

Resource Allocation and Job Execution

To run a job, computational resources for this particular job must be allocated. This is done via the PBS Pro job workload manager software, which efficiently distributes workloads across the supercomputer. Extensive informations about PBS Pro can be found in the official documentation here, especially in the PBS Pro User's Guide.

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. The resources are accessible via several queues for queueing the jobs. The queues provide prioritized and exclusive access to the computational resources. Following queues are available to Anselm users:

- **qexp**, the Express queue
- **qprod**, the Production queue****
- **qlong**, the Long queue, regula
- **qnvvidia**, **qmic**, **qfat**, the Dedicated queues
- **qfree**, the Free resource utilization queue

Check the queue status at <https://extranet.it4i.cz/anselm/>

Read more on the Resource Allocation Policy page.

Job submission and execution

Use the **qsub** command to submit your jobs.

The qsub submits the job into the queue. The qsub command creates a request to the PBS Job manager for allocation of specified resources. The **smallest allocation unit is entire node, 16 cores**, with exception of the qexp queue. The resources will be allocated when available, subject to allocation policies and constraints. **After the resources are allocated the jobscript or interactive shell is executed on first of the allocated nodes.**

Read more on the Job submission and execution page.

Capacity computing

Use Job arrays when running huge number of jobs. Use GNU Parallel and/or Job arrays when running (many) single core jobs.

In many cases, it is useful to submit huge (>100+) number of computational jobs into the PBS queue system. Huge number of (small) jobs is one of the most effective ways to execute embarrassingly parallel calculations, achieving best runtime, throughput and computer utilization. In this chapter, we discuss the the recommended way to run huge number of jobs, including **ways to run huge number of single core jobs**.

Read more on Capacity computing page.