

Scheduler

Learning Goals

- ? What is Slurm and why do we need it?
- ▶ How do I submit and manage jobs?
- ⚙ How do I request the right resources?
- ★ What are the best practices for efficient usage?
- 🛟 Where can I get help and find examples?



Job Scheduler (Slurm)

What is Slurm?

Slurm scheduler automates job management by allocating resources effectively.

Key Features:

- **Job Queuing & Prioritisation:** Slurm maintains a queue of pending jobs, deciding the run order based on priorities (FIFO, fairshare, job size, etc.).
- **Resource Allocation:** Ensures exclusive use of requested resources (e.g., CPU, memory, GPU) for each running job, preventing conflicts.
- **Policy Enforcement:** Enforces cluster policies such as maximum runtime limits, user quotas, and fair distribution among user groups.
- **Job Monitoring & Logging:** Provides information on job progress and resource usage.
- **Dependency Management:** Allows users to set dependencies, enabling workflows with multiple sequential job execution.

[More on Using Slurm on CoSTAR](#)



Slurm workload manager



How Slurm Works – User Perspective

Slurm Entities:

Job: Your unit of work (single process or parallel tasks).

Node: A physical machine (you may get all or part of it).

Partition: A grouping of nodes with specific time limits and priorities (e.g., shared, debug, high_mem, gpu).

Priority: Determines job run order based on factors like fairshare, job age, and size.

Max Runtime Enforcement:

Slurm enforces time limits; jobs exceeding these limits are terminated. Always set an appropriate --time for your job.



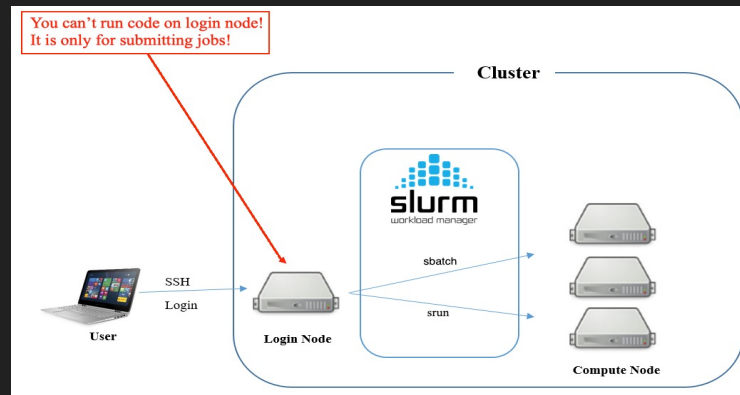
How Slurm Works – User Perspective

Basic Workflow:

1. Log in to the cluster.
2. Prepare your job script or interactive commands.
3. Submit the job to Slurm (sbatch for scripts, srun for interactive).
4. Monitor the job using commands like squeue.
5. Job executes until completion or time limit.
6. Retrieve results upon job completion (.out files).

Interactive vs. Batch Jobs:

- Interactive jobs (srun): Ideal for testing and debugging, can run for the max time of the partition the session runs on.
- Batch jobs (sbatch): Suitable for production tasks running unattended.



Slurm commands & Tips

Common Slurm Commands:

- sbatch: Submit batch jobs.
- srun: Run interactive sessions.
- squeue: Check job status.
- scancel: Cancel jobs.
- sacct: View detailed job accounting and history.
- sinfo: Overview cluster node and partition status.



Writing a Slurm Job Script (Batch Jobs)

A Slurm job script is a shell script with `#SBATCH` directives at the top that define resource needs:

Key `SBATCH` Directives:

- `#SBATCH --job-name=<name>`: Name your job for easier identification.
- `#SBATCH --partition=<name>`: Specify the queue (partition) to submit to (e.g., `shared`, `gpu`).
- `#SBATCH --nodes=X`: Number of nodes required.
- `#SBATCH --cpus-per-task=<N>`: Number of CPU cores allocated to the job.
- `#SBATCH --mem=<amount>`: Memory needed per node (e.g., `--mem=4G`).
- `#SBATCH --time=DD-HH:MM:SS`: Set a wall time limit for the job.
- `#SBATCH --gpus=<N>`: Number of GPUs required.
- `#SBATCH -o <file>` and `-e <file>`: Define file paths for standard output and error.

After the directives, you add the shell commands to execute your program:

echo "helloworld" or python test.py

Submit the script using the `sbatch` command.



Best Practices

Efficiency Tips:

- Only request necessary resources.
- Select appropriate partitions based on job requirements.
- Regularly monitor job resource usage.
- Stage large data files to appropriate storage to optimise performance.



Support & Resources

1.  [Main CoSTAR docs \(Scheduler\)](#)
2.  [CoSTAR: Slurm Examples](#)
3.  [CoSTAR Support Portal](#)
4.  [CoSTAR MS Teams channel](#)

