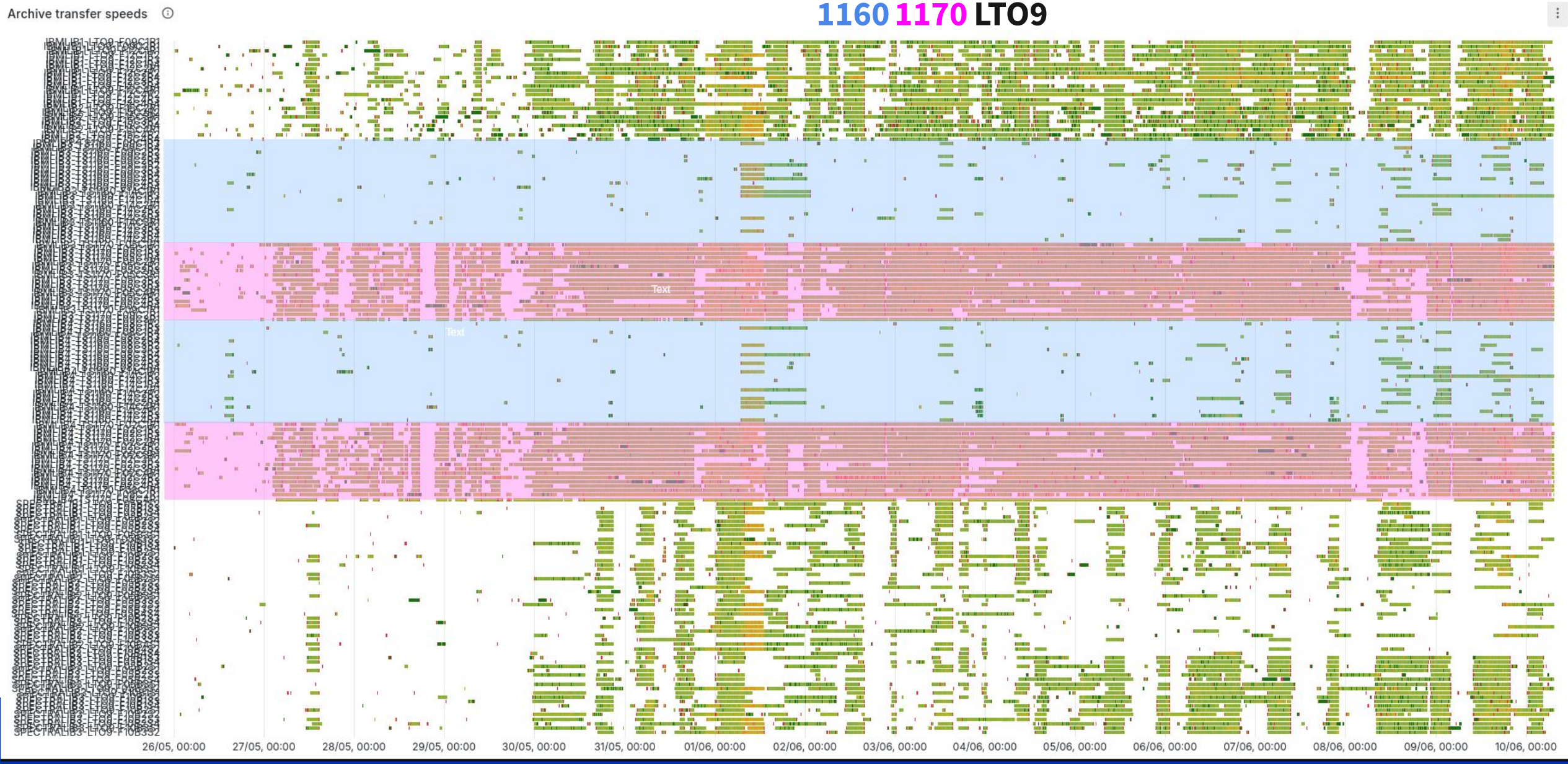
Data Taking Performance

Writing data efficiently is the critical first step

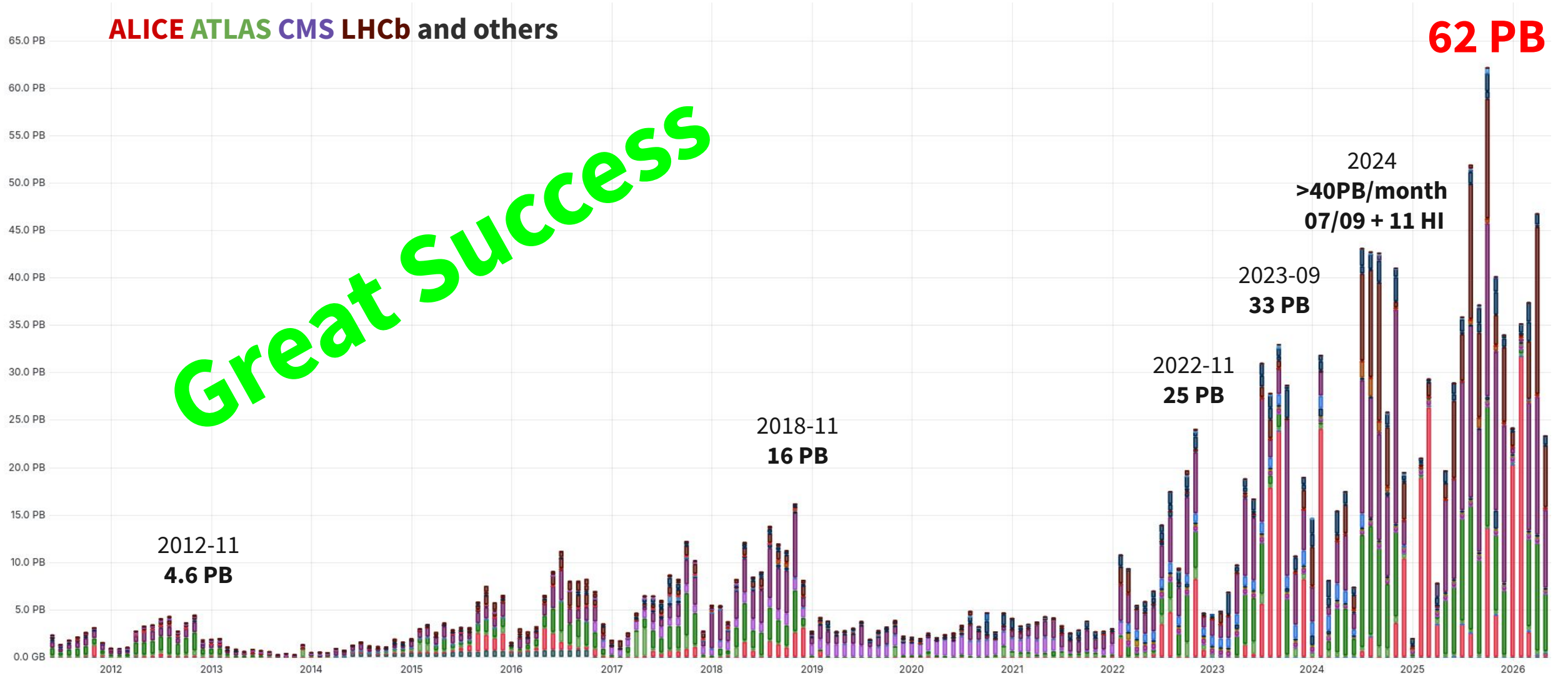
Write efficiency 2026-06



Write efficiency 2026-06

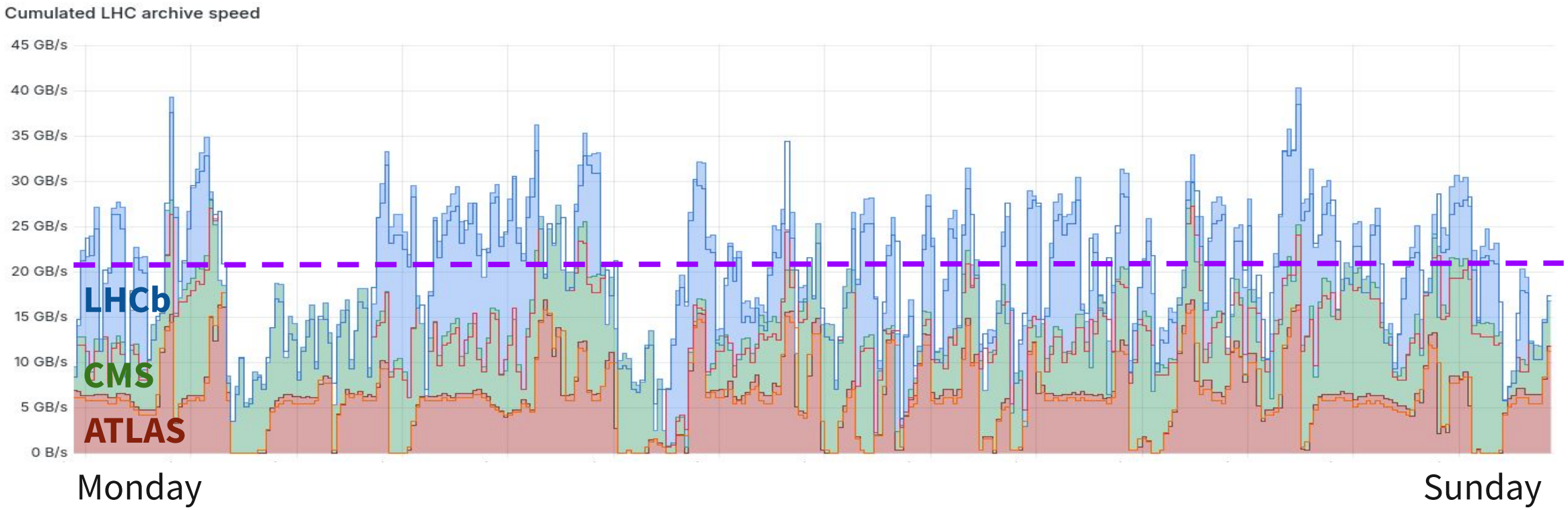


