

## Job scheduling

### Job execution priority

Scheduler gives each job an execution priority and then uses this job execution priority to select which job(s) to run.

Job execution priority is determined by these job properties (in order of importance):

1. queue priority
2. fairshare priority
3. eligible time

### Queue priority

Queue priority is priority of queue where job is queued before execution.

Queue priority has the biggest impact on job execution priority. Execution priority of jobs in higher priority queues is always greater than execution priority of jobs in lower priority queues. Other properties of job used for determining job execution priority (fairshare priority, eligible time) cannot compete with queue priority.

Queue priorities can be seen at <https://extranet.it4i.cz/rsweb/salomon/queues>

### Fairshare priority

Fairshare priority is priority calculated on recent usage of resources. Fairshare priority is calculated per project, all members of project share same fairshare priority. Projects with higher recent usage have lower fairshare priority than projects with lower or none recent usage.

Fairshare priority is used for ranking jobs with equal queue priority.

Fairshare priority is calculated as

$$MAX\_FAIRSHARE * (1 - \frac{usage_{Project}}{usage_{Total}})$$

Figure 1:

where MAX\_FAIRSHARE has value 1E6, usage<sub>Project</sub> is cumulated usage by all members of selected project, usage<sub>Total</sub> is total usage by all users, by all projects.

Usage counts allocated corehours (ncpus\*walltime). Usage is decayed, or cut in half periodically, at the interval 168 hours (one week). Jobs queued in queue qexp are not calculated to project's usage.

Calculated usage and fairshare priority can be seen at <https://extranet.it4i.cz/rsweb/salomon/projects>.

Calculated fairshare priority can be also seen as Resource\_List.fairshare attribute of a job.

### Eligible time

Eligible time is amount (in seconds) of eligible time job accrued while waiting to run. Jobs with higher eligible time gains higher priority.

Eligible time has the least impact on execution priority. Eligible time is used for sorting jobs with equal queue priority and fairshare priority. It is very, very difficult for >eligible time to compete with fairshare priority.

Eligible time can be seen as eligible\_time attribute of job.

### Formula

Job execution priority (job sort formula) is calculated as:

$$1000 * \text{queue\_priority} + \frac{\text{fairshare\_priority}}{1000} + \frac{\text{eligible\_time}}{864000}$$

Figure 2:

### Job backfilling

The scheduler uses job backfilling.

Backfilling means fitting smaller jobs around the higher-priority jobs that the scheduler is going to run next, in such a way that the higher-priority jobs are not delayed. Backfilling allows us to keep resources from becoming idle when the top job (job with the highest execution priority) cannot run.

The scheduler makes a list of jobs to run in order of execution priority. Scheduler looks for smaller jobs that can fit into the usage gaps around the highest-priority

jobs in the list. The scheduler looks in the prioritized list of jobs and chooses the highest-priority smaller jobs that fit. Filler jobs are run only if they will not delay the start time of top jobs.

It means, that jobs with lower execution priority can be run before jobs with higher execution priority.

It is **very beneficial to specify the walltime** when submitting jobs.

Specifying more accurate walltime enables better scheduling, better execution times and better resource usage. Jobs with suitable (small) walltime could be backfilled - and overtake job(s) with higher priority.

### **Job placement**

Job placement can be controlled by flags during submission.