

Computación GPU en TeideHPC

El clúster TeideHPC tiene una serie de nodos que tienen conectadas unidades de procesamiento de gráficos de propósito general (GPGPU) de NVIDIA. Es posible usar herramientas CUDA para ejecutar trabajo computacional en ellas y, en algunos casos de uso, ver aceleraciones muy significativas.

Como explicamos en la [sección cómo ejecutar trabajos](#) podemos utilizar 3 formas diferentes de enviar un trabajo a la cola de trabajos: utilizando una sesión interactiva, iniciando la aplicación en tiempo real o medio de un script de ejecución.

GPU en Slurm

Para solicitar una sola GPU en slurm simplemente agregue la directiva `#SBATCH --gres=gpu:<model>` a su script de envío de trabajo y le dará acceso a una GPU. Para solicitar varias GPU, agregue `#SBATCH --gres=gpu:<modelo>:n`, donde 'n' es la cantidad de GPU.

Entonces, si desea 1 CPU y 2 GPU de nuestros nodos GPU de uso general en la partición 'gpu', debe especificar:

```
#SBATCH -p batch
#SBATCH -n 1
#SBATCH --gres=gpu:<model>:1
```

Si prefiere usar una sesión interactiva, puede usar:

```
salloc -p express --mem 8000 --gres=gpu:<model>:1
```

Mientras está en el nodo GPU, puede ejecutar ***nvidia-smi*** para obtener información sobre las GPU asignadas.

Especificación del tipo de GPU o partición MIG a usar.

Los modelos de GPU actualmente disponibles en nuestro clúster se pueden encontrar [aquí](#) pero como explicamos en [MIG section](#) podemos especificar el tipo de GPU o la partición MIG para usar.

Hay **dos métodos** que se pueden usar. Visite la página de [solicitud recurso GPU y computo](#) para una explición más detallada:

Select GPU using `--constraint=`

```
salloc -p express --mem 8000 --constrains=gpu,a100
```

Seleccionar GPU usando `--gres=gpu::1` or `--gres=gpu::1`

```
salloc -p express --mem 8000 --gres=gpu:a100:1
```

Tenga en cuenta que `--gres` especifica los recursos por nodo, por lo que para el trabajo de varios nodos solo necesita especificar cuántos gpus necesita por nodo.

Listar modelos de GPU y particiones MIG

Para averiguar qué tipos específicos de gpus están disponibles en una partición, muestre la definición de una partición con `scontrol` busque en la categoría **TRES**.

```
scontrol show partition express
```

```
PartitionName=express
...
MaxNodes=UNLIMITED MaxTime=03:00:00 MinNodes=0 LLN=NO
MaxCPUsPerNode=UNLIMITED
Nodes=node0303-2,node0304-[1-4],node1301-[1-4],node1302-[1-4],node1303-[1-4],
...
State=UP TotalCPUs=2424 TotalNodes=88 SelectTypeParameters=NONE
...
DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
TRES=cpu=2424,mem=7565306M,node=88,billing=2424,gres/gpu=79,gres/gpu:1g.
5gb=2,gres/gpu:2g.10gb=1,gres/gpu:3g.20gb=1,gres/gpu:a100=71,gres/gpu:t4=4
```

Nvidia A100

```
gpu:a100
```

a100-mig

- `gpu:1g.5gb`
- `gpu:2g.10gb`
- `gpu:3g.20gb`

Nvidia Tesla T4

- gpu:t4

Scripts de ejemplo para un trabajo de GPU:

- Ejemplo usando una GPU Nvidia A100 completa

```
#!/bin/bash

#SBATCH --partition=batch
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --gres=gpu:a100:1
#SBATCH --mem=8G
#SBATCH --time=1:00:00

module purge
module load CUDA/12.0.0

nvidia-smi
sleep 20
```

```
sbatch 01_gpu_basic_a100.sbatch
```

- Ejemplo usando una partición MIG en una Nvidia A100

```
#!/bin/bash

#SBATCH --partition=batch
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --gres=gpu:2g.10gb:1
#SBATCH --mem=16G
#SBATCH --time=1:00:00

module purge
module load NVHPC/22.11-CUDA-11.8.0

nvidia-smi
sleep 20
```

```
sbatch 02_gpu_basic_mig_partition.sbatch
```

Más ejemplos.

Visita nuestro github https://github.com/hpciter/user_codes