

# Sistema de colas

## launch.sh

```
sbatch --exclusive -p cola-corta,thinnodes -n 1 -N 1 -c $1 --time 03:59:00 --mem 4G $2 $1 $3 $4 $5
```

- \$1: número nodos
- \$2: script a encolar
- \$3: máscara
  - para 1 thread: 1
  - para 2 threads: 1001
  - para 4 threads: 6006
  - para 8 threads: F00F
  - para 16 threads: FF0FF
  - para 24 threads: FFFFFFF
- \$4, \$5: versión gadget
  - sin indicar \$4 y \$5: versión secuencial
  - -parallel spe: versión especulativa
  - -parallel rep: versión reproducible

## ompLongrun.sh

```
module load gcc/6.3.0
```

```
export GADGET_HOME=$HOME/mareframe/trunk/gadget  
export PATH=$GADGET_HOME:$PATH  
cat /proc/cpuinfo | grep 'model name'  
gcc -v
```

```
cd $HOME/mareframe/trunk/gadgetMerluza2015/model/
```

```
export OMP_NUM_THREADS=$1
```

```
cp params.final input.1
```

```
srunk -n 1 -N 1 -c $1 --cpu-freq=High --cpu_bind=verbose,mask_cpu:$2 gadget -l -main hke.main -i  
input.1 -opt optinfo.long -p params.out_omp$1$4 -nowarnings -seed 1234 $3 $4
```

## Ejemplo de uso

```
sh launch.sh 24 ompLongrun.sh FFFFFFF -parallel spe
```

## Compilar

```
module load gcc/6.3.0  
make gadget
```

## Clases

- Simulated annealing
  - siman.cc: paralelo
  - seq\_optimize\_template.h : secuencial
  - optinfosimann.cc: declaración, lectura parámetros entrada algoritmo
- Hooke and Jeaves
  - hoke.cc: implementación
  - optinfohooke.cc: declaración, lectura parámetros entrada algoritmo
- Particle swarm Optimization
  - pso.cc: implementación
  - pso.h: definición variables
  - optinfopso.cc: declaración, lectura parámetros entrada algoritmo
- optinfo.h: declaraciones variables algoritmos
- interrupinterface.cc: ctrl+c
- ecosystem.cc: clase encargada del ecosistema (optimizar, simular, imprimir resultado)
- maininfo.cc: lectura parámetros entrada aplicación
- gadget.cc: clase principal

## Fichero opciones

optinfo.long

[simann] → opciones para simulated and annealing

[hooke] → opciones para Hooke and Jeaves

[pso] → opciones pso