Karlsruhe Institute of Technology

GridKa

# Tape @ KIT

Artur Gottmann, Andreas Petzold

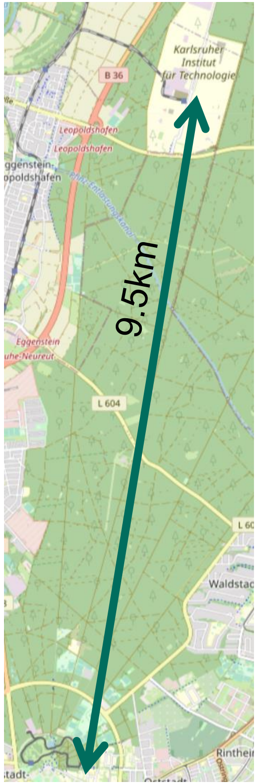
# Tape @ KIT

- 2 sites ~8 J-E tapes apart
- Libraries
  - 2 SL8500, 1 TS3500 to be retired
  - 1 TS4500, 1 TFinity, 1 TS4500 being purchased
- TS1155/TS1160 drives
  - TS1155 to be replaced by TS1160 soon
- J-D/E cartridges 15/20TB
- Customers
  - GridKa Tier-1 (Pledge 135PB, 90PB on tape)
  - Disaster recovery copy of Large Scale Data Facility (14PB)
  - Archive Service bwDataArchive (7PB)
  - Server backup (9PB) (TSM)

GridKa

LSDF

BWDATA ARCHIVE



# HPSS @ KIT

- HPSS used as tape platform for large scale applications
- Proven system
  - WLCG: CC-IN2P3, BNL
- Not w/o challenges
  - Performance
  - occasional bugs
- Many interface options
  - FUSE mount/API calls/pftp/GPFS integration/...
- Transparent aggregation

# Lessons learned

- High performance requirements for tape
  - Data transfer/processing chain very complex with many possible bottlenecks
  - Involve all experts
- Drives can be operated streaming at ~nominal speeds
  - 380/400MB/s (w/r)
  - if disk buffer has enough IOPS ➔ NVMe
  - if aggregation works well
- Latest generation of drives/tapes very sensitive to environment conditions
  - Humidity
- Constant development required

