Deutsches Klimarechenzentrum
(German Climate Computing Centre)
DKRZ

Carsten Beyer

© Carsten Beyer

German Climate Computing Centre

Non-profit limited company since 1987

- Share-holders MPG (55%), FHH/UHH (27%), AWI (9%), Hereon (9%)
- 100+ staff at DKRZ
- 4+ staff at university research group

HLRE-4 – Levante (2022-2028)

BullSequana, 3,000+ nodes, 370,000+ cores, AMD Milan, 14 PFLOPS
815 TB main memory, 130 PB disk storage,
60 GPU nodes (visualization, machine learning, faster codes)
hot liquid cooling with high efficiency

Current Lustre Storage in Levante

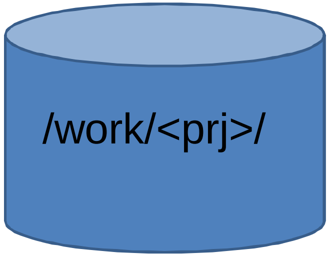
- HOME
  - 120 TiB NVMe
    - Home directories and software tree (User Quota)
    - Small files, fast access
- PROJECT
  - 118 PiB HDD based storage
    - Project directories (Lustre Project Quota)
    - SCRATCH directories of user (Lustre Project Quota)
- FASTDATA
  - Hybrid storage 200 TiB NVMe / 3 PiB HDD
    - Collaboration with DDN for testing new workflows / concepts
- All connected by Infiniband

Infiniband

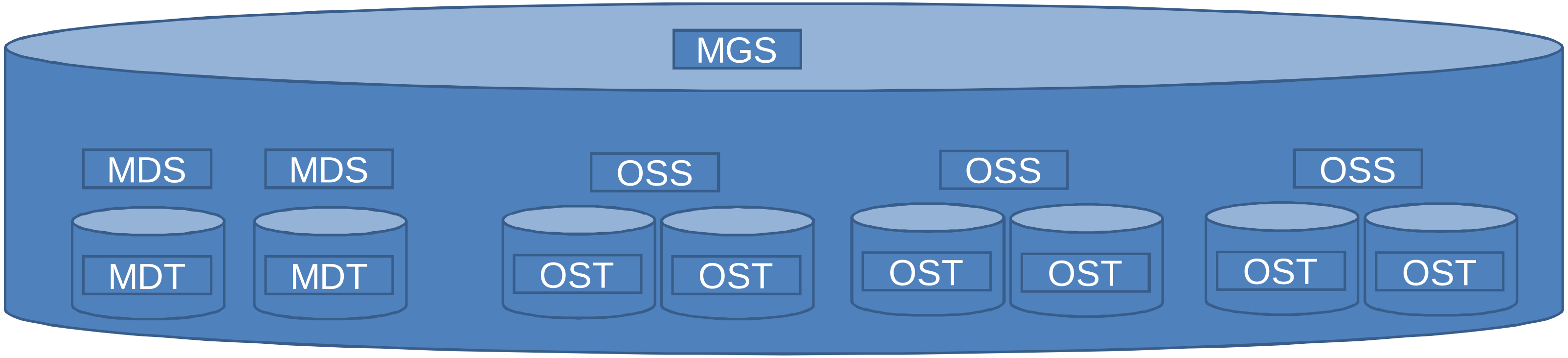
- Infiniband fabric of Levante has 100 Gb/s (EDR)
  - Internally the fabric is divided by Virtual Lanes
    - One lane for Lustre traffic and another for MPI/other traffic
    - Each Virtual Lane has 50% of the bandwidth
      - Other setups are possible e.g. 30/70
  - Before that separation Lustre traffic was in some cases disturbing the MPI traffic of simulations
    - Causing large runtime variations for the jobs

What’s your view to storage

- Large place and lots of space to store my data, I don’t care too much about structure or how the filesystem hardware is organized.



