

## Storage

There are two main shared file systems on Anselm cluster, the HOME and SCRATCH. All login and compute nodes may access same data on shared filesystems. Compute nodes are also equipped with local (non-shared) scratch, ramdisk and tmp filesystems.

## Archiving

Please don't use shared filesystems as a backup for large amount of data or long-term archiving mean. The academic staff and students of research institutions in the Czech Republic can use CESNET storage service, which is available via SSHFS.

## Shared Filesystems

Anselm computer provides two main shared filesystems, the HOME filesystem and the SCRATCH filesystem. Both HOME and SCRATCH filesystems are realized as a parallel Lustre filesystem. Both shared file systems are accessible via the Infiniband network. Extended ACLs are provided on both Lustre filesystems for the purpose of sharing data with other users using fine-grained control.

## Understanding the Lustre Filesystems

(source <http://www.nas.nasa.gov>)

A user file on the Lustre filesystem can be divided into multiple chunks (stripes) and stored across a subset of the object storage targets (OSTs) (disks). The stripes are distributed among the OSTs in a round-robin fashion to ensure load balancing.

When a client (a compute node from your job) needs to create or access a file, the client queries the metadata server (MDS) and the metadata target (MDT) for the layout and location of the file's stripes. Once the file is opened and the client obtains the striping information, the MDS is no longer involved in the file I/O process. The client interacts directly with the object storage servers (OSSes) and OSTs to perform I/O operations such as locking, disk allocation, storage, and retrieval.

If multiple clients try to read and write the same part of a file at the same time, the Lustre distributed lock manager enforces coherency so that all clients see consistent results.

There is default stripe configuration for Anselm Lustre filesystems. However, users can set the following stripe parameters for their own directories or files to get optimum I/O performance:

1. `stripe_size`: the size of the chunk in bytes; specify with k, m, or g to use units of KB, MB, or GB, respectively; the size must be an even multiple of 65,536 bytes; default is 1MB for all Anselm Lustre filesystems
2. `stripe_count` the number of OSTs to stripe across; default is 1 for Anselm Lustre filesystems one can specify -1 to use all OSTs in the filesystem.
3. `stripe_offset` The index of the OST where the first stripe is to be placed; default is -1 which results in random selection; using a non-default value is NOT recommended.

Setting stripe size and stripe count correctly for your needs may significantly impact the I/O performance you experience.

Use the `lfs getstripe` for getting the stripe parameters. Use the `lfs setstripe` command for setting the stripe parameters to get optimal I/O performance. The correct stripe setting depends on your needs and file access patterns.

```
$ lfs getstripe dir|filename $ lfs setstripe -s stripe_size -c stripe_count -o stripe_offset dir|filename
```

Example:

```
' $ lfs getstripe /scratch/username/ /scratch/username/ stripe_count: 1 stripe_size: 1048576 stripe_offset: -1
```

```
$ lfs setstripe -c -1 /scratch/username/ $ lfs getstripe /scratch/username/ /scratch/username/ stripe_count: 10 stripe_size: 1048576 stripe_offset: -1 '
```

In this example, we view current stripe setting of the `/scratch/username/` directory. The stripe count is changed to all OSTs, and verified. All files written to this directory will be striped over 10 OSTs

Use `lfs check osts` to see the number and status of active OSTs for each filesystem on Anselm. Learn more by reading the man page

```
$ lfs check osts $ man lfs
```

## Hints on Lustre Stripping

Increase the `stripe_count` for parallel I/O to the same file.

When multiple processes are writing blocks of data to the same file in parallel, the I/O performance for large files will improve when the `stripe_count` is set to a larger value. The stripe count sets the number of OSTs the file will be written to. By default, the stripe count is set to 1. While this default setting provides for efficient access of metadata (for example to support the `ls -l` command), large

files should use stripe counts of greater than 1. This will increase the aggregate I/O bandwidth by using multiple OSTs in parallel instead of just one. A rule of thumb is to use a stripe count approximately equal to the number of gigabytes in the file.

Another good practice is to make the stripe count be an integral factor of the number of processes performing the write in parallel, so that you achieve load balance among the OSTs. For example, set the stripe count to 16 instead of 15 when you have 64 processes performing the writes.

Using a large stripe size can improve performance when accessing very large files

Large stripe size allows each client to have exclusive access to its own part of a file. However, it can be counterproductive in some cases if it does not match your I/O pattern. The choice of stripe size has no effect on a single-stripe file.

Read more on [http://wiki.lustre.org/manual/LustreManual20\\_HTML/ManagingStripingFreeSpace.html](http://wiki.lustre.org/manual/LustreManual20_HTML/ManagingStripingFreeSpace.html)

## **Lustre on Anselm**

The architecture of Lustre on Anselm is composed of two metadata servers (MDS) and four data/object storage servers (OSS). Two object storage servers are used for file system HOME and another two object storage servers are used for file system SCRATCH.

