

Use of tmp by Running Jobs

/tmp on compute nodes is a memory-resident temporary file system. Used with discretion, this offers the opportunity to stage temporary files, or frequently accessed read-only data, in memory. This can offer significant improvements in performance compared with I/O from other filesystems.

For applications that use large amounts of main memory, the use of /tmp should be avoided. In these cases, using up memory via /tmp may result in unpredictable application failure or increased time-to-solution.

To use memory for temporary storage, simply create files in (or copy existing files to) the directory /tmp. This must be done on the back-end (ie., via srun).

The amount of space available on /tmp is limited to one half the memory available on the node (giving 32 GB on Magnus). If this is exceeded, the application will fail with a message of the form

```
cp: writing `/tmp/file.dat': No space left on device
cp: failed to extend `/tmp/file.dat': No space left on device
```

(in this particular case as a result of a standard "cp" command).

It is important to understand the memory requirements per node of applications, together with any space used in /tmp. It must also be remembered that files in /tmp are local to the node, and not available to all nodes in a job (use /scratch). Files in /tmp on the back end are deleted automatically at the end of a job.

Fortran users should be aware that /tmp is commonly the default destination for files with status="scratch" used at run time. Please check the situation for your application or contact the helpdesk for assistance if you are unsure.

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](#)