

Using O2 GPU resources

- [About GPU Resources in O2](#)
- [GPU Partition Limits](#)
- [How to compile cuda programs](#)
- [How to submit a GPU job](#)

About GPU Resources in O2

The first 6 GPU nodes are now available on O2, including: 8 Tesla V100, 8 Tesla M40 and 16 Tesla K80 GPUs. To list information about all the nodes with GPU resources you can use the command:

```
login01:~$ sinfo --Format=nodehost, cpusstate, memory, gres | grep
'HOSTNAMES\|gpu'
HOSTNAMES          CPUS(A/I/O/T)      MEMORY             GRES
compute-g-16-254   0/32/0/32          373760             gpu:teslaV100:4
compute-g-16-255   0/32/0/32          373760             gpu:teslaV100:4
compute-g-16-175   11/9/0/20          257548             gpu:teslaM40:4
compute-g-16-176   18/2/0/20          257548             gpu:teslaM40:4
compute-g-16-194   5/15/0/20          257548             gpu:teslaK80:8
compute-g-16-177   0/24/0/24          257548             gpu:teslaK80:8
```

GPU Partition Limits

The following limits are applied to this partition in order to facilitate a fair use of the limited resources:

GPU hours

The amount of GPU resources that can be used by each user at any time in the O2 cluster is measured in terms of **GPU hours / user**, currently there is an **active limit of 160 GPU hours for each user**.

For example at any time each user can allocate* at most 1 GPU card for 120 (due partition wall time limit), 2 GPU cards for 80 hours, 16 GPU cards for 10 hours or any other combination that does not exceed the total GPU hours limit.

* as resources allow

Memory

The total amount of memory, from all running GPU jobs, that each user can get allocated is set to 420GB

CPU cores

The total amount of CPU cores, from all running GPU jobs, that each user can get allocated is set to 34

Those limits will be adjusted as we migrate additional GPU nodes from the older cluster to O2.

How to compile cuda programs

In most cases a cuda library and compiler module must be loaded in order to compile cuda programs. To see which cuda modules are available use the command `module spider cuda`, then use the command `module load` to load the desired version. Currently only the latest version of Cuda toolkit (V 9) is available

```
login01:~ module spider cuda
```

```
-----  
-----  
-----  
cuda: cuda/9.0  
-----  
-----
```

You will need to load all module(s) on any one of the lines below before the "cuda/9.0" module is available to load.

```
gcc/6.2.0
```

Help:

For detailed instructions, go to:

<https://developer.nvidia.com/about-cuda>

```
login01:~ module load gcc/6.2.0 cuda/9.0
```

How to submit a GPU job

To submit a GPU job in O2 you will need to use the partition **gpu** and must add the flag **--gres=gpu:1** to request a GPU resource. The example below shows how to start an interactive bash job requesting 1 CPU core and 1 GPU card:

```
login01:~ srun -n 1 --pty -t 1:00:00 -p gpu --gres=gpu:1 bash
```

```
srun: job 6900282 queued and waiting for resources
```

```
srun: job 6900282 has been allocated resources
```

```
compute-g-16-176:~
```

While this other example shows how to submit a batch job requesting 2 GPU cards and 4 CPU cores:

```
login01:sbatch gpujob.sh
Submitted batch job 6900310
```

where gpujob.sh contains

```
#-----
-----
#!/bin/bash
#SBATCH -c 4
#SBATCH -t 6:00:00
#SBATCH -p gpu
#SBATCH --gres=gpu:2

module load gcc/6.2.0
module load cuda/9.0

./deviceQuery #this is just an example

#-----
-----
```

It is also possible to request a specific type of GPU card by using the **--gres** flag. For example **--gres=gpu:teslaM40:3** can be used to request 3 GPU Tesla M40 cards. Currently two GPU types are available: **teslaM40** and **teslaK80**