How to request memory

This article lists the memory limits for jobs launched through the Slurm job manager, and suggests how to request memory resources for different types of jobs.

Available memory per node:

1. Standard partition:
    a. Thin nodes (*pirineus1-44, -p std*): 192 GB/node (4GB/core)
    b. Fat nodes (*pirineus45-50, -p std-fat*): 384 GB/node (8GB/core)
2. Shared memory partition (*canigo1-2, -p mem*): 4.6 TB/node (24 GB/core)
3. GPGPU partition (*pirineusgpu1-4, -p gpu*): 192 GB/node (4 GB/core)
4. KNL partition (*pirineusknl1-4, -p knl*): 384 GB/node (8 GB/core)

How to request memory:

- Memory is not charged as a separate resource.
- Jobs are assigned the fraction of the node's memory corresponding to the fraction of CPUs requested; for example, if you request four cores in the ***std*** partition, your job will be allocated 4*4=16GB of memory.
- Alternatively, you can request a total memory size, and your job will be assigned enough cores to fulfil that requirement; be mindful of the number of nodes that translates into.
- If you can limit or estimate the memory-per-core requirement of your program, the ***std***, ***std-fat*** and ***mem*** partitions are structured in such a way that switching partitions to a more memory-intensive one is less expensive than requesting additional, unnecessary cores in a less memory-intensive partition.
- It is sensible to demand slightly less in-program memory than the memory demanded to Slurm, to account for auxiliary processes, operating system, etc. We recommend to leave a couple of GBs per job free for this purpose.

There are two ways of specifying memory requirements in Slurm: demanding memory per core or demanding total memory per node.

Total memory per node is demanded with option **--mem=size[unit]**, where *unit* is one of ***M*** for megabytes, ***G*** for gigabytes or ***T*** for terabytes.

```
sbatch --mem=48G [other parameters] your_script.slm # Request a total memory of 48 GB per node
sbatch --mem=1T [other parameters] your_script.slm # Request a total memory of 1 TB per node
```

If no unit is specified, the value is interpreted as a number of **megabytes**.

The upper limit for this value is the total memory of the node (see table above). Note that if more than one node is requested, it is not possible to demand different memory sizes for different nodes – the same memory size will be requested in each node.

Memory per core is demanded with option **--mem-per-cpu=size[unit]**, where again unit is one of *M* for megabytes, ***G*** for gigabytes or ***T*** for terabytes.

```
sbatch --mem-per-cpu=6G [other parameters]          # Request a total memory of 6 GB per core
```

If the value requested for this option is above the per-core memory limit of the partition (see table above), **--cpus-per-task** is automatically defined so that enough cores are assigned to each task to satisfy the memory requirement: for instance, if a job sent to a thin node of the ***std*** partition (48 cores, max. 4 GB per core) specifies -n 4 and --mem-per-cpu=8G, it will be interpreted as a job with 4 tasks, each running on 2 cores with 4GB per core.

As a final note, the Slurm implementation of memory limits relies on **Resident Set Size (RSS)**; if you need to fine-tune your program's memory requirements, you might be interested in reading more about how that metric is computed.

> **Restricted queues**
>
> Remember that jobs sent to the ***gpu*** partition are always assigned whole 24-core sockets, and jobs sent to the ***knl*** partition are always allotted whole nodes. Jobs sent to the ***std***, ***std-fat*** and ***mem*** partitions, on the other hand, are free to specify any number of cores, and allow for greater flexibility.

Best practices:

- Prioritise less memory-intensive partitions if possible, as they tend to have a higher turnout. Request the partition with the smallest memory-per-core profile that you can fit your job in.
- If you manually specify memory requirements in excess of the partition default, try to incorporate and utilise the additional cores rather than have them idly stand by, if possible.
- Don't demand exactly as much memory as your calculation requires. Reserve a portion of your requested memory for system processes; 1-3 GB is fairly reasonable, depending on your calculation.

Related articles

- *How to run GUI applications using GPUs and VNC*
- *Benchmarks 2022*
- *How to request GPUs*
- *How to request memory*
- *How to run Gromacs*