

PRIMEROS TRABAJOS EN SLURM

SBATCH, SQUEUE Y MONITOREO BÁSICO

Equipo docente ClústerLab

13 de agosto de 2025

- Enviar el primer trabajo en lote con `sbatch`.
- Consultar la cola con `squeue` y `sacct`.
- Interpretar estados: **PENDING**, **RUNNING**, **COMPLETED**.
- Introducir arreglos de trabajos y límites de recursos.

1. Script `sbatch` mínimo.
2. Opciones comunes: `-p`, `-N`, `-n`, `-mem`, `-time`.
3. Seguimiento con `squeue` y `sacct`.
4. Ejemplo Python: calcular π con Monte-Carlo.
5. Actividad práctica grupal.

SCRIPT hello.slurm

```
1  #!/usr/bin/env bash
2  #SBATCH -p main      # partición
3  #SBATCH -N 1         # nodos
4  #SBATCH -n 4         # tareas
5  #SBATCH --time=01:00 # límite
6  #SBATCH --job-name=hello
7
8  hostname
9  sleep 5
```

Enviar con:

```
1  $ sbatch hello.slurm
```

```
1 $ squeue -u $USER
2 JOBID PARTITION NAME   USER   ST TIME   NODES NODELIST(REASON)
3 123    main      hello jper PD 0:00    1    (Resources)
```

- Estado PD: pendiente. R: ejecutándose.

```
1 $ sacct -j 123 --format=JobID,State,Elapsed,MaxRSS
```

Útil para facturación y análisis de memoria.

```
1 $ sbatch --array=1-10%3 pi_mc.slurm
```

Ejecuta 10 réplicas Monte-Carlo, max 3 simultáneas.

```
1 import random, sys
2 N = int(sys.argv[1])
3 inside = sum((random.random()**2 + random.random()**2) < 1 for _ in range(N))
4 print(4 * inside / N)
```

Script `pi.slurm`:

```
1 #!/usr/bin/env bash
2 #SBATCH -p main -n 1 -t 00:02:00 -o pi_%A_%a.out
3 python3 pi.py 1000000
```

1. Envía `hello.slurm` y confirma ejecución.
2. Modifica límites de tiempo y observa el cambio en `squeue`.
3. Lanza un arreglo Monte-Carlo de π (10 tareas).
4. Comparte la desviación estándar de los resultados en el canal.

- `sbatch` es la puerta de entrada a Slurm.
- `squeue` y `sacct` permiten diagnosticar trabajos.
- Arreglos facilitan barrer parámetros.

- *Slurm Quick Start Guide* – SchedMD.
- <https://slurm.schedmd.com/sbatch.html>.
- <https://slurm.schedmd.com/queue.html>.