

Taiga User Guide

- [Lustre Documentation](#)
 - [Capacity to inode Ratio](#)
 - [Block Size](#)
 - [Default Stripe Size](#)
 - [Progressive File Layout \(PFL\)](#)
- [Taiga Access Methods](#)
 - [Lustre Native Mount](#)
 - [Globus](#)
 - [NFS](#)
- [Data Recovery](#)
 - [Snapshots \(Coming Soon\)](#)
 - [Backups](#)
- [VM](#)

Lustre Documentation

Capacity to inode Ratio

Ratio: **1TB Quota to 1,500,000 iNodes**

An inode is a record that describes a file, directory, or link. This information is stored in a dedicated flash pool on *Taiga* and is of a finite capacity. To ensure that the inode pool does not run out of space before the capacity pools, this quota ratio is implemented. So for example if your project has a 10TB quota on *Taiga*, it will also have a quota of 7.50 million inodes. If you have any questions about this ratio please contact the storage team by opening a support ticket.

Block Size

File System Block Size: **2MB**

For a balance of throughput performance and file space efficiency, a block size of 2MB has been chosen for the *Taiga* file system. Larger file sizes help larger streaming data movement go faster, in general doing large I/O to file systems is encouraged if possible.

Default Stripe Size

Stripe Size: **1**

Number of Flash OSTs in *Taiga*: **24**

Number of HDD OSTs in *Taiga*: **28**

Lustre is capable of striping data over multiple OSTs to increase performance and help balance data across the disks. The default stripe for *Taiga* is set to 1 but this value is overridden as the file being written gets larger and larger; this behavior is determined by the PFL configured for *Taiga* which is described in the next section. If a user wants to see how many OSTs a file is striped across they can run the below command; the example shows a file that is striped across 2 OSTs.

Check Stripe Size

```
user@client# lfs getstripe -y /taiga/nsf/delta/abc123/testfile
lmm_stripe_count: 2
lmm_stripe_size: 4194304
lmm_pattern: raid0
lmm_layout_gen: 0
lmm_stripe_offset: 3
lmm_objects:
  - l_ost_idx: 0
    l_fid: 0x100000000:0x2:0x0
  - l_ost_idx: 1
    l_fid: 0x100010000:0x2:0x0
```

Progressive File Layout (PFL)

Taiga deploys a Progressive File Layout (PFL) that performs a couple key functions.

First, it allows us to keep the initial 64KB of every file on NVME flash; this increases performance for small file I/O by keeping it on faster media and keeps that noisy traffic off the spinning media that prefer larger I/O patterns. This helps improve the throughput for workloads doing large I/O by letting them have clearer access to the HDDs that make up the bulk of *Taiga's* capacity.

Second, it allows us to dynamically set the stripe size of files so that the bigger a file grows the more stripes it gets. This helps improve the performance of the system, and helps keep the OST usage rates more balanced which leads to overall better system responsiveness. The stripe count of a file can be overridden by using either "lfs setstripe" or by using "lfs migrate" to change an existing file's stripe count; however these actions are very much discouraged. Users should use the system defaults except in rare cases.

PFL Implementation Details:

NVME Capture Size: **64KB**

Stripe for Files 0 bytes to 256MB: **1**

Stripe for Files 256MB to 4GB: **4**

Stripe for Files 4GB and Above: **All HDD** (Currently 28)

Taiga Access Methods

Lustre Native Mount

Taiga is available via a native Lustre mount on the below systems:

- Delta
- HAL
- Radiant
- HOLL-I
- NCSA Industry Systems

Native sub-directory mounts can also be requested for one-off machines via an SVC ticket, with the **STO: Taiga** component. SET will be providing a streamlined guide for Lustre client installation and configuration here (link coming soon). Requests for one-off mounts of *Taiga* will only be allowed for machines that have gone through the NCSA security hardening process.

Globus

Taiga is accessible via Globus at the endpoint name "NCSA Taiga" and the endpoint is open to the public internet for transfers from anywhere with another Globus endpoint. Authentication to the endpoint is handled by NCSA's CILogon service and requires two-factor via Duo. For shared collections, or other questions submit a ticket to help+globus@ncsa.illinois.edu. Information about Globus can be found at their site <https://www.globus.org>

NFS

Native mounts of *Taiga* using the Lustre client are greatly preferred for superior performance and increased stability, however sub-directories of *Taiga* can be mounted via NFS in cases where that is necessary. The NFS service is accessed via the HA *taiga-nfs.ncsa.illinois.edu* endpoint. The NFS endpoint is currently comprised of 4 servers that are directly connected to the 100GbE public-facing storage network and via redundant links to *Taiga's* HDR Infiniband core fabric. If you need an NFS export please file a ticket with the **STO: Taiga** component flag and the storage team will assist in getting the exported provisioned.

Data Recovery

Snapshots (Coming Soon)

Snapshots are run on *Taiga* once per day, and are retained for a **14** day period. The creation of new and removal of old snapshots is run automatically without intervention, it is not possible to recover deleted data from more than 14 days prior. Users wishing to recover data from snapshots should open a ticket with the storage team.

In general snapshots are designed to protect against and are useful for:

- Recovery in case of accidental delete
- Restore in case of file corruption due to application error
- Restore in case of encryption due to ransomware

Snapshots are **not** designed to protect against:

- Catastrophic file system hardware/software failure

Snapshots are also not designed to act as version control for software; all code changes should be kept in a git repository or similar version control tool.

Backups

Data on *Taiga* is not backed up; and is only a single copy. It is recommended to back up critical data to an allocation on the *Granite* tape archive system or on another system on which you have an allocation.

VM

For VM attached storage, we recommend utilizing the Radiant openstack service. This is a separate service but will allow projects to have VMs to be able to handle their data pipeline/other requirements