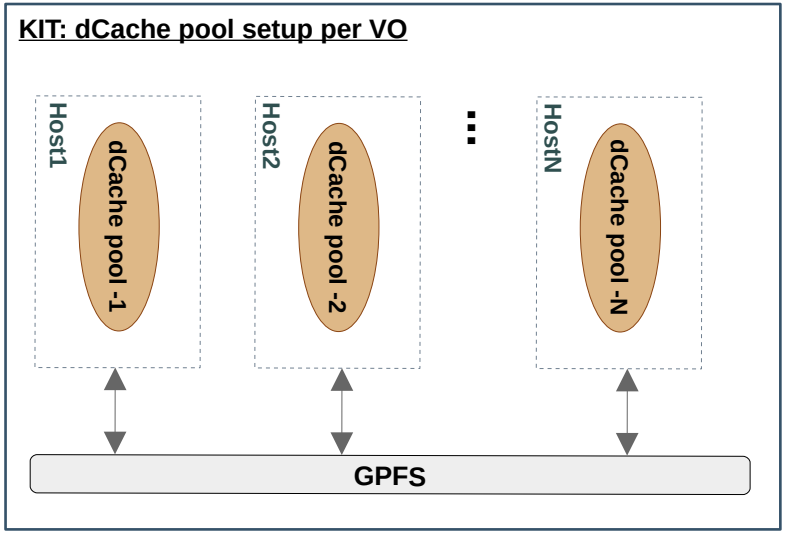
Karlsruhe Institute of Technology

Write/Read data to/from HPSS at GridKa

Haykuhi Musheghyan & Gridka Team

dCache file servers

- Separate dCache instances per VO
- dCache pools are directories in a shared GPFS
- Stage/Write tape pools
  - 1 primary and 1 fallback pools for ATLAS
  - 1 pool for Belle2
  - 2 pools for CMS/LHCb (same priority pools)



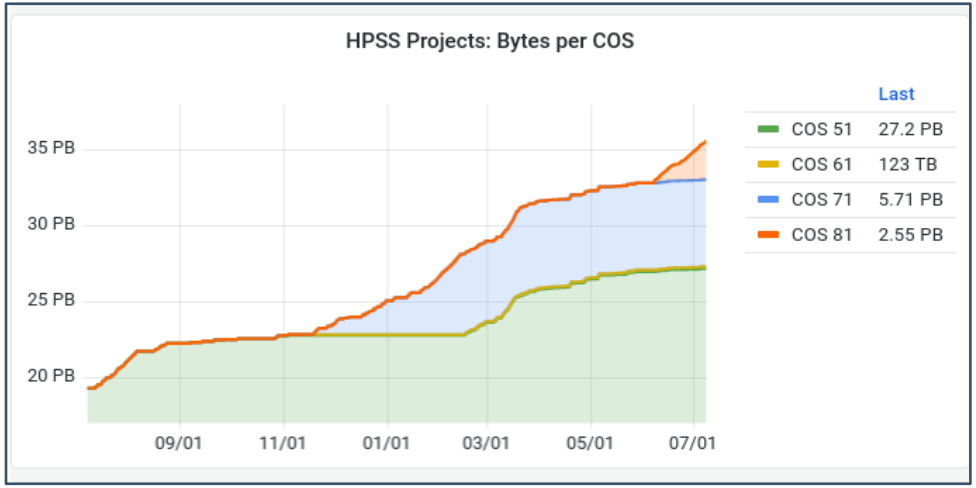
Current HPSS Setup

HPSS Tape Library

- SpectraLogic Tfinity
- 39 TS1160 tape drives (Native sustained data rate (uncompr.) → 400 MB/s)
- 20TB tape capacity

