UNIVERSITY
OF OSLO

Queue System

Table of Contents

This page documents the queue system on the Colossus (HPC for TSD) cluster.

In TSD, each project has its own Linux Virtual Machine (VM) for submitting jobs to the cluster.

Submitting a job

To run a job on the cluster, you submit a job script into the *job queue*, and the job is started when one or more suitable *compute nodes* are available. The job queue is managed by a queue system (scheduler and resource manager) called Slurm (Slurm's documentation page).

Please note that jobscript names should not contain sensitive information.

Job scripts are submitted with the `sbatch` command:

```
sbatch YourJobscript
```

The `sbatch` command returns a *jobid*, an id number that identifies the submitted job. The job will be waiting in the job queue until there are free compute resources it can use. A job in that state is said to be *pending* (*PD*). When it has started, it is called *running* (*R*). Any output (stdout or stderr) of the job script will be written to a file called `slurm-jobid.out` in the directory where you ran `sbatch`, unless otherwise specified.

All commands in the job script are performed on the compute-node(s) allocated by the queue system. The script also specifies a number of requirements (memory usage, number of CPUs, run-time, etc.), used by the queue system to find one or more suitable machines for your job.

You can cancel running or pending (waiting) jobs with `scancel`:

```
scancel jobid # Cancel job with id jobid (as returned from sbatch)
scancel --user=MyUsername    # Cancel all your jobs
scancel --account=MyProject  # Cancel all jobs in MyProject
```

See `man scancel` for more details.

Project Quota

On Colossus, each user only has access to a single project, the name of which (p*NN*) is the prefix of the user name. Projects can have access to up to 3 allocations of computational resources in TSD:

- TSD allocation
- Sigma2 allocation
- Dedicated allocation

TSD allocation

Any TSD project with HPC access can use these resources for free with a 200k CPUh limit.

To use this resource, jobs should be submitted with the "`--account=pNN_tsd`" argument. Where pNN is your project number.

Sigma2 allocation

Only TSD projects with cpu hour quota from Sigma2 can use this pool. We advice any project with substantial computational needs to request CPU (and disk) quota from Sigma2 as described here. Sigma2 quota are valid for 6 month periods, starting April 1 and October 1.

To use this resource, jobs should be submitted with the "`--account=pNN`" argument. Where pNN is your project number. If you submit to this resource, but the project doesn't have a Sigma2 quota, jobs will remain pending (PD) with reason "AssocGrpBillingMinutes".

Dedicated allocation

All other compute and gpu nodes in Colossus were acquired by individual projects for privileged access but are maintained by TSD.

To use this resource, jobs should be submitted with the "`--account=pNN_reservationname`" argument. Where pNN is your project number and the reservationname the name of the dedicated resource.

Inspecting Quota

The command `cost` can be used to inspect the quota and see how much of it has been used and how much remains. This only applies to projects with an accountable Sigma2 quota. For projects *without* a Sigma2 quota, the cost command will list a zero quota and a corresponding message:

```
Report for account p11 on Colossus
Allocation period 2021.1 (2021-04-01 -- 2021-10-01)
Last updated on Mon Apr 12 10:14:02 2021
============================================================
Account  Description              Billing hours  % of quota
============================================================
p11      Used (finished)                   0.00          NA
p11      Reserved (running)                0.00          NA
p11      Pending (waiting)                 0.00          NA
p11      Available                         0.00          NA
p11      Quota                             0.00         100
============================================================
The project does not have a Sigma2 quota for the current period.
See https://www.uio.no/english/services/it/research/sensitive-data/use-tsd/hpc/
for information.
```

For the projects having a Sigma2 cpu hour quota, a typical output will look like:

```
Report for account p11 on Colossus
Allocation period 2021.1 (2021-04-01 -- 2021-10-01)
Last updated on Mon Apr 12 10:14:02 2021
================================================================
Account     Description             Billing hours     % of quota
================================================================
```

```
p11         Used (finished)                  0.95          0.0 %
p11         Reserved (running)               0.00          0.0 %
p11         Pending (waiting)              192.00          0.4 %
p11         Available                    49807.05         99.6 %
p11         Quota                        50000.00        100.0 %
============================================================
```

Notice that "Available" shows the "Quota" minus the "Used", the "Reserved" and the "Pending". So "Available" it is what is expected to be available if all running and pending jobs use the hours they specify. (Usually, jobs specify longer `--time` than they actually use, so "Available" will typically increase as jobs finish.)

One can also list the use per user within the project by adding "`--details`":

```
cost --details
```

This will append the quota usage per user:

```
Report for account p11 on Colossus
Allocation period 2020.2 (2020-10-01 -- 2021-04-01)
Last updated on Fri Oct  2 08:34:02 2020
============================================================
Account  Description              Billing hours  % of quota
============================================================
p11      Used (finished)                164.55       0.2 %
p11      Reserved (running)               0.00       0.0 %
p11      Pending (waiting)             1680.00       1.7 %
p11      Available                    98155.45      98.2 %
p11      Quota                       100000.00     100.0 %
============================================================
Account  User                Used billing hours  % of quota
============================================================
p11      p11-bartt                      162.32       0.2 %
p11      root                             2.22       0.0 %
============================================================
```

See `man cost` for details about this command.

