

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			
walltime	— —	qexp no none required 32 nodes, max 8 per user 24 >150	
	no 1 / 1h	qprod yes > 0 >1006 nodes, max 86 per job 24 0 no 24 / 48h	
	qlongLong queue yes > 0 256 nodes, max 40 per job, only non-accelerated		
	nodes allowed 24 0 no 72 / 144h	qmppMassive parallel queue yes > 0 1006	
	nodes 24 0 yes 2 / 4h	qfatUV2000 queue yes > 0 1 (uv1) 8 0 yes 24	
	/ 48h	qfree yes none required 752 nodes, max 86 per job 24 -1024 no 12	
	/ 12h	qvizVisualization queue yes none required 2 (with NVIDIA Quadro	
	K5000) 4 150 no 1 / 2h		

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.

- **qexp**, the : 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.* 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 *: **This queue is intended for normal production runs. It is required that active project with nonzero remaining re-**

sources is specified to enter the qprod. All nodes may be accessed via the qprod queue, however only 86 per job. Full nodes, 24 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 336 nodes without acceleration may be accessed via the qlong queue. Full nodes, 24 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)*
- ***qmpp**, the massively parallel queue. This queue is intended for massively parallel runs. It is required that active project with nonzero remaining resources is specified to enter the qmpp. All nodes may be accessed via the qmpp queue. ^{**} Full nodes, 24 cores per node are allocated. The queue runs with medium priority and no special authorization is required to use it. The maximum runtime in qmpp is 4 hours. An PI > *needs explicitly* ask support for authorization to enter the queue for all users associated to her/his Project.
- **qfat**, the UV2000 queue. *This queue is dedicated to access the fat SGI UV2000 SMP machine. The machine (uv1) has 112 Intel IvyBridge cores at 3.3GHz and 3.25TB RAM. An PI > needs explicitly* ask support for authorization to enter the queue for all users associated to her/his Project.****
- **qfree**, the ^{***}: 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, 24 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.
- **qviz**, the Visualization queue^{**}: *Intended for pre-/post-processing using OpenGL accelerated graphics. Currently when accessing the node, each user gets 4 cores of a CPU allocated, thus approximately 73 GB of RAM and 1/7 of the GPU capacity (default "chunk").* If more GPU power or RAM is required, it is recommended to allocate more chunks (with 4 cores each) up to one whole node per user, so that all 28 cores, 512 GB RAM and whole GPU is exclusive. This is currently also the maximum allowed

allocation per one user. One hour of work is allocated by default, the user may ask for 2 hours maximum.*

To access node with Xeon Phi co-processor user needs to specify that in job submission select statement.

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).

Salomon users may check current queue configuration at <https://extranet.it4i.cz/rsweb/salomon/queues>.

Queue status

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

Display the queue status on Salomon:

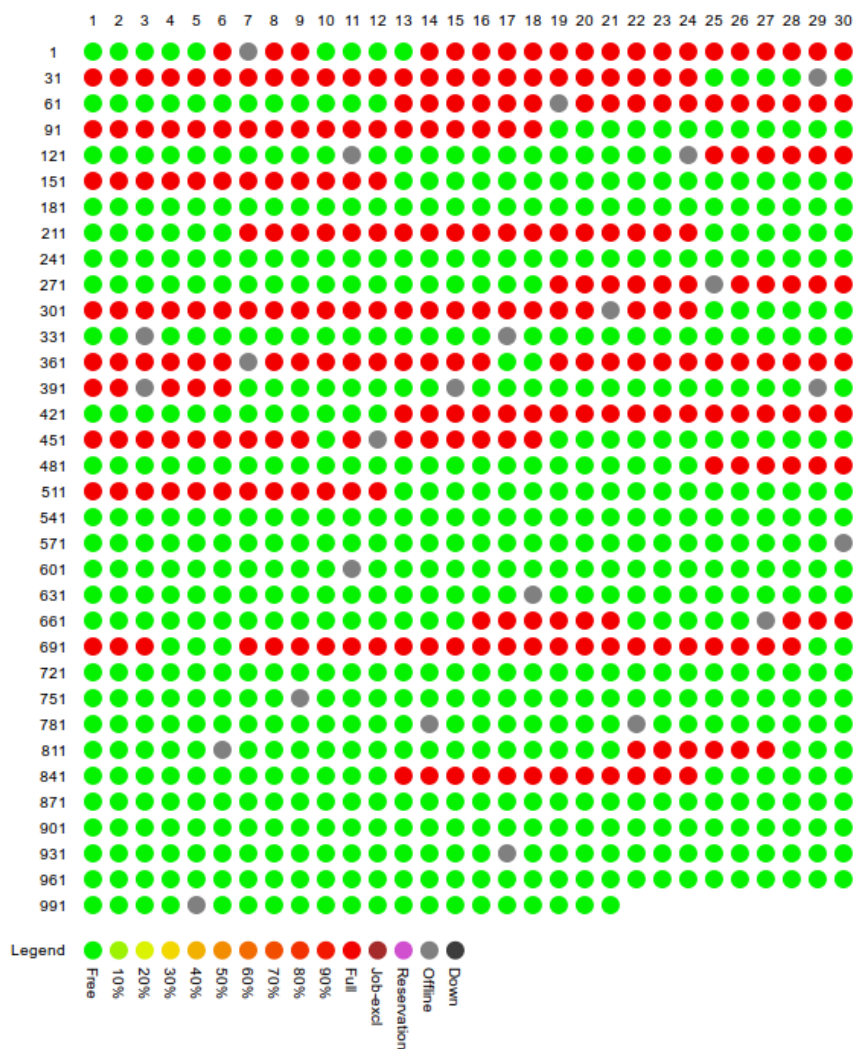
```
$ qstat -q
```

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-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

Allocation Load CPU Load|CPU Analysis



PBS data refreshed 2015-08-14 13:45:27

User: ALL Project: ALL Queue: ALL

Figure 1: RSWEB Salomon

```

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-nodesc  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 -summary          Print summary -get-node-ncpu-
chart                               Obsolete. Print chart of allocated ncpus per node
-server=SERVER      Use given PBS server -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 (24 cores) for 1 hour accounts to 24 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  -----
-----
1099356   DD-13-1   10000   2606   2606   7394   225265

```