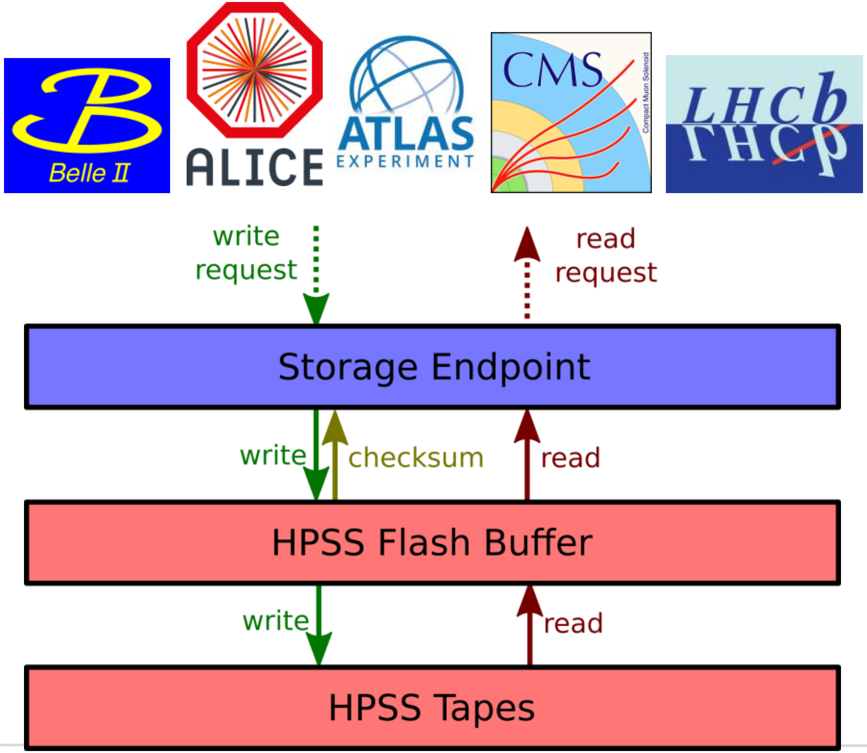
# Outlook

- Plan for new library for Tier-1
  - no decision yet for type or site
- Enterprise vs. LTO
  - recent offers show only very small price differences
- Next generation drives
  - TS1170 doubles capacity but still 400MB/s – not ideal
  - LTO10/TS1180 rumored to have improved throughput
- Cost of tape system
  - Dominated by media, but performance requires many drives, fast buffer system, and network

# HPSS @ GridKa

# Overview

- Multi-experiment setup with
  - dCache: Belle 2, ATLAS, CMS, LHCb
  - XRootD: ALICE
- Individual experiment storage endpoints (SE’s) with disks for transfer requests
- Shared HPSS flash buffer (500TB) to collect data for/from tapes
  - Useful for file aggregation to up to 300 GB before writing to HPSS tapes
- File families in HPSS used to combine data to be written to the same tape
- File namespace available in HPSS
- File attributes can be set in HPSS



# Writing to HPSS from dCache

Benefits of using dCache for the SE’s:

- Write requests managed by dCache & obtain a unique identifier (pnfsid)
  - Transfer requests collected in queues
  - Number of concurrent active transfers to tapes
- Database with information on files in dCache:
  - file path & size
  - checksums, pnfsid
  - additional tape system info via URI
- Experiments can provide additional information:
  - Logical file names (LFN) reflecting dataset structures,
  - checksums of files
  - extended attributes (in use only by ATLAS currently); only dataset size used for HPSS at GridKa

For each single file, dCache is calling a script (dc2hpss.py) to write to HPSS a.s.a.p.

# File families and aggregates for datasets

- LFN structure of files might reflect how the data belongs together
- Based on such LFN’s, datasets can be identified & matched to a file family
- Writing to tapes with 1 drive at a time per file family
  → files from the **same dataset** are written to the **same tape**
- File aggregation up to 300 GB only within the same directory
  → aggregates considered as a single entity when written to tapes

Illustrative example:

- /mc/winter23/ztautau/0/output_1.file → dataset: /mc/winter23/ztautau/ → file family: 91 (mc: 91-94)
  ...
- /mc/winter23/ztautau/1/output_n.file → dataset: /mc/winter23/ztautau/ → file family: 91 (mc: 91-94)
- /data/run3/tau/output_1.file → dataset: /data/run3/tau/ → file family: 99 (data: 95-98)
  ...
  → Files from /mc/winter23/ztautau/{0,1} aggregated up to 300 GB within each directory individually

# Dealing with large datasets

- File family assignment discussed before good for **small** datasets
  → Files end up on a single tape → Low number of mounts
- However: **large** datasets would only be written with one drive (max. 400 MB/s) at a time to tapes
  → HPSS flash buffer **might fill up**

Solution:

- Assign **several** special file families for a big dataset
  → Need dataset size hint on a file-by-file basis → Can be provided via extended attributes
- Currently for ATLAS in use for datasets > 40 TB: 8 special file families
  → 8 drives used to write to tapes → rate increased to 3.2 GB/s

# Recalling files from HPSS

Main goal: recall files efficiently from tapes for O(50k) requests

- Best for tapes: mount only once and read from front to end
- Best for experiments: obtain files at stable rates of O(1GB/s)
- Experiments recall large fractions of datasets during recall activities