HPSS Storage: Current Status

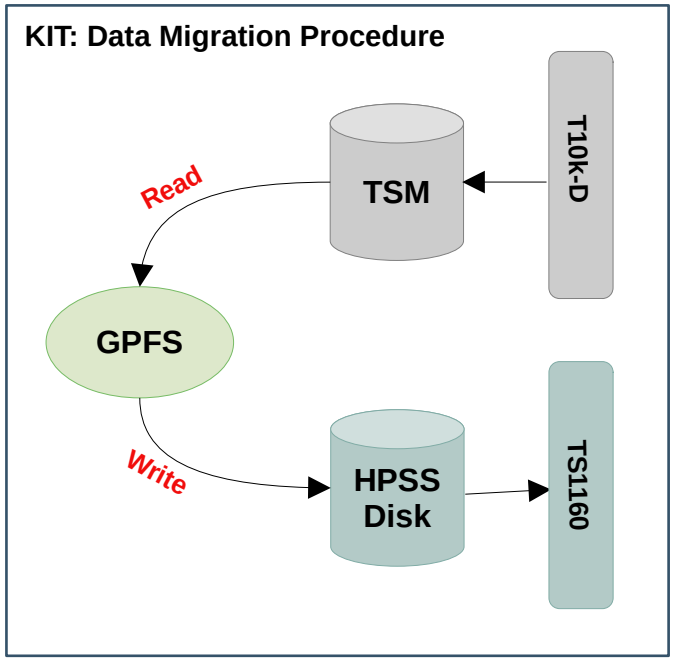
- CMS/LHCb/Belle2 are migrated to HPSS
- CMS/LHCb are currently in production using HPSS
- Currently available ~35PB in total



- ATLAS data migration is currently in progress
- *Next step is to migrate ALICE*

Data Migration from TSM to HPSS

- Data is transferred outside of dCache
- Query chimera DB for file names
- Reading a complete datasets from TSM to GPFS
- Writing a dataset via pftp to HPSS
- Verifing the checksum after writing to HPSS disk
- Using 1 tape drive per dataset to write files to tape



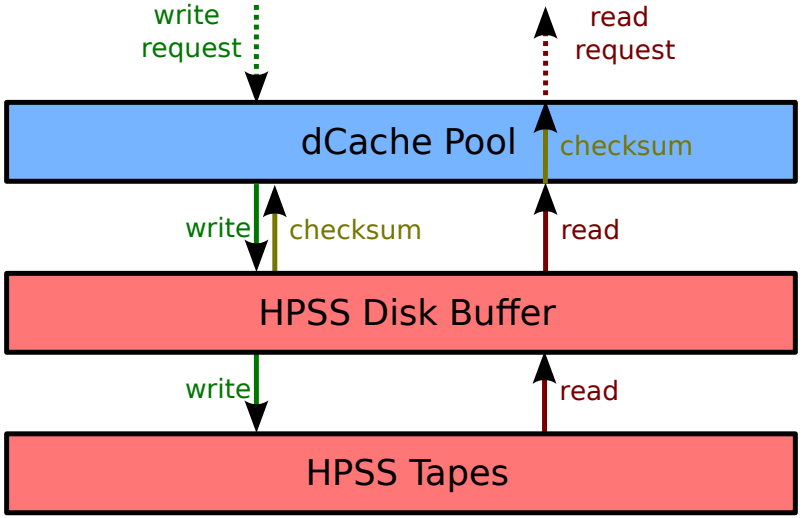
HPSS Tape System Overview

Writing:

- Incoming file transfer at dCache pool
- File written from dCache to HPSS disk buffer
- Read back for checksum test
- Within HPSS, writing to tapes initiated afterwards in file aggregates

Reading:

- File read requests collected for file aggregates
- Entire aggregates read from tapes to HPSS disk buffer (Full Aggregate Recall mechanism)
- Files read from HPSS disk buffer to dCache pool
- Before sending out, checksum test performed by dCache

