PACE is easier than you think to use. Here’s one way to do it.

Steps:

1. Create job script.
2. Create slurm script
3. Upload your files
4. SSH into PACE, submit slurm script to queue
5. Wait
6. Get results
7. (Optional) Using interactive sessions

1. You have some file that does a job you want on the cluster. Let’s call it test.py.

2. Make a Slurm script. Slurm scripts have a .sub extension.

This is an example of a slurm script. In the cluster, it should be added to same folder as the job you want to put in queue.

============

```
#!/bin/bash
#SBATCH -JSlurmPythonExample                  # Job name
#SBATCH --account=hive-gburdell3              # Tracking account
#SBATCH -n4                                   # Number of cores required
#SBATCH --mem-per-cpu=1G                      # Memory per core
#SBATCH -t15                                  # Duration of the job (Ex: 15 mins)
#SBATCH -phive                                # Queue name (where job is submitted)
#SBATCH -oReport-%j.out                       # Combined output and error messages file
#SBATCH --mail-type=BEGIN,END,FAIL            # Mail preferences
#SBATCH --mail-user=gburdell3@gatech.edu      # E-mail address for notifications
cd $SLURM_SUBMIT_DIR                          # Change to working directory

module load anaconda3/2022.05                 # Load module dependencies
srun python test.py                           # Example Process
```

==============

The above is a slurm script that does not request GPU. The below requests GPU. Note the different options.

==============

```
#!/bin/bash
#SBATCH -JGPUExample                          # Job name
#SBATCH -Ahive-gburdell3                      # Charge account
#SBATCH -N1 --gres=gpu:1                      # Number of nodes and GPUs required
#SBATCH --gres-flags=enforce-binding          # Map CPUs to GPUs
```

```
#SBATCH --mem-per-gpu=12G                    # Memory per gpu
#SBATCH -t15                            # Duration of the job (Ex: 15 mins)
#SBATCH -phive-gpu                        # Partition name (where job is submitted)
#SBATCH -oReport-%j.out                    # Combined output and error messages file
#SBATCH --mail-type=BEGIN,END,FAIL                # Mail preferences
#SBATCH --mail-user=gburdell3@gatech.edu          # e-mail address for notifications

cd $HOME/slurm_gpu_example                      # Change to working directory created in
$HOME

module load tensorflow-gpu/2.9.0                 # Load module dependencies
srun python $TENSORFLOWGPUROOT/testgpu.py gpu 1000  # Run test example
==============
```

3. Now you have your beautiful scripts. Add them to the cluster.

Make a directory on the cluster to hold this file:

```
  ssh [your_gatech_name]@login-ice.pace.gatech.edu mkdir -p yourusername/yourdirectory
```

Copy your local file(s) to the cluster

```
  scp -r test.py slurm.sub [your_gatech_name]@login-
ice.pace.gatech.edu:yourusername/yourdirectory
```

==

***Beware. Before scp * ing all your local files to PACE, know what files are in the directory you are in. ***

==

4. SSH to PACE
   `ssh [your_gatech_name]@login-ice.pace.gatech.edu`
   Cd your way to your directory where you uploaded the files. Submit the sbatch file to pace with “`sbatch slurm.sub`”
5. Wait patiently. You can check the status of your job with the “squeue” command.
6. After you see that the job is finished, the results will in the directory of the slurm job (or wherever you instructed it to output results).
7. Let’s say you don’t want to submit an sbatch job, you want to run an interactive GPU session. You can do this in the command line or via an .sh script.

Command line:

```
salloc -A hive-gburdell3 -N1 --mem-per-gpu=12G -phive-gpu-short -
t0:15:00 --gres=gpu:1 --gres-flags=enforce-binding
```

This is allocating for an interactive job using the `hive-gburdell3` account on the `gpu-short` queue.

It will send a response such as:

```
salloc: Pending job allocation 1187
salloc: job 1187 queued and waiting for resources
```

You will then wait until the request is available to you.

```
salloc: job 1187 has been allocated resources
salloc: Granted job allocation 1187
salloc: Waiting for resource configuration
salloc: Nodes atl1-1-03-002-35 are ready for job
---------------------------------------
Begin Slurm Prolog: Aug-03-2022 08:01:10
Job ID:    1187
User ID:   gburdell3
Account:   hive-gburdell3
Job name:  interactive
Partition: hive
---------------------------------------
```

You now have GPU resources.

One other way to do it is to create a .sh script to handle this for you.

```
./pace_interactive.sh
     [wait till the node is acquired]
```

Example code for pace_interactive.sh

“ ” ”

```
#!/bin/sh
salloc -A hive-gburdell3 -N1 --mem-per-gpu=12G -phive-gpu-short -t0:15:00 --gres=gpu:1 --gres-flags=enforce-binding
exit 0
```

“””

===

This was written with help from the official PACE documentation: https://gatech.service-now.com/technology?id=kb_article_view&sysparm_article=KB0042003 and CSE6220 project 1 instructions (not publicly available). If you have any questions or problems please refer to the official documentation.