Monthly written data over the past 15 years

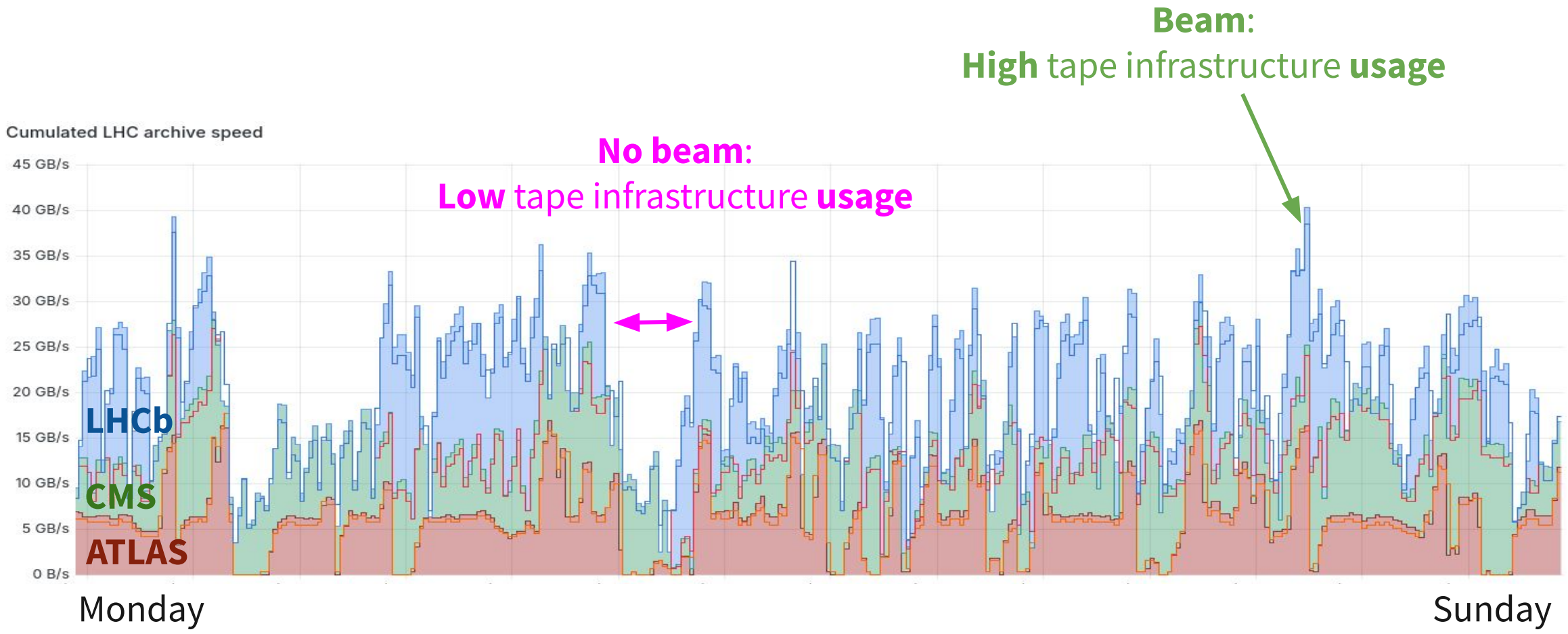


Synchronous data taking model *can't* go faster

60 PB per month = 22GB/s average



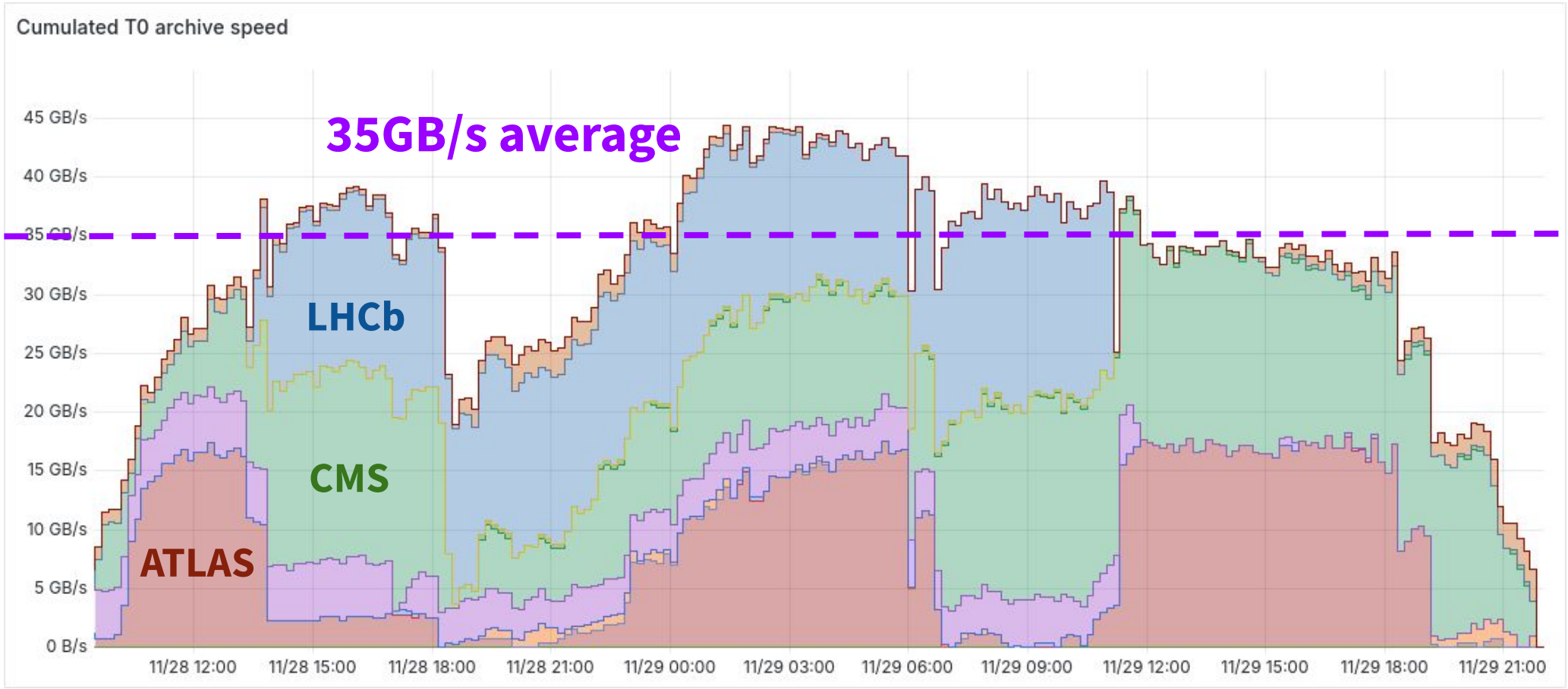
Synchronous data taking model *can't* go faster