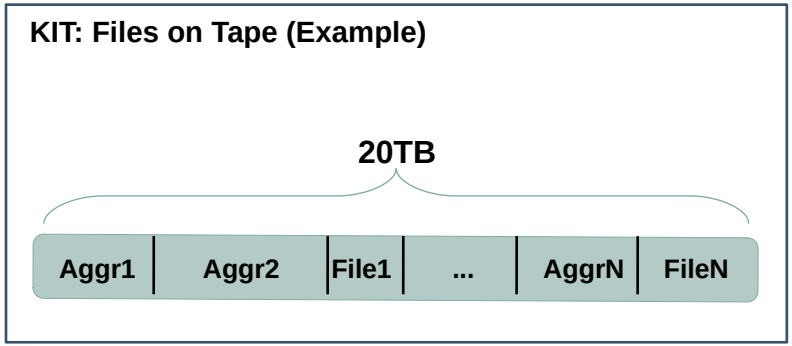


Up to 100 files ≤ 10 GiB in the same directory collected into aggregates

- HPSS has its own disk buffer/cache, shared between all VOs.

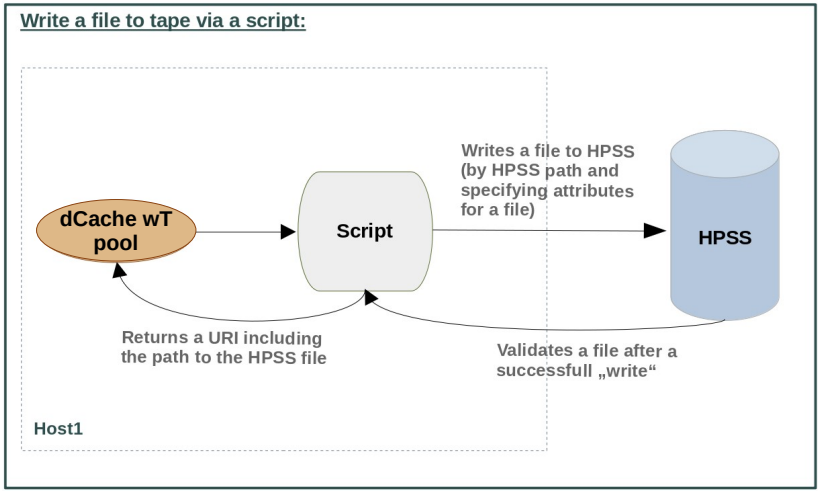
Files on Tape

- The files on tape are a mixture of individual files and aggregates.



Writing Files to HPSS (Production Setup)

- Get LFN from dCache and converting it to HPSS path
- Write a file to HPSS disk via pftp and by setting specific HPSS attributes such as
  - VO id, FF number, checksum type/value
  - Calculate file family number on the fly per file
    - Dataset name is in the path
    - Dataset information is used in conjuction with existing file families of VOs
    - FFs are integers and are reused within the same VO, but unique per VO
    - VO specific setup
    - Currently we use 1 tape drive per FF for writing to HPSS



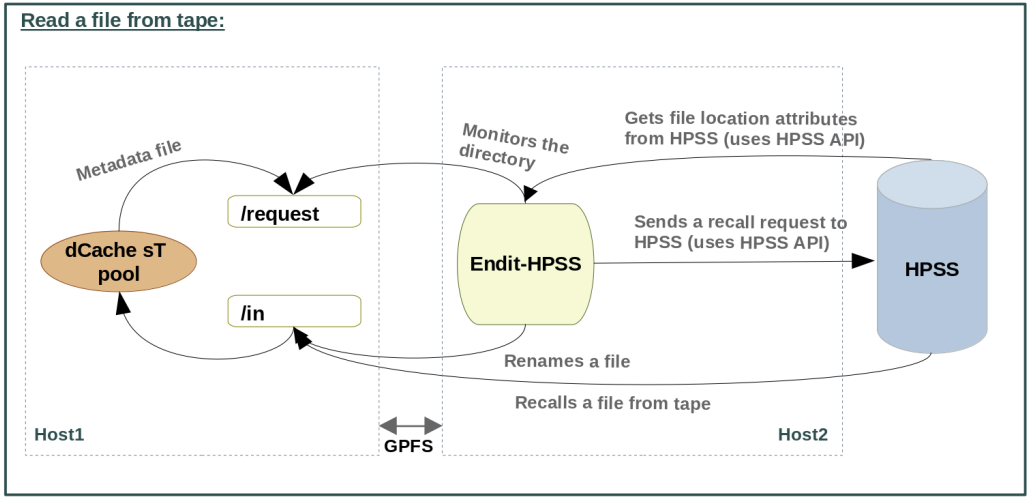
This currently works via a script, which is based on the current migration setup (developed at GridKa).

