

Jobtypes, Queues, Queue Factors, and CPU Points

(Last update: Jul 11, 2025)

- [jobtype](#)
- [Queues](#)
- [CPU Points and Queue Factors](#)
- [Group Limits \(# of CPU cores, # of GPUs, # of Jobs\)](#)

jobtype

A jobtype is assigned for each job. Number of available computation nodes depends on the jobtype. CPU points per hour are also defined by the jobtype. Jobtype is currently defined as follows.

jobtype	definition
largemem	jobs with jobtype=largemem for g16sub, if -j largemem is specified
vnode	jobs with ncpus=64 or ncpus=128 for g16sub, (-np 64 or -np 128) and not largemem
core	jobs with ncpus < 64
gpu	jobs with ngpus > 0

Queues

Computing nodes can be used per node (128 cores), per vnode (64 cores), or per core basis. TypeG nodes are equipped with GPUs.

jobtype	node type	memory	utilization unit	per job limit	total # of vnodes (# of cores)
largemem	TypeF	7.875 GB/core	vnode or node	1-14 vnode(s) (64-896 cores)	28 vnodes (1,792 cores)
vnode	TypeC	1.875 GB/core	vnode or node	1-50 vnode(s) (64-3,200 cores)	1,248+ vnodes (79,872+ cores)
core	TypeC	1.875 GB/core	core	1-63 core(s)	200+ vnodes (12,800+ cores)
gpu	TypeG	1.875 GB/core	core	1-32 GPU(s) 1-16 core(s)/GPU	32 vnodes (2,048 cores 128 GPU)

- The maximum walltime for a job is up to next scheduled maintenance. Only about half of computation nodes can be used for jobs which run more than one week.
- You can omit jobtype in the jobscript except for "jobtype=largemem"; other types can be judged from the resource specification.
- 80 nodes (160 vnodes) of Type C nodes are shared by "vnode" and "core" type jobs.
- Short "vnode" jobs might run on TypeF nodes.
- Short "core" jobs might run on TypeG nodes.
- [In exclusive-use case, the limits above can be loosened.](#) (English page is not yet available, sorry.)

CPU Points and Queue Factors

CPU points per hour (Queue Factor) depend on the jobtype as follows.

jobtype	CPU Queue Factor	GPU Queue Factor
largemem	60 points / (1 vnode * 1 hour)	-
vnode	45 points / (1 vnode * 1 hour)	-

jobtype	CPU Queue Factor	GPU Queue Factor
core	1 point / (1 core * 1 hour)	-
gpu	1 point / (1 core * 1 hour)	60 points / (1 GPU * 1 hour)

- On ccfep, CPU points are calculated from cpu time.
- On other nodes, CPU points are calculated from elapsed time.
- If you run out of CPU points, jobs of your group (running and waiting jobs included) will be removed CPU and your new job submission will be rejected.
- [CPU points usage status can be checked with "showlim" command.](#)
- It never actually costs money.

Point calculation example

- 64 core job for 3 hours => 1 (vnode) * 45 (points/vnode*hours) * 3 (hours) = 135 points
- 8 node job (128*8=1024 cores) for 1 week (168 hours) => 2 (vnodes/nodes) * 45 (points/vnodes*hours) * 8 (nodes) * 168 (hours) = 120,960 points
- 16 core + 1 GPU job for 24 hours => (16 (cores) * 1 (points/cores*hours) + 1 (gpus) * 60 (points/GPU*hours)) * 24 (hours) = 1,824 points

Group Limits (# of CPU cores, # of GPUs, # of Jobs)

The group limits of available number of CPU cores, GPUs are determined from the initially allocated points. Limitation on number of jobs is common for all the groups.

Group Limit			
initially assigned CPU points	# of CPU cores	# of GPUs	# of jobs
7,200,000+	7,680	32	5,000
2,400,000+	6,400	24	
720,000+	4,096	20	
240,000+	3,200	12	
-240,000	768	8	

- There are additional limits for core jobs (ncpus<64), jobtype=largemem jobs, and vnode jobs within single node (total ncpus <= 128). [Those limit values can be checked with "jobinfo -s" command.](#)
- Group limit is determined from initially allocated CPU points. CPU points from additional resource requests are not considered in principle.
- The limit values may be changed depending on the congestion status of the queue. The current limit values can be shown with "jobinfo -s" command.