

Resources Allocation Policy

Resources Allocation Policy

The resources are allocated to the job in a fairshare fashion, subject to constraints set by the queue and resources available to the Project. The Fairshare at Anselm ensures that individual users may consume approximately equal amount of resources per week. Detailed information in the Job scheduling section. The resources are accessible via several queues for queueing the jobs. The queues provide prioritized and exclusive access to the computational resources. Following table provides the queue partitioning overview:

queue	active project	project resources
nodes		
min ncpus*		
priority		
authorization		
walltimedefault/max	— —	qexpExpress queue no none required 2 reserved, 31 totalincluding MIC, GPU and FAT nodes 1 >150 no 1h qprod-Production queue yes > 0 >178 nodes w/o accelerator 16 0 no 24/48h qlongLong queue yes > 0 60 nodes w/o accelerator 16 0 no 72/144h qnvidia, qmic, qfatDedicated queues yes
> 0	23 total qnvidia4 total qmic2 total qfat	16 >200 yes 24/48h qfreeFree resource queue yes none required 178 w/o accelerator 16 -1024 no 12h

The qfree queue is not free of charge**. Normal accounting applies. However, it allows for utilization of free resources, once a Project exhausted all its allocated computational resources. This does not apply for Directors Discretion's projects (DD projects) by default. Usage of qfree after exhaustion of DD projects computational resources is allowed after request for this queue.

The qexp queue is equipped with the nodes not having the very same CPU clock speed.** Should you need the very same CPU speed, you have to select the proper nodes during the PSB job submission. **

- **qexp**, the Express queue: This queue is dedicated for testing and running very small jobs. It is not required to specify a project to enter the qexp. >>There are 2 nodes always reserved for this queue (w/o accelerator), maximum 8 nodes are available via the qexp for a particular user, from a pool of nodes containing **Nvidia** accelerated nodes (cn181-203), **MIC** accelerated nodes (cn204-207) and **Fat** nodes with 512GB RAM (cn208-209). This enables to test and tune also accelerated code or code with higher RAM requirements. The nodes may be allocated on per core basis.

No special authorization is required to use it. The maximum runtime in qexp is 1 hour.

- **qprod**, the Production queue****: This queue is intended for normal production runs. It is required that active project with nonzero remaining resources is specified to enter the qprod. All nodes may be accessed via the qprod queue, except the reserved ones. *>>178 nodes without accelerator are included.* Full nodes, 16 cores per node are allocated. The queue runs with medium priority and no special authorization is required to use it. The maximum runtime in qprod is 48 hours.
- **qlong**, the Long queue****: This queue is intended for long production runs. It is required that active project with nonzero remaining resources is specified to enter the qlong. Only 60 nodes without acceleration may be accessed via the qlong queue. Full nodes, 16 cores per node are allocated. The queue runs with medium priority and no special authorization is required to use it. *> The maximum runtime in qlong is 144 hours (three times of the standard qprod time - 3 48 h).**
- **qnvidia, qmic, qfat**, the Dedicated queues****: The queue qnvidia is dedicated to access the Nvidia accelerated nodes, the qmic to access MIC nodes and qfat the Fat nodes. It is required that active project with nonzero remaining resources is specified to enter these queues. 23 nvidia, 4 mic and 2 fat nodes are included. Full nodes, 16 cores per node are allocated. The queues run with *> very high priority*, the jobs will be scheduled before the jobs coming from the *> qexp* queue. An PI *> needs explicitly* ask support for authorization to enter the dedicated queues for all users associated to her/his Project.
- **qfree**, The Free resource queue****: The queue qfree is intended for utilization of free resources, after a Project exhausted all its allocated computational resources (Does not apply to DD projects by default. DD projects have to request for permission on qfree after exhaustion of computational resources.). It is required that active project is specified to enter the queue, however no remaining resources are required. Consumed resources will be accounted to the Project. Only 178 nodes without accelerator may be accessed from this queue. Full nodes, 16 cores per node are allocated. The queue runs with very low priority and no special authorization is required to use it. The maximum runtime in qfree is 12 hours.

Notes

The job wall clock time defaults to **half the maximum time**, see table above. Longer wall time limits can be set manually, see examples.

Jobs that exceed the reserved wall clock time (Req'd Time) get killed automatically. Wall clock time limit can be changed for queuing jobs (state Q) using the qalter command, however can not be changed for a running job (state R).

Anselm users may check current queue configuration at <https://extranet.it4i.cz/>

anselm/queues.

