

Ejecutar trabajos en el cluster

Lanzar una ejecución

Para enviar un trabajo a Slurm se puede hacer de 3 formas diferentes: usando una sesión interactiva, lanzando la aplicación en tiempo real o por medio de un script de ejecución.

En cualquier caso, siempre habrá que informar a Slurm qué recursos se están solicitando.

Ejecución de un trabajo con sesión interactiva

Mediante el comando `salloc` Slurm le dará al usuario un nodo de cómputo donde, de manera interactiva, puede trabajar y ejecutar software cualquier software que necesite. Es una opción muy útil para probar software nuevo o nuevos datos con los que trabajar, sin necesidad de lanzar trabajos a la cola, con el riesgo de que fallen nada más empezar.

- Solicitamos un nodo:

```
salloc -N 1
```

- Solicitamos un nodo de una partición concreta, se le asigna un nombre al trabajo y una duración:

```
salloc -p <partition> -J <job_name> -t <HH:MM:SS> -N 1
```

- Una vez concedido el nodo y el trabajo, recibirás el siguiente mensaje:

```
salloc: Granted job allocation 968978
```

- Y cuando el trabajo comienza:

```
srun: Job 968978 step creation temporarily disabled, retrying
srun: Step created for job 968978
The authenticity of host 'node1404-3.hpc.iter.es (10.0.14.15)' can't be established.
RSA key fingerprint is 6b:6f:2e:33:2c:92:11:d6:d6:9d:aa:55:9d:c1:a9:bb.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node1404-3.hpc.iter.es,10.0.14.15' (RSA) to the list of known hosts.
xxxxxx@node1404-3.hpc.iter.es's password:
Welcome to node1404-3. Deployment version 0.5-hpc. Deployed Wed May 13 19:58:32 WEST 2022.
```

Info

Una vez tenemos un nodo mediante `salloc`, podemos, desde otra terminal, acceder a dicho nodo por ssh para tener varias sesiones abiertas sobre el mismo nodo y trabajar en varias cosas a la vez.

- Para salir de la sesión interactiva:

```
[xxxxxx@node0101-1 ~]$ exit
```

Una vez salgamos del nodo desde la terminal donde hicimos `salloc`, Slurm liberará el nodo y dará por completado el trabajo. Obviamente, una vez salgamos, se cerrarán de forma automática todas aquellas sesiones extras que hayamos abierto mediante SSH.

Ejecutar un trabajo en tiempo real

Con el comando `srun` es posible lanzar un trabajo a la cola directamente.

```
srun -p <partition> -J <job_name> -t <days-HH:MM:SS> <aplicacion>
```

Más opciones disponibles:

Nombre	Dirección IP
-p <partition>	partición en la que se ejecutarán los trabajos
-N <nodes>	nodos que se solicitan para la ejecución
-n=<num_tasks>	número de tareas
--tasks-per-node=<number>	número de tareas a ejecutar en cada nodo (en combinación con -N)
-J <job_name>	nombre que se le asignará al trabajo
-t <days-HH:MM:SS>	tiempo estimado que durará la ejecución
-d=<type:job_id[:job_id]>	el trabajo depende de otros para ejecutarse (opcional)
-o </path/to/file.out>	fichero para la salida estándar de las ejecuciones

Nombre	Dirección IP
-e </path/to/file.out>	fichero para la salida de errores de las ejecuciones
-D <directory>	directorio en donde se realizará la ejecución
--mail-user=<email>	email al que slurm notificará eventos del trabajo
--mail-type=<eventos>	lista de eventos de notificación

Ejecutar en SLURM mediante un script

El comando `sbatch` envía un trabajo a la cola para ser ejecutado por uno o varios nodos, según los recursos que se hayan especificados.

```
sbatch [-p <partition>] [-J <job_name>] [-t <days-HH:MM:SS>] mi_script.sh
```

La estructura más básica de un script es la siguiente.

```
#!/bin/bash

#SBATCH -J <job_name>
#SBATCH -p <partition>
#SBATCH -N <nodes>
#SBATCH --tasks=<number>
#SBATCH --cpus-per-task=<number>
#SBATCH --constrains=<node architecture> # sandy, ilk (icelake)... architecture
#SBATCH -t <days-HH:MM:SS>
#SBATCH -o <file.out>
#SBATCH -D .
#SBATCH --mail-user=<cuenta_de_correo>
#SBATCH --mail-type=BEGIN,END,FAIL,TIME_LIMIT_50,TIME_LIMIT_80,TIME_LIMIT_90
#####

module purge
module load <modules>

srun <aplicacion>
```