dCache @ KIT

20th International dCache Workshop, May 6-7, 2026, Nikhef

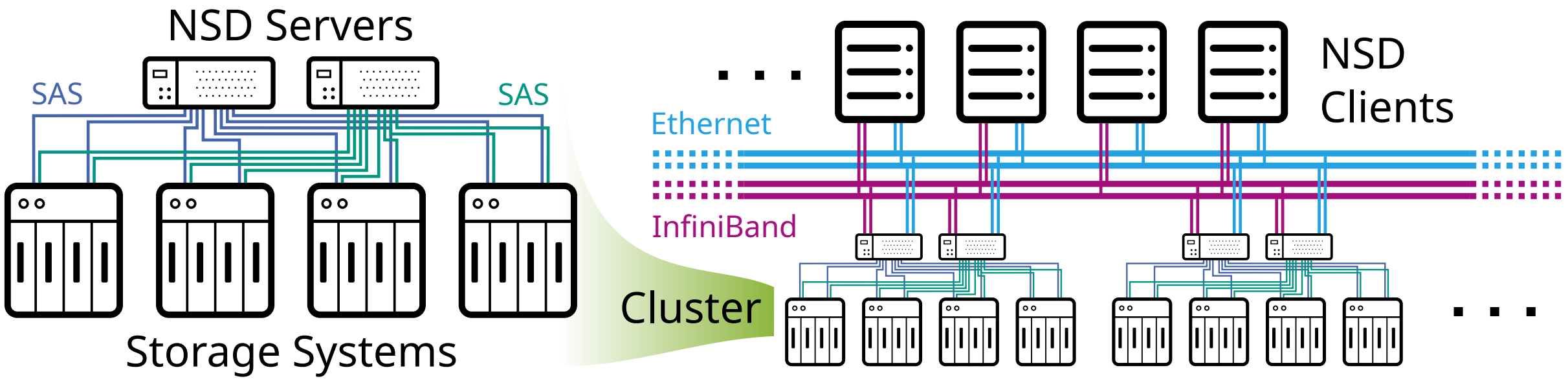
Artur Gottmann on behalf of the GridKa Online Storage Team | May 6, 2026

dCache Setup at GridKa in Numbers

Characteristic	Experiment				
	ATLAS	Belle II	CMS	LHCb	Others
dCache	11.2.3 Golden Release				
Disk pledge [PB]	29.0	2.3	20.5	17.9	4.6
Disk pools [#] × [size]	2 × 2.1 PB 10 × 2.5 PB	1 × 250 TB 2 × 1.0 PB	2 × 1.4 PB 7 × 2.4 PB	7 × 2.6 PB	3 × $O(100)$ TB 2 × 2 PB
Tape pools [#] × [size]	4 × 175 TB	2 × 50 TB	4 × 140 TB	4 × 100 TB	–

- All 2026 disk pledges deployed in time. Upgrade to `dCache` 11.2.3 without problems
- Typically, each experiment is provided with a dedicated `dCache` setup
- Our parallel file system allows for **large pools** of several PB