Configuration of the storages

- HOME Lustre object storage
  - One disk array NetApp E5400
  - 22 OSTs
  - 227 2TB NL-SAS 7.2krpm disks
  - 22 groups of 10 disks in RAID6 (8+2)
  - 7 hot-spare disks
- SCRATCH Lustre object storage
  - Two disk arrays NetApp E5400
  - 10 OSTs
  - 106 2TB NL-SAS 7.2krpm disks
  - 10 groups of 10 disks in RAID6 (8+2)
  - 6 hot-spare disks
- Lustre metadata storage
  - One disk array NetApp E2600
  - 12 300GB SAS 15krpm disks
  - 2 groups of 5 disks in RAID5

– 2 hot-spare disks

## HOME

The HOME filesystem is mounted in directory /home. Users home directories /home/username reside on this filesystem. Accessible capacity is 320TB, shared among all users. Individual users are restricted by filesystem usage quotas, set to 250GB per user. >If 250GB should prove as insufficient for particular user, please contact support, the quota may be lifted upon request.

The HOME filesystem is intended for preparation, evaluation, processing and storage of data generated by active Projects.

The HOME filesystem should not be used to archive data of past Projects or other unrelated data.

The files on HOME filesystem will not be deleted until end of the users lifecycle.

The filesystem is backed up, such that it can be restored in case of catastrophic failure resulting in significant data loss. This backup however is not intended to restore old versions of user data or to restore (accidentaly) deleted files.

The HOME filesystem is realized as Lustre parallel filesystem and is available on all login and computational nodes. Default stripe size is 1MB, stripe count is 1. There are 22 OSTs dedicated for the HOME filesystem.

Setting stripe size and stripe count correctly for your needs may significantly impact the I/O performance you experience.

HOME filesystem Mountpoint /home Capacity 320TB Throughput 2GB/s User quota 250GB Default stripe size 1MB Default stripe count 1 Number of OSTs 22 ###SCRATCH

The SCRATCH filesystem is mounted in directory /scratch. Users may freely create subdirectories and files on the filesystem. Accessible capacity is 146TB, shared among all users. Individual users are restricted by filesystem usage quotas, set to 100TB per user. The purpose of this quota is to prevent runaway programs from filling the entire filesystem and deny service to other users. >If 100TB should prove as insufficient for particular user, please contact support, the quota may be lifted upon request.

The Scratch filesystem is intended for temporary scratch data generated during the calculation as well as for high performance access to input and output files. All I/O intensive jobs must use the SCRATCH filesystem as their working directory.

Users are advised to save the necessary data from the SCRATCH filesystem to HOME filesystem after the calculations and clean up the scratch files.

Files on the SCRATCH filesystem that are **not accessed for more than 90 days** will be automatically **deleted**.

The SCRATCH filesystem is realized as Lustre parallel filesystem and is available from all login and computational nodes. Default stripe size is 1MB, stripe count is 1. There are 10 OSTs dedicated for the SCRATCH filesystem.

Setting stripe size and stripe count correctly for your needs may significantly impact the I/O performance you experience.

SCRATCH filesystem Mountpoint /scratch Capacity 146TB Throughput 6GB/s  
User quota 100TB Default stripe size 1MB Default stripe count 1 Number of OSTs 10 ### Disk usage and quota commands

User quotas on the file systems can be checked and reviewed using following command:

```
$ lfs quota dir
```

Example for Lustre HOME directory:

```
$ lfs quota /home Disk quotas for user user001 (uid 1234):
Filesystem kbytes quota limit grace files quota limit
grace      /home 300096      0 2500000000      -    2102      0
500000      - Disk quotas for group user001 (gid 1234):
Filesystem kbytes quota limit grace files quota limit grace
/home 300096      0      0      -    2102      0      0      -
```

In this example, we view current quota size limit of 250GB and 300MB currently used by user001.

Example for Lustre SCRATCH directory:

```
$ lfs quota /scratch Disk quotas for user user001 (uid 1234):
Filesystem kbytes quota limit grace files quota limit
grace      /scratch      8      0 1000000000000      -
3          0          0      - Disk quotas for group user001 (gid
1234): Filesystem kbytes quota limit grace files quota limit grace
/scratch      8      0      0      -    3      0      0      -
```

In this example, we view current quota size limit of 100TB and 8KB currently used by user001.

To have a better understanding of where the space is exactly used, you can use following command to find out.

```
$ du -hs dir
```

Example for your HOME directory:

```
$ cd /home $ du -hs * .[a-zA-z0-9]* | grep -E "[0-9]*G|[0-9]*M" |
sort -hr 258M      cuda-samples 15M      .cache 13M      .mozilla
5,5M      .eclipse 2,7M      .idb_13.0_linux_intel64_app
```

This will list all directories which are having MegaBytes or GigaBytes of consumed space in your actual (in this example HOME) directory. List is sorted in descending order from largest to smallest files/directories.

