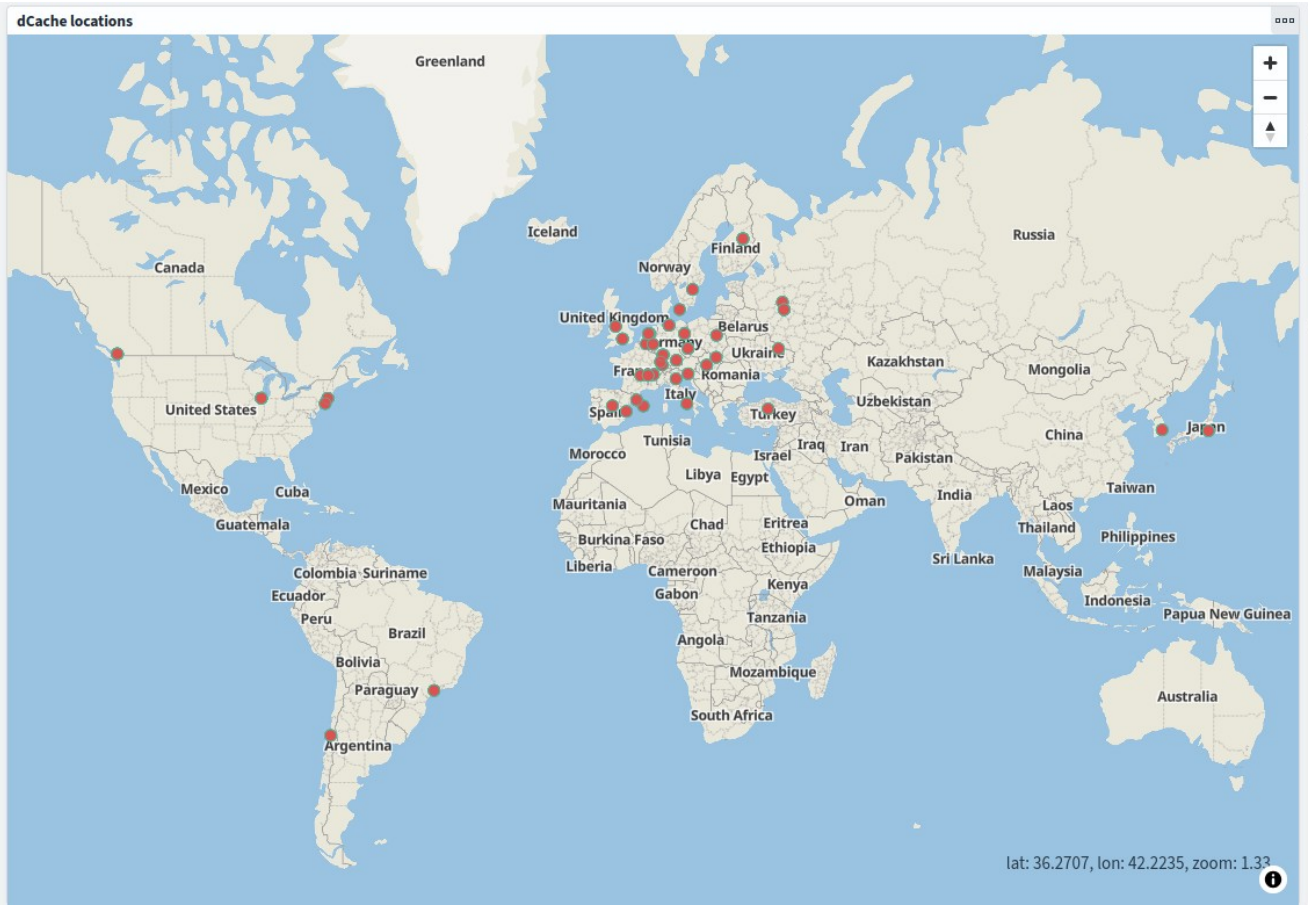
dCache Project Updates

Anastasiia Chub on behalf of dCache team

20th International dCache Workshop

Overview

The Geography



Last update

Apr 9, 2026 @ 17:50:04.850

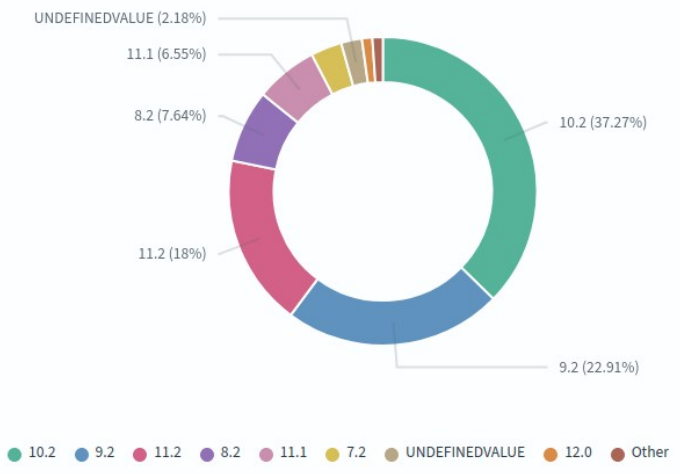
Total Capacity

583.7PB

Number of sites

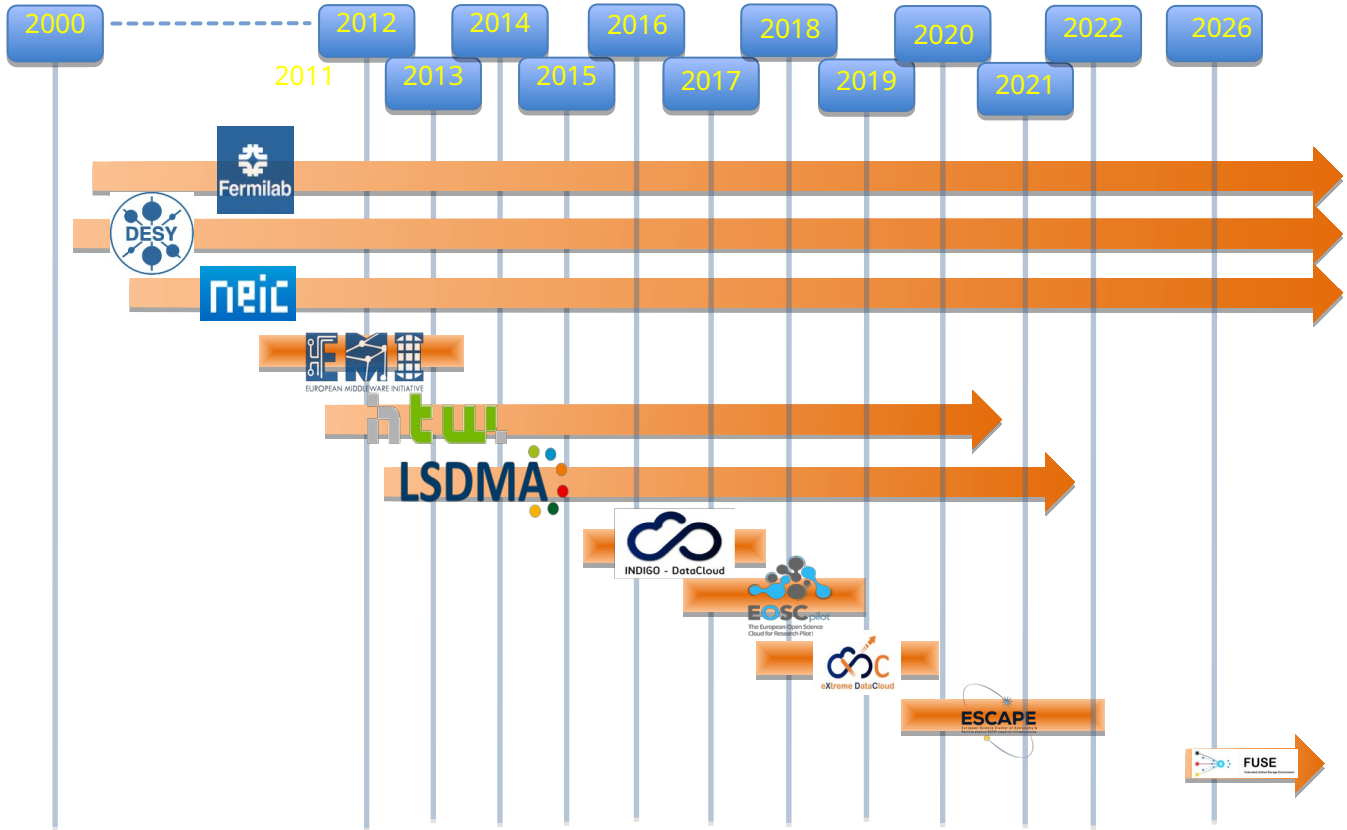
74

Versions pie



Project Funding & Team

- **DESY**
  - Anastasiia Chub
  - Karen Hoyos
  - Tigran Mkrtchyan
  - Lennart Sack
  - Marina Sahakyan
- **FermiLab**
  - Dmitry Litvintsev
  - Chris Green
- **NeIC**
  - Darren Starr

