

**Name :** Sudip Konde

**Class :** LY AIEC

**Roll No. :** 2223118

# ~~High Performance Computing Lab~~

## Experiment2:FamiliarizationwithSLURMCommands

### **Objective:**

To gain practical understanding of basic SLURM commands used for job submission, monitoring, and management in a High-Performance Computing (HPC) environment.

---

### **1. Introduction to SLURM**

SLURM (Simple Linux Utility for Resource Management) is a workload manager used in HPC clusters to schedule, allocate, and monitor jobs efficiently. It allows users to submit batch jobs, run interactive sessions, and manage resources on a cluster.

---

### **2. Basic SLURM Commands**

- **Job Submission (sbatch):**

sbatch job\_script.sh

e.g: sbatch my\_job.sh

```
[mit103@login01 Square]$ ls
run_square.sh  square  square.c  square_error.txt  square_output.txt
[mit103@login01 Square]$ sbatch run_square.sh
Submitted batch job 30362
```

- **Job Status (squeue):**

squeue

Example with user filter: squeue -u username

```
[mit103@login01 Square]$ squeue -u mit103
      JOBID PARTITION      NAME      USER ST       TIME  NODES NODELIST(REASON)
      29848      cpu dot_seri    mit103 PD      0:00      1 (QOSMaxWallDurationPerJobLimit)
```

- **Interactive Job (srun):**

srun --pty bash

Example: srun -n 4 --pty bash

- **Cancel Job (scancel):**

scancel

Example: scancel 12345

```
[mit103@login01 Square]$ scancel 30362
[mit103@login01 Square]$
```

- **Job Information:**

scontrol show job <job\_id>

Example: scontrol show job 12345

sacct -j <job\_id>

Example: sacct -j 12345

- **Node Information (sinfo):**

sinfo

```
[mit103@login01 Square]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
standard*   up   4-00:00:00      2  down* rdcn[22,26]
standard*   up   4-00:00:00      1  drain rdcn31
standard*   up   4-00:00:00     33  idle rdcn[01-21,23-25,27-30,32-36]
cpu        up   8-00:00:00      2  down* rdcn[22,26]
cpu        up   8-00:00:00      1  drain rdcn31
cpu        up   8-00:00:00     33  idle rdcn[01-21,23-25,27-30,32-36]
gpu         up   8-00:00:00      1    mix rdgpu01
gpu         up   8-00:00:00      1    idle rdgpu02
[mit103@login01 Square]$
```

---

### 3. ExampleJobScript

A simple SLURM batch script 'my\_job.sh':

```
#!/bin/bash  #SBATCH --job-
name=TestJob  #SBATCH --
output=result.out #SBATCH -
-error=result.err #SBATCH -
-time=00:10:00 #SBATCH --
ntasks=4       #SBATCH --
partition=short  # Load
required modules module load
python/3.10      # Run the
program python my_script.py
```

```
[mit103@login01 Square]$ cat run_square.sh
#!/bin/bash
#SBATCH --job-name=square_job
#SBATCH --output=square_output.txt
#SBATCH --error=square_error.txt
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --time=00:01:00
#SBATCH --partition=cpu

# Load the GCC module (example: change to your cluster's module)
module load gcc/9.3.0

# Run the program
./square

[mit103@login01 Square]$
```

---

#### 4. Notes

- Use 'man ' for detailed information about each SLURM command. Example: man sbatch
  - Job scripts must start with #!/bin/bash and include SLURM directives beginning with #SBATCH.
  - Check the cluster documentation for partition names, maximum resources, and time limits
-