This p-p setup was not enough for 2025 Heavy Ion Run

Asynchronous data taking model *can* go faster

2025 Heavy Ion Run

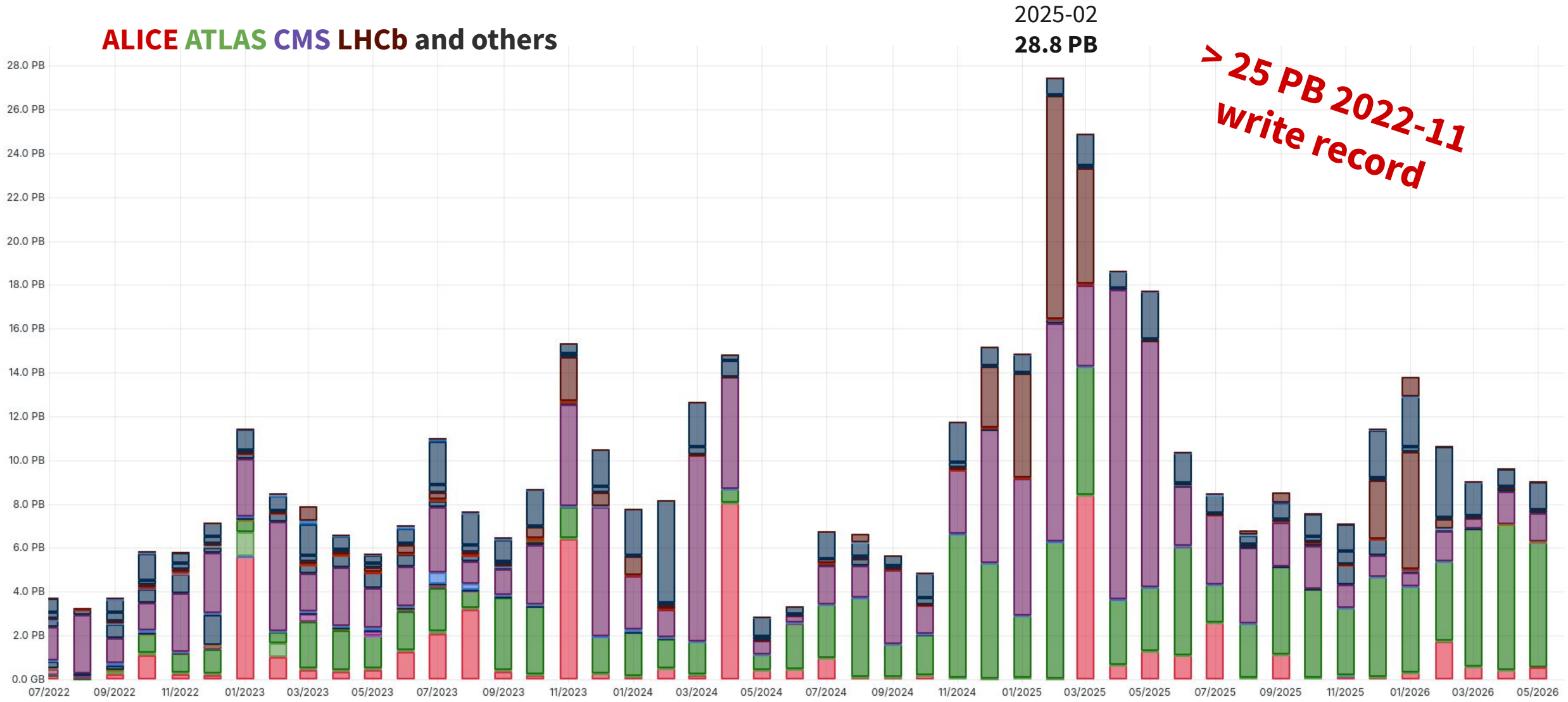


We worked with experiments to tune the data flow

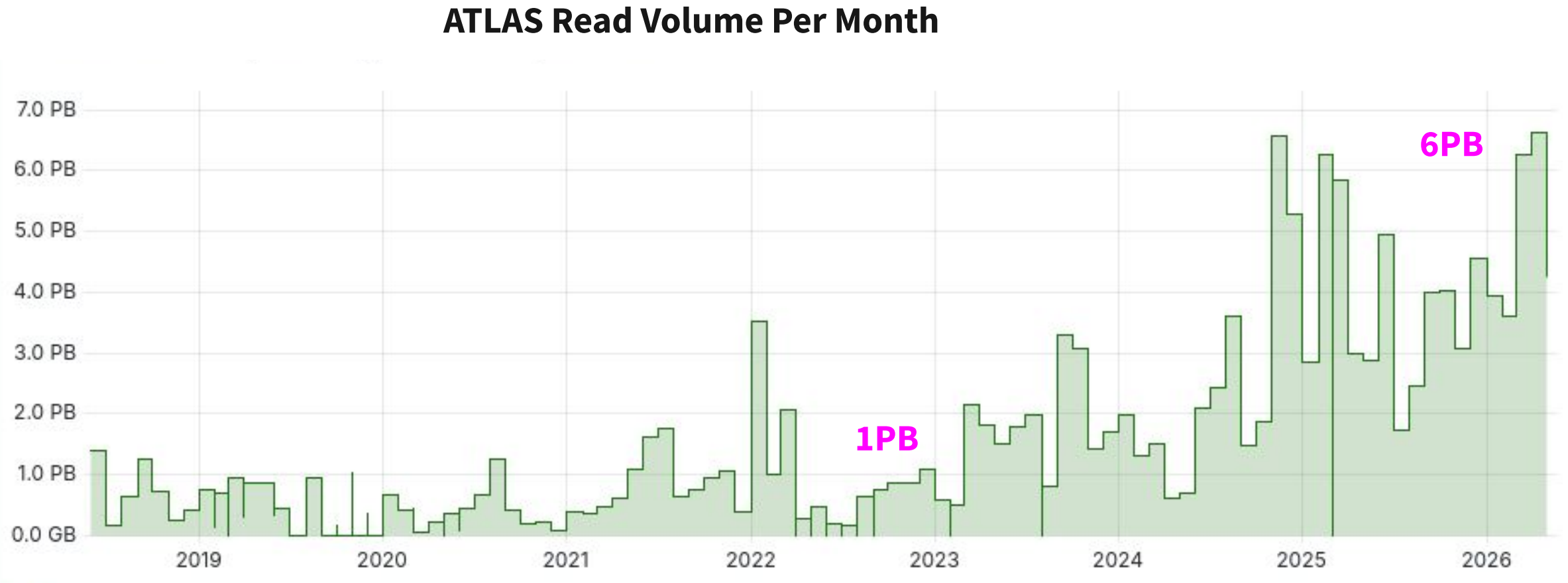
CTA Read Performance

Writing is critical, but is pointless without read performance

Monthly read data during Run-3