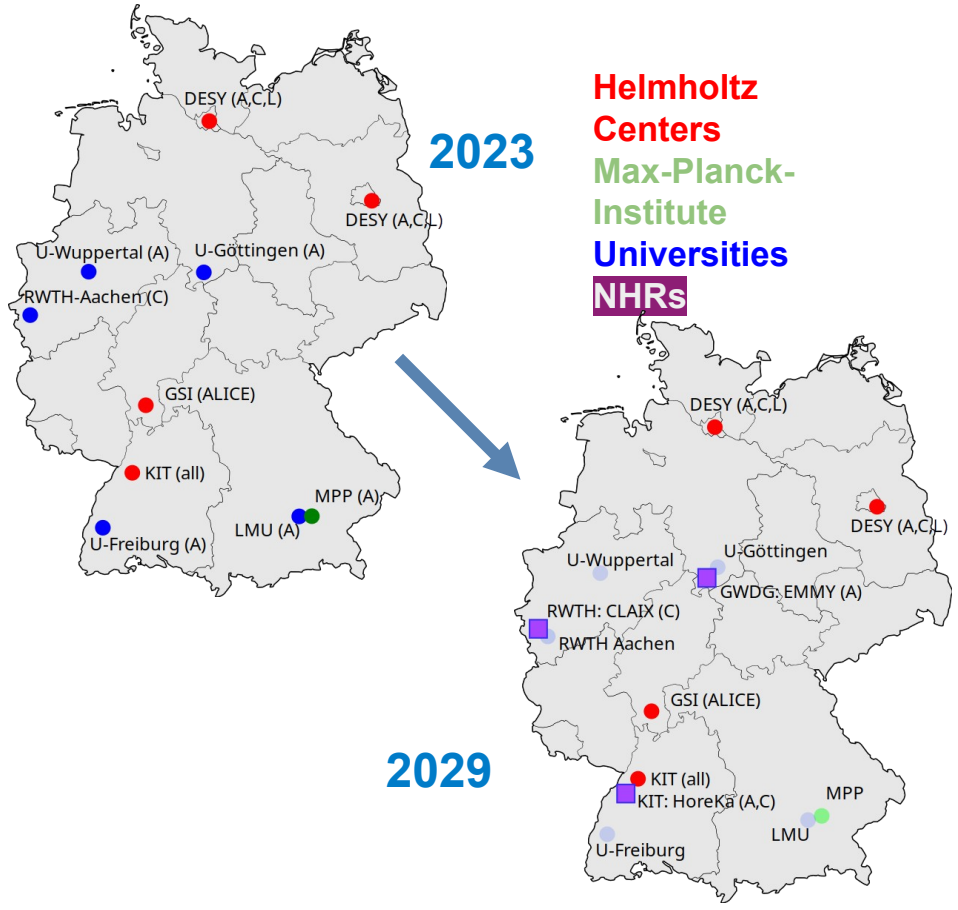


Main Focus

- HL-LHC
  - Distributed operation
  - Multi-purpose monitoring
  - Data challenges
- Petra-IV
  - ~4PB/day to tape
- EuXFEL
  - Data life-cycle

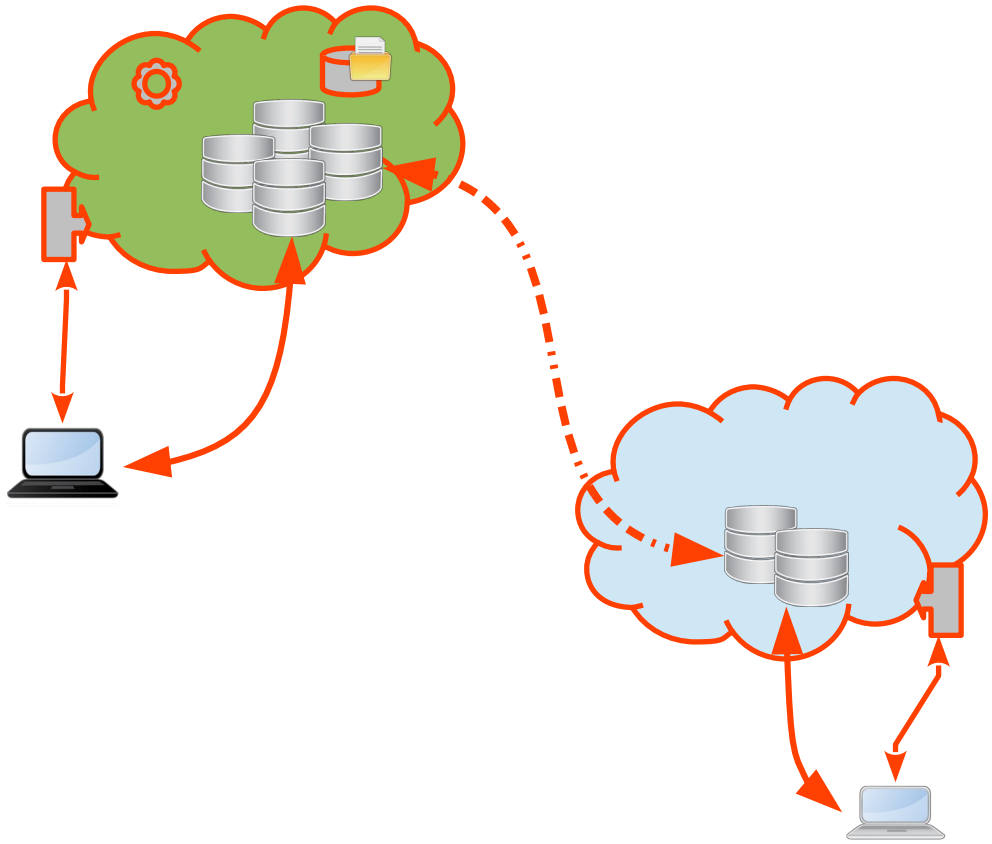
The Shifting German WLCG Model

- University Tier-2 centres are being phased out
- Gradual replacement towards the HL-LHC
- Several large German HPC sites (NHR) will take over the CPU share and increasing CPU pledges
- Helmholtz sites DESY/KIT will take over the storage shares
- Process will started in 2025
  - Annual ramp down of 20% of the University shares
  - Increase of the storage pledges taken over by DESY&KIT



Data-lake Déjà vu…

- Federated deployments
  - TLS enhances interconnect
  - Remote deployments
  - Remote monitoring
  - Performance evaluation
  - Traffic shaping



Development Directions

Scale-out

Namespace scalability for billions of files.
Increasing number of pools (SW/HW).
Distributed architecture for petascale sites.

BULK Operations

Continued performance improvements to bulk service.
HA mode stability.
Expanded API coverage for large-scale operations.

QoS Evolution

Policy-driven lifecycle for complex multi-copy, multi-media scenarios.
Better integration with HEP and photon science workflows.

Token-Based Auth

Full WLCG JWT token support, OIDC integration, elimination of X509 dependency for modern deployments.

Analysis Facility Support

Better POSIX compliance.
HPC workload support (DDoS protection).
NFS read-delegations for interactive analysis environments.

Tape Integration

Broader CTA adoption, improved HSM connector stability.
Tape REST API as full SRM replacement across all Tier-1s.