Reading Files from HPSS (Production Setup) -1

- Continue using dCache Endit-Provider plugin
- ENDIT-HPSS: interacts with the dCache ENDIT-Provider plugin and the HPSS API (developed at GridKa).
- Continue using GPFS

Endit-HPSS:

- Extracts a file HPSS path from the dCache URI
- Queries file location attributes from HPSS
- Groups files per tape and per aggregate on that tape and creates a list(s) of them
- Iterates through the created lists of aggregates and recalls files sequentially from tape
- Allows to control the number of used drives per VO
- **Supports multiple dCache pools** (next slide)

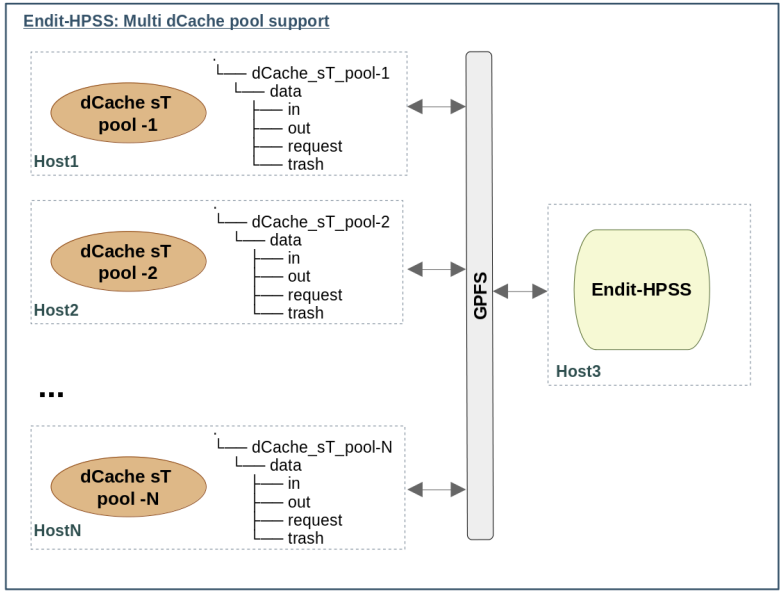


Reading Files from HPSS (Production Setup) -2

- Continue using dCache Endit-Provider plugin
- ENDIT-HPSS: interacts with the dCache ENDIT-Provider plugin and the HPSS API (developed at GridKa).
- Continue using GPFS

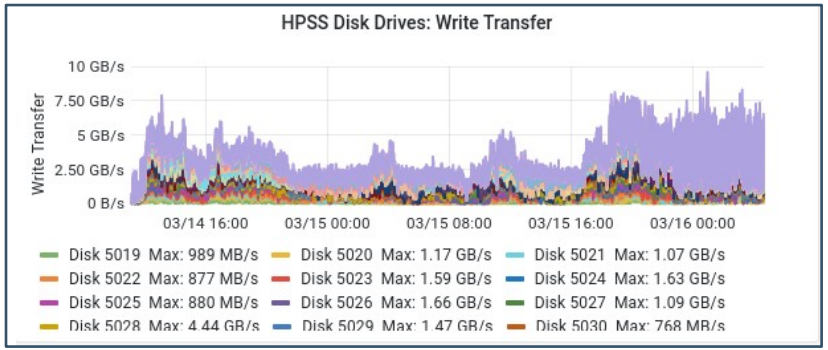
Endit-HPSS:

