Submitting a job to SoC slurm

To submit to a job to slurm, you need to prepare two bash scripts, a slurm script (in this example we call `slurm_job.sh`) and your training script (`train.sh`). This document will cover how to prepare the slurm script and submit the slurm job. You can then submit the job to one of the three types of queues (partitions): standard, medium, or long. Choose the partition that suits your need. The partitions have specifications as follows:

Partition Name	Default Running Time	Max Running Time	Priority Factor
standard	5 mins	10 mins	4x
medium	30 mins	3 hours	2x
long	3 hours	3 days	1x

Currently, each user is permitted to have a maximum of 8 running jobs, and 16 queued jobs.

1. ssh to one of slurm login nodes (xlog0/xlog1/xlog2) and install Miniconda.
2. Create a Miniconda virtual environment, activate it, and install the packages you need.
3. Remember to activate your virtual environment before submitting a job, or add the following command at the start of your training script (`train.sh`) to automatically activate it.
   ```
   source {path to conda activation script}
   conda activate {conda environment name or path}
   ```
   The conda activation script path depends on where you install it, but by default it looks like `/home/m/mrqorib/miniconda3/etc/profile.d/conda.sh`
4. Check the available nodes by running `sinfo`. The `sinfo` command will show you the nodes for each partition along with their states. Choose nodes that have the state `idle` (ideal) or `mix`. Do not choose the ones that are `down`. The hardware specification of each node is described at https://dochub.comp.nus.edu.sg/cf/guides/compute-cluster/hardware. The current strongest nodes (fastest GPU and largest GPU memory) are the **xgpg** and **xgph** nodes.

```
mrqorib@xlog0:~/projects/BART-GEC$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
standard*    up      10:00      1    mix xgph5
standard*    up      10:00     38   idle amdgpu[0-1],xcnb[0-9],xcnc[1-9],xgpc[0-2],xgpd[0,2],xgpe[0-2],xgph[0-4,6-9]
standard*    up      10:00      1   down xgpd1
medium       up    3:00:00     11    mix xgpc3,xgpd[3-4],xgpe[3-4],xgph[5,10-14]
medium       up    3:00:00      5  alloc xcnb[10-14]
medium       up    3:00:00     24   idle amdgpu1,xcnb[5-9],xcnc[5-14],xgpc[2,4],xgpd2,xgpe2,xgph[6-9]
medium       up    3:00:00      1   down amdgpu2
long         up 3-00:00:00      1  down* amdgpu3
long         up 3-00:00:00     15    mix xgpc3,xgpd[3-4],xgpe[3-4],xgph[10-19]
long         up 3-00:00:00      6  alloc xcnb[10-14,16]
long         up 3-00:00:00     13   idle xcnb[17-19],xcnc[10-15,17-19],xgpc4
long         up 3-00:00:00      1   down amdgpu2
```

5. Prepare your slurm script as follows (the script must start with `#!/bin/sh`)
   ```
   #!/bin/sh

   #SBATCH --time={execution_time_in_minutes}
   #SBATCH --job-name={job_name}
   #SBATCH --mail-type={email_notification}
   #SBATCH --mail-user={email_address}
   #SBATCH --gpus={number_of_gpus_requested}
   #SBATCH --partition={partition_type}

   srun train.sh
   ```

6. An example of a slurm_job.sh that runs for 3 hours (180 minutes) on 1 NVIDIA A100 40GB GPU is as follows:

```
#!/bin/sh

#SBATCH --time=180
#SBATCH --job-name=myjob
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --mail-user=youremail@comp.nus.edu.sg
#SBATCH --gpus=a100:1
#SBATCH --partition=medium

srun train.sh
```

   There are only 10 xgpg nodes that have NVIDIA A100 40GB GPU. If your program can fit in 10 GB GPU memory, you can use the xgph node by changing the `--gpus` argument from "`--gpus=a100:1`" to "`--gpus=a100mig:1`". The cluster has many more xgph nodes, so it is recommended that you use xgph if it fits your needs. You can also use older GPUs by replacing the argument with "`--gpus=v100:1`" or other GPU types in this document (Hardware Selection): https://dochub.comp.nus.edu.sg/cf/guides/compute-cluster/slurm-info

7. Note that it is **important** to specify the running time and the number of GPUs. If you do not specify the running time, it will use the default time (5 minutes for the standard queue, 30 mins for the medium queue, and 3 hours for the long queue), and if you do not specify the GPU, the job will not have access to a GPU.
8. After preparing the slurm script (e.g., `slurm_job.sh`), submit the job by running `sbatch slurm_job.sh`. After you successfully submit a job, the command will give you the job ID.
9. Check the status of your job with the command: `squeue -u {your_username}`. The status (column ST) can have two types of value, R for running and PD for pending (queueing). If it's no longer in the table, that means the job has finished. You can cancel your job by running `scancel {job_ID}`.
10. You can check the output of your job on the `slurm-{job_ID}.out` file.

More information on the SoC slurm system can be found at

https://dochub.comp.nus.edu.sg/cf/guides/compute-cluster/slurm-quick

https://dochub.comp.nus.edu.sg/cf/guides/compute-cluster/slurm-new