File System at GridKa: IBM Storage Scale



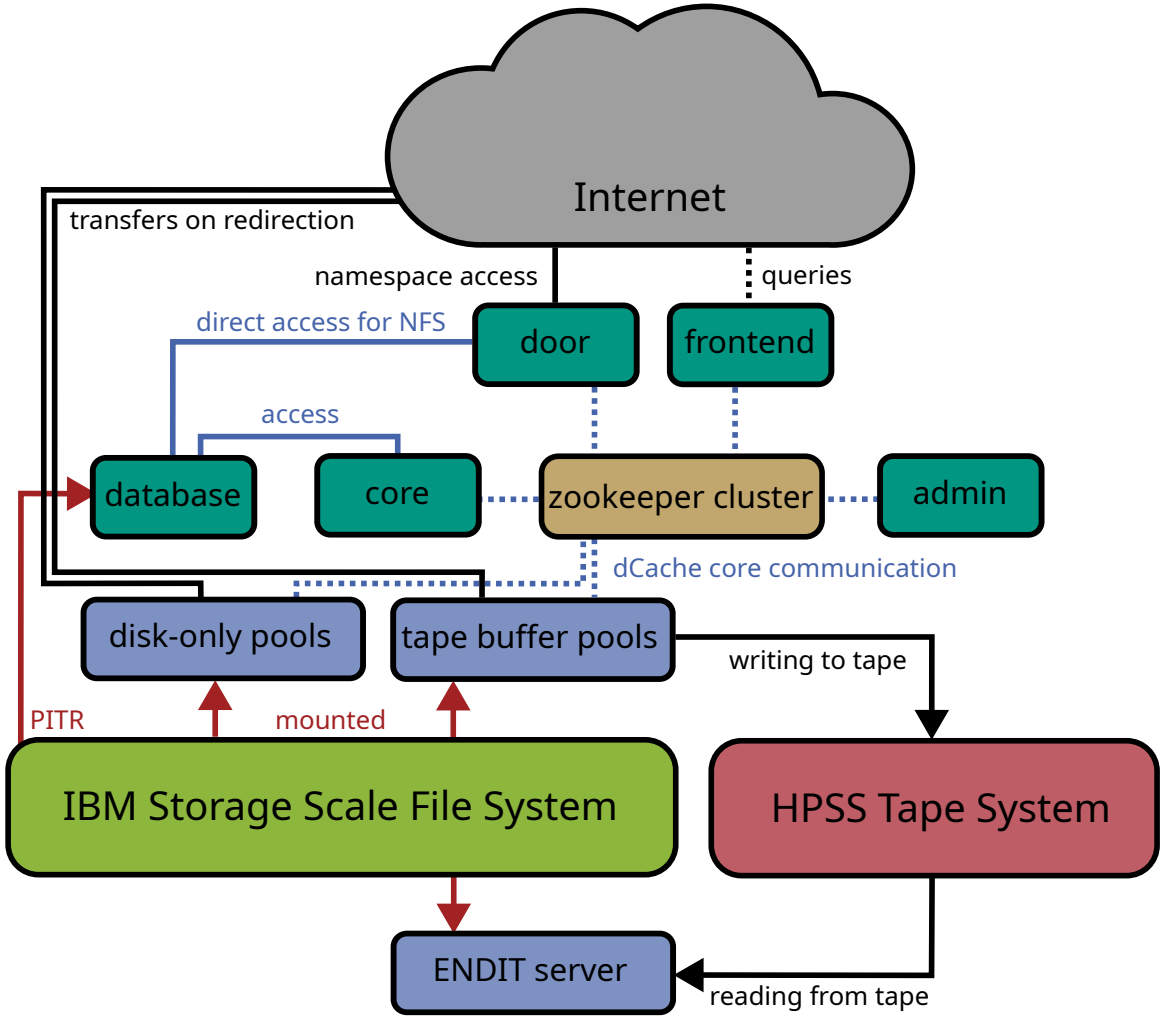
- NSD servers and storage systems connected redundantly via SAS to a NSD building block (left)
- Several building blocks are composing a storage cluster of for example $O(100)$ PB (right)
- Such a storage cluster contains several file systems, typically one for each experiment
- (Meta-)Data exchange via InfiniBand (including RDMA), server administration via Ethernet, each redundant
- $16 + 2$ data redundancy scheme deployed, similar to a RAID 6 setup