- How is the filesystem / hardware organized and how it could help in performance



Lustre WORK / SCRATCH

- PROJECT (aka /work and /scratch at DKRZ)
  - 2 MGS (Management Server, in our case the first 2 MDS)
  - 8 MDS (Metadata server) with one MDT (Metadata Target)
    - lfs df /work | grep MDT
  - 80 OSS (Object Storage Server) with 2 OST (Object Storage Targets)
    - Each OST is approx. 755 TiB in size
    - Total 160 OST’s available

      lfs df /work | grep OST
    - Lustre is distributing data nearly balanced over all OST’s

Concepts of stripping

- Metadata
  - Done by SysAdmin if new top level directory is created e.g. Project/Scratch/Home directory. Command is not available for normal users
    - lfs mkdir –c 8 /work/<prj> (Stripping over all MDS)
    - lfs mkdir –i [0,1,…,7] /work/<prj>/<subdir>[/…] (bind directory to explicit MDS)
  - Afterwards subdirectories will be distributed to one of the 8 MDS
    - Theoretically, in practice we found out, that it in some cases it does not work
    - It could happen that a subdirectory tree still sticks to the same MDS
    - Result: unbalanced MDT usage (better balancing/rebalancing with next Lustre version ?)

Concepts of stripping

- Files / Data
  - Lustre standard stripping with one OST
    - The stripping is set on the directory and will be inherited by files
    - Stripcount is 1 for a directory/file
    - Files are stored on only one Lustre OST (depending on the usage of OST's)
    - How to check:
      - Directory: lfs getstripe -d <dir>

        ```
        stripe_count:  1 stripe_size:   1048576 pattern:       0 stripe_offset: -1
        ```
      - File: lfs getstripe <file> | grep stripe_count

        ```
        lmm_stripe_count:  1
        ```
    - If changed afterwards to a directory, only new files will inherit new stripping

Concepts of stripping

- Progressive File Layout (PFL)
  - Default is set by SysAdmin on all new toplevel directories (e.g. /work/<prj>) and will be inherited by new subdirectories/files
    - lfs setstripe -E 1G -c 1 -S 1M -E 4G -c 4 -S 1M -E -1 -c 16 -S 1M
    - Files up to 1G size => 1 OST
    - Files up to 4G size => 4 OST's
    - Files >4G size => 16 OST's
  - Lustre is analyzing the filesize during writing and stripes it automatically
  - How to check:
    - lfs getstripe -d <dir> | grep stripe_count

```
stripe_count:  1      stripe_size:   1048576      pattern:       raid0      stripe_offset: -1
stripe_count:  4      stripe_size:   1048576      pattern:       raid0      stripe_offset: -1
stripe_count:  16      stripe_size:   1048576      pattern:       raid0      stripe_offset: -1
```

Concepts of stripping

- Manual set of stripping
  - User could also set an individual stripping, but be careful
  - Could be done as PFL (dynamic stripping)
    - e.g. if no PFL is set or standard stripping with one OST
    - lfs setstripe -E 1G -c 1 -S 1M -E 4G -c 4 -S 1M -E -1 -c 16 -S 1M TARGET_FOLDER
  - Or same stripping for all files
    - lfs setstripe –c 16 –S 1M TARGET_FOLDER
  - Comments
    - Only new files in that directory will get that stripping
    - If you want to stripe an existing folder with data
      - Create a new folder with your stripping setup and copy the data from the other one

Concepts of stripping

- History at DKRZ for Levante
  - The PROJECT filesystem was build with only 4 MDS/MDT in the beginning and expanded with 4 additional MDS/MDT later
    - Projects created in 2022 only have Metadata stripe of 4
    - Projects created in 2023 and later have Metadata stripe of 8
    - It CAN'T be easily changed from 4 to 8 for older Project directories
  - Distribution of Metadata not equally on all MDS/MDT
  - Progressive File Layout (PFL) was also created after the system was already in production and data was copied from previous HPC system Mistral
    - Only new directories under the toplevel /work/<prj> will inherit this PFL
    - Other older directories could still have a stripecount of 1 or what somebody maybe has set manually.

