A practical introduction to the effective use of compute cluster storage space and dataset management

Maciej Pawlik, Jacek Budzowski, Joanna Pająk, Patryk Lasoń
ACC Cyfronet UST

Zakopane, 03.04.2025

A practical introduction to the effective use of compute cluster storage space and dataset management

1. Practical, i.e. interactive bash oriented
2. supercomputer,
3. data, files, storage.

- Not about data discoverability, metadata management systems, etc.
- This training contains information specific to supercomputers operated by ACC Cyfronet UST.
- Theory & practice, nothing is set in stone.

Plan:

1. Basic information
   a. Storage spaces of a cluster
   b. Some behind the scenes of a distributed storage system
   c. Typical lifecycle of a dataset
2. Demonstration of available tools
3. Hands on exercise of parallel copy

Basic information

Storage spaces available on a supercomputer

Start by consulting dedicated documentation:

- https://docs.cyfronet.pl/ares
- https://docs.cyfronet.pl/athena
- https://docs.cyfronet.pl/helios
- https://kdm.cyfronet.pl/

- above answer questions like:
  - what storage is available?
  - what are individual storage policies?

Storage spaces available on a supercomputer

- Home directories
  - env variable: $HOME
  - NFS filesystem
  - Purpose: small number of small files, not high performant
- **Group storage**
  - env variable: $PLG_GROUPS_STORAGE/<group name>
  - Lustre filesystem
  - Purpose: data living for the duration of a grant, data sharing
  - Moderate performance, not to be used for intensive IO
- **Scratch space**
  - env variable: $SCRATCH
  - Lustre filesystem
  - Purpose: data used for current computations, moderate number of large files, individual use
  - Data can be removed after 30 days of not being used
  - Highest performance, strict quota

And other storage services

- S3-like, object storage
  - https://guide.s3p.cloud.cyfronet.pl/index.html
  - Available though PLGrid grant system
  - Remote storage, http api, backups, data sharing,
- Tape
  - Glacier like services, really really long term storage

Behind the scenes of a Supercomputer's "Superstorage"