ErUM-Data FUSE Project

FUSE
Federated Unified Storage Environment

Gefördert durch:

Bundesministerium
für Forschung, Technologie
und Raumfahrt

- WP1: “federated dCache” with central services hosted at DESY.
  - Task Area 1/2: integration of semi-permanent sites
  - Task Area 3: integration of opportunistic sites
  - Task Areas 4: monitoring and self-healing

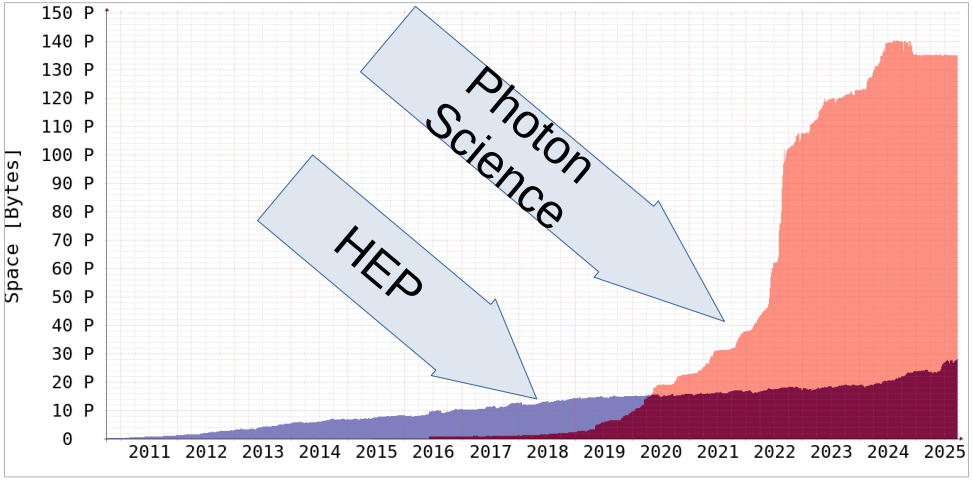
- WP2: "Caching Services and Cloud Storage Integration"
  - Task Area 1: Orchestration, monitoring and feature evaluation of the XCache proxy server
  - Task Area 2: S3 Frontend for POSIX-compliant storage

Large-scale Deployments Today

- EuXFEL
  - Total on DISK capacity ~120 PB
  - ~400 physical hosts (~1000 dCache pools)
  - 20-40 GB/s ingest
- PETRA-III
  - Two tape copies, different media type
- US-CMS
  - 1.2 B files
  - 300K directories
  - 80 PB disk capacity
  - 800 TB HSM transfers / day

- ATLAS
  - dir/file → 1/3
- NextCloud
  - File lifetime < 1s

Data on DISK (DESY)

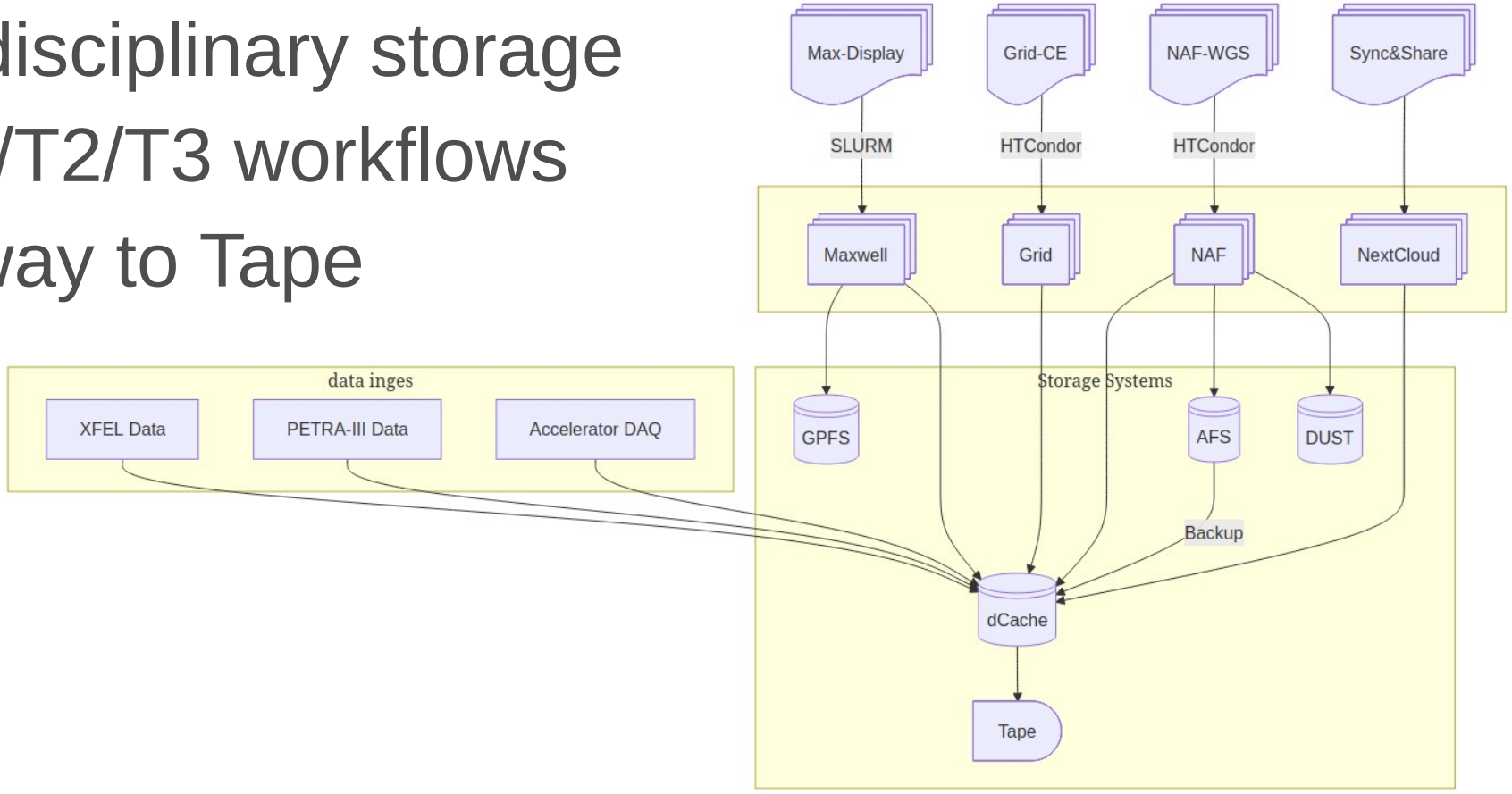


Large Instance Issues

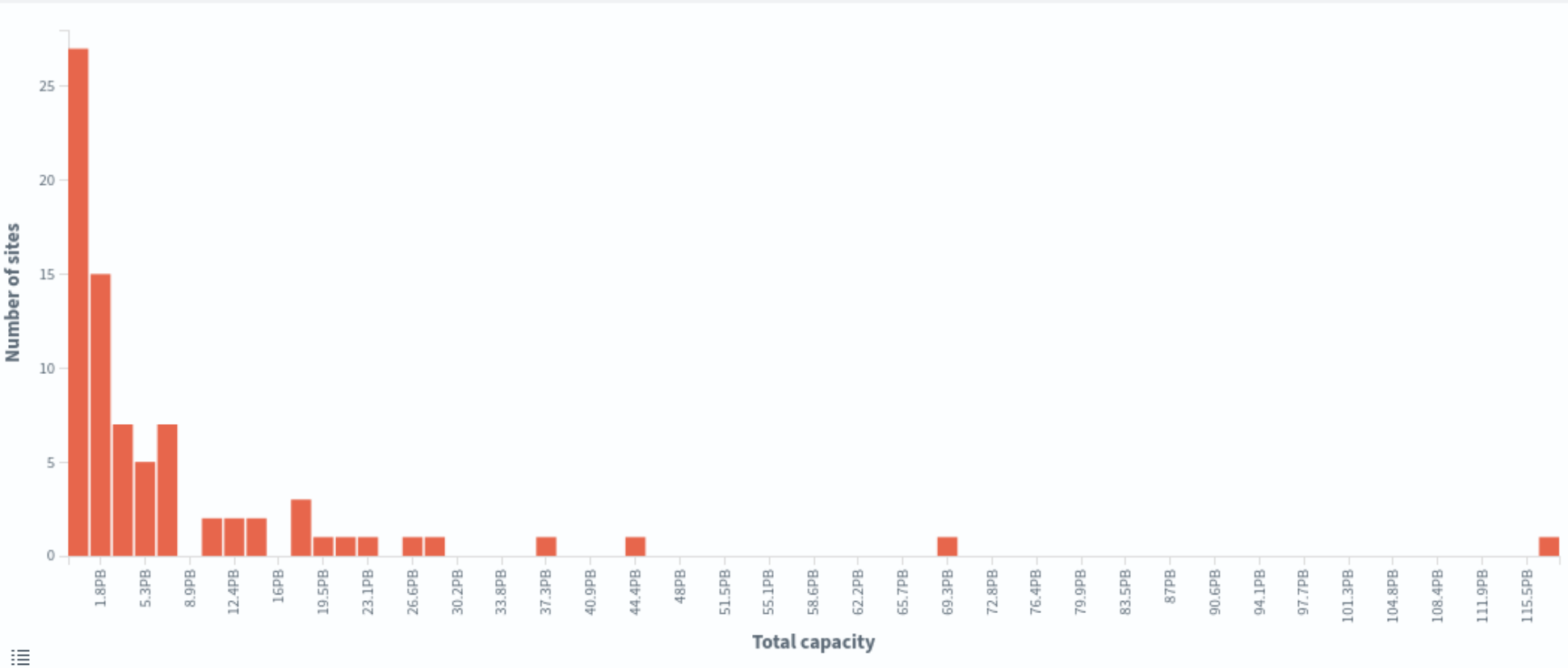
- Operational overhead
  - Monitoring
  - Installation
- Large HSM request queues
  - In-dCache HSM driver
- Single node failure
  - Replica
  - CEPH
  - Cluster filesystems
- I/O starvation (DDoS)
  - On-demand replication
  - Zero-copy
  - Direct-IO
- Long rebuild times
  - ZFS+dRAID
  - EC

