CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

STORAGE: A story about the lifecycle of a project

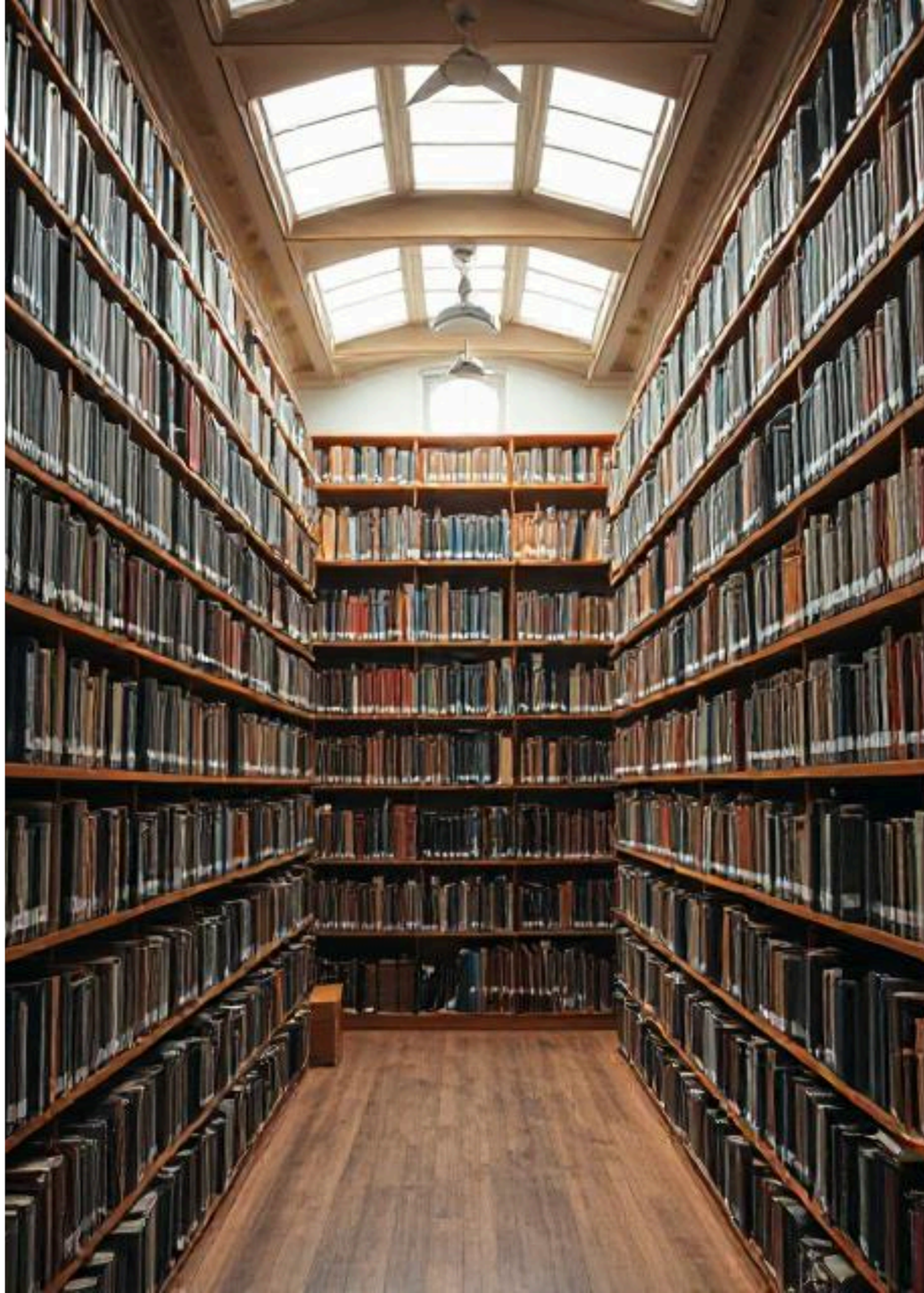
UserLab Day 2025

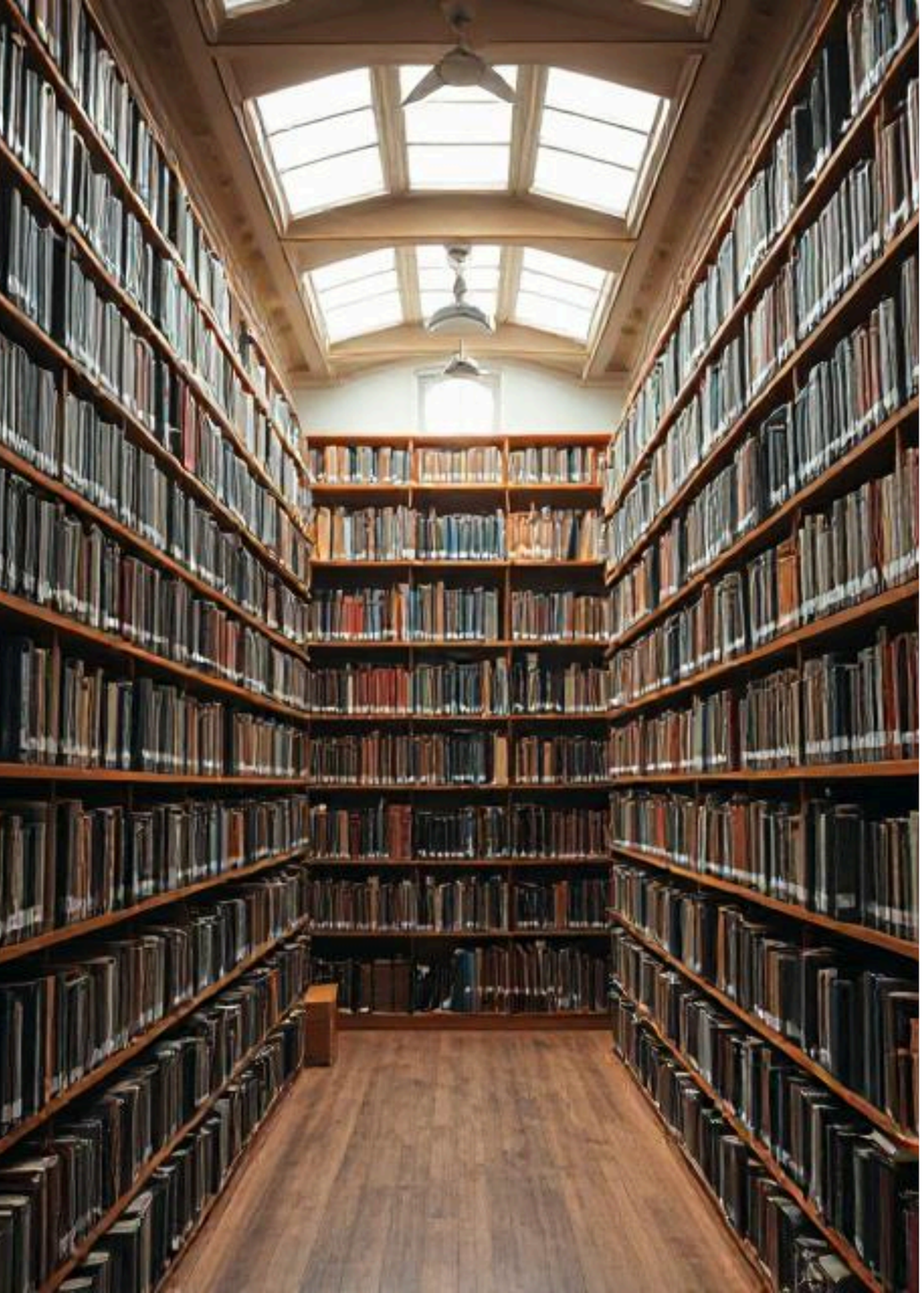
Dr. Fawzi Mohamed & Dr. Matthias Kraushaar

Storage

come for the compute, stay for the storage

- all compute needs inputs and generates outputs
- those normally come from and go to storage
- without storage no compute
- often neglected





Storage: Not Just Space – A System to Understand

- Storage is not a passive resource. It's a system with architecture, behaviour, and performance characteristics that directly affect your scientific workflow.
  - **Not all storage is equal**— different tiers serve different purposes.
  - **Performance varies**— actual transfer rates depend on many factors.
  - **Efficiency requires understanding**— how files are stored, accessed, and moved matters.
  - **Planning is essential**— just like compute, storage needs to be budgeted and tested.
- **To use storage efficiently, you need to understand how it works — not just how much you have.**

Storage tiers

- 🔥 **Hot Storage**
  - High-performance, short-term
  - This is *$SCRATCH* - CAPSTOR/IOPSSTOR

➢ **Here you can actively tune I/O performance through system-level features!**

- 🌡️ **Warm Storage**
  - Medium-performance, mid-term
  - This is *$STORE*, but also *$SCRATCH* – CAPSTOR/IOPSSTOR

