

# Managing your Files

- 1 [Hyak File Systems](#)
  - 1.1 [GSCRATCH](#)
  - 1.2 [Home Directories](#)
  - 1.3 [Node Local Scratch](#)
- 2 [Note on filesystem permissions](#)
- 3 [Transferring files in and out of Hyak – Examples](#)
  - 3.1 [Transferring files \(all commands executed on the remote host\)](#)
- 4 [Notes On I/O Performance](#)
- 5 [Remote Filesystems - lolo](#)
  - 5.1 [Enabling lolo Access](#)
  - 5.2 [Notes on Using the Archive](#)
  - 5.3 [SSH Access](#)
  - 5.4 [Quotas](#)
  - 5.5 [Purchasing Additional Capacity](#)
  - 5.6 [Performance](#)
- 6 [Cloud Storage](#)

## Hyak File Systems

### GSCRATCH

Each Hyak group has an eponymous shared scratch directory under /gscratch which is shared among all Hyak nodes. If your group is hyak-mygroup, your directory would be /gscratch/mygroup. You can determine your group by looking at the output from the groups command. **Please use your group's /gscratch directory in most situations** requiring large data volume, high throughput, or high transaction rates, or in any situation where you will need to share data with other users. Some situations might benefit from the use of node-local scratch space described below. Gscratch is not backed up.

Some of you have additional shared scratch space dedicated to your projects. Please ask your PI if you're in doubt.

Shared scratch directories are configured "setgid". This ensures any files you create within inherit the group association of the parent directory. Nonetheless, it's worthwhile to check to be sure the group association of files you create is correct.

File modification time (mtime) is only periodically updated and file access time (atime) is disabled on the /gscratch filesystem. Both these configurations are to improve the performance on /gscratch. Running find with any of the time options such as mtime or atime may not produce expected results.

#### Quotas

Each hyak group is assigned a filesystem quota based on the number of nodes it owns. Quotas in the shared scratch space are managed based on file group association. Check your quotas with the mmlsquota command:

```
MOX: mmlsquota -j mygroup --block-size G gscratch
IKT: mmlsquota -j hyak-mygroup --block-size G suppscr
```

Usage reporting is written to each group's space hourly in a file at the root of the directory called usage\_report.txt. Reports can also be sent by e-mail weekly (Monday morning) and daily when your group is over their quota. If you're interested in e-mail reporting, please contact us.

### Home Directories

Each Hyak user also has a home directory which is shared among all Hyak nodes. Quotas are set to 10GB with a limit of 5,000 files. Hyak home directories are intended for files you want to keep completely private (ssh keys) or which are completely unique to you (your login scripts). They are not for storing big code source trees or doing any computation. **The home filesystem is small and slow, so please limit your work to the scratch filesystems.** Home directories are not backed up.

Check your /home quota with the mmlsquota command:

```
MOX: mmlsquota --block-size G gscratch:home
IKT: mmlsquota --block-size G suppscr:home
```

### Node Local Scratch

Each node in Hyak has an /scr filesystem, typically with ~100GB of space. These filesystems are accessible only to the local node and are not shared with any other Hyak nodes. This is available to you for use in cases where your application performance is limited by the shared scratch. mpiBLAST, for example, partitions a reference database and distributes it among the nodes participating in a calculation to achieve better I/O scaling. The local scratch disk is cleaned up after each job completes. If you would like to keep data on the local scratch disk of your nodes, create a directory with group name and put your data there (if your UNIX group is hyak-mygroup, you would create /scr/mygroup). Node local scratch filesystems are not backed up.

## Note on filesystem permissions

Hyak uses the standard Unix/Posix authentication and authorization model. Each file is associated with an owner and a group. Users may set the default permissions for all files they create by using a umask in their ~/.bashrc environment configuration files.

- Read permissions for all group members (default on hyak) \*

The default settings on hyak are such that users belonging to a group XYZ can read the files of other group members inside /gscratch/XYZ. These permissions should be sufficient for most groups.