Use Cases (DESY)

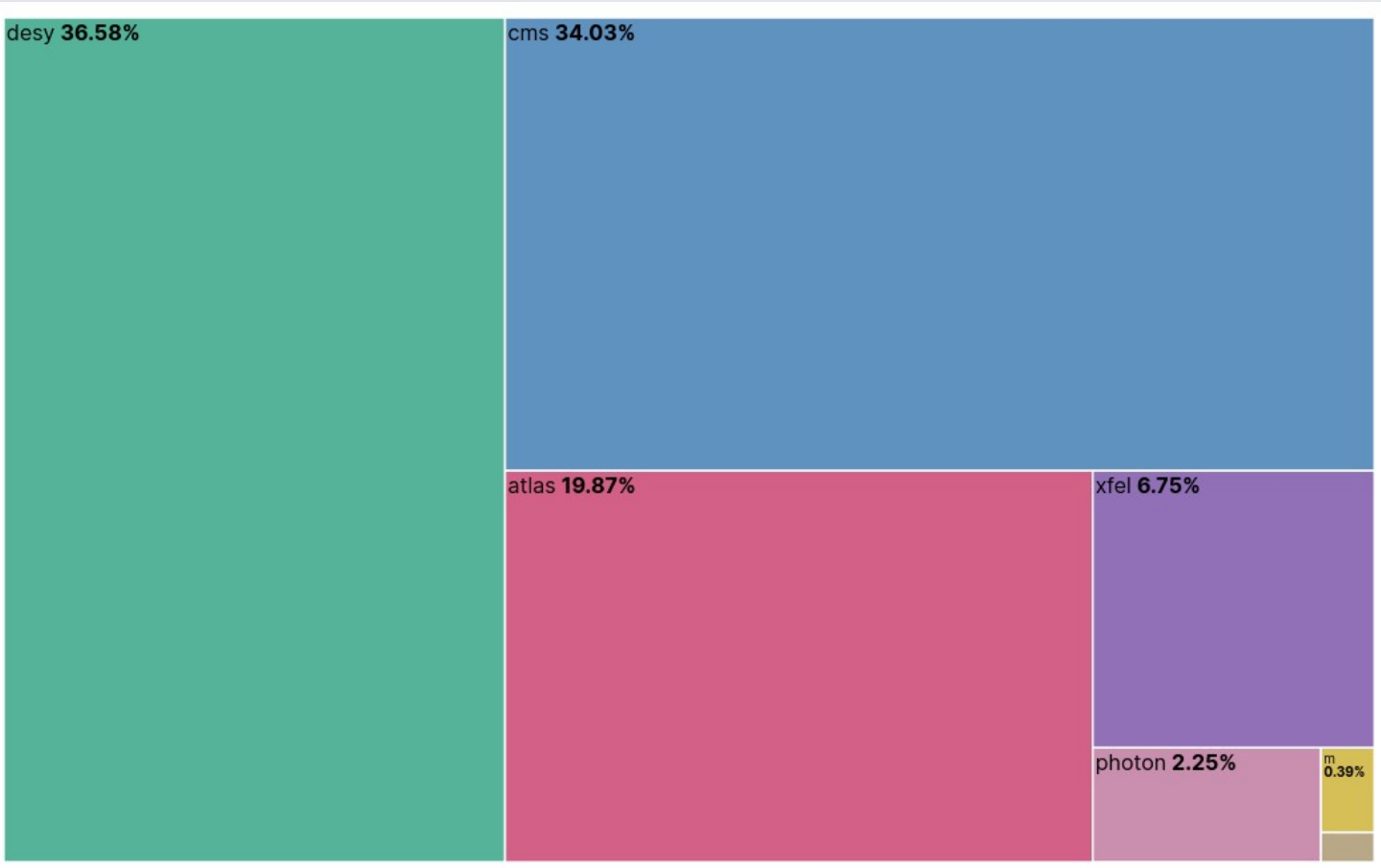
- Inter-disciplinary storage
- T0/T1/T2/T3 workflows
- Gateway to Tape