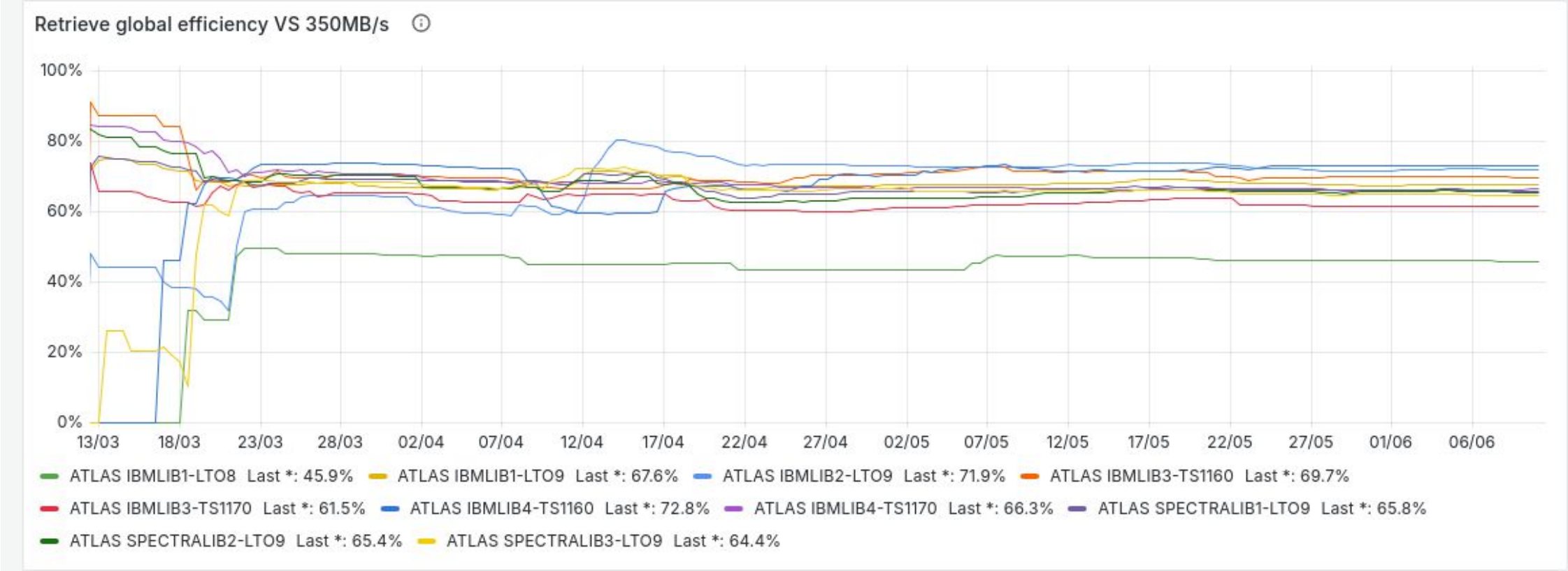
Tape Reads: The new trend



Reading from tape increased by a factor 6 and is continuous now!

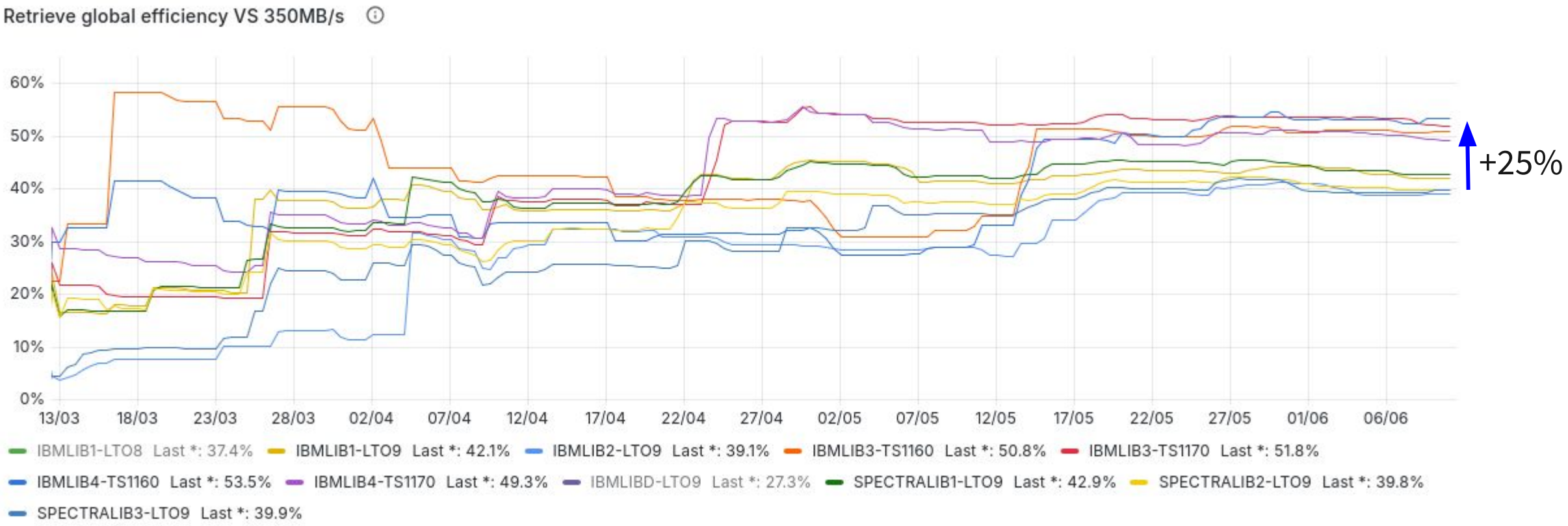
Tape Read: oRAO VS hdRAO

From this morning discussions: what we see in production for a large experiment over 3 months.



Tape Read: oRAO VS hdRAO

From this morning discussions: what we see in production globally.