➢ **Workflow optimization is key here!**

- ❄️ **Cold Storage** (not by default provided by CSCS)
  - Low-performance, long-term storage
  - *Tape Library, Ceph (S3)*

➢ **Plan early for data reduction and transfer logistics!**

Mapping storage tiers to a typical HPC workflow

1. Project Initialization
   ($HOME is not part of the storage tiers!)
2. Data Preparation
   $STORE/$SCRATCH
3. Simulation execution
   $SCRATCH
4. Post-processing & Analysis
   $STORE/$SCRATCH
5. Archiving & Sharing

- Storing scripts, configs, small persistent files
- Staging data, pre-processing, and simulation setup
- High-performance I/O during compute jobs
- Analyze results and prepare data for archiving
- Transfer data to new project/long-term storage

Storage Kinds

docs.cscs.ch/storage/filesystems/

compute volant, storage manent

Storage Kinds

docs.cscs.ch/storage/filesystems/

compute volant, storage manent

... unless it is scratch

Storage Kinds

docs.cscs.ch/storage/filesystems/
compute volant, storage manent
... unless it is scratch

- **File Systems**

  There are three *types* of file system that are provided on Alps clusters:

	backups	snapshot	cleanup	access
Home `/user/$USER`	yes	yes	no	user
Scratch `$SCRATCH`	no	no	yes	user
Store `/capstor/store/cscs/<org>/<project>`	yes	no	no	project

Storage Kinds...

	Capstor	Iopsstor	VAST
Model	HPE ClusterStor E1000D	HPE ClusterStor E1000F	VAST
Type	Lustre	Lustre	NFS
Capacity	129 PB raw GridRAID	7.2 PB raw RAID 10	1 PB
Number of Drives	8,480 16 TB HDD	240 * 30 TB NVMe SSD	N/A
Read Speed	1.19 TB/s	782 GB/s	38 GB/s
Write Speed	1.09 TB/s	393 GB/s	11 GB/s
IOPs	1.5M	8.6M read, 24M write	200k read, 768k write
file create/s	374k	214k	97k

Home

docs.cscs.ch/alps/storage/#home

- per user
- relatively small (50GB, 500K inodes)
- uses VAST
- scripts,... but not large git repos

Scratch

docs.cscs.ch/alps/storage/#scratch

- per user
- for ephemeral data
- can handle many files and large files
- $SCRATCH points to you personal space
- cleanup policy: files not touched by longer than 30 days get deleted, less if filesystem is too full
- different platform have different scratch
  `/capstor/scratch/cscs/$USER` vs
  `/iopstor/scratch/cscs/$USER`

Store

docs.cscs.ch/alps/storage/#store

- **Project** related
- has backup
- place to keep inputs and outputs
- lustre based: better fewer larger files
  - consider to archive/compress tar/squashfs
- how to get it?

Project lifetime

- write proposal
- submit proposal
- proposal accepted

- project start
- computation
- computational budget expired (latest at project end)
- backup disabled (project end)
- access blocked (end + 1 month)
- data deleted (end + 3 months)

Requesting storage

- quantify usage and request it in the proposal
- project data allocation is confirmed at acceptance
- exceptions can be discussed via tickets

Other Storage Kinds

- Archive/tape
- Object Storage
- Long-Term Storage
- In memory storage
- squashfs

How to use storage

- storage management can get out of hand
- permissions: use newgrp if you are/were member of multiple projects
- you can use acl if required docs.cscs.ch/guides/storage/#sharing-files-and-data
- many small files docs.cscs.ch/guides/storage/#many-small-files-vs-hpc-file-systems
- lustre tuning docs.cscs.ch/guides/storage/#lustre-tuning

Data Transfer Realities – Understanding the Challenge

Transfer Speed: Theory vs. Reality

- **Theoretical bandwidth ≠ actual performance**
- Real-world transfer speed depends on:
  - File size and count
  - Source and destination file systems
  - Network congestion
  - Transfer method (e.g., rsync, scp, tar)
- **Data movement is part of your workflow** — plan for it!
  - Timelines needed for the migration of data sets can easily reach days or weeks

Best Practices for Internal and External Transfers

Internal Transfers (within CSCS systems)

- Use the xfer Slurm queue for moving data between project, scratch, and archive
- Supports commands like rsync, cp, mv, rm
- Allows automation and job chaining with compute jobs
  - Does **not** consume project resources
- Optimized for performance and scheduling

External Transfers (to/from CSCS)