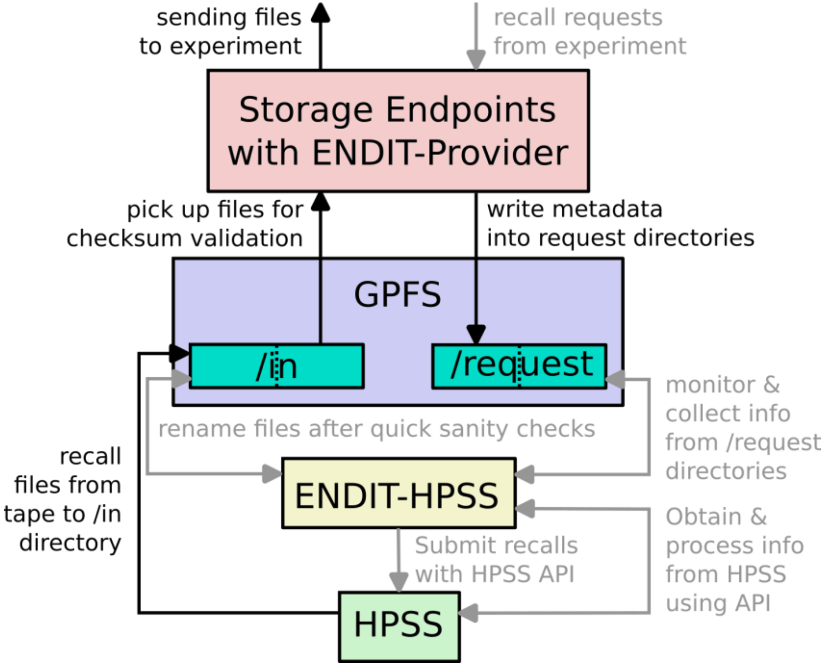
→ Optimize based on these boundary conditions:

- full aggregate recall (FAR) in HPSS
  → faster reading of files on a tape from the same aggregate
- recommended access order (RAO) in HPSS
  → multiple aggregates are recalled in most efficient order from a tape
- number of used drives per experiment configurable
  → remaining flexible w.r.t. the load on HPSS

Deployed in an adapted dCache ENDIT-Provider and dedicated ENDIT-HPSS interface
→ technical details to be published in CHEP 2023 proceedings

# Recall workflow of ENDIT

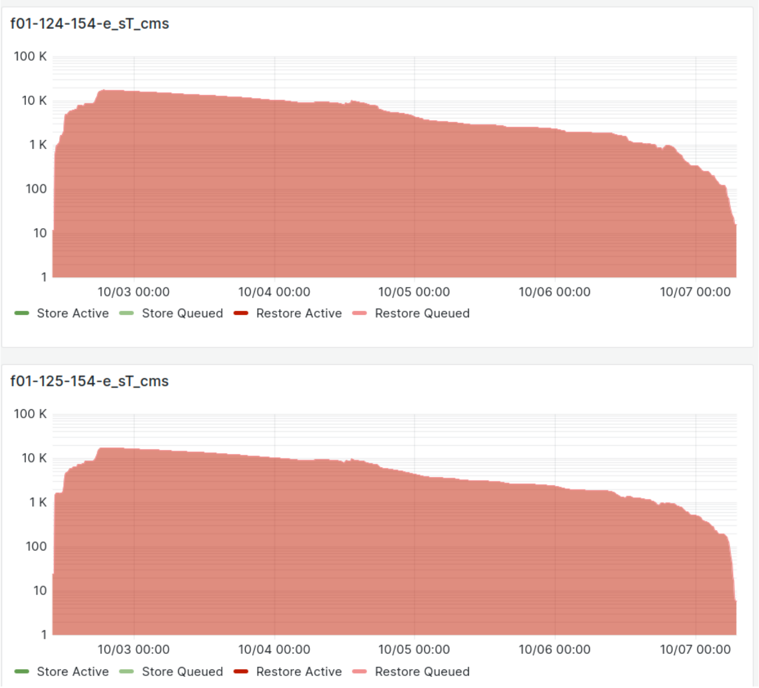
1. Collect metadata from recall requests
2. Use provided URI to obtain info from HPSS
3. Group requests by tape and aggregate
4. Put tapes in a processing queue
5. Process multiple tapes concurrently (e.g. 14) → number controls used drives
6. For each tape, submit to HPSS a recall for **one file** per aggregate
   → triggers FAR for these aggregates
7. Once submitted file recalled: iterate through remaining files from the same aggregate to recall them quickly
8. Once no aggregates left for a tape, pick new one from processing queue


sending files to experiment
recall requests from experiment
Storage Endpoints with ENDIT-Provider
pick up files for checksum validation
write metadata into request directories
GPFS
/in
/request
rename files after quick sanity checks
monitor & collect info from /request directories
recall files from tape to /in directory
ENDIT-HPSS
Obtain & process info from HPSS using API
Submit recalls with HPSS API
HPSS