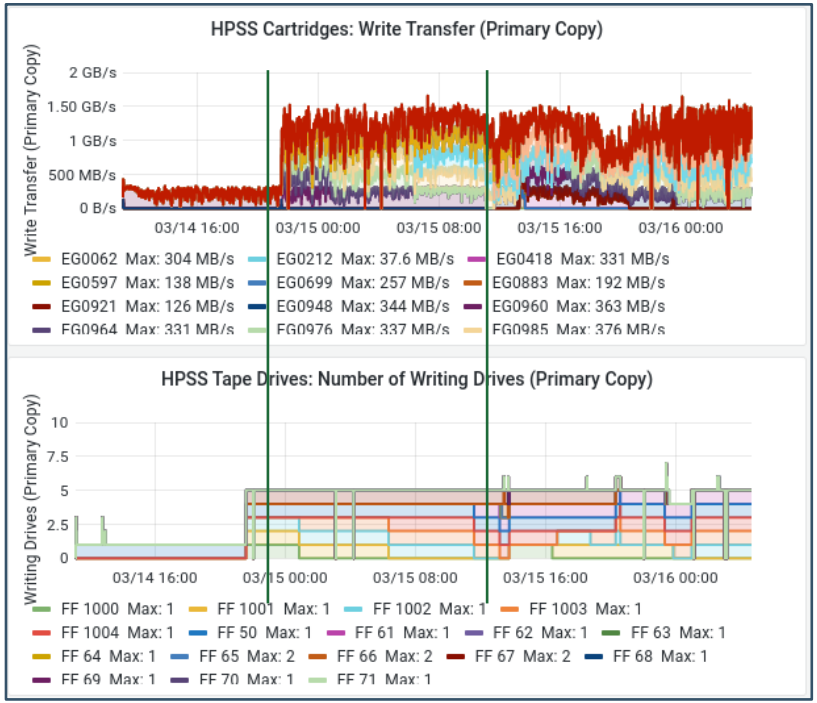
- Extracts a file HPSS path from the dCache URI
- Queries file location attributes from HPSS
- Groups files per tape and per aggregate on that tape and creates a list(s) of them
- Iterates through the created lists of aggregates and recalls files sequentially from tape
- Allows to control the number of used drives per VO
- **Supports multiple dCache pools**



Writing to Tape: Results (Production Setup)

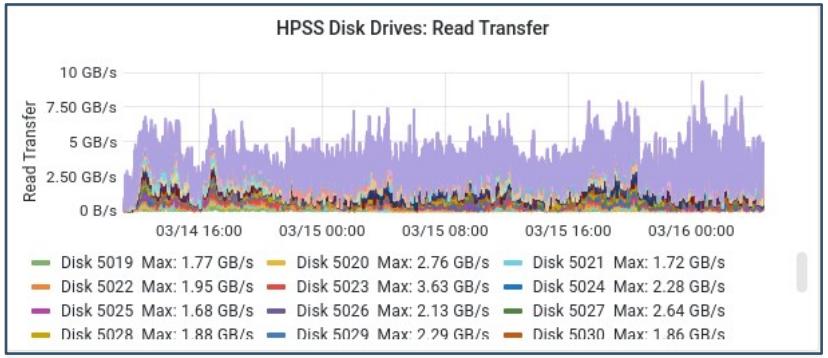
- ***Max write rate (disk):*** ~8.0 GB/s
- ***Max write rate (disk → tape):*** ~1.5 GB/s (~300 MB/s per tape drive)
- ***Drives:*** 5