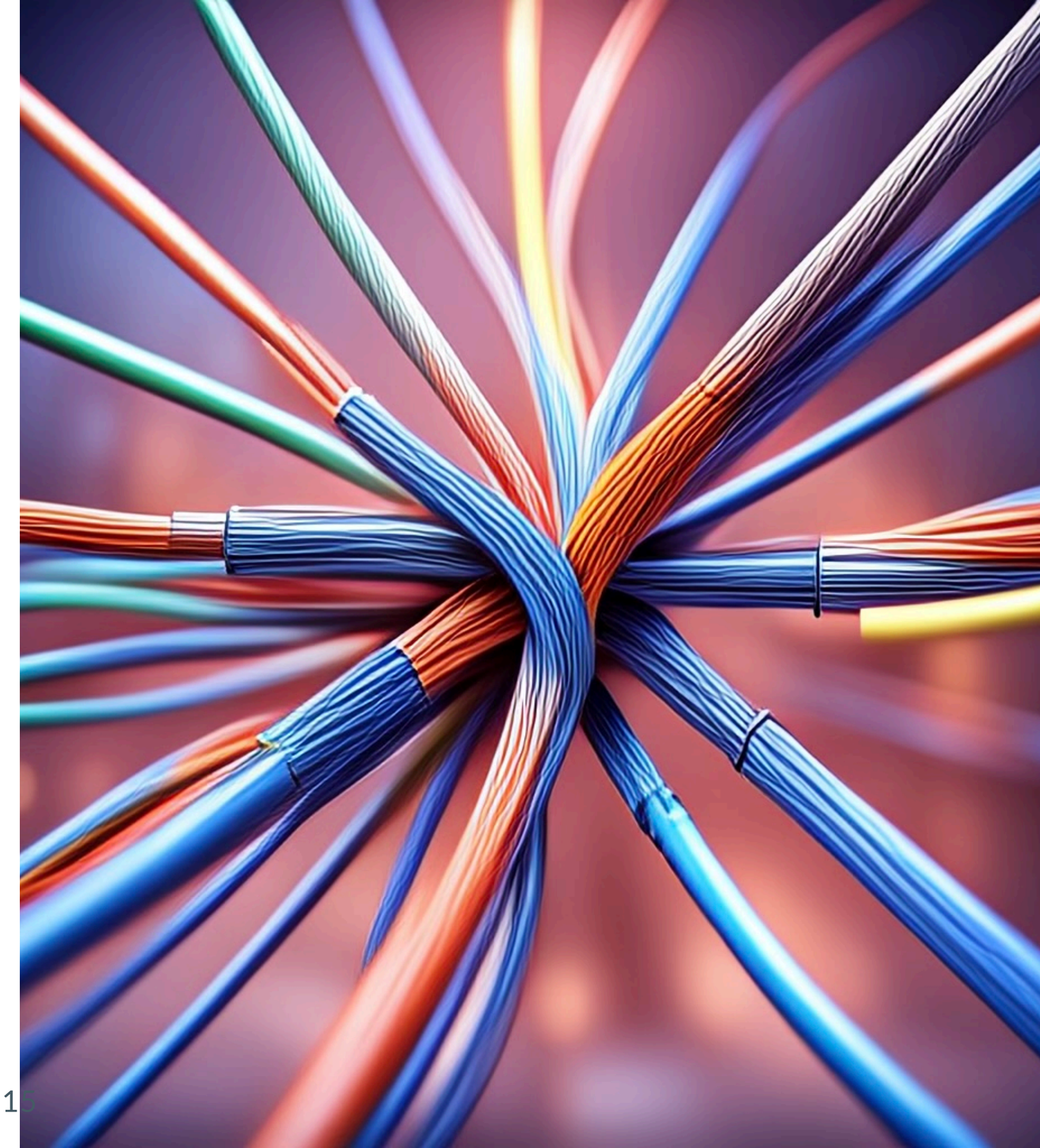
- Use **Globus** as the primary tool for secure, efficient transfers
- Requires endpoint setup and authentication
- Handles large datasets and supports restart/resume
- May be slower due to external network limits and institutional firewalls

For more details and example scripts, please check out the CSCS documentation: https://docs.cscs.ch/storage/transfer/

Internet Access on Alps

docs.cscs.ch/guides/internet-access/#communicating-with-external-services

- Alps has an execellent connection (400 GB/s)to the wider internet
- Netowrk is a shared resource
- Provider of a service might blacklist CSCS
- Ethical use is required



Thanks for your attention!