NSD clients - like `dCache` - see a **single** large, mountable, and POSIX-compliant file system

dCache Setup at GridKa: Schematic View

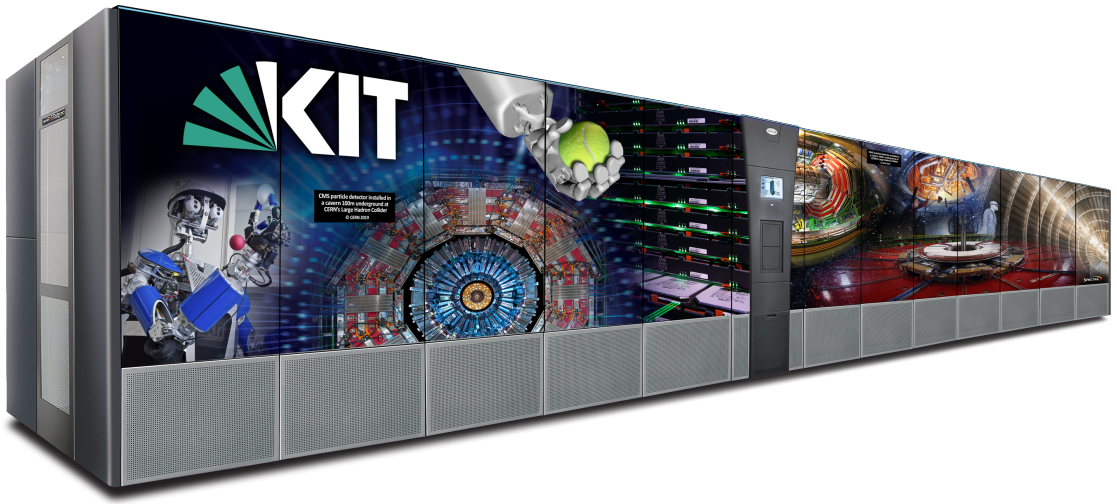
- dCache services running on single VMs:
  - door: WebDAV, XRootD, NFS
  - frontend: frontend, httpd, statistics, telemetry
  - admin: admin, billing, bulk
  - core: gplazma, cleaner-disk, topo, managers (pin, pnfs, pool, transfer, space)
- postgresql dCache database runs on a single VM → PITR backup on IBM Storage Scale FS
- 3-VM zookeeper cluster to establish dCache core communication via core ↔ service tunnels
- pools and endit servers the only hardware machines
- On the right: schematic view of GridKa dCache setup

