How are jobs charged?

This page details how are jobs charged, and how to figure out how many UCs do your jobs consume.

Computational Hours (HCs) vs. Computational Units (UCs)

The central concept to understand how jobs are charged are **Computational Units** (UCs, from its acronym in Catalan).

Every group with access to our HPC service will be granted a number of UCs. That is the upper limit of consumption for a group in a given year, and both the assigned limit and the number of UCs consumed so far can be checked with the command

```
consum
```

This information is also printed when logging into our cluster.

Our cluster is heterogeneous, and in order to both rationalise resource usage and establish an accounting equivalence between different architectures, all charges are translated from computational hours (HCs) into computational units (UCs).

- **Computational hours** are the straightforward **wallclock** utilisation of resources. A job, in any particular architecture, will be defined by its wall-clock length (in hours) times the number of CPUs it has been assigned. Therefore, a job using 6 CPUs for 2.2 hours will consume, in total, 6*2.2=13.2 HCs.
- **Computational units** are a universal equivalent that allows us to measure and charge jobs in different architectures against the same account. This is done by introducing a conversion factor for each **partition** that serves as an equivalence table between architectures:

Partition	Conversion factor
std	1 UC/HC
std-fat	1.5 UC/HC
mem	2 UC/HC
gpu	1 UC/HC
knl	0.5 UC/HC

In simple terms, UCs act as a virtual *currency* in which jobs are charged, and conversion factors are the *price* of a single computational hour (1 CPU * 1 hour) for each architecture.

What is charged for, and what isn't?

The only resource that is charged for is CPU time. The cost of any job, in any architecture, is determined by the number of CPUs it uses times the wall-clock length of the job.

Note that this means that all of the following elements are not taken into account when charging a job:

- Memory use (memory utilisation is determined by the queue, but is not charged separately - see *How to request memory* for more details)
- GPU use (GPU utilisation itself is free of charge - GPU jobs are only charged for the accompanying CPUs - see *How to request GPUs* for more details)
- Storage or disk use
- Communication, data transfer, or other network use
- Actual CPU utilisation (i.e. actual CPU time is not a factor - only wall-clock time is considered)

The formula to find out how much a job costs you is therefore very simple:

UCs charged = CPUs assigned to the job * Job length in hours * Conversion factor of the partition you're using

---

A more detailed look at each architecture

Standard partition (std) - 1 UC/HC

The standard partition is the reference for all conversion factor. As a result, jobs that run on the std partition are charged at a rate of 1 UC per HC - which is to say, 1 UC per core per hour.

Standard-fat partition (std-fat) - 1.5 UC/HC

The standard-fat partition contains "fat" nodes with identical CPUs but twice as much memory per core as those in the standard partition. To encourage a rational use of those resources for jobs that actually require them, the conversion factor for this partition is 1.5 UC/HC - meaning that 1.5 UC is charged per core per hour.

Note that this is less expensive than running the same job in twice as many cores in the standard partition to have access to the same amount of memory.

Shared memory partition (mem) - 2 UC/HC

The shared memory partition corresponds to ***canigo1*** and ***canigo2***, our shared memory machines. These machines have a considerably larger memory-per-core ratio, up to 24GB/core. Again, in order to encourage a responsible use of these resources, these are charged at a rate of 2 UCs per core per hour.

GPU partition (gpu) - 1 UC/HC

The GPU partition contains four nodes with identical CPUs and memory to standard nodes, but which additionally contain two Tesla P-100 GPGPUs each. The GPUs themselves are free of charge, but to ensure a responsible use of resources, each GPU is only allocated together with an entire CPU socket containing 24 cores (see *How to request GPUs* for more details). The conversion factor is 1 UC per core per hour, but taking into account that jobs in the ***gpu*** partition are only assigned multiples of 24 cores, in practice these jobs are charged at an effective rate of 24 UCs per each set of P-100 GPU + 24-core socket per hour.

KNL partition (knl) - 0.5 UC/HC

This partition contains four Intel Knight's Landing (KNL) nodes with specialty, highly parallel CPUs tailored to certain tasks. To ensure an efficient use of those resources, jobs can only be assigned whole nodes of 68 physical cores. Since the conversion factor is 0.5 UC/HC, in practice these jobs are charged at an effective rate of 34 UCs per node per hour.

Related articles

- *What disk storage locations are available?*
- *Which partition should I use for my jobs?*
- *Can I use my LSF scripts?*
- *How are jobs charged?*
- *write error: Disk quota exceeded*