# Fraction of requested vs. recalled files

- We assume, that each recall activity is for an entire dataset
- This is not true in reality
  → extreme cases requesting a **single file**
- **Main reason: large fraction of dataset already on disk somewhere**
- For larger recall campaigns, the coverage is better, but not ideal:
  - About 35k requested files
  - Due to FAR, ca. 102k recalled files

  → **only 34.4%** of files used in the end

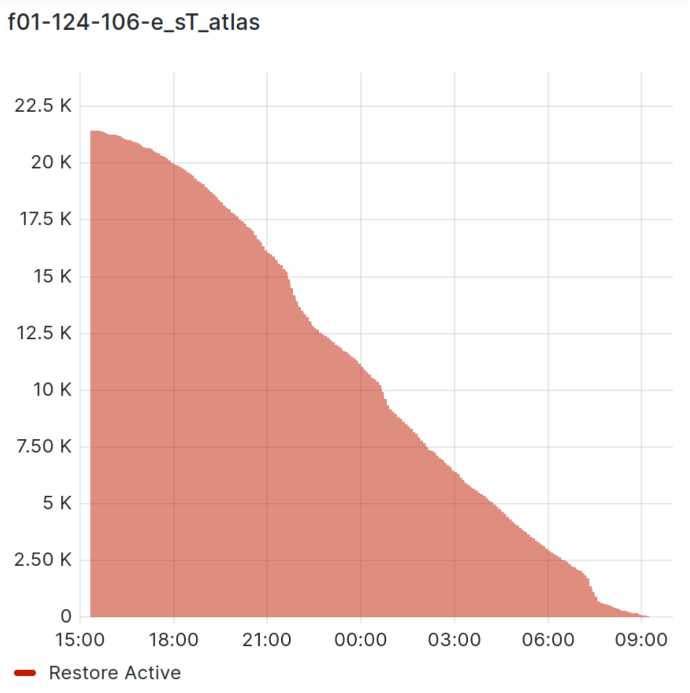


CMS recall campaign from 02.10.2023

# Backup

# Recall experience: ATLAS tests

- ATLAS recently performed recall tests at GridKa
- Good performance in handling several 10k requests of ~ 100 TB volume
- 320 - 340 MB/s per drive on average
- More details presented by Xin Zhao