Lessons Learned: Writes are efficient, Reads must be improved

- **Tape Write Efficiency is excellent and deterministic**
  - CTA reliably saturates the drive speed on writes

- **Tape Read Efficiency is not deterministic**
  - Heavily relies on initial data placement set at write time
  - *Tape family* data placement results in roughly half the drive speed on reads

More reads means we need to put more focus on improving read efficiency

If we want to go further we have to do it together

- Tape is part of a bigger system

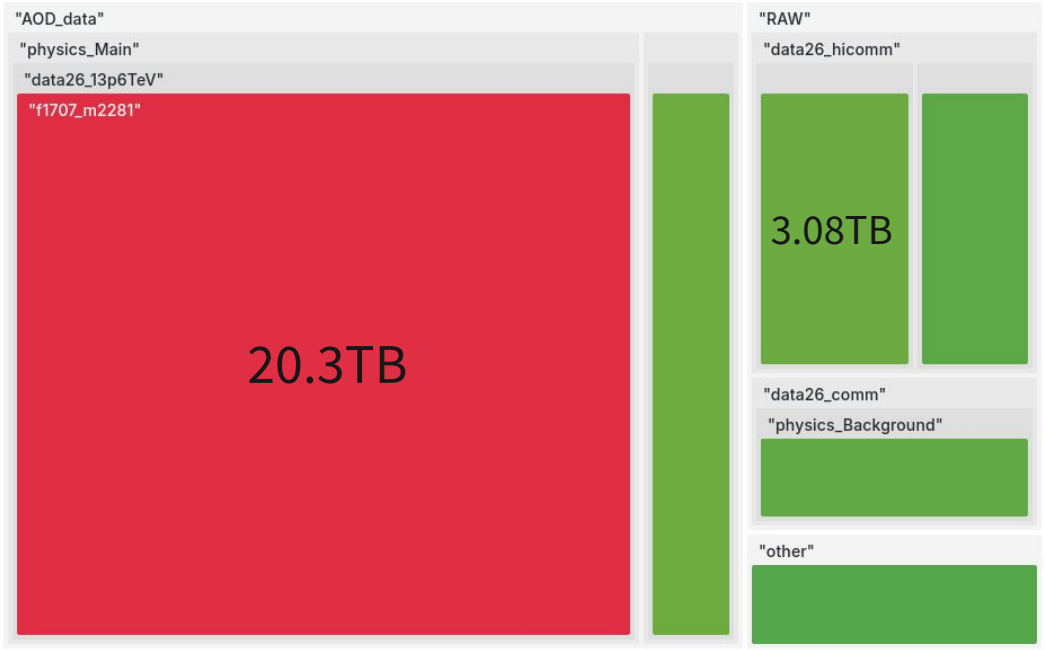
- **The tape systems should allow users to express their intentions**
  - Read pattern intention
  - Priority intention
  - Deadline intention
  - Give feedback?

Disclaimer: that does not mean users should have direct control over tape resources!

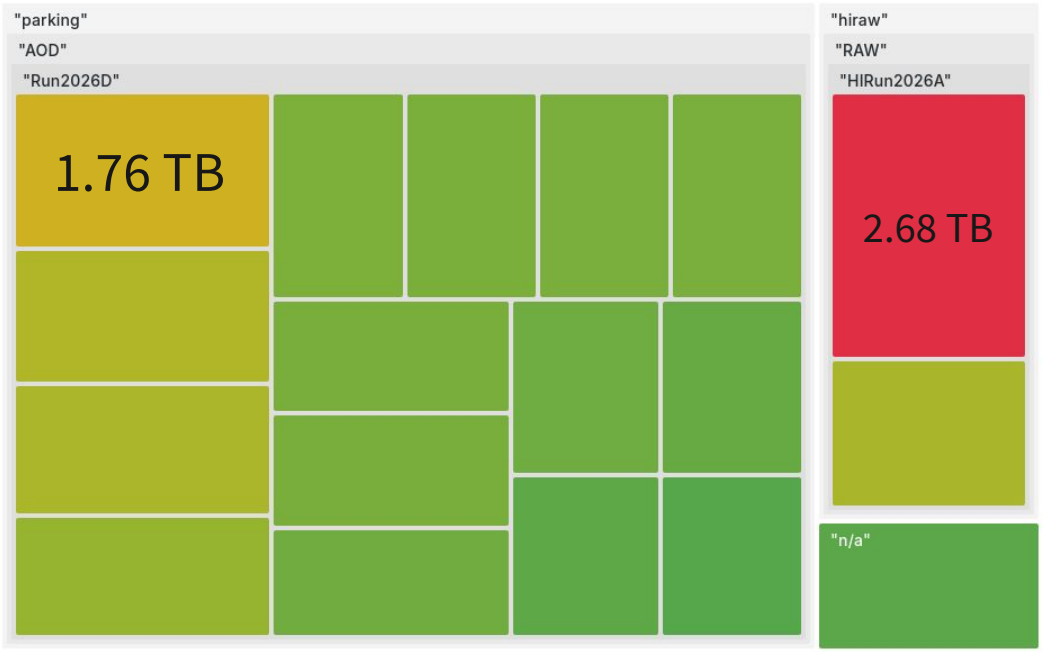
Read Pattern Intention: Archive Metadata

- Put a color on a file to group them
- Tape software understands and will take action at the data placement level

ATLAS



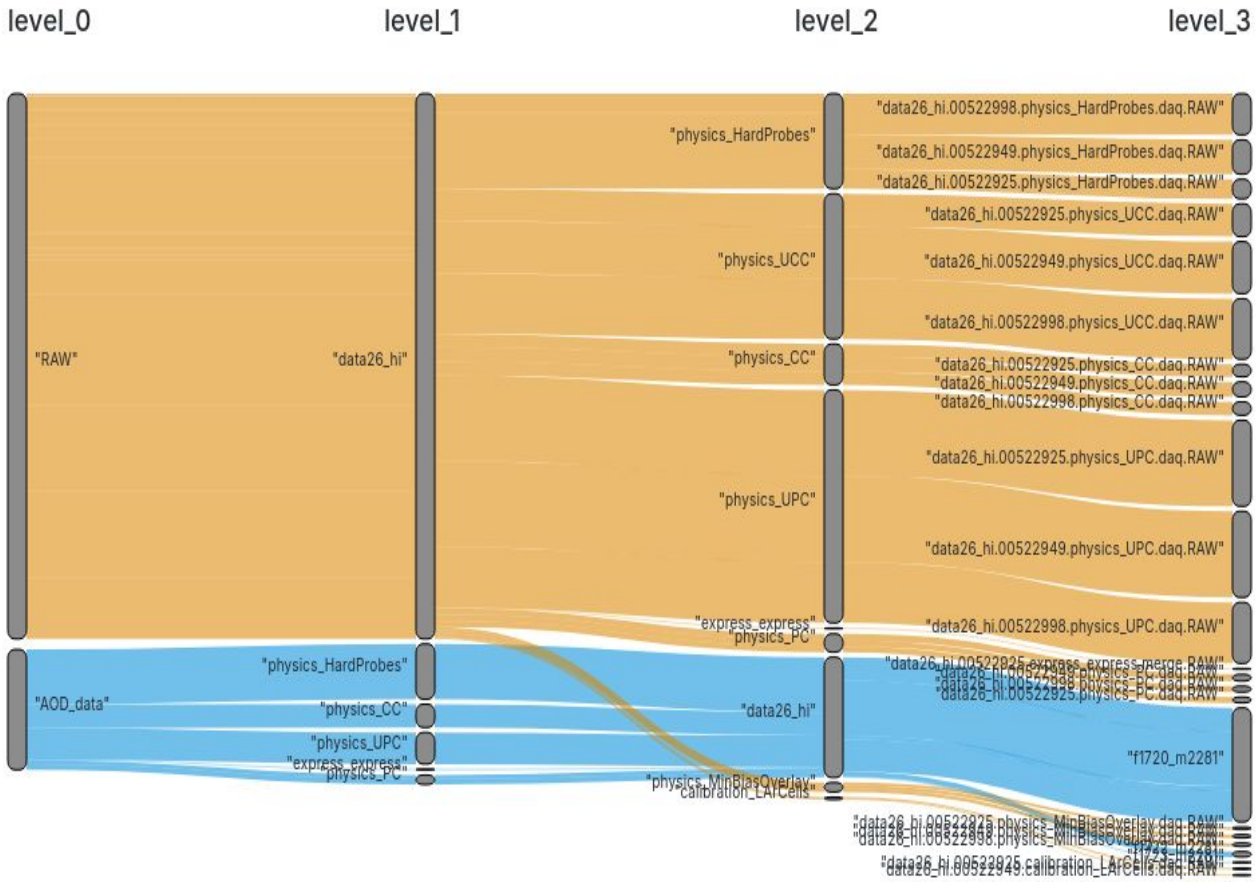
CMS



See CHEP 2024 paper: [Archive Metadata for Efficient Data Collocation on Tape]