VMs with snapshot backup within our VM provider → allows fast recovery

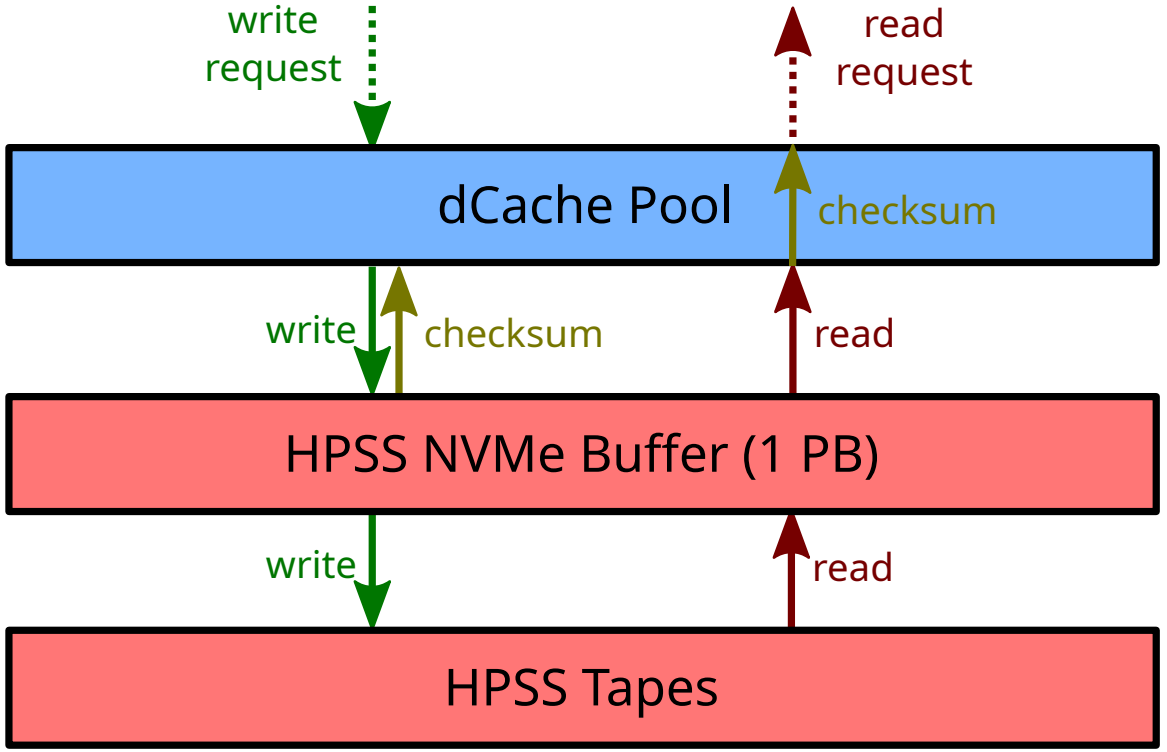


HPSS Tape System

- Tape system: TFinity Spectralogic with 48 TS1160 drives and more than 200 PB capacity
- Currently used tapes have 20 TB
- Drive rates up to 400 MB/s