during AOD recall test 26.10 - 27.10

# ATLAS setup for HPSS at GridKa

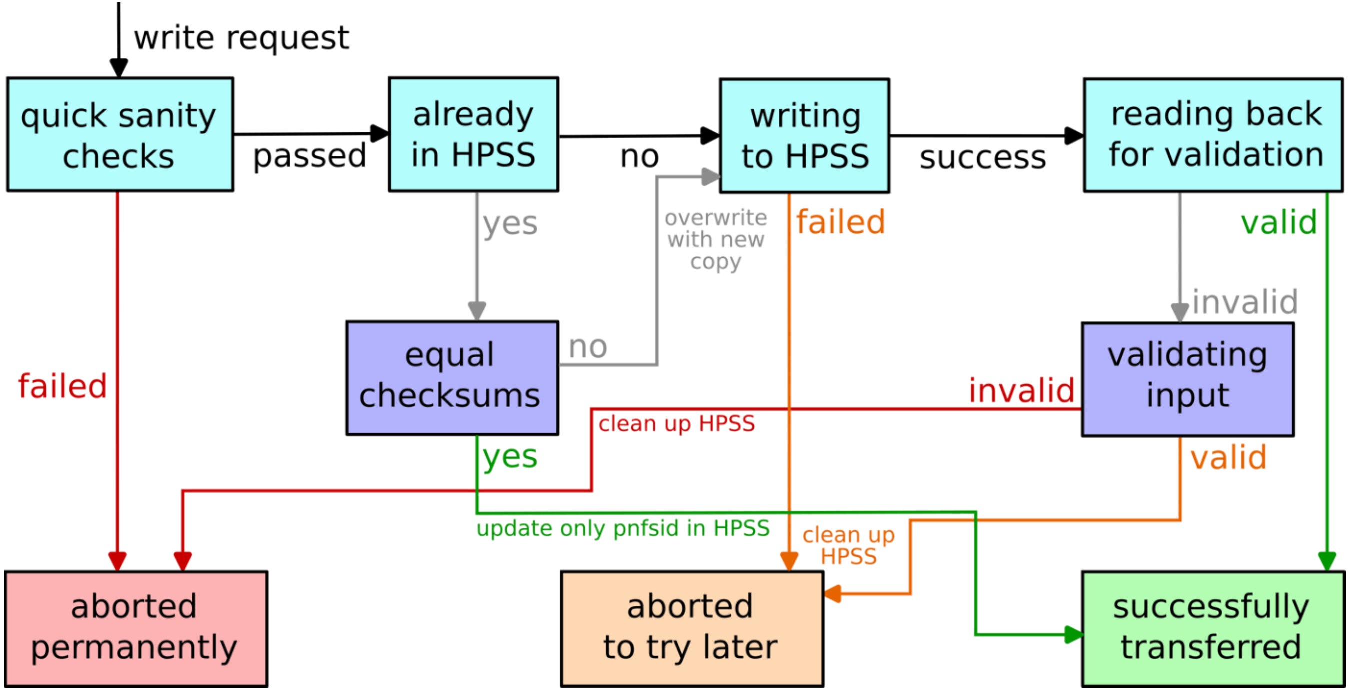
## Writing

- Transfer protocol used is WebDAV
- Extended attributes currently provided from Rucio over FTS using URL parameters
- Further plans discussed by Xin Zhao in more detail
- CMS approached ATLAS to adapt a similar solution

## Reading

- Using Tape Rest API in production since 19.04.2023
- Submission of requests orchestrated by API
- Old SRM-bring-online decommissioned

# Workflow of the script dc2hpss.py



# Writing to HPSS for ALICE from XRootD

ALICE experiment is using XRootD as solution for the SE

→ Adaptions taken into account for writing to HPSS workflow:

- **Compute** the checksum for verification before writing file to HPSS flash buffer
- LFN directory structure from ALICE without a dataset meaning

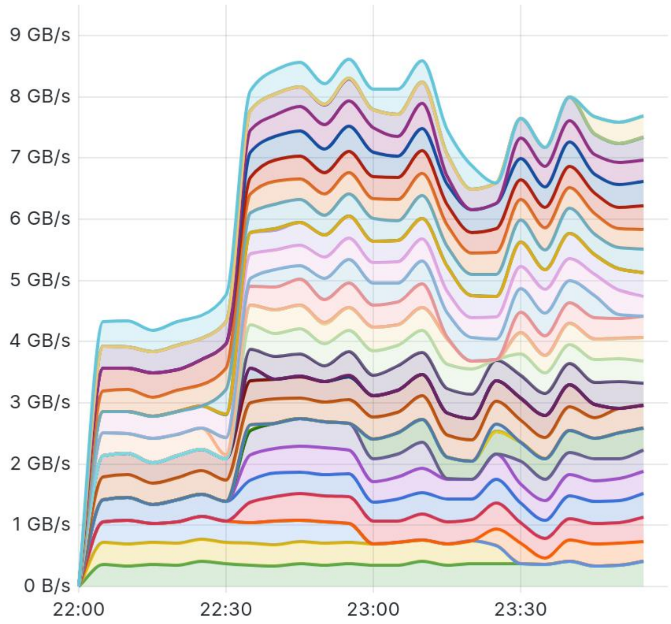
  → File families discarded, aggregation performed **time-based**