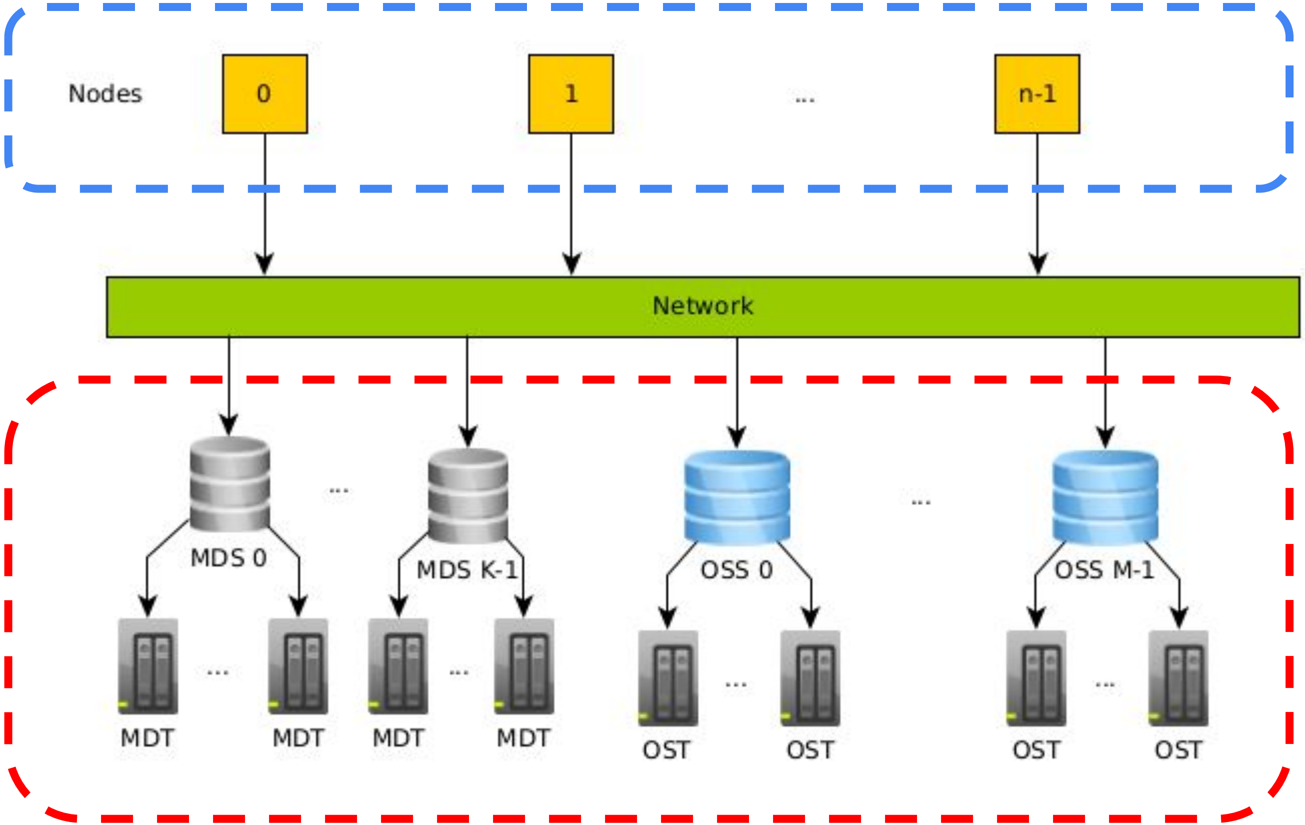


Image source: https://docs.csc.fi/computing/lustre/

Behind the scenes of a supercomputer storage cont.

- Every storage is available on all nodes
  - great, but can lead to bad practices
- Shared environment
  - consider handling files in an interactive job on a worker node
- High availability
- Backups?
  - High availability storage is not a backup
  - Please do proper backups of important data on your own
  - use external storage, some other location

- Single thread throughout is limited
- Parallelization and simultaneous file access is key to achieve high performance
  - Do multiple things at once
- Use proper storage for the task
  - Data might need to be moved around

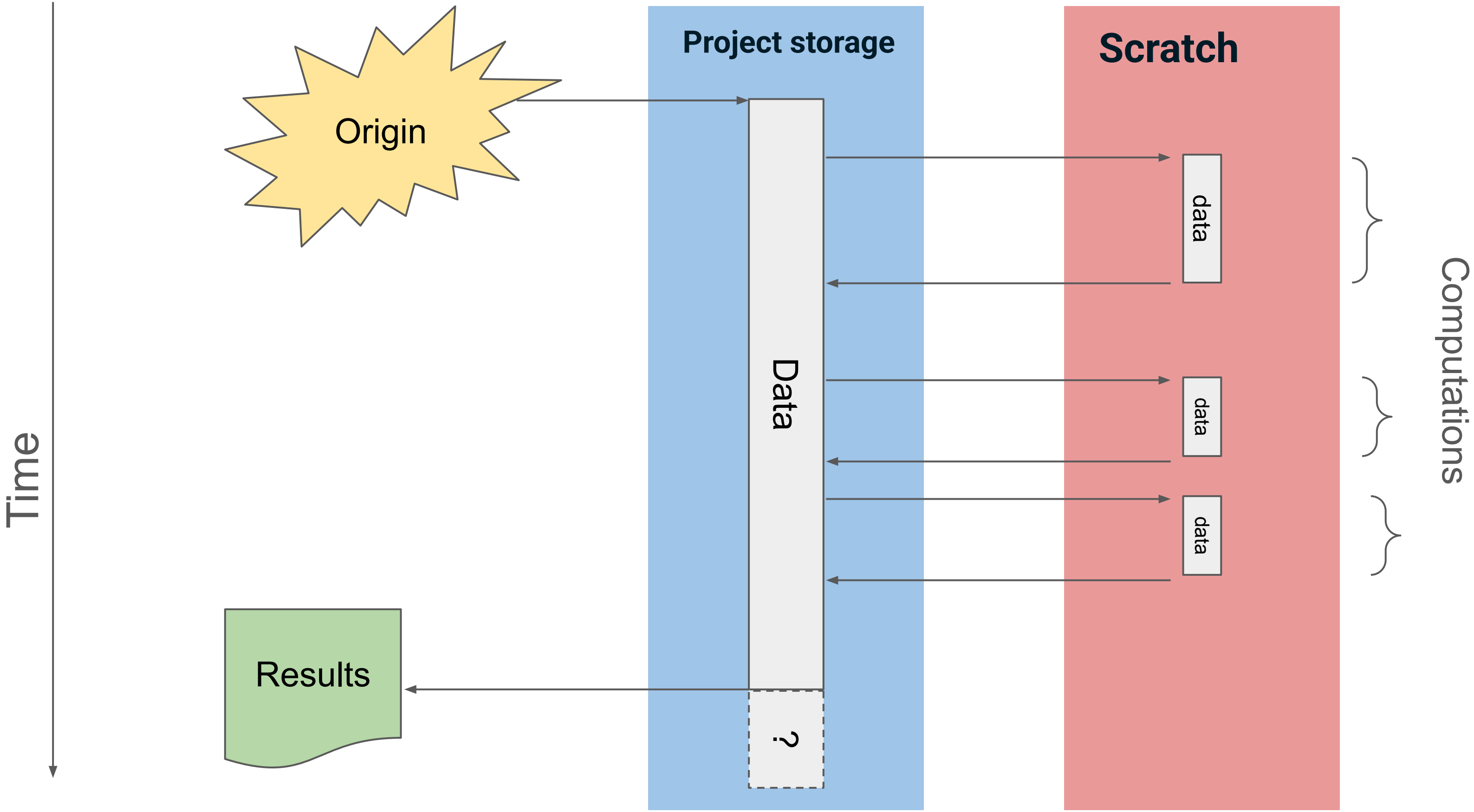
- Cleanup and removing data
  - Large amount of files can easily become overwhelming
  - Might improve performance
  - Be very careful about removing files