- Files on HPSS buffer are subject to aggregation
  - Aggregates are constructed per directory
  - Up to 300 GB are collected into an aggregate
  - An aggregate is written as a single entity to tape



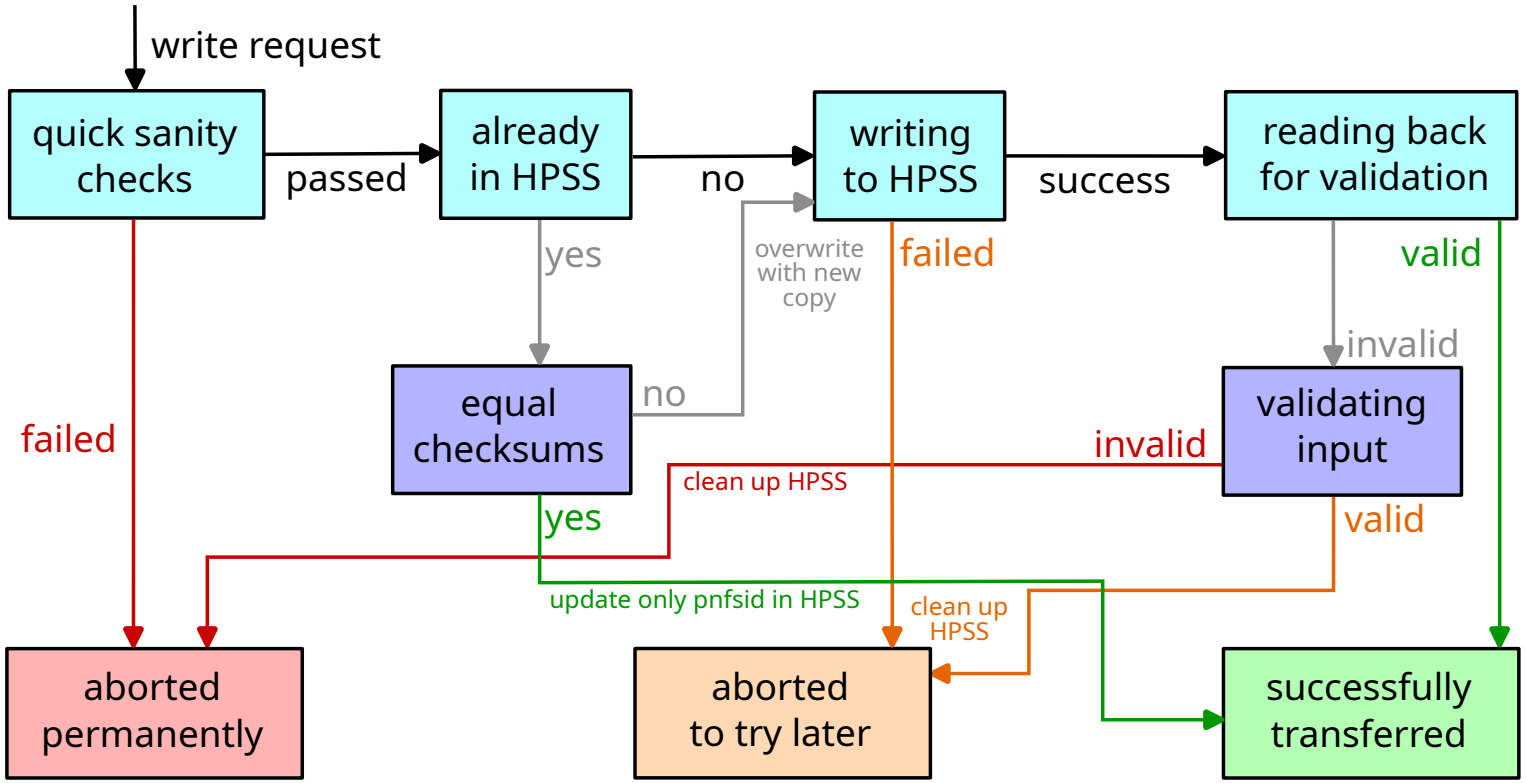
Schematic view of HPSS in the dCache Ecosystem