Accounting

Accounting is done in terms of *billing units*, and the quota is in *billing unit hours*. Each job is assigned a number of billing units based on the requested CPUs, memory and GPUs. The number that is subtracted from the quota is the number of billing units multiplied with the (actual) wall time of the job.

The number billing units of a job is calculated like this:

1. Each requested CPU is given a cost of 1.

2. The requested memory is given a cost based on a *memory cost factor* (see below).
3. The requested GPU is given a cost based on a *GPU cost factor* (see below).
4. The number of billing units is the maximum of the CPU cost, memory cost and GPU cost.

The *memory cost factor* and the *GPU cost factor* vary between nodes.

- For regular compute nodes, the *memory cost factor* is 0.12749 units per GiB. Thus the memory cost of a job asking for all memory on a node will be 64, the number of CPUs on the node.
- For GPU nodes, the *memory cost factor* is 0.09775967 units per GiB, and the *GPU cost factor* is 24 per GPU. The means that a job asking for all memory, or all GPUs on a node, get a cost of 96, the number of CPUs on the node.
- For bigmem nodes, the *memory cost factor* is 0.0323641 per GiB. Thus the memory cost of a job asking for all memory on a node will be 128, the number of CPUs on the node.

When a project has exceeded the Quota limit, jobs will be left pending with the reason "AssocGrpBillingMinutes".

Inspecting Jobs

To get a quick view of the status of a job, you can use `squeue`:

```
squeue –j JobId
```

where *JobId* is the job id number that `sbatch` returns. To see more details about a job, use

```
scontrol show job JobId
```

See `man squeue` and `man scontrol` for details about these commands.

Inspecting the Job Queue

There are several available commands to inspect the job queue:

- `squeue`: list jobs in the queue
- `pending`: list the pending (waiting) jobs in the queue
- `qsumm`: show summary of queue usage

To see the list of running or pending jobs in the queue, use the command `squeue`. `squeue` will only show the jobs in your own project. Useful `squeue` options:

```
[-j jobids]   show only the specified jobs
[-w nodes]    show only jobs on the specified nodes
[-t states]   show only jobs in the specified states (pending, running,
              suspended, etc.)
[-u users]    show only jobs belonging to the specified users
```

All specifications can be comma separated lists. See `man squeue` for details. Examples:

```
squeue -j 14132,14133    # shows jobs 4132 and 4133
squeue -w c1-11         # shows jobs running on c1-11
squeue -u foo -t PD      # shows pending jobs belonging to user 'foo'
```

Squeue status (ST)

Status	**Text**
PD	Pending
R	Running
S	Suspended
CG	Completing
CDhttps://www.prosjektveiviseren.no/god-praksis/viktige-tema-i-alle-faser/interessenter/hva-er-en-interessent/interessentgrupper	Completed
CF	Configuring
CA	Cancelled
F	Failed
TO	Timeout

PR	Preempted
NF	Node failed

You can use the `pending` command to list only the pending jobs. It lists the jobs in descending priority order, and includes an estimate of when the job will start, when such an estimate is available. `pending` is simply a wrapper around `squeue`, and accepts all options that `squeue` takes. It will also just show the jobs belonging to your own project.

To see the resource situation of the cluster, use the command `qsumm`. It shows how many CPUs (or rather, Processor Equivalents) are used by running jobs and are requested by pending jobs. The output has two lines, one for your project, and one showing the total usage for all projects (including your project). An example output:

```
--------------------------------
Account        Limit  nRun nPend
--------------------------------
p11             1536    20    11
Total           1536  1550   200
--------------------------------
```

See `qsumm --help` for explanations of each column. The output is updated every 5 minutes, so it can take a couple of minutes after jobs are submitted/started/finished before it shows in the `qsumm` output.

General Job Limitations

Default values when nothing is specified

- 1 core (CPU)

The rest (time, mem per cpu, etc.) must be specified.

Limits

- The max wall time is 4 weeks, but do ***not*** submit jobs that will run for more than 7 days unless they implement checkpointing: ***None of the nodes on Colossus have dual power, and we reserve the right to shutdown any node for maintenance at any time with 7 days notice!***
- Max 4500 submitted jobs (running or pending) per project (`--account`) at the same time.

- Max size of job arrays: 4000. That also means that the largest job array index is 4000. Note that an job array of size *N* will count as *N* jobs wrt. the total number of submitted jobs per project.

Scheduling

Jobs are started by priority. Pending jobs are prioritized according to

- Queue time
- How many jobs each user has pending in the queue

Colossus starts jobs by priority + backfilling, so small, short jobs can start earlier than jobs with higher priority, as long as they do not delay the higher priority jobs. In addition, we have added a limit on how many jobs belonging to a user increase in priority over time, to avoid a single user preventing all other users from getting jobs run by submitting a large number of jobs at the same time. In this way, the priority will in effect increase for users running few jobs relative to users running many jobs. This is a trade-off, and we will adjust the limit (currently 10) if we see that the effect is too large/small.

Did you find what you were looking for?

Give us feedback

Published June 21, 2021 10:35 AM - Last modified June 17, 2022 3:01 PM

Share by E-mail

Share on Facebook

Share on Twitter

UNIVERSITY OF OSLO

CONTACT INFORMATION

Contact us
Find us

ABOUT THE WEBSITE

Cookies
Accessibility statement (in Norwegian only)

RESPONSIBLE FOR THIS PAGE

Web editor USIT

Log in