Behind the scenes of a supercomputer storage

Ideal scenario for lustre I/O workload:

- Sequential read/writes
- Large block size (writes/reads in range of MB)
    - What is the block size?
- DO: a number large files
    - DON’T: many small files
- Parallel file access
- Reduce to minimum:
    - metadata operations, stat, etc.
    - open, close operations
    - using locks

Typical lifecycle of a dataset



Tools

Simplicity over complexity, tools available on a cluster

- Coreutils:
  - cp, mv, rm
  - e.g. cp and mv are optimized for performance
- Generate a list of files (note passed arguments)
  - find
- File copy
  - rsync (not great, but not bad either, with the -B argument)
- Parallelization and scripting:
  - xargs
  - parallel
- Lustre tuning tools:
  - lfs getstripe
  - lfs setstripe
  - Discussed during KUKDM2019

Simplicity over complexity

General guidelines for moving data in and around the cluster:

- Avoid doing heavy IO on a login node

- Prioritize copying over moving data
- Delete after copy integrity is verified
- Be very, very cautious when using rm (and possible automation/parallelization)
  - Common pitfall is to use commands like:
    rm -rf mydata/$setone
    with $setone unset
- Don’t resort to tuning Lustre parameters too early
- If things go slow, investigate why, optimize or create a support ticket.

Hands on

Real life challenges:

- How to efficiently copy a large number of files?
  - large number of files
  - limited single thread throughput
- Solution:
  - **work in an interactive job on a worker node**
  - work in $SCRATCH directory
  - parallelize the process

- Follow along, using your account and grant from the previous training (plgtraining2024-cpu)

Some useful commands

- Download openmpi sources from: https://www.open-mpi.org/software/ompi/v5.0/
- wget https://download.open-mpi.org/release/open-mpi/v5.0/openmpi-5.0.7.tar.gz
- tar -xzf openmpi-5.0.7.tar.gz
- find openmpi-5.0.7 -type f
- time cat files.txt | parallel --will-cite -X -n100 -j8 --eta rsync -a -R /net/afscra/people/ybpawlik/kukdm2025/./{} /net/afscra/people/ybpawlik/kukdm2025/copy/

Thank you!