Read Pattern Intention: Archive Metadata

- Ongoing modelling and simulation work
  - Tape model developed
    - tape infrastructure emulation
  - Use simulation to evaluate next generation CTA tape scheduler
    - Replay Run-3 write traces
    - Generate tape topologies
    - Evaluate recall efficiency gains VS additional cost

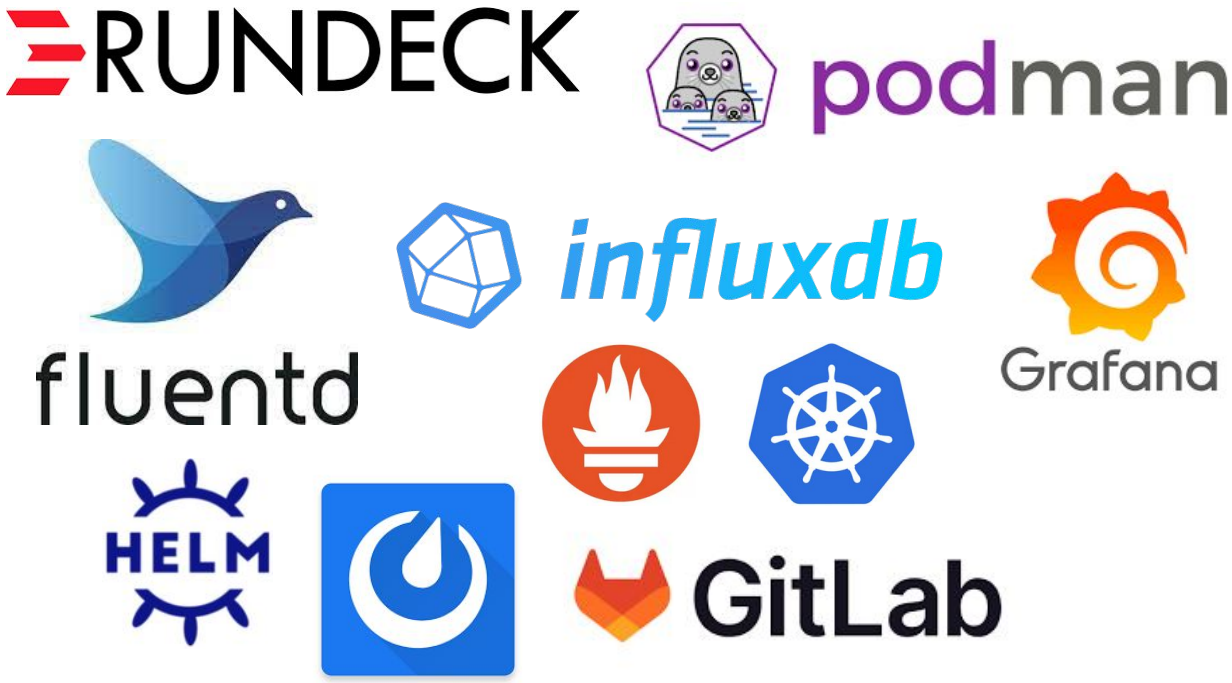


CTA Service operations

CTA comes with consistent set of tools for a common tape operations framework

CERN continuous improvement of EOSCTA operations

- **Operations monitoring**
  - real time, short lived, wipe and replace
  - sends alerts in mattermost
- **Operations issues in gitlab**
  - tracking incidents, specific activities, postmortem
  - follows up, dev_ticket needed,...
  - Reviewed once a week at CTA operations meeting
    - minutes, rota calendar in gitlab wiki
- **Operations procedures**
  - automated workflows in rundeck scheduled jobs or containers
  - CTA catalogue upgrade container
  - Weekly EOSCTA namespace dump per vo
    - json list of **healthy files on tape**/**files on BROKEN tapes**
- **Evolving toward Kubernetes**



What is tape REPACK?

Read from:

- **problematic tapes**
- **partially written tapes (user deletion, expired backups…)**
- **large repack campaigns to remove old media (LTO7M, JD…)**
  - next large repack for JD+JE will be around 400PB (1.5 y at 2x10GB/s)

Write to:

- **current high capacity media (JF, LTO9) liberating library slots in the process**

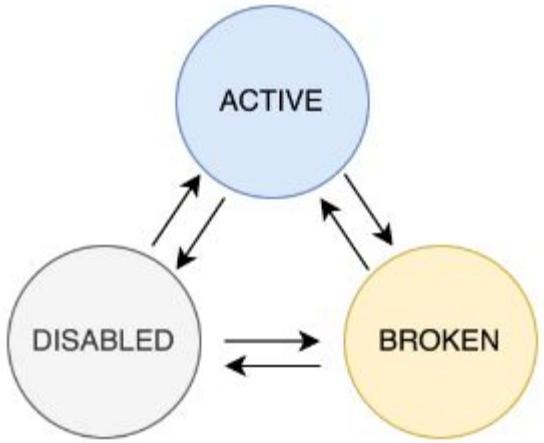
Strategies for repack at CERN:

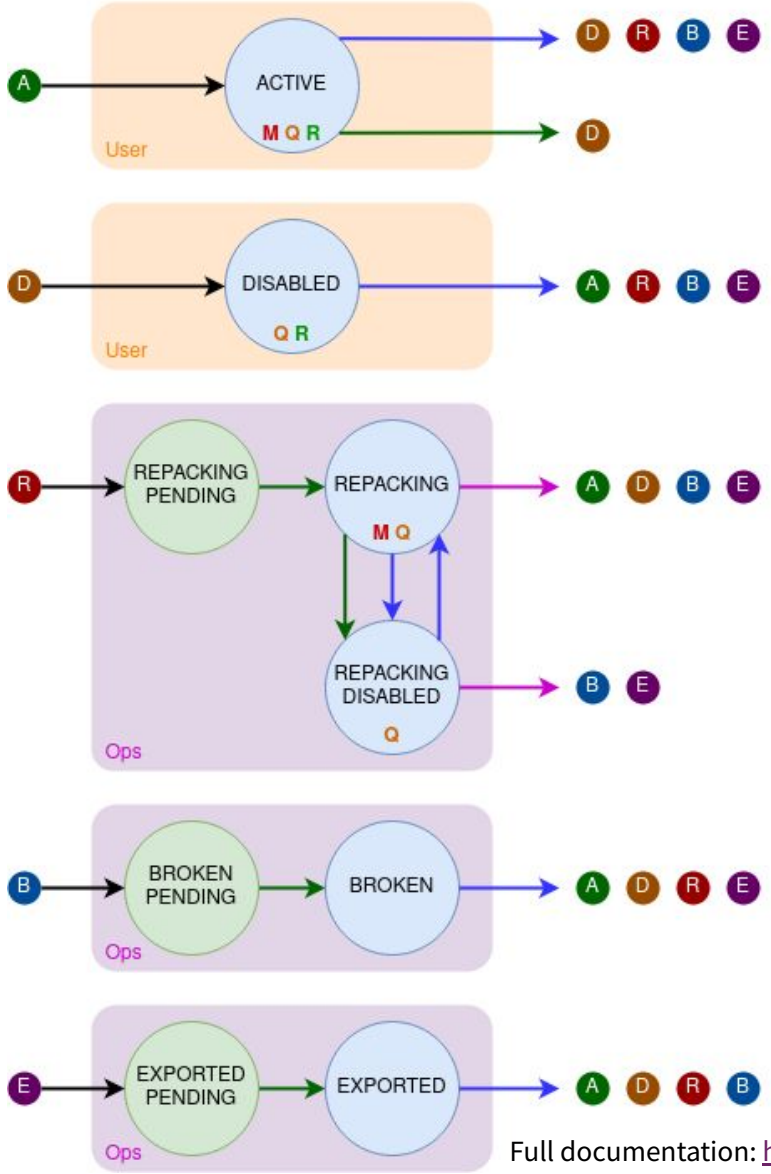
- **tape to tape - unpredictable and inefficient**
- **tape to disk cache to tape - requires too much capacity to reach repack nominal throughput**
- **tape to SSD to tape**

