

# Tutorial: Memory Requests in Slurm

SubMIT Users Group Meeting 3/5/2024 10:00am

<https://indico.mit.edu/event/1017/>



# Outline: New Additions to Users Guide

- FAQ: "How can I see the resources each node has to offer in Slurm?"
  - <https://submit.mit.edu/submit-users-guide/running.html#how-to-see-the-available-resources>
  
- FAQ: "How much memory should I request in Slurm?"
  - <https://submit.mit.edu/submit-users-guide/running.html#requesting-memory>



# "How can I see the resources each node has to offer in Slurm?"

```
sinfo -Ne -O "PARTITION:.20,NodeHost:.10,StateLong:.11,NodeAIOT:.15,CPUSState:.15,Memory:.9,AllocMem:.9"
```

<https://submit.mit.edu/submit-users-guide/running.html#how-to-see-the-available-resources>

## How to see the available resources

The *sinfo* command can give information about the Slurm partitions and nodes. For detailed information about this command, view its manual page by typing *man sinfo*.

In particular, to view the resources in the subMIT Slurm cluster, the following command can be handy

```
sinfo -Ne -O "PARTITION:.20,NodeHost:.10,StateLong:.11,NodeAIOT:.15,C
```

This will list each node on a separate line. As described in *man sinfo*, the CPUS column gives the count of the node's CPUs in each state: "A/I/O/T" ("Allocated/Idle/Other/Total"). The MEMORY column gives the total memory for each node, while the ALLOCMEM gives the amount of memory which is currently allocated on that node. Thus, with this command, you can see the total inventory of resources on each node, as well as what happens to be available at the moment.



PARTITION	HOSTNAMES	STATE	NODES(A/I/O/T)	CPUS(A/I/O/T)	MEMORY	ALLOCMEM
submit*	submit00	reserved	0/1/0/1	0/32/0/32	96076	0
submit*	submit01	mixed	1/0/0/1	6/42/0/48	191694	3072
submit*	submit02	mixed	1/0/0/1	2/126/0/128	128383	1024
submit*	submit03	mixed	1/0/0/1	2/30/0/32	96457	1024
submit*	submit04	mixed	1/0/0/1	4/36/0/40	128547	2048
submit*	submit05	mixed	1/0/0/1	6/58/0/64	128392	3072
submit*	submit06	inval	0/0/1/1	0/0/64/64	128393	0
submit*	submit07	idle	0/1/0/1	0/40/0/40	128543	0
submit*	submit08	mixed	1/0/0/1	2/46/0/48	191694	1024
submit-gpu	submit20	mixed	1/0/0/1	2/46/0/48	128393	65536
submit-gpu	submit21	idle	0/1/0/1	0/48/0/48	128395	0
submit-gpu	submit22	idle	0/1/0/1	0/48/0/48	128395	0
submit-gpu-alma9	submit23	idle	0/1/0/1	0/48/0/48	127836	0
submit*	submit50	mixed	1/0/0/1	2/62/0/64	128393	1024
submit*	submit51	mixed	1/0/0/1	14/50/0/64	128393	7168
submit*	submit52	idle	0/1/0/1	0/64/0/64	128393	0
submit*	submit53	mixed	1/0/0/1	8/56/0/64	128393	4096
submit*	submit54	mixed	1/0/0/1	2/62/0/64	128393	1024
submit*	submit55	mixed	1/0/0/1	6/58/0/64	128393	3072
submit*	submit56	mixed	1/0/0/1	6/58/0/64	112265	3072
submit*	submit57	idle	0/1/0/1	0/64/0/64	128393	0
submit*	submit58	mixed	1/0/0/1	6/58/0/64	128393	3072
submit*	submit59	mixed	1/0/0/1	2/62/0/64	128393	1024
submit-gpu1080	submit60	idle	0/1/0/1	0/24/0/24	128332	0
submit-gpu1080-alma9	submit62	idle	0/1/0/1	0/24/0/24	128332	0
submit-gpu1080-alma9	submit63	idle	0/1/0/1	0/24/0/24	128332	0
submit-gpu1080-alma9	submit64	idle	0/1/0/1	0/24/0/24	128332	0
submit-gpu1080-alma9	submit66	idle	0/1/0/1	0/24/0/24	128332	0
submit-gpu1080-alma9	submit67	down*	0/0/1/1	0/0/24/24	128332	0
submit-gpu1080-alma9	submit70	down*	0/0/1/1	0/0/24/24	128332	0

# "How much memory should I request in Slurm?"

<https://submit.mit.edu/submit-users-guide/running.html#requesting-memory>

CPUs & Memory are "consumable resources" in Slurm

- These are reserved for your job (entire node is not)
- If you request more than you need, you can unnecessarily block other jobs from running
  - You can block your own jobs as well as other users'
  - It is OK & recommended to request a bit more *memory* than you need, though ... just not a lot
- Jobs using more memory than requested will be *killed*
  - but the offending job can still disrupt other jobs if not killed soon enough
  - so you still need to be careful to request enough memory
- Best practices: It is **recommended** to request a bit more memory than you actually need, so as to allow a "**safety cushion**" for variations ... so that a job is not killed unnecessarily
- Request memory using --mem OR --mem-per-cpu
  - e.g. adding "#SBATCH --mem=2G" to a job file requests 2GB/node
  - Note: --mem is memory per node!



# "How much memory should I request in Slurm?"

<https://submit.mit.edu/submit-users-guide/running.html#requesting-memory>

If you are unsure of your memory requirements ...

1. start an interactive slurm job with a significant memory request

```
$ salloc --mem=10g
```

2. prepend /usr/bin/time -v to the command you wish to run in your job:

```
$ /usr/bin/time -v python example.py
```



(Alternatively, instead of doing an interactive job, put the above line inside a batch job script and submit it using sbatch, then look at the error output file.)

3. Look at "Maximum resident set size" in the output.

```
Command being timed: "python example.py"
User time (seconds): 23.29
System time (seconds): 2.29
Percent of CPU this job got: 99%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:25.65
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 7934008
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 1809783
Voluntary context switches: 11
Involuntary context switches: 47
Swaps: 0
File system inputs: 8
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 0
```