Submitting GPU jobs on ACCRE

For more information and instructions of using ACCRE visit:
https://www.vanderbilt.edu/accre/documentation/parallel/

Follow these steps to submit jobs on GPU nodes

Step 1: Find which of your groups has access to GPU

In order to find your user group that has GPU access use 'slurm_groups' and find the one with 'pascal' or 'maxwell' partitions.

```
[gainara@gw345 ~]$ slurm_groups

Accounts Partitions
------------ ------------
sc3260 debug
sc3260 production
sc3260_acc maxwell
sc3260_acc pascal
vuscl debug
vuscl production

You have access to accelerated GPU resources.
As a usage example, if you wanted to request 2 GPUs for a job with
account "sc3260_acc" on the partition "pascal",
then you would add the following lines to your SLURM script:

#SBATCH --account=sc3260_acc
#SBATCH --partition=pascal
#SBATCH --gres=gpu:2
```

Step 2: Decide if you need interactive access to a GPU node or if you want to submit a job through the scheduler

Interactive jobs

If you only need one GPU/CPU node, use the 'salloc' command to get access to this node, after which you can try running codes multiple time on the same architecture.

- You should change the group name to your group
- You can change the type of GPU from pascal to maxwell
- Specifying how much memory per node is not required

```
salloc --partition=pascal --account=sc3260_acc --gres=gpu:1 --time=0:10:00 --mem=20G
```

Once access has been granted to the GPU node, compile and execute your code in an interactive fashion.

Load the CUDA compiler:

```
module load GCC/5.4.0-2.26
module load CUDA/8.0.61
```

Compile the code

```
make
```

Run the code

```
[gainara@gpu0017 tiled]$ ./mat_multiply
kernel time (ms) : 544.26697
```

Job submission

If you want to submit your job on one GPU/CPU node or run your code on multiple nodes (either multiple CPU and one GPU or multiple CPU+GPU nodes) you can use ‘sbatch‘ to submit your script.

```
[gainara@gpu0023 tiled-GPU-version]$ cat run_my_code.sh
#!/bin/bash
#SBATCH --partition=pascal
#SBATCH --account=sc3260_acc
#SBATCH --gres=gpu:1
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=0:10:00
#SBATCH --output=gpu-job.log


module load GCC/5.4.0-2.26
module load CUDA/8.0.61

make
./mat_multiply

[gainara@gpu0023 tiled-GPU-version]$ sbatch run_my_code.sh
```

Once your code finished running, the results will be generated in the gpu-job.log file in the current directory.

```
[gainara@vm-infr-portal tiled]$ cat gpu-job.log
kernel time (ms) : 530.67419
```