Deleting large numbers of files on scratch and group

Deletion of files is a permanent action.

Deletion of files in linux system is a non-recoverable action. Please execute any deletion command with care.

rm may overload the metadata server when deleting a large number of files

Using the standard Linux command **`rm`** to delete multiple files on a Lustre filesystem **is not recommended**. The `rm` command will generate a `stat()` operation on for each file it removes, meaning all of the attributes of a file (filetype, owner, permission, modification time, etc.) will be returned from the metadata server. A large number of `stat()` operations can place an increased load on the metadata server, resulting in lower performance and instabilities with the filesystem.

Instead, use **`munlink`**, a Lustre-specific command that will simply delete a file without performing a stat() operation. Below is an example of our recommended approach which consists of two steps.

The **first step** deletes all the files and soft links within a directory (and its subdirectories) with the use of `munlink`:

```
find -P ./processor0 -type f -print0 -o -type l -print0 | xargs -0 munlink
```

Here is an overview of each step in that command:

- `find`
  is a command that will search the indicated directory (and subdirectories within). The syntax defines a search for files and soft links.
- `-P`
  this option restricts the search within the indicated directory tree and forces NO dereference of symbolic links. This warranties that the find command will not look for files within the links.
- `./processor0`
  this argument is the directory from which the search will start
- `-type f -print0 -o -type l -print0`
  These options indicate that the find command will search for anything that is a file (`-type f`) or (`-o`) a soft link (`-type l`, this is the lower letter `l`). Needs `-print0` on both parts of the "or" (its meaning is explained below).
- `-print0`

  This option indicates the format of the result of the "`find`" command. This particular format is able to catch strange file names, and ensures that they are readable for the following command (`xargs`) which has been concatenated with the pipe. It needs to appear twice due to the "or" option (`-o`) explained in the previous bullet point.
- `|`
  The Pipe command (represented by a single pipe line: `|`) concatenates two commands. This makes the output of the previous command "`find`" to serve as input to the following command (`xargs` in this case).
- `xargs -0`
  `xargs` will then convert the received list of files, line by line, into an argument for whatever command is specified at the end (in this case: `munlink`). The `-0` flag is related to the format of the listed files; if you use `-print0` in the `find` command you must use `-0` in the `xargs` command.
- `munlink`
  deletes each file and soft link in the list without overloading the metadata server. In this case, the list is the one received by `xargs`.

The **second step** is to remove the empty directories and subdirectories in the tree. Once all of the files and soft links were deleted, you can remove the empty directories with a similar command:

```
find -P ./processor0 -type d -empty -delete
```

Again, the `find` command will search the directory `processor0` and all subdirectories for any empty directories (`-type d -empty`) and delete them. The `-delete` action deletes the empty directories that has been found. The `-delete` option uses the `-depth` option implicitly (so the `-depth` option does not need to be explicitly given to this `find` command). The `-depth` option (used implicitly here) instructs to process each directory's contents before the directory itself, then the most distant branches in the directory tree will be processed first.

Use the given order of the options

Please note that the flags passed to `find` are evaluated as an expression, so if you pass `-delete` before `-type d -empty`, `find` will attempt to delete EVERYTHING below your starting directory and not only the empty directories, as the rest of the options are interpreted afterwards.

We **strongly recommend** that all OpenFOAM users make use of `munlink` for deleting simulation output. Doing so will not only help keep a stable filesystem, but also provide a speedup over using the standard `rm -rf` command.

Related articles

- Deleting large numbers of files on scratch and group
- Exploring files in Lustre
- Use of tmp by Running Jobs
- Python 2 End of Life
- Installing Python modules on supercomputers