# Writing experience

- With 500 TB HPSS flash buffer, **stable** conditions
  → 340-380 MB/s per drive, even under load

- Up to 8 drives for writing to tapes per experiment
  → observed 24 active drives

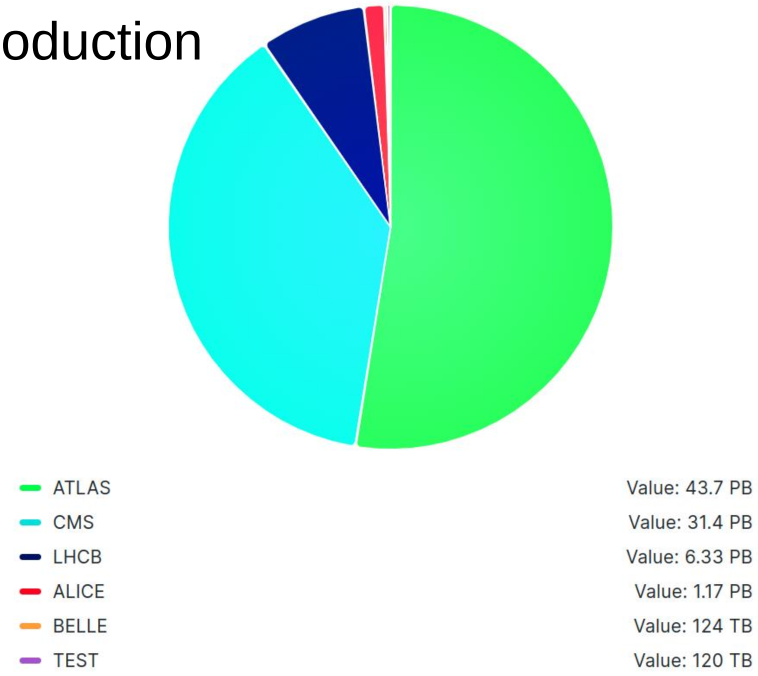
HPSS Tape Drives: Write Transfer per Drive/Cartridge

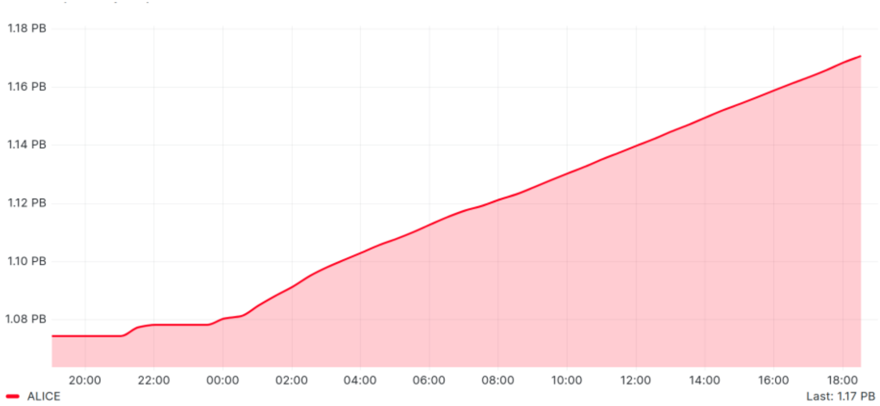


16.10.2023: writing load on HPSS from ALICE, ATLAS and CMS

# Current HPSS usage status at GridKa

- ATLAS, Belle 2, CMS, and LHCb in production
- ALICE migration of 11.7 PB to HPSS ongoing with ~ 100 TB/d





# HPSS Hardware Setup

- Core server with DB2 (+standby)
- disk and tape mover servers for different projects
- Disk buffer systems
  - NetApp E5600/E5700 with HDDs (~2.5PB)
  - Dell ME5024 with SSDs (250TB)
  - Server with local NVMe storage (250TB) HEPiX 2023 Presentation