# Handling large datasets within the cluster

While working on the cluster, you may need to perform operations such as *copy*, *delete*, *sync*, *find*, etc. on a very large number of files or on files that are very large in size. While we do recommend to first have a look whether you can reduce the number of files you need to handle e.g. by packing them (because both the file system Lustre that is driving BIGWORK and the controllers of the disk drives have limits how many files they can handle at once. That means that if you need more IOPs, the overall performance you get may deteriorate significantly), we also realize that this frequently is more a long-term project. For this reason, we provide some parallel tools provided by the MPI-based package mpiFileUtils. The standard Unix commands like `cp`, `rm` or `find` are often comparatively slow, as they are implemented as single process applications.

As a typical example, consider copying directories containing a large number of files from your $BIGWORK to the local $TMPDIR storage on the compute nodes you allocated for further processing by your job, or/and transferring computation results back to your $BIGWORK.

Another example would be a quick freeing up of space in your $BIGWORK by first copying files to your $PROJECT storage and then deleting them from $BIGWORK.

Also, you could utilize the command `dsync`, if you use your $PROJECT storage for backing up directories on your $BIGWORK.

Below we will look at some of the mpiFileUtils tools in these and other practical examples.

In order to speed up the recursive transfer of the **contents** of the directory `$BIGWORK/source` to `$TMPDIR/dest` on a compute node, put the lines below in your job script or enter them at the command prompt of your interactive job - we assume that you've requested 8 cores for your batch job:

```
module load GCC/8.3.0 OpenMPI/3.1.4 mpifileutils/0.11
mpirun -np 8 dcp $BIGWORK/source/ $TMPDIR/dest
```

Please note a trailing slash (`/`) on the source path, which means "**copy the contents of the source directory**". If the `$TMPDIR/dest` directory does not exist before copying, it will be created. The command `mpirun` launches `dcp` (distributed copy) in parallel with 8 MPI processes.

The directory `$BIGWORK/dir` and its contents can be removed quickly using the `drm` (distributed remove) command:

```
mpirun -np 8 drm $BIGWORK/dir
```

**Note**: here and below we assume that the module `mpifileutils/0.11` is already loaded.

The command `drm` supports the option `--match` allowing to delete files selectively. See `man drm` for more information.

The next useful command is `dfind` - a parallel version of the unix command `find`. In this example we find all files on $BIGWORK larger than 1GB and write them to a file:

```
mpirun -np 8 dfind -v --output files_1GB.txt --size +1GB $BIGWORK
```

To learn more about other `dfind` options, type `man dfind`.

If you want to synchronize the directory `$BIGWORK/source` to `$PROJECT/dest` such that the directory `$PROJECT/dest` has content, ownership, timestamps and permissions of `$BIGWORK/source`, execute:

```
mpirun -np 8 dsync -D $BIGWORK/source $PROJECT/dest
```

Note that for this example the `dsync` command has to be launched on the login node where both $BIGWORK and $PROJECT are available.

The last mpiFileUtils tools we consider in this section are `dtar` and `dbz2`. The following creates a compressed archive `mydir.tar.dbz2` of the directory `mydir`:

```
mpirun -np 8 dtar -c -f mydir.tar mydir
mpirun -np 8 dbz2 -z mydir.tar
```

Please note: If the directory to be archived is located on your $HOME, the archive file itself should be placed on $BIGWORK.

Please note: Transferring a large number of files from the cluster to an external server or vice versa as a single (compressed) tar archive is much more efficient than copying files individually.

Some other useful commands are:

- `dstripe` - restripe(lustre) files in paths
- `dwalk` - list, sort, summarize files
- `ddup` – find duplicate files

A complete list of mpiFileUtils utilities and their description can be found at http://mpifileutils.readthedocs.io/.

You might also consider alternative tools like GNU parallel, pigz or pcp. They are all available as modules on the cluster.