Queue status

Check the status of jobs, queues and compute nodes at <https://extranet.it4i.cz/anselm/>

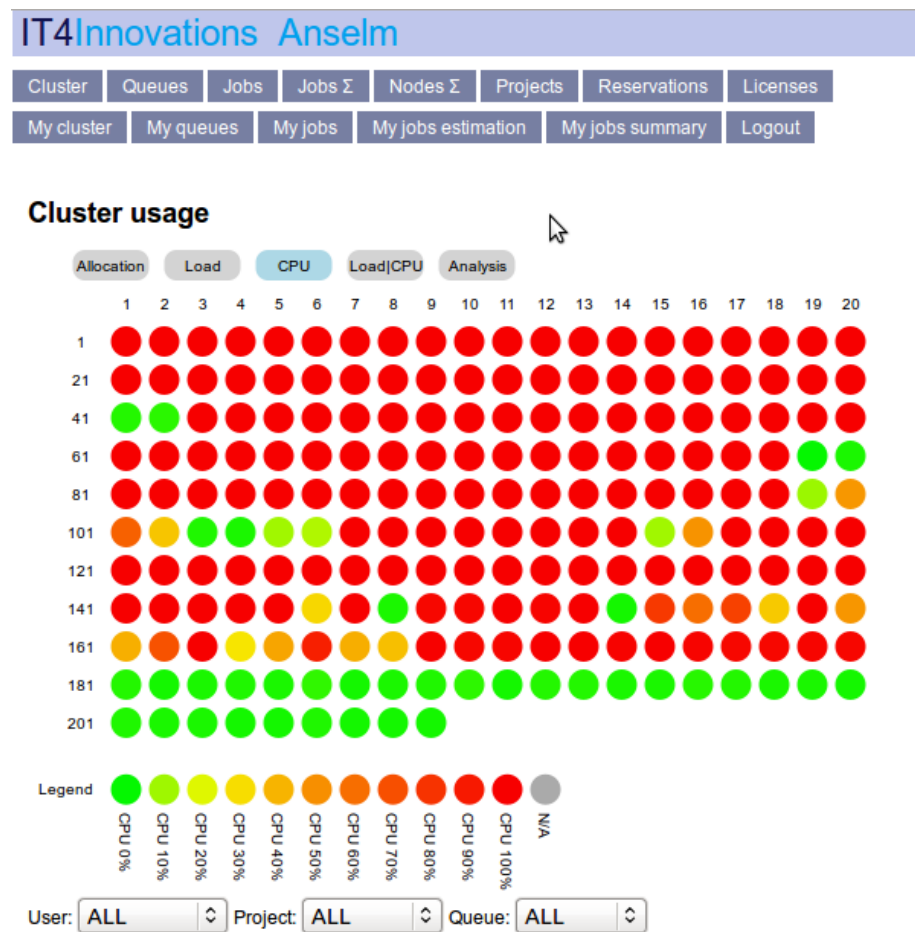


Figure 1: rspbs web interface

Display the queue status on Anselm:

```
$ qstat -q
```

The PBS allocation overview may be obtained also using the rspbs command.

```
$ rspbs
```

Usage: rspbs [options]

Options:

--version	show program's version number and exit
-h, --help	show this help message and exit
--get-node-ncpu-chart	Print chart of allocated ncpus per node
--summary	Print summary
--get-server-details	Print server
--get-queues	Print queues
--get-queues-details	Print queues details
--get-reservations	Print reservations
--get-reservations-details	Print reservations details
--get-nodes	Print nodes of PBS complex
--get-nodeset	Print nodeset of PBS complex
--get-nodes-details	Print nodes details
--get-jobs	Print jobs
--get-jobs-details	Print jobs details
--get-jobs-check-params	Print jobid, job state, session_id, user, nodes
--get-users	Print users of jobs
--get-allocated-nodes	Print allocated nodes of jobs
--get-allocated-nodeset	Print allocated nodeset of jobs
--get-node-users	Print node users
--get-node-jobs	Print node jobs
--get-node-ncpus	Print number of ncpus per node
--get-node-allocated-ncpus	Print number of allocated ncpus per node
--get-node-qlist	Print node qlist
--get-node-ibswitch	Print node ibswitch
--get-user-nodes	Print user nodes
--get-user-nodeset	Print user nodeset
--get-user-jobs	Print user jobs
--get-user-jobc	Print number of jobs per user
--get-user-nodc	Print number of allocated nodes per user
--get-user-ncpus	Print number of allocated ncpus per user
--get-qlist-nodes	Print qlist nodes
--get-qlist-nodeset	Print qlist nodeset
--get-ibswitch-nodes	Print ibswitch nodes
--get-ibswitch-nodeset	Print ibswitch nodeset
--state=STATE	Only for given job state
--jobid=JOBID	Only for given job ID

```

--user=USER          Only for given user
--node=NODE          Only for given node
--nodestate=NODESTATE
                    Only for given node state (affects only --get-node*
                    --get-qlist-* --get-ibswitch-* actions)
--incl-finished      Include finished jobs

```

Resources Accounting Policy

The Core-Hour

The resources that are currently subject to accounting are the core-hours. The core-hours are accounted on the wall clock basis. The accounting runs whenever the computational cores are allocated or blocked via the PBS Pro workload manager (the qsub command), regardless of whether the cores are actually used for any calculation. 1 core-hour is defined as 1 processor core allocated for 1 hour of wall clock time. Allocating a full node (16 cores) for 1 hour accounts to 16 core-hours. See example in the Job submission and execution section.

Check consumed resources

The **it4ifree** command is a part of it4i.portal.clients package, located here: <https://pypi.python.org/pypi/it4i.portal.clients>

User may check at any time, how many core-hours have been consumed by himself/herself and his/her projects. The command is available on clusters' login nodes.

```

$ it4ifree
Password:
      PID      Total   Used   ...by me Free
-----
OPEN-0-0 1500000 400644   225265 1099356
DD-13-1   10000    2606     2606    7394

```