Multi-Namespace Motivation

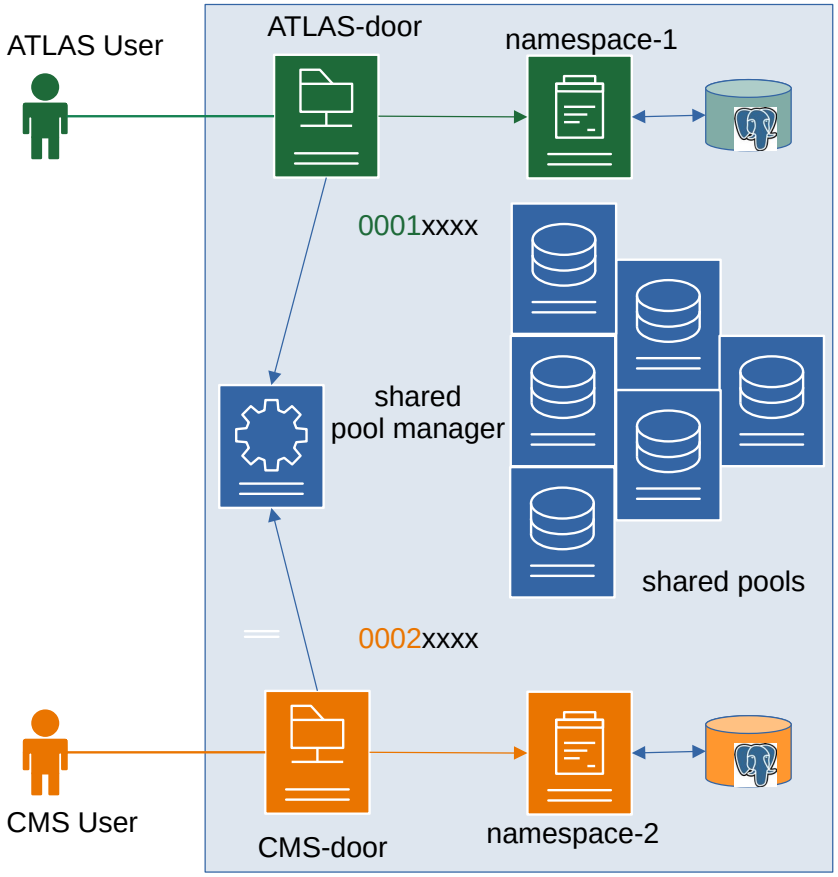


Large ≠ Busy

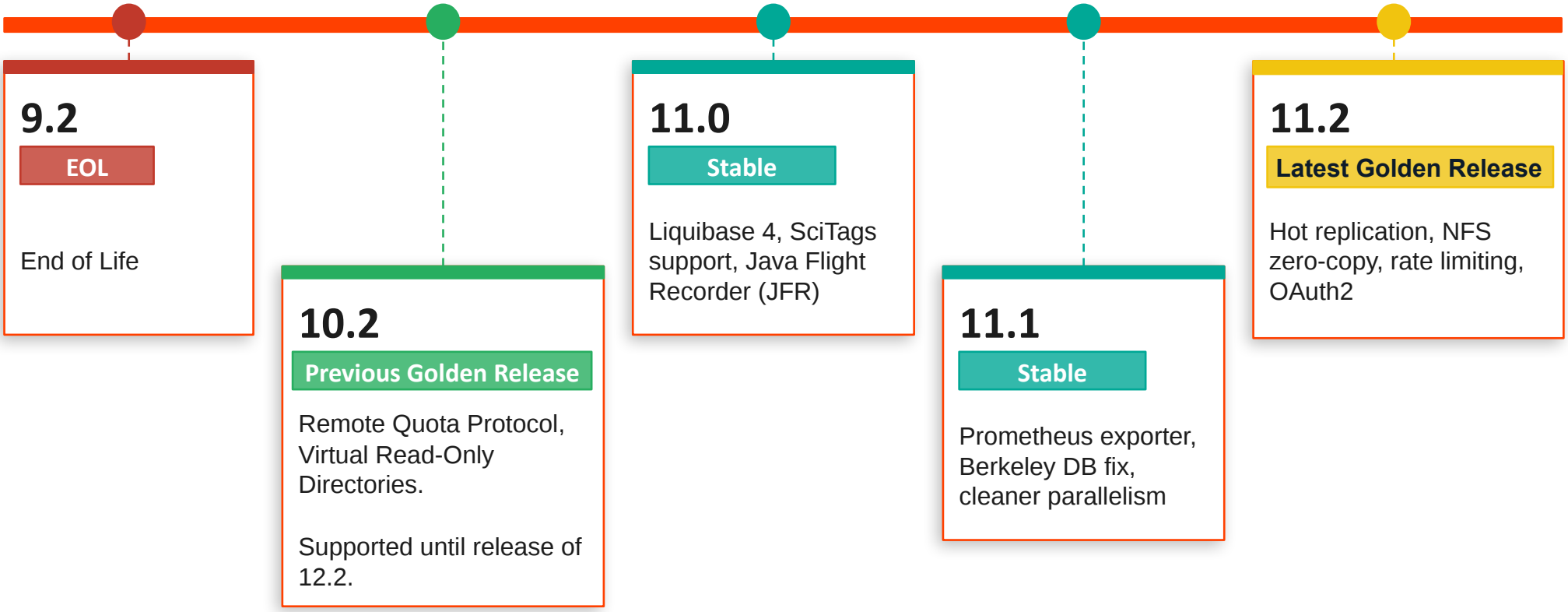


Multi-namespace Implementation

- Each door talks to a specific namespace
- PnfsID generated by namespace include filesystem ID
- Pool and PoolManager use PnfsIDs as-is
- Based on filesystem ID pool register files at corresponding PnfsManager
- Admin friendly deployment is in development



Release Timeline



Java 17 minimum · Java 21 supported · Java 25 planned in 12.2

Highlights: What's New in 11.2

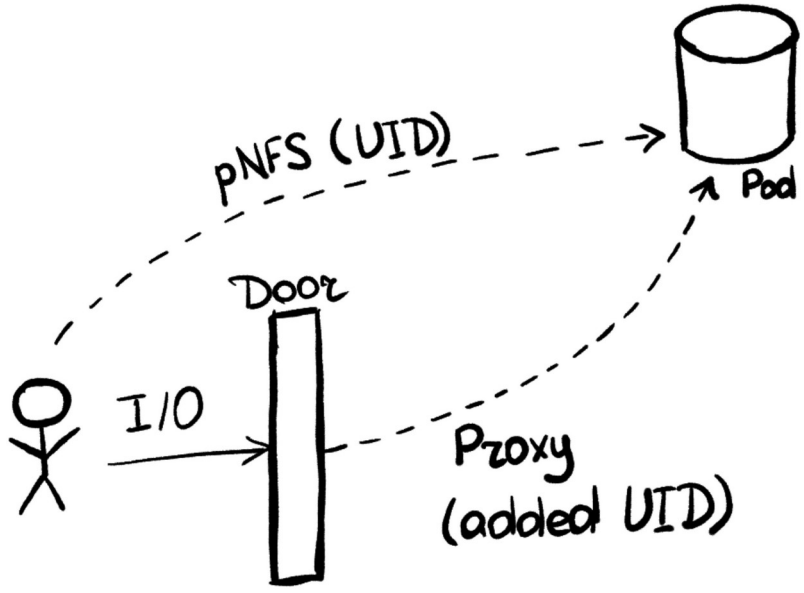
- NFSv4.1 Read Delegations
- NFS Zero-Copy I/O
- Hot File Replication
- Prometheus Exporter
- Non-Blocking Kafka
- Java 21 Support

NFS Improvements

Proxy-IO Credential Fix

NFS door now passes **original client credentials** to pools instead of **door credentials**.

Fixes permission mismatches in multi-user environments.



Remote Quota (rquota over UDP)

NFS door exposes **user/group quota info via rquota protocol**.

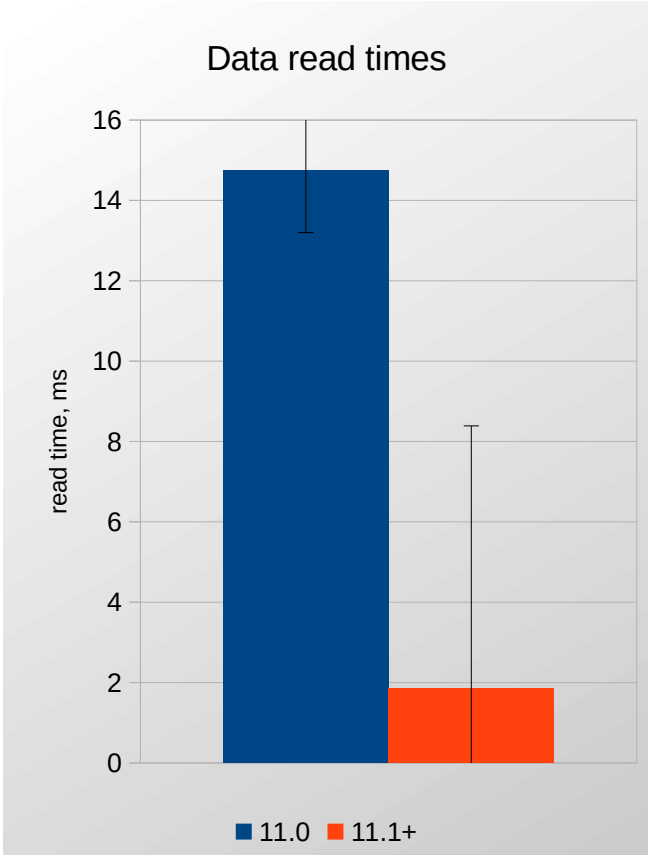
Configure port with nfs.net.rquota.port.

NFS: Read Delegations

NFSv4.1 Read Delegations

[11.1]

- Server delegates OPEN → CLOSE lifecycle to **client**
- Client handles **repeated opens locally**, no server round-trips
- Billing records reflect **delegation periods** (not individual open/close cycles)
- **nfsv4_1_files** layout type removed, was required to enable delegations
- NFS movers may remain **active longer** while clients retain delegations
- Dramatically **reduces latency** for HPC workloads



NFS: Zero-Copy I/O

Zero-Copy I/O for NFS READ

[11.2]

Before (with copy):

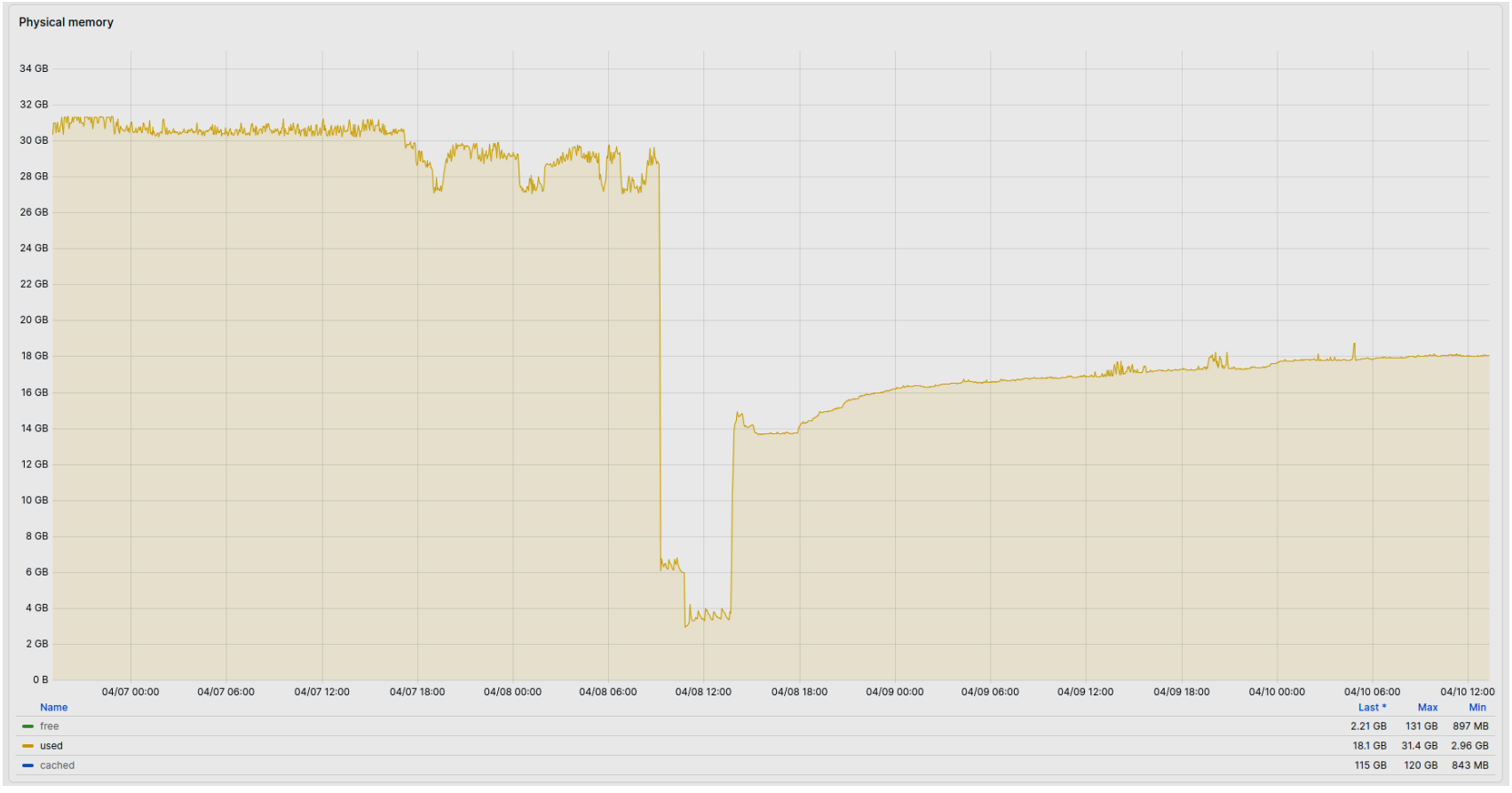
Disk → Kernel buffer → JVM heap → Kernel buffer → Network
`(extra copy in JVM heap = latency + GC pressure)`

