

Single core

```
#SBATCH --ntasks=X donde X => 1
```

Multi-core

```
#SBATCH --ntasks=X donde X => 1
```

```
#SBATCH --cpus-per-task=Y donde Y > 1
```

OMP, MPI y Trabajos Híbridos

Es posible encontrar diferentes tipos de paralelismo: OpenMP, OpenMPI o una solución híbrida combinando ambos. Por un lado, utilizará OpenMP para el paralelismo dentro de un nodo multinúcleo.

Dado que es una implementación de subprocesos múltiples, se debe definir una variable OMP_NUM_THREADS. En cambio, si el paralelismo es entre nodos, usarás OpenMPI. Entonces, en nuestro script de muestra, será necesario especificar la cantidad de nodos, la cantidad de tareas en cada nodo y cada CPU.

OpenMP

```
#SBATCH --cpus-per-tasks=X  
donde X > 1  
export OMP_NUM_THREADS=X  
donde X > 1
```

OpenMPI

```
#SBATCH --ntasks=X donde  
X => 1  
#SBATCH --cpus-per-task=Y  
donde Y>1  
#SBATCH --nodes=Z donde Z  
=>2  
#SBATCH --ntasks-per-  
node=W donde W>1  
#SBATCH --ntasks-per-  
socket=U donde U>1  
module load OpenMPI/2.0.2-  
GCC-6.3.0-2.27
```

Hybrid

Combine both
options

Arrays de Jobs

Con el uso de arrays puede ejecutar múltiples trabajos con los mismos parámetros. En su script debe especificar la opción `--array`.

Array

```
#SBATCH --array=1-X donde X > 1
```

Trabajos de GPU

Es imprescindible utilizar el parámetro `--gres` para reservar un recurso GPU y cargar el módulo CUDA.

Nvidia A100

```
#SBATCH --gres=gpu:a100:1  
module load CUDA/8.0.61
```

Nvidia Tesla T4

```
#SBATCH --gres=gpu:t4:1  
module load CUDA/8.0.61
```