

## Sesión de Prácticas 2

Computación de Altas Prestaciones en Bioinformática  
Master Universitario en Bioinformática para Ciencias de la Salud

Curso 2021/2022



1 Trabajo con varios nodos del FTII

2 Ejercicio



# 1 Trabajo con varios nodos del FTII

## 2 Ejercicio



