

## ALTAMIRA

Se recomienda la lectura la [?guía de usuario](#) del Supercomputador Altamira antes de la realización de las prácticas.

### Práctica 1

Realizar el cálculo del número Pi mediante openMP y openMPI. El objetivo de esta práctica es comprobar que opción es más eficiente desde el punto de vista del tiempo de ejecución. Para este propósito usaremos los códigos [?PiOpenMP.c](#) y [?PiMPI.c](#).

#### 1. Job OpenMP

Haciendo uso del programa **module** cargar el compilador *gcc*:

```
[user@login1 ~]$ module load gcc
load gcc/4.6.3 (PATH, MANPATH, LD_LIBRARY_PATH)
```

Compilar el programa *PiOpenMP.c* y generar un ejecutable con el nombre *PiOpenMP*.

```
[user@login1 ~]$ gcc PiOpenMP.c -fopenmp -o PiOpenMP
```

#### • Plantilla a ejecutar en el supercomputador:

```
#!/bin/bash
#@ job_name = pi_openmp_%j
#@ initialdir = .
#@ output = pi_openmp_%j.out
#@ error = pi_openmp_%j.err
#@ total_tasks = 16
#@ wall_clock_limit = 00:02:00

export OMP_NUM_THREADS=16

echo "Numero de procesos: ${SLURM_NPROCS}"
echo "Numero de nodos: ${SLURM_JOB_NUM_NODES}"
echo "Numero de procesos por nodo: ${SLURM_JOB_CPUS_PER_NODE}"
echo "Nodos: ${SLURM_JOB_NODELIST}"

time ./PiOpenMP
```

#### • Envío del job PiOpenMP:

```
$ module load gcc
load gcc/4.6.3 (PATH, MANPATH, LD_LIBRARY_PATH)
$ msubmit PiOpenMP_template
Submitted batch job 621356
```

#### 1. Job openMPI:

Haciendo uso del programa **module** cargar el compilador *OPENMPI*:

```
[user@login1 ~]$ module load OPENMPI
load OPENMPI/1.8.3 (PATH,LD_LIBRARY_PATH,MANPATH)
```

Compilar el programa *PiMPI.c* y generar un ejecutable con el nombre *PiMPI*:

```
[user@login1 ~]$ mpicc PiMPI.c -o PiMPI
```

#### • Plantilla a ejecutar en el supercomputador:

```
#!/bin/bash
#@ job_name = pi_mpi_%j
#@ initialdir = .
#@ output = pi_mpi_%j.out
#@ error = pi_mpi_%j.err
#@ total_tasks = 16
#@ wall_clock_limit = 00:02:00

echo "Numero de procesos: ${SLURM_NPROCS}"
echo "Numero de nodos: ${SLURM_JOB_NUM_NODES}"
echo "Numero de procesos por nodo: ${SLURM_JOB_CPUS_PER_NODE}"
echo "Nodos: ${SLURM_JOB_NODELIST}"

time mpirun -np 16 ./PiMPI
```

- Envió del job PiMPI:

```
$ module load OPENMPI
load OPENMPI/1.8.3 (PATH,LD_LIBRARY_PATH,MANPATH)
$ mnsbmit PiMPI_template
Submitted batch job 621357
```

- ¿Qué job ha tardado menos? ¿Qué ocurre si el job openMPI, con el mismo número de procesos(16),se ejecuta en 2 dos nodos en vez de 1?