- Write permissions for all group members (not recommended) \*

This is not recommended but we are documenting it in case some group wants to use this approach. With these permissions, user A can overwrite files written by user B. This can happen inadvertently under various circumstances e.g. user A gets some script from user B and runs it without modifying the paths such as /gscratch/XYZ/B... This will overwrite the files belonging to user B.

To set the permissions for a directory to ensure all members of the group can read and write all files created within it, do this:

```
chmod 2770 dirname
chgrp groupname dirname
```

All members of the group must also enter below line in their .bashrc file:

```
umask 007
```

## Transferring files in and out of Hyak – Examples

### Transferring files (all commands executed on the remote host)

- **Into Hyak**
  - via scp, e.g. command line:  
scp filename user@ikt.hyak.uw.edu:path/to/destination/directory
  - via tar, e.g. command line:  
tar -czf - local\_directory | ssh user@ikt.hyak.uw.edu 'tar -xvzf -'
  - via tar, e.g. command line through ssh channel (see [Logging In](#))  
tar -czf - local\_directory | ssh -S /tmp/\$USER-hyak-socket ikt.uw.edu "tar -xvzf -"
  - via bbcp  
bbcp mydatafile ikt1.hyak.uw.edu:/gscratch/mygroup/myuserid  
bbcp mydatafile ikt2.hyak.uw.edu:/gscratch/mygroup/myuserid
  - via rsync  
insert example here
- **Out of Hyak**
  - via scp, to local working directory  
scp user@hyak.washington.edu:path/to/file .
  - via tar  
ssh user@hyak.washington.edu '(cd path/to/destination/directory; tar -czf -)' | tar -xvzf -
  - via tar, e.g. command line through ssh channel (see [Logging In](#))  
ssh -S /tmp/\$USER-hyak-socket ikt.hyak.uw.edu "tar -czf /path/to/remote/directory" | tar -xvzf -
  - via rsync  
insert example here

## Notes On I/O Performance

You will see substantially better I/O performance by using I/O buffers 4MB or larger and by avoiding lots of open/close operations on files. Not following these guidelines will result in poor I/O performance for all Hyak users, not just you. The shared storage on Hyak is a limited resource in capacity and performance: Please try to be a good neighbor.

## Remote Filesystems - lolo

Groups participating in Hyak have access to the [lolo archive](#) scalable storage services.

The Archive filesystem is intended for long term storage of data that changes infrequently, if ever. It is a tape-based system with a small disk buffer to improve write performance. It should **NEVER** be used for **ANY** sort of interactive access. It is equally important that you **NEVER** store thousands of **SMALL FILES**. Instead, collect directories of small files into tar files **BEFORE** transferring them to the Archive. The Archive quota limits you to 10,000 files/TB - an average file size of 100MB. This means if you store even a few tiny files, the remaining files must be very large - typically several GB in size.

The filesystem will perform best when used with large (> 1MB) files. For this and other reasons, we strongly encourage you to use tar to combine collections of small files into tar files before transferring them to the Archive filesystem for safekeeping. Be sure to read the section on Quotas below: limits are in place on the capacity each group can use and on the number of files.

Hyak groups have disk space on lolo at /archive/xyz where xyz is the group name.

However, hyak (ikt and mox) and lolo are separate systems. Hyak users can transfer data between them by using sftp or scp from the login nodes. See below link for more details on how to transfer files from hyak to lolo.

[Hyak lolo file transfer](#)

## Enabling lolo Access

Enable lolo access at below link:

<http://uwnetid.washington.edu/manage/?service>

## Notes on Using the Archive

Files written to the archive are cached on disk before being migrated to tape. Once migrated, a backup is made on tapes stored in a separate data center. Filesystem metadata is always maintained on disk, allowing a consistent view of the filesystem regardless of where the data blocks reside (i.e. 'ls -l' will list your files the same way whether they're in disk or tape). Copy data to or from the archive using the same commands as you for any other filesystem. For the Archive to function well, however, there are some **important guidelines**:

The Archive should **NEVER** be used for **ANY** sort of interactive access

This means it's OK to recall the files associated with a project a few months or years after copying them to the Archive. It is **NOT** OK to recall the same files from the Archive on a weekly basis.