2-level buffer system (dCache and HPSS) makes writing to tapes very flexible

Interface to HPSS: Writing to Tape System

- Once on dCache tape buffer pools, files written FIFO to HPSS buffer
- Using dCache's ScriptNearlineStorageProvider
- File-families **on-the-fly** for tape assignment, computed from file paths
  - Once available: plan to extend the logic with archive metadata
- On the right: schematic description of the writing script dc2hpss.py

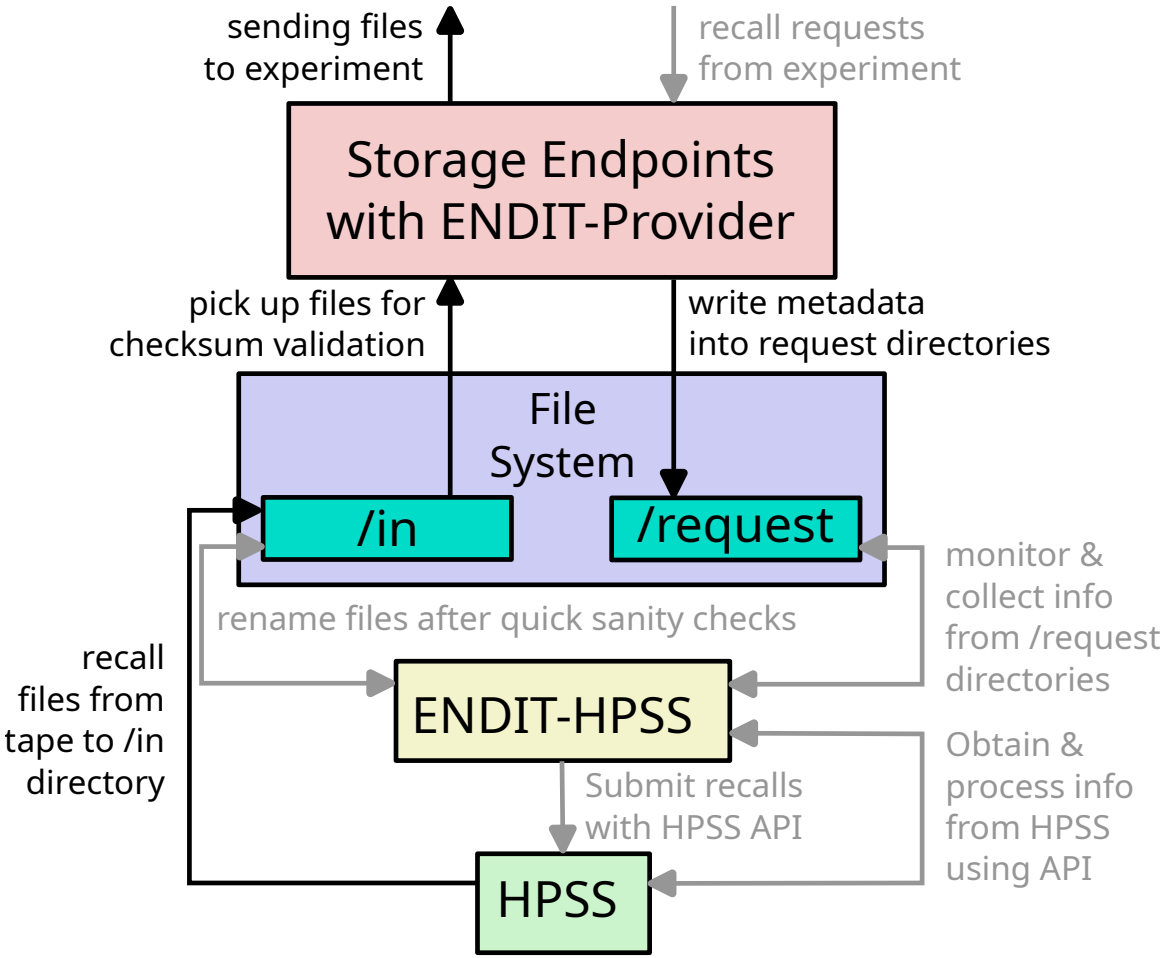


- If enough data on HPSS buffer (usually a few TB): migration to tapes with up to 8 drives in total per experiment
  - After waiting period expires, migration starts in any case
- For good co-location on tapes, we use **one** drive per file-family at a time

Interface to HPSS: Reading from Tape System

- Using **adapted** versions of ENDIT and its provider
  - For staging orchestration, using shared file system available to both, `dCache` tape pools and ENDIT servers
  - Before triggering recalls, requests are grouped by tape ID and aggregate
  - Making use of **Full Aggregate Recall** (FAR) by sending at the same time only **one** recall per aggregate
  - Recalls triggered by HPSS API read command calls → call processes remain active upon completion
- On the right: schematic view of ENDIT workflow at GridKa

- Making efficient use of data co-location on tapes → Recent demonstration: CMS tape challenge 2026
- Using up to 18 drives for recalls per experiment → up to 7.2 GB/s total read rate possible



Side Quest: dCache for bwLTS

- Pre-production setup for the successor of bwDataArchive for data archival on HPSS
- Much more diverse userbase compared to GridKa
  - Most of the time: individual participant entities like universities, small experiments, etc.
  - HPSS file-families created statically per directory of a participant upon registration
  - Technical implementation of file-families in `dCache` in the `OSMTemplate` directory tag
- Authentication and authorization via OIDC with RegApp
  - Making use of Code Flow, also in the `dCache` View. Required an implementation in `dCache` View to include PKCE → Big thanks to `dCache` developers!
  - In case of basic API access, using `oidc-agent` works out-of-the-box

We are happy to test future `dCache` improvements for Code Flow

Summary

- Running latest dCache Golden Release (11.2.X) series at GridKa
- Consistent setup scheme for multiple experiments
- Slim and robust setup making use of VMs with snapshots for most of its services
- File system setup allows for large dCache pools of several PB size
- 2-level buffer system for the tape setup allows for large flexibility

GridKa Online Storage Team

- Department leads: Dr. Doris Ressmann, Andreas Petzold
- Technical lead: Dr. Max Kühn
- Team lead: Dr. Serge Sushkov Evdoshenko
- IBM Storage Scale administrators: Jolanta Bubeliene, Dr. Uwe Falke
- dCache administrators: Xavier Mol, Dr. Artur Gottmann