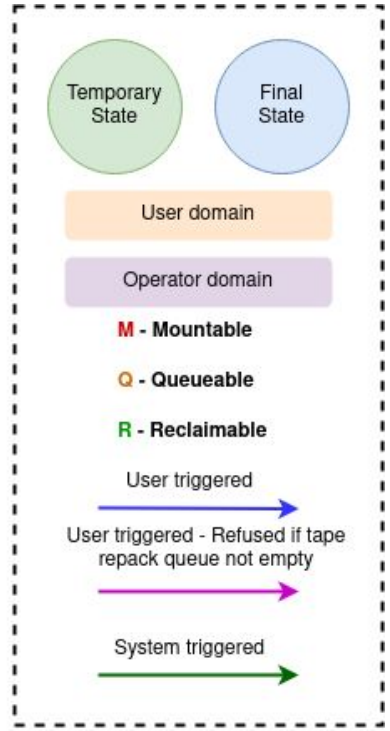
LARGE TAPE REPACK CAMPAIGNS ALLOW TO TEST THE NEXT TAPE SERVICE ARCHITECTURE

Automating repack

New tape lifecycle (CTA >= 4.8.0)



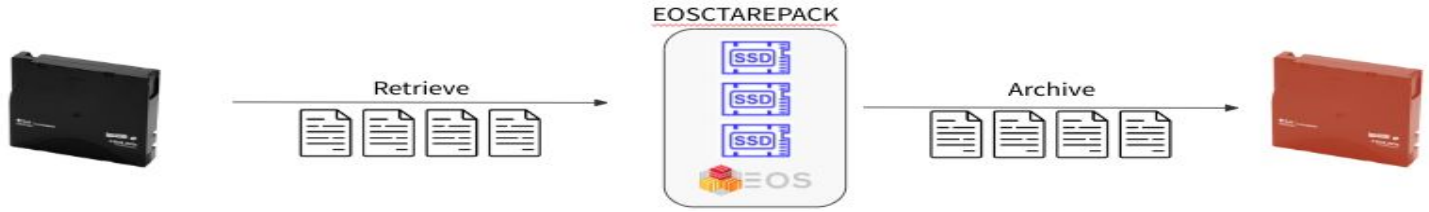


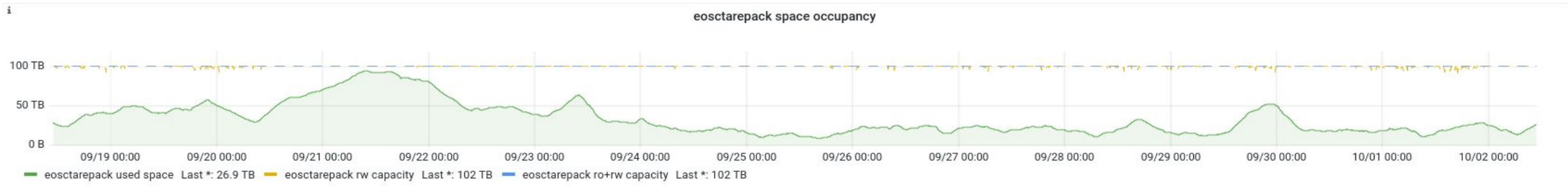


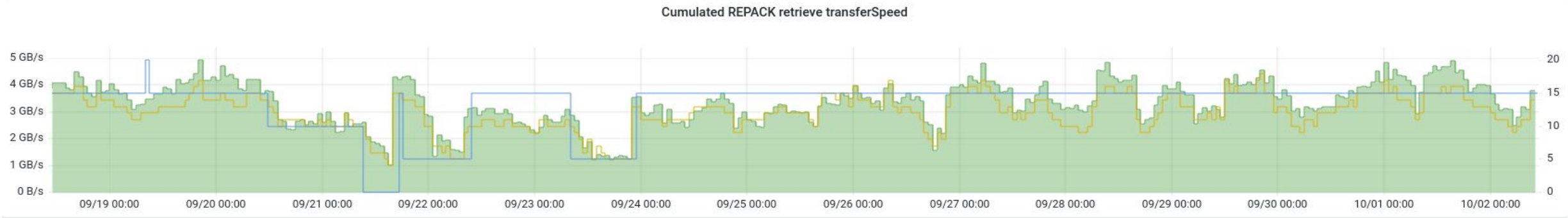
- No clear state for Repack
- Mix of user-requests and repack-requests
- User requests may be queued indefinitely
- No state for exported tapes

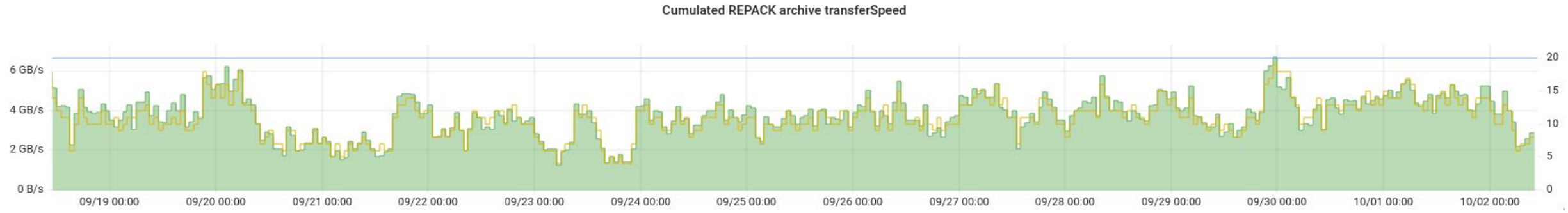
Full documentation: https://eoscta.docs.cern.ch/tape/tape_lifecycle/

Repack in production

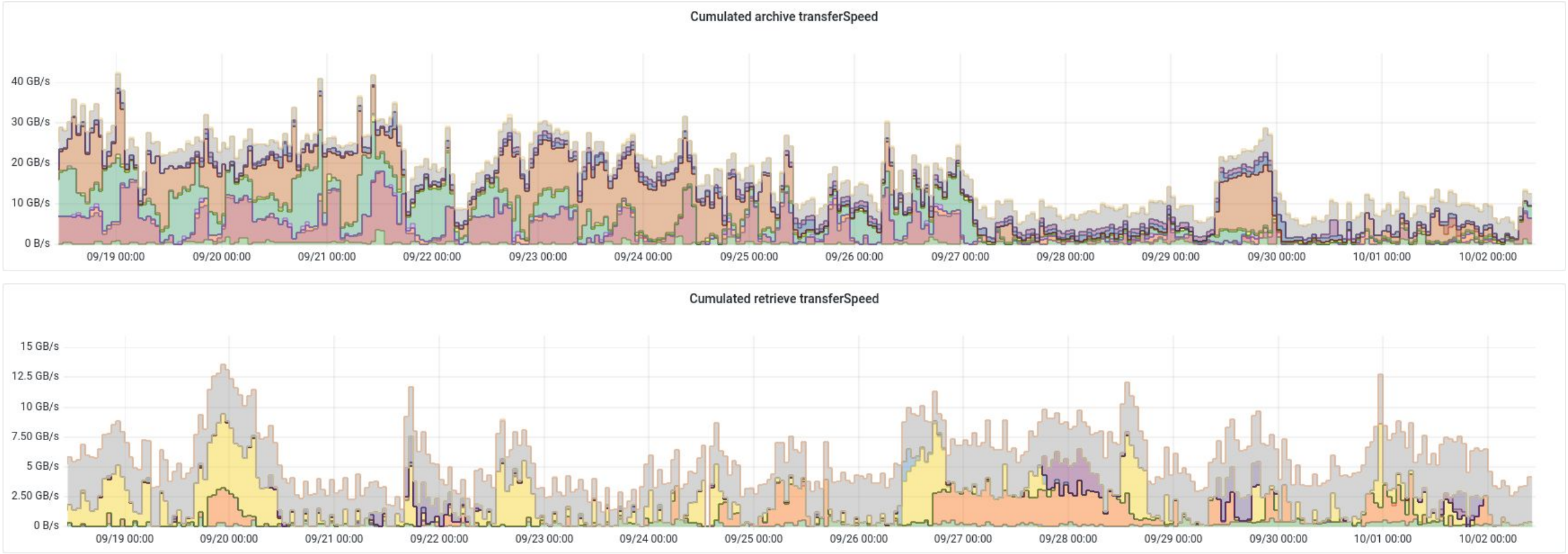








Repack along with production



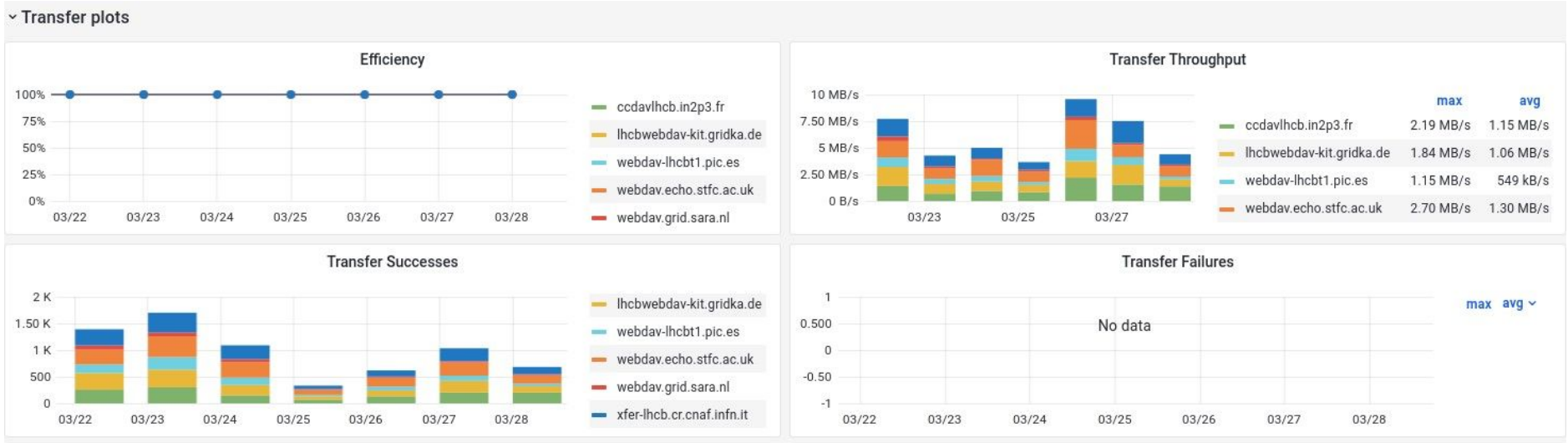
>35PB and 2000 tapes repacked during 2024-07/10 while writing >130PB to tape

Driving standards for tape in WLCG

Common tape backend to solve common tape issues with collective input

HTTP protocol consolidation on tape

- **Remove few sub-optimal data flows**
  - xrootd TPC with delegation transfers
  - 1 gridftp use case in CTA T0 (low priority)
- **Experiments moving to HTTP protocol on WLCG**
  - HTTP TAPE REST API version 1.0 specifications implemented in EOSCTA software stack in CTA 5/4.8.7-1
    - **Critical for check on tape** (implemented with fileinfo method in GFAL2)
  - Deployed at T0 on HTTP oriented EOSCTA LHC instances earlier this month
    - tested with RUCIO ATLAS team in preproduction
    - **archive transfers to eosctalhcb ongoing in production for LHCb using checkOnTape**



CTA outside High Energy Physics?

Common tape backend to solve common tape issues with WIDER collective input

dCache migrations to CTA

- **CTA provides metadata migration tools**
  - if it can read the source tape format
- **DESY migrated from OSM to CTA tape backend**
  - Configured CTA backend for all new writes
    - OSM was still there for tape reads
  - OSM metadata migration 1 pool at a time
    - CTA can read OSM tapes
- **Fermilab migrated from enstore to CTA tape backend**
  - Stopped dCache tape for 1 week
  - enstore metadata migration performed during that week
    - CTA can read enstore tapes
- **If it cannot read the source tape format**
  - Sites can implement read methods for their tape format
  - Write all new data to CTA and migrate old tape backend data through dCache
    - For example during a large media migration

CTA Beyond core physics?

At CERN, custom usages for various Backups (50PB):

- **CERN AFS, HDFS, filers backups**
- **EOS namespace backups**
- **List and volume is growing**

Evolve to offer a more standard backup solution?

Offer additional protocols in front of CTA

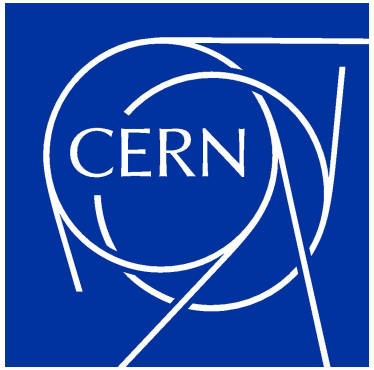
- **S3 glacier**

HPC community sees a growing tape usage

- **Other communities? PLEASE COME AND TALK WITH ME!**

Conclusion

- **CTA delivers nominal archival performance for Run-3 with significant write efficiency improvements**
  - Run-4 write performance rates already demonstrated during Run-3
- **NEXT STEPs**
  - **clearly oriented toward monitoring and improving data placement to improve tape read efficiency**
    - **Archive Metadata project**
  - **Next MASS repack campaign during LS3 will be interesting (>400PB)**
    - No media reformatting nor backward compatibility adds on the challenge
- **Tape and protocol consolidation ongoing on WLCG**
  - Opportunity to consolidate tape dataflows and build a stronger tape community based expertise should not be missed
- **More use cases coming to CTA = more opportunities to work together**
  - On-premise open source data storage for all!



home.cern

Expanding CTA community