To have a better understanding of previous commands, you can read manpages.

```
$ man lfs
```

```
$ man du
```

## Extended ACLs

Extended ACLs provide another security mechanism beside the standard POSIX ACLs which are defined by three entries (for owner/group/others). Extended ACLs have more than the three basic entries. In addition, they also contain a mask entry and may contain any number of named user and named group entries.

ACLs on a Lustre file system work exactly like ACLs on any Linux file system. They are manipulated with the standard tools in the standard manner. Below, we create a directory and allow a specific user access.

```
‘ [vop999@login1.anselm ~]$ umask 027 [vop999@login1.anselm ~]$ mkdir test
[vop999@login1.anselm ~]$ ls -ld test drwxr-x— 2 vop999 vop999 4096 Nov  5
14:17 test [vop999@login1.anselm ~]$ getfacl test # file: test # owner: vop999
# group: vop999 user::rwx group::r-x other::—

[vop999@login1.anselm ~]$ setfacl -m user:johnsm:rwx test [vop999@login1.anselm
~]$ ls -ld test drwxrwx—+ 2 vop999 vop999 4096 Nov  5 14:17 test
[vop999@login1.anselm ~]$ getfacl test # file: test # owner: vop999 # group:
vop999 user::rwx user:johnsm:rwx group::r-x mask::rwx other::— ‘
```

Default ACL mechanism can be used to replace setuid/setgid permissions on directories. Setting a default ACL on a directory (-d flag to setfacl) will cause the ACL permissions to be inherited by any newly created file or subdirectory within the directory. Refer to this page for more information on Linux ACL:

[http://www.vanemery.com/Linux/ACL/POSIX\\_ACL\\_on\\_Linux.html](http://www.vanemery.com/Linux/ACL/POSIX_ACL_on_Linux.html)

## Local Filesystems

### Local Scratch

Every computational node is equipped with 330GB local scratch disk.

Use local scratch in case you need to access large amount of small files during your calculation.

The local scratch disk is mounted as /lscratch and is accessible to user at /lscratch/\$PBS\_JOBID directory.

The local scratch filesystem is intended for temporary scratch data generated during the calculation as well as for high performance access to input and output files. All I/O intensive jobs that access large number of small files within the calculation must use the local scratch filesystem as their working directory. This is required for performance reasons, as frequent access to number of small files may overload the metadata servers (MDS) of the Lustre filesystem.

The local scratch directory /lscratch/\$PBS\_JOBID will be deleted immediately after the calculation end. Users should take care to save the output data from within the jobscript.

local SCRATCH filesystem Mountpoint /lscratch Accesspoint /lscratch/\$PBS\_JOBID  
Capacity 330GB Throughput 100MB/s User quota none ### RAM disk

Every computational node is equipped with filesystem realized in memory, so called RAM disk.

Use RAM disk in case you need really fast access to your data of limited size during your calculation. Be very careful, use of RAM disk filesystem is at the expense of operational memory.

The local RAM disk is mounted as /ramdisk and is accessible to user at /ramdisk/\$PBS\_JOBID directory.

The local RAM disk filesystem is intended for temporary scratch data generated during the calculation as well as for high performance access to input and output files. Size of RAM disk filesystem is limited. Be very careful, use of RAM disk filesystem is at the expense of operational memory. It is not recommended to allocate large amount of memory and use large amount of data in RAM disk filesystem at the same time.

The local RAM disk directory /ramdisk/\$PBS\_JOBID will be deleted immediately after the calculation end. Users should take care to save the output data from within the jobscript.

RAM disk Mountpoint /ramdisk Accesspoint /ramdisk/\$PBS\_JOBID Capacity  
60GB at compute nodes without accelerator

90GB at compute nodes with accelerator

500GB at fat nodes

Throughput over 1.5 GB/s write, over 5 GB/s read, single thread over 10 GB/s write, over 50 GB/s read, 16 threads

User quota none ### tmp

Each node is equipped with local /tmp directory of few GB capacity. The /tmp directory should be used to work with small temporary files. Old files in /tmp directory are automatically purged.

#### Summary

---

Mountpoint	Usage	Protocol	Net	Capacity	Throughput	Limitations	Access Services
/home	home directory	Lustre	320 TiB	2 GB/s	Quota 250GB	Compute and login nodes backed up	/scratch
cluster shared jobs' data	Lustre	146 TiB	6 GB/s	Quota 100TB	Compute and login nodes files older 90 days removed	/lscratch	node local jobs' data local 330 GB 100 MB/s none
Compute nodes purged after job ends	/ramdisk	node local jobs' data local 60, 90, 500 GB	5-50 GB/s	none	Compute nodes purged after job ends	/tmp	local temporary files local 9.5 GB 100 MB/s none
Compute and login nodes auto purged							