After (zero-copy):

`Disk → Network`
`(JVM heap bypass = up to 20% throughput gain)`

- NFS mover **eliminates memory copies** during READ operations.
- Delivers up to **20% throughput** improvement for data-intensive workloads.
- Most impactful for sequential large-file reads
- Reduces GC pressure in JVM, improves overall pool stability
- Combined with read-delegations: significant HPC latency reduction
- **No configuration required**, enabled by default in 11.2

Memory Usage 10.2 vs 11.2



Pool Improvements

[11.0]

Parallel Multi-Checksum

Pools calculate multiple checksums simultaneously on HTTP upload or TPC-pull if requested by client.

No configuration needed.

[11.0]

Percentage-Based Disk Space

set max diskspace 90% — heterogeneous pool configs simplified.

Pools **auto-shrink** max space if filesystem shrinks.

[11.0]

Migration: wait/limit Strategies

The migration module can now spread files to a **desired number of pools**.

If insufficient pools are online, strategies available: wait and limit.

[11.1]

Berkeley DB Open-File Fix

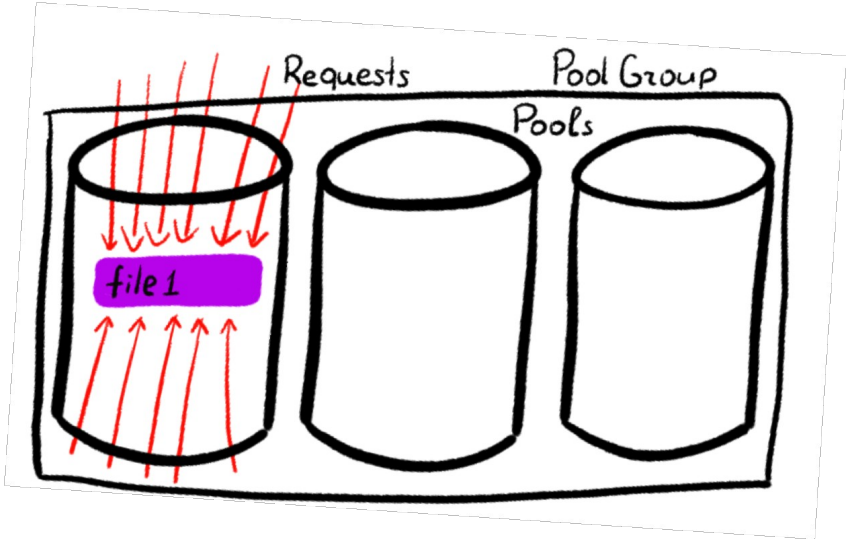
Open-file limit raised from **100** → **512**.
Can dramatically improve startup time.

No action required.

Pool: Hot File Replication Deep-Dive

How It Works

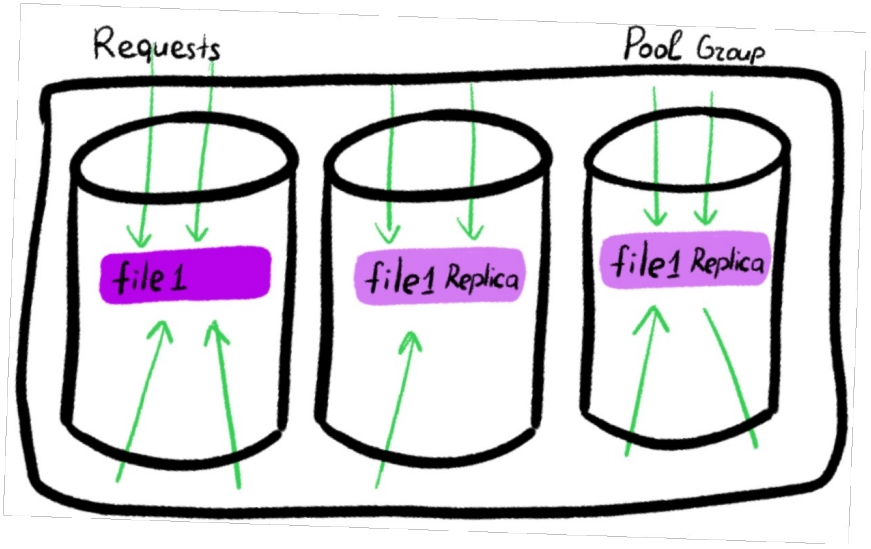
- Pool tracks concurrent in-flight transfers per file
- When transfers exceed configurable threshold → **migration triggered**
- New replica created on **another pool** in the same pool group
- Subsequent requests **distributed across replicas**
- Particularly valuable for: calibration files, shared reference data, popular datasets, libraries
- Replicas are managed by **normal pool lifecycle** (LRU eviction)



Pool: Hot File Replication Deep-Dive

Use Cases

- HEP: shared calibration constants, libs
- Analysis: frequently-read reference data
- XFEL: hot raw data during beam periods
- CMS/ATLAS: common alignment files



Enable

```
pool.hotfile.monitoring.enable=true
```

Tune via admin:

```
\s <pool> hotfile
\s <pool> hotfile set threshold 3
```

Info commands:

```
hotfile show
info -a
```

QoS — Dynamic Replica Reduction

Quality of Service Typical Requirements

HEP (typical Tier-1)

- Single copy (tape or disk)
- Tape REST API replacing SRM

Photon Science (DESY PETRA)

- 2 tape copies
- Different media types (Jaguar + LTO)

XFEL

- 2 media copies: DISK+TAPE → TAPE+TAPE
- Time-based lifecycle (DISK+HSM, then 2×HSM)

NextCloud / Multi-User

- 2 disk copies + tape
- Custom labeling and metadata policies

Disabled dynamic replica reduction [11.0]

A race condition was observed between PIN and UNPIN operations, which might be sent and processed in different order because of concurrency.

Expected: PIN → UNPIN (file is safe)
Risk: UNPIN → PIN (file may be lost)

The reduction of PINned replicas on migration event is skipped now. Then handled by Scanner on the next run.

gPlazma Improvements

[10.2]

Python Scripting gplazma2-pyscript

Custom **auth/map plugins via Python** without Java compilation.

Dynamically loaded at runtime.

[10.2, 11.0]

Consistent RolePrincipal

Sets RolePrincipal whenever **role attribute is present**.

Fixes inconsistent admin access enforcement.

[11.1]

VO: atlas
role: atlas_admin, *cms_admin*

VOMs Plugin FQAN match

Prevents unauthorized access by ensuring a **user's role** strictly **matches their VO name**.

This closes a security loophole that allowed (malicious) servers to grant roles they didn't actually own.

[11.2]

LDAP Plugin Flexible Search DN

More **flexible search base DN configurations** for complex LDAP directory structures.

Reduces need for custom workarounds.

Frontend: Request Rate Limiter

Request Rate Limiter

Protects against **brute-force attacks**.

Rate limiting on authentication errors by default.

Per-source-IP configuration available.

Main triggers:

- Too many auth attempts
- Too many requests
- Too many errors

New properties* are introduced (See full list at backup slides):

frontend/webdav.limits.max-blocked-clients – The maximum number of clients that can be blocked at any one time.

frontend/webdav.limits.rate.overall – The maximum number of requests per second allowed from all clients.

frontend/webdav.limits.rate.per-client.fractions – The maximum share of requests a single client is allowed.

frontend/webdav.limits.error.max-allowed – The maximum number of errors allowed within a time window before client is blocked.