Monitor your data

- We have small wrapper script, to show infos about Quota, number of Files for your HOME/SCRATCH or for projects where you belong to
  - /sw/bin/lfsquota.sh -u <username> | -p projectname
- In some cases we see user with millions of files in one directory e.g. in their personal /scratch
  - ls -f | wc –l (-f => no sort)

    3089954
  - This could problems with your IO, Linux commands (e.g. ls, rm with ‘Argument list too long’)
  - We get problems to go through the files/directories for deleting data older th1n 14 days

Monitor you data

- Last year we had a user creating temporary files in his HOME by PyCharm with a rate of approx. 2500 creates/sec.
  - Was running for some time unnoticed
  - In the end => approx. 130 million temporary files in one directory
  - Quota was exceeded in HOME, but Quota mechanism could not stop the process, because it was too fast

```
[root@levante6 ~]# lfs quota -h -u xxxxxxx /home
Disk quotas for usr xxxxxxx (uid yyyyyy):
   Filesystem    used   quota   limit   grace   files   quota   limit   grace
        /home  33.36G*    30G     30G       - 120588351      0       0       -
```

  - Stopped by SysAdmin, deletion of directory took about 23 hours

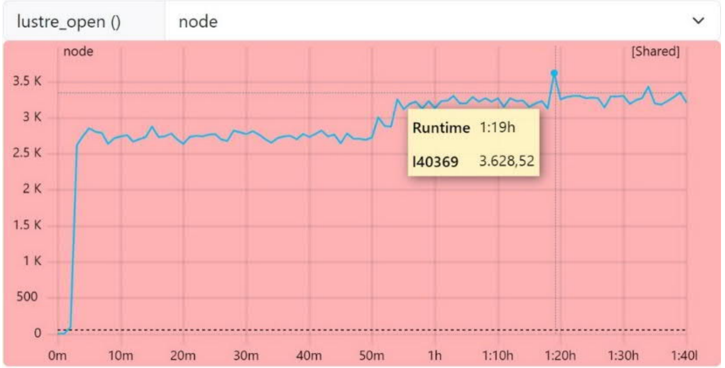
Monitor your data

- For running or finished SLURM jobs you could monitor also your IO by ClusterCockpit at DKRZ
  - https://clustercockpit.dkrz.de/
  - You get a lot information about your job incl.
    - IB bandwidth
    - Lustre bandwidth
    - Lustre close files
    - View from MDS/MDT side

```
interval=10 seconds, top10
                                     mknod/s    unlink/s      open/s     close/s     mkdir/s     rmdir/s   setattr/s  gettattr/s  setxattr/s  getxattr/s    statfs/s      sync/s
project                               125.38        0.00      125.85    15701.05        0.27        0.00       13.10        0.00        0.00      125.18        0.00        0.00
 MDT0003                              125.38        0.00      125.85    15701.05        0.27        0.00       13.10        0.00        0.00      125.18        0.00        0.00
  <slurm jobid>:<userid>:<nodename>     0.00        0.00        0.00     3492.30        0.00        0.00        0.00        0.00        0.00        0.00        0.00        0.00
```

Monitor your data

- Other view for opening files from ClusterCockpit



- If you have several thousands open/close per second during the whole runtime of the job, it might be worth to think about the workflow
- And if you have maybe several jobs of the same type running in parallel
    - In this case the user had five jobs like this running at same time and 4 of them on one GPU node

Quota and it’s bad habits

- On Levante we use two different kinds of Quota in Lustre
  - User Quota on HOME (Default 30 GB, no inode/file quota)
  - So called ‘Project’ Quota for WORK and SCRATCH
    - Each project in WORK gets a unique id (3000000 + ldap group id of project)
      - Request once per year at steering committee
    - Each User in SCRATCH gets a unique id (2000000 + ldap userid)
      - 15 TB per user
  - Currently no inode/file quota in WORK/SCRATCH

Quota and it’s bad habits

- Bad habits of Lustre Project Quota
  - If you are in more than one project , which is always case (SCRATCH+WORK), you have a problem
  - You can’t easily move data from one project to another due to the different quota id’s
  - For this you could request help by sending an Email to support@dkrz.de (e.g. >5TB)
  - We could manipulate the Quota id on the source side and then move the data to the new target
  - Also if you want to copy large datasets (several TB) between WORK projects or from SCRATCH to WORK, we have a special copy tool with parallel IO to help

Questions

Questions ?

Carsten Beyer (beyer@dkrz.de)