The Archive should **NEVER** store thousands of **SMALL FILES**.

Instead, collect directories of small files into tar files **BEFORE** transferring them to the Archive - like so:

```
$ tar -cvf subdir.tar subdir
```

After this you can use sftp or scp to transfer subdir.tar to lolo.

More detail:

When interacting with the Archive filesystem it is very important to avoid unnecessarily recalling files from tape. In most cases, opening an archived file requires recalling it from tape. This triggers a sequence of events:

1. The tape library fetches the appropriate tape and loads it into a tape drive (< 2 minutes)
2. The tape drive seeks to the location of your file (< 2 minutes)
3. The tape drive begins streaming your file to the disk cache (> 100MBs for large files)

As you can see, this is a heavyweight operation and should be avoided when possible. For example, recalling a file simply for appending is almost never a good thing. Instead, we encourage everyone to make "whole file" copies (use the "~~W~~--whole-file" flag with rsync). This will replace the earlier version of the file with the new version without triggering a recall from tape. For example, to update your archive of a very important directory from your laptop:

```
$ rsync -av --whole-file ImportantDir lolo.washington.edu:/archive/hyak/GROUPDIR/
```

Where GROUPDIR is the name of your Hyak group.

Anytime you use rsync with the Archive filesystem it is very important to use the "--whole-file" argument to prevent rsync from recalling previous versions of each file from tape for comparison. You can rerun this command anytime you want to update the archive copy. It will not delete any files from the archive and will replace any previously archived versions of files with the updated versions.

Files deleted from the Archive persist on tape for some time. This is accomplished by two means. First, it is an (nearly) unavoidable consequence of working with tape. When you issue a delete command, the filesystem metadata is updated to reflect the new state of the file (deleted), but the blocks occupied by that file on tape are not immediately reclaimed. Periodically, an automatic reclamation process transfers data from tapes with deleted file "holes" to new tapes, returning the tapes with the deleted file blocks to the empty tape pool for reuse. Second, we back up the migration tape copy to a second tape copy. The backup retention policy is to preserve all files for at least one month beyond their date of deletion. This means you can be confident that when you replace a file in-place (as with the rsync example, above), the previous version of the file will be recoverable if something goes wrong.

Be aware of the quota on the number of files you can store in the Archive: 10,000 files per TB. This works out to an average file size of 100MB. If you are using the Archive to store directories containing many small files, consider using tar to combine these files into one big file before transferring them to the Archive. Not only will this help manage your quota, but large files ensure much better tape drive performance, resulting in quicker recall of your data when you need it.

## SSH Access

You may use ssh to access lolo from Hyak, Campus, or the Internet.

## Quotas

Groups are entitled to at about 450GB of archive space for each node they operate in Hyak. Quotas are applied based on the contents of each group's directory.

When logged into lolo, you can report your group's quotas with `mmlsquota`.

```
-bash-3.2$ mmlsquota -j hyak-groupname archive
```

where *groupname* is your group's name.

Quotas are also set on the number of files you can have. The limit is 10,000 files per TB in the Archive.

## Purchasing Additional Capacity

lolo is a UW-IT service to the UW research community, not just Hyak. Details and pricing are available directly from the [UW-IT Service Catalog](#).

## Performance

We have measured transfers from peers on es.net as fast as 650MB/s. Transfers with Teragrid sites have typically been in the 50MB/s range (we're looking into it). Copying a single 100GB file between lolo and /gscratch takes less than seven minutes (~285MB/s). Aggregate transfer rates are higher.

## Cloud Storage

We provide the Google `gsutil` tool for transferring data from Google Cloud Storage and Amazon S3. If you would like more details on its usage, please see the Google [documentation](#).