Frontend: Metalink Format

Metalink Format Support

Functionality: Generates a Metalink description for all files within a target directory.

Accessible via HTTP content-negotiation or a specific "Link" header in directory descriptions.

Use Case: Originally designed to demonstrate data access via DOIs for datasets hosted on dCache.

Current Workflow: aria2c https://webdav-door.example.org/path/to/dir/ → Client fetches Metalink → All files in directory are downloaded.

Future Opportunities (Requires Development):

Label-Based Downloads: Integrating with filesystem labels to allow users to download all files associated with a specific tag/label via Metalink.

Monitoring Improvements

Prometheus Exporter [11.1]

Integrated Prometheus exporter **exposes JVM memory**, thread counts, and open file handles.

Enable and configure endpoint:

```
dcache.enable.prometheus.exporter=true
dcache.enable.prometheus.exporter
   .endpoint=localhost:9876
```

Expose via: GET http://localhost:9876/metrics

Let us know which metrics you use and want to include?

Java Flight Recorder (JFR) [9.1, 11.0]

JFR started **via admin interface** for low-overhead performance profiling.
Captures JVM performance stats, CPU load, memory consumption, file descriptor leaks.

```
# Start recording
System@thecell jfr start

# Stop and download
System@thecell jfr stop
```

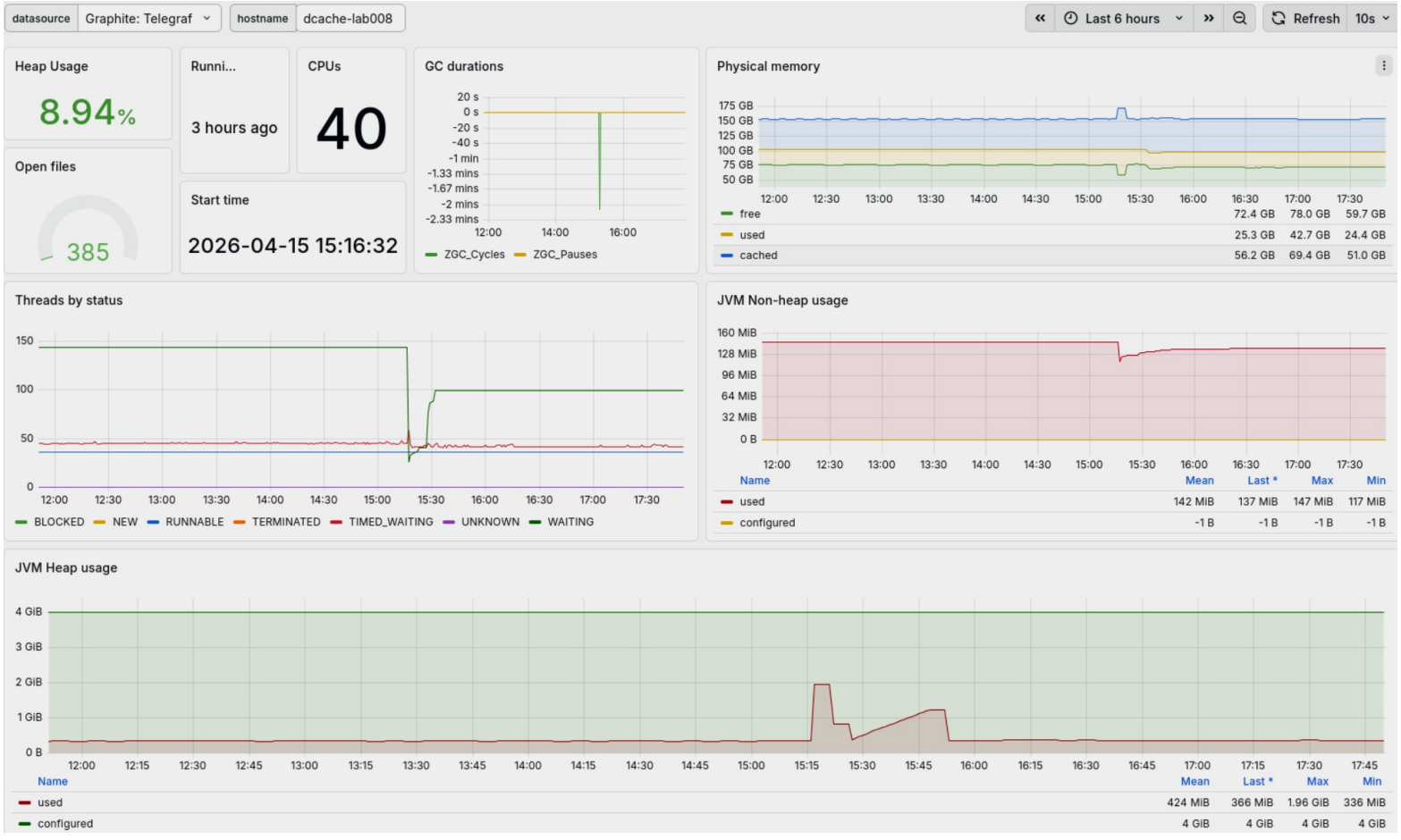
perf top

Enables observability from dCache into kernel. View Java stack traces via Linux perf.

JFR Door Message Timing [11.0]

```
org.dcache.door.Messages {
  startTime   = 09:45:35.672
  duration    = 37.8 ms
  message     = "PnfsGetFileAttributes"
  Source      = "NFS-nairi@dCacheDomain"
  Dest        = "PnfsManager"
}
```

Prometheus Stats



Billing Improvements

Non-Blocking Kafka Logging [11.2]

Kafka producers added to **the Billing** service. You can configure it to work only in billing **and/or** in door/pool.

Previously Kafka outages would block door/pool threads.

```
dcache.enable.kafka = false
dcache.billing.enable.kafka = false
```

Billing JSON transferTag [11.2]

Billing JSON records now include **transferTag** from XRootD and HTTP clients.

Enables **per-experiment metadata** tracking (experiment ID, workflow ID).

JSON billing logs also available with `billing.format.json=true`.

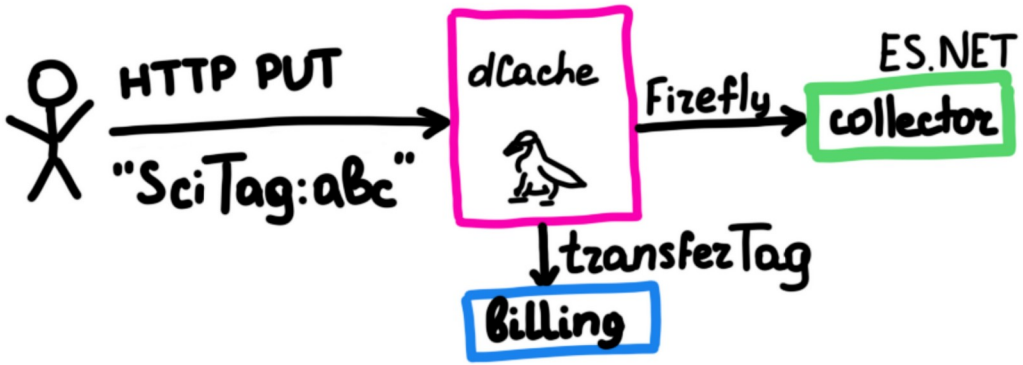
SciTags & Firefly

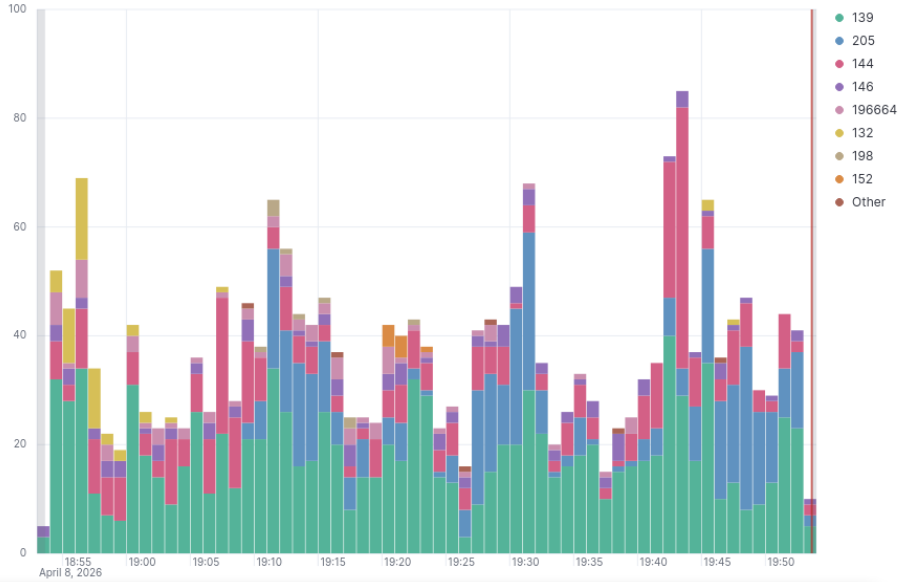
SciTags / Firefly [10.2, 11.0]

SciTag standard for **monitoring data movement** between sites.

dCache sends UDP Firefly packets on transfer begin/end.

