Cheyenne use policies

Page contents

Appropriate use of login nodes

Users may run short, non-memory-intensive processes interactively on the Cheyenne system's login nodes. These include tasks such as text editing or running small serial scripts or programs.

However, the login nodes **may not** be used to run processes that consume excessive resources. This is to ensure an appropriate balance between user convenience and login node performance.

This applies to individual processes that consume excessive amounts of CPU time, more than a few GB of memory, or excessive I/O resources. It also applies collectively to multiple concurrent tasks that an individual user runs.

Processes that use excessive resources on the login nodes are terminated automatically. Affected users are informed by email that their sessions were terminated. They are also advised to run such processes in batch or interactive jobs on the Casper cluster.

See Checking memory use for how to use the **peak_memusage** utility.

Fair share policy

CISL manages scheduling priorities to ensure fair access to the system by all of these stakeholder groups: the university community, the NCAR community, the Community Earth System Model (CESM) community, and the Wyoming community.

The **fair-share policy** takes the community-wide usage balance into account along with several additional factors. These include the submitting users' currently running jobs and recently completed jobs. The scheduling system uses a dynamic-priority formula to weigh these factors, calculate each job's priority, and make scheduling decisions.

Job scheduling priorities

The PBS Pro workload management system scheduling policy for running jobs in the Cheyenne environment requires balancing several factors, as summarized above. Jobs generally are sorted in the following order:

1. Current processor use
2. Job priority
3. Queue priority
4. Job size

Job sorting is adjusted frequently in response to varying demands and workloads. PBS examines the jobs in sorted order in each scheduling cycle and starts those that it can. Jobs that cannot be started immediately are either scheduled to run at a future time or bypassed for the current cycle. Under typical system usage, multiple scheduling cycles are initiated every minute.

The scheduler may not start a job for a number of reasons, including:

- Your job is waiting for resources.
- The system has been reserved for a scheduled outage.
- Your job has been placed on hold.
- You have reached your concurrent core-usage limit when using the share queue.

Note that a high-priority job might be delayed by one of the limits on the list, while a lower-priority job from a different user or a job requesting fewer resources might not be blocked.

If your job is waiting in the queue, you can run the **qstat** command as shown to obtain information that can indicate why it has not started running. (Use this command only sparingly.)

```
qstat -s jobID
```

Note: To prevent jobs from languishing in the queues for an indefinite time, PBS reserves resources for the top-priority jobs and doesn't allow lower-priority jobs to start if they would delay the start time of a higher-priority job.

PBS sorting order

Stakeholder shares

CISL manages scheduling priorities to ensure fair access to the system by these stakeholder groups: the university community, the NCAR community, the Climate Simulation Laboratory, and the Wyoming community.

Each stakeholder group is allocated a certain percentage of the available processors. A job cannot start if that action would cause the group to exceed its share, unless another group is using less than its share and has no jobs waiting. In this case, the high-use group can "borrow" processors from the lower-use stakeholder group for a short time.

When jobs are sorted, jobs from groups that are using less of their share are picked before jobs from groups using more of their shares.

Job priority

Job priority has three components.

- native priority
- queue priority
- waiting time

If the native priority is 0, a further adjustment is made based on how long the job has been waiting for resources. Waiting jobs get a "boost" of up to 20 priority points, depending on how long they have been waiting and which queue they are in.

Queue priority

Cheyenne queue priorities are shown in the table on this web page: Job-submission queues and charges.

Job size

Jobs asking for more nodes are favored over jobs asking for fewer. The reasoning is that while it is easier for small jobs to fill gaps in the schedule, larger jobs need help collecting enough CPUs to start.

Backfilling

As mentioned above, when PBS cannot start a job immediately, if it is one of the first such jobs, PBS sets aside resources for it before examining other jobs to see if any of them can run as backfill. That is, PBS looks at running jobs to determine when they will finish based on wall-time requested. From those finish times, PBS decides when enough resources (such as CPUs, memory, and job limits) will become available to run the top job. PBS then reserves the resources that the job requests at that identified time.

When PBS looks at other jobs to see if they can start immediately, it also checks whether starting any of them would collide with one of these resource reservations. Only if there are no collisions will PBS start the lower-priority jobs.