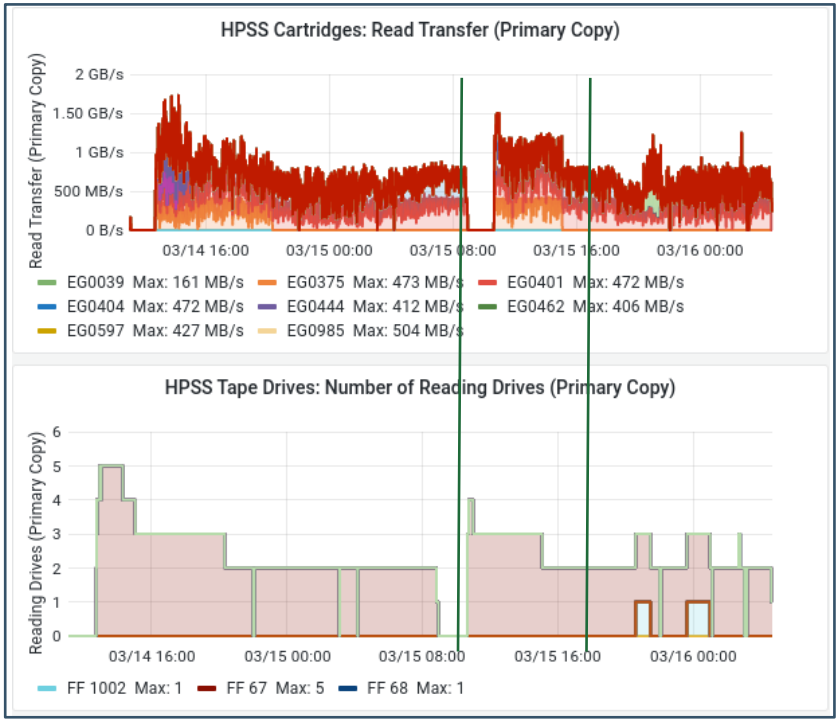




Reading from Tape: Results (Production Setup)

- ***Max read rate (disk):*** ~8.0 GB/s
- ***Max read rate (disk → tape):*** ~1.2 GB/s (~400 MB/s per tape drive)
- ***Drives:*** 3





Tape Challenge Results

Tape Challenge 2022

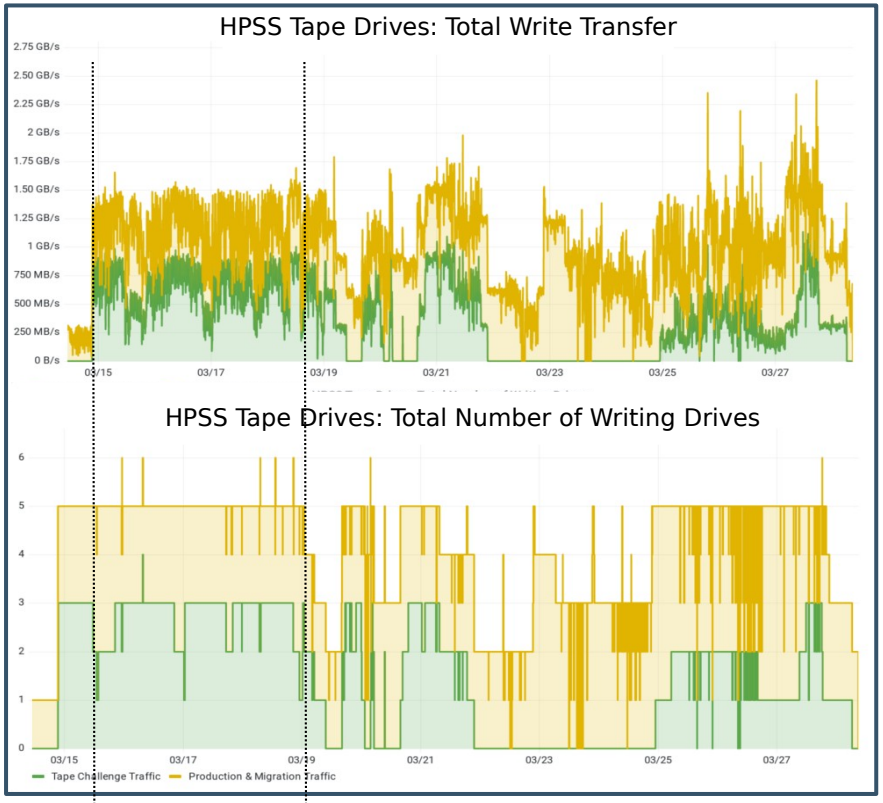
- ***CMS:*** 2nd week of March (7th~11th) for A-DT test
- ***ATLAS/CMS/LHCb:*** 3rd week of March (14th~18th) for DT test
- ***ATLAS/LHCb:*** 4th week of March (21st~25th) for A-DT test

Just before the „Tape Challenge 2022“, CMS & LHCb were switched from TSM to HPSS.

CMS results – Writing

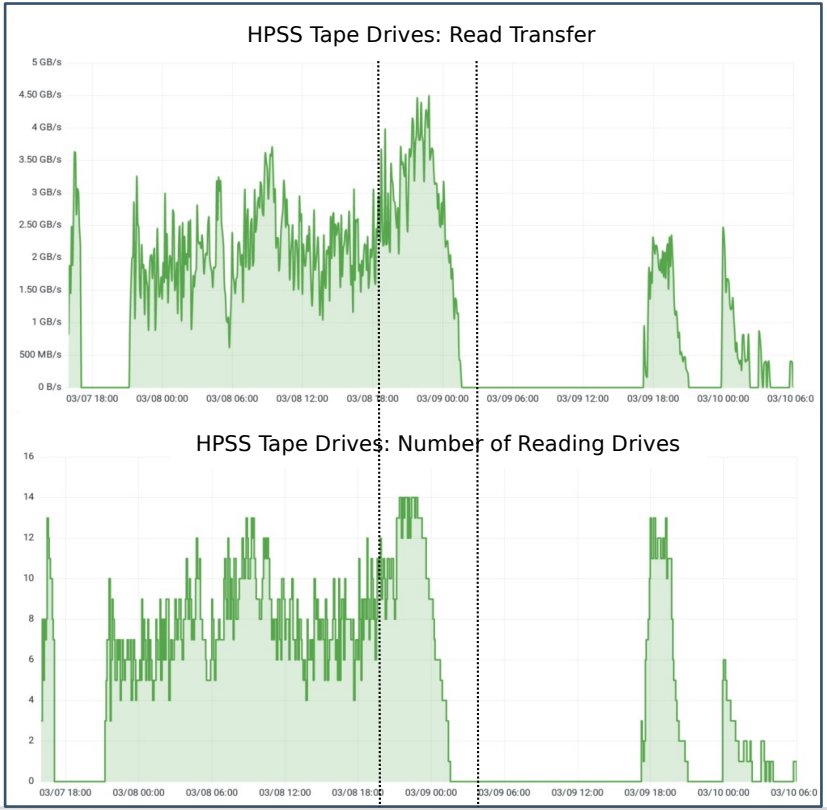
- Good opportunity to test the new HPSS setup for CMS

- ***Avg write rate:*** ~1,5GB/s
  (~300MB/s per drive)
- ***Tape drives:*** 5



CMS results – Reading

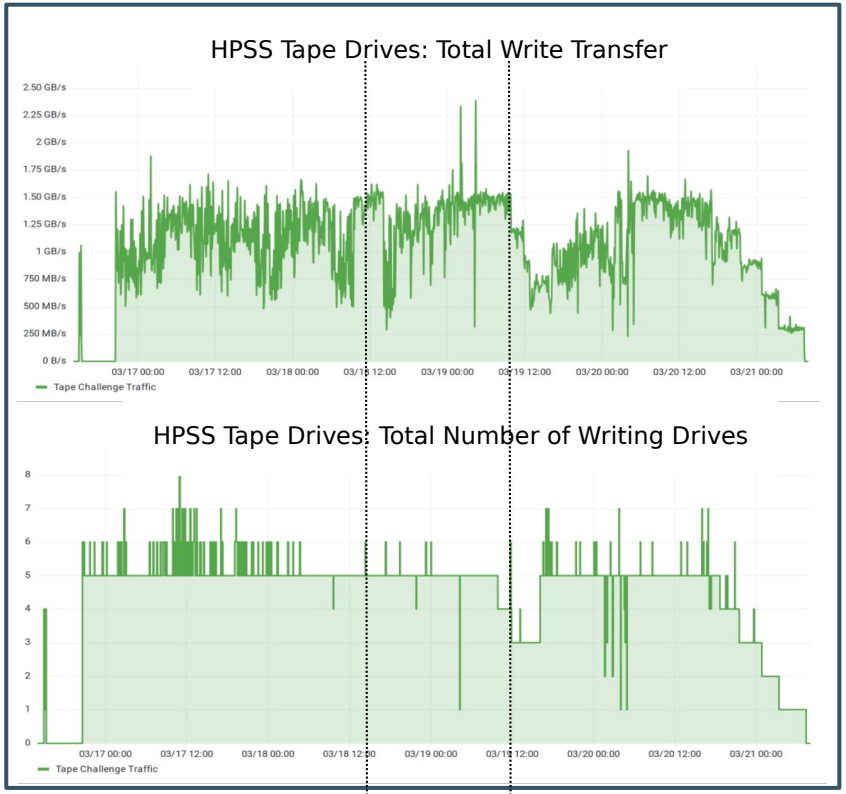
- ***Avg read rate:*** ~4,2GB/s
  (~300MB/s per drive)
- ***Tape drives:*** 14



LHCb results – Writing

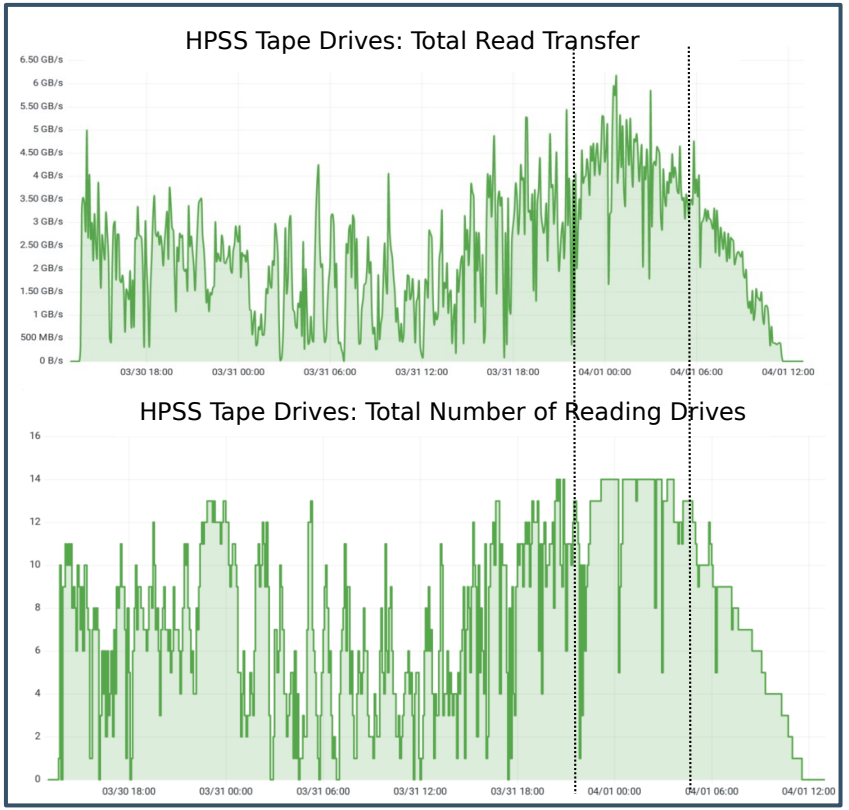
- Good opportunity to test the new HPSS setup for LHCb

- ***Avg write rate:*** ~1,5GB/s
  (~300MB/s per drive)
- ***Tape drives:*** 5



LHCb results – Reading

- ***Avg read rate:*** ~4,2GB/s (~300MB/s per drive)
- ***Tape drives:*** 14



Conclusions

- During the Tape Challenge (see KIT_report) we were able to test our new setup, tune and fix it accordingly.
- The HPSS setup together with TSM setup is currently in production for CMS/LHCb/Belle2.
- As for reads, within the new setup, we achieved max ~400 MB/s per tape drive, which is not the case for the TSM use case (max ~180 MB/s per drive).
- Grouping files into aggregates before writing them to tape and reading those aggregates works very efficiently.
- Having well performing HPSS disk cache is essential!

The new setup works very well and the overall tape rate is improved by more than factor of 2 per tape drive.

THANK YOU

Backup

dCache/Endit-Provider plugin

- Adapted for HPSS use case