Propagates client-supplied SciTag to network monitoring infrastructure.






```
pool.enable.firefly=true
pool.firefly.destination=<endpoint>
pool.firefly.excludes=a.b.c.d/16
pool.firefly.vo-mapping=atlas:2,cms:3,lsst:9
pool.firefly.storage-statistics=false
```

Tape: HSM Cleaner

Cleaner-HSM Parallel Deletion [11.1]

Delete files from tape in **parallel across multiple pools** instead of sequentially through one pool.

Decrease cleaner-hsm.period to **increase the number of parallel cleaning pools.**

```
cleaner-hsm.period=60
cleaner-hsm.period.unit=SECONDS
```

HSM-Only Files: Access Denied [10.2]

If a file has QoS policy HSM-only, dCache rejects READ access even if the file is still cached on disk. This enforces intended lifecycle policy.

→ Stage file to disk first, or adjust QoS policy to allow temporary caching before reading

Breaking change from 10.2

stage-allowed Defaults to yes

Pool Manager partitions now allow **staging by default**. Add stage-allowed=no to partition definitions if you need the old blocking behavior.

Retry on Pool-Up During Stage

When pool comes back online during active stage, PoolManager **retries to use pool copy**.
Clients get data from whichever source (tape or pool) is ready first.

Legacy Farewell

SRM — Are We Done?

Tier-2 sites (no tape): Should NOT run SRM

→ Use WebDAV / XRootD / NFS instead

Tier-1 sites (tape): Should NOT run SRM

→ Use Tape REST API (WLCG standard)

SRM planned for full removal in 12.0.

If we find out that some sites still need/use SRM, support of 11.2 will be extended for longer.

Legacy gPlazma Modules Being Removed

The following legacy gPlazma modules are being removed in upcoming releases. Convertation is trivial. Examples in docs. Migrate now:

Remove	→	Migrate to
`argus`	→	`multimap`
`nis`	→	`multimap or ldap`
`nsswitch`	→	`multimap or ldap`
`gridmapfile`	→	`multimap`
`vorolemap`	→	`multimap`
`kpwd`	→	`multimap`

Migrate before next major dCache upgrade!

Breaking Changes

CRITICAL

Full Instance Upgrade Required [11.0+]

Release 11.0 removed backward compatibility with pre-11.0 message encoding. Stop ALL services, upgrade ALL packages, run DB migrations, restart all.

Irreversible Liquibase 4 Migration [11.0+]

Liquibase 4.29.2 migrations cannot be rolled back. **Back up all databases** before running dCache database update.

REQUIRED

Remove nfsv4_1_files from Exports [11.1+]

nfsv4_1_files layout type removed entirely. Must remove from NFS export files before upgrading or NFS doors will fail to start.

Remove pool.enable.hsm-flag [11.1]

Property obsolete and fully removed. **Remove from pool configuration** files to avoid startup errors.

XRootD prepare → Unsupported [11.2]

xrdfs prepare -s no longer works. Migrate tape staging workflows to WLCG Tape REST API (POST /api/v1/tape/stage) before upgrading.

Legacy Admin Shell Removed [11.1]

cd PnfsManager no longer works. Update all scripts to **use \c** PnfsManager syntax or direct \s commands.

OPTIONAL

File Flags Commands Removed [11.1]

set/get/delete flags removed from all components. Update scripts. Storage info from directory tags only, table empty. Optionally: TRUNCATE t_level_2; TRUNCATE t_storageinfo;

SQL SHOW [11.1]

Command is not working without update due to checksum mismatches.
Should start working immediately after updating to 11.2.

Technical Directions

Where are moving right now:

- **Code Coverage** – was **lost** while migrating from Jenkins to GitLab CI, now we are **implementing it again** using JaCoCo.
  Removal of dead code or adding tests to uncovered classes.

JaCoCo Coverage Report

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed
org.dcache.ftp.door		25%		17%	813	1,000	2,134
diskCacheV111.poolManager		50%		44%	926	1,700	2,261
org.dcache.util		64%		61%	872	2,468	1,767
diskCacheV111.services		2%		0%	503	518	1,884
org.dcache.pool.classic		38%		24%	771	1,192	1,835
dmg.cells.nucleus		48%		38%	824	1,467	1,829
org.dcache.pool.movers		27%		18%	579	779	1,883
org.dcache.pool.migration		50%		39%	594	973	1,585
dmg.cells.services.login.user		3%		1%	485	498	1,351
diskCacheV111.doors		31%		17%	433	570	1,267
org.dcache.chimera		66%		58%	546	1,288	1,214
org.dcache.ftp.client		22%		27%	468	623	1,377
org.dcache.auth		51%		42%	568	1,053	1,172
org.dcache.chimera.namespace		42%		41%	411	666	1,093
diskCacheV111.namespace		43%		30%	390	614	1,120
diskCacheV111.services.space		38%		23%	437	648	974
org.dcache.webdav		41%		26%	435	710	1,099
dmg.cells.services.login		64%		26%	337	511	833
org.dcache.services.bulk		15%		5%	338	469	920
org.dcache.chimera.nfsv41.door		48%		31%	363	608	908
diskCacheV111.services.web		14%		12%	196	232	666
org.dcache.webdav.transfer		6%		1%	284	313	850
org.dcache.xrootd.door		39%		20%	296	448	835

- **Anomaly Detection** – Transitioning from manual to **automated monitoring**.
  - Long Short-Term Memory recurrent neural networks to learn **usage patterns from billing** files and automatically **flag unexpected deviations**.
  - A pipeline developed using Apache Spark for large-scale data retrieval and PyTorch for training models that compare **predicted activity** against actual system behavior.

Work in Progress

- Enhancing Codeflow Authorization
- Enhancing Hot File Replication
- Federated deployments
- HPC Workflows

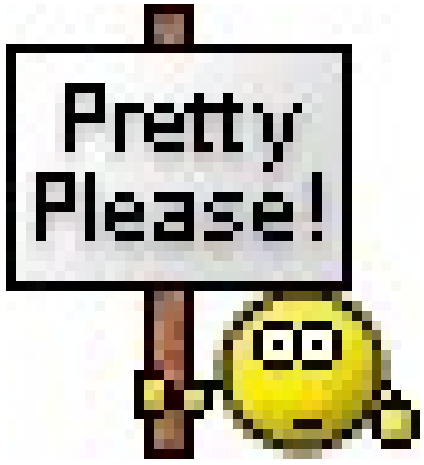
Documentation

- Release Notes 11.2
- The Ultimate Golden Release Upgrade Guide X - How to get from 10.2 to 11.2
- The Book 11.2
- User Guide 11.2

Faster and more documentation, thanks to LLMs.

Call to Action

- You can contribute with ...
  - Code
  - Configuration
  - Testing
  - HW setup
  - Knowledge
  - Hosting Events
- You can make dCache visible with …
  - Sharing your use case
  - Demonstrate dCache use in various events



Thank you!

support@dcache.org

User request tracking system. Accessible by all team members. Best way to get help from devs.

security@dcache.org

Report security issues or incidents. Restricted access — selected team members only.

user-forum@dcache.org

Mailing list for sysadmins. Self-help group — share experiences, ask for advice.

dev@dcache.org

Developers shared mailbox. Contact point for developers, not for support.

Questioner

Please take part in our small questioner regarding SRM usage and gPlazma modules usage.

https://forms.gle/iEa4r8bcc7obHnpK8

Frontend: Request Rate Limiter

Request Rate Limiter

Protects against brute-force attacks. Rate limiting on authentication errors by default. Per-source-IP configuration available.

New properties are introduced:

frontend.limits.max-blocked-clients – The maximum number of clients that can be blocked at any one time. Preferably a power of 2.

frontend.limits.rate.overall – The maximum number of requests per second allowed from all clients.

frontend.limits.rate.per-client.fractions – The maximum share of requests a single client is allowed, expressed as a percentage of the total requests across all clients.

frontend.limits.error.max-allowed – The maximum number of errors allowed within a time window before client is blocked.

frontend.limits.error.block.window.time(.units)

frontend.limits.rate.per-client.block.window.time(.units)

frontend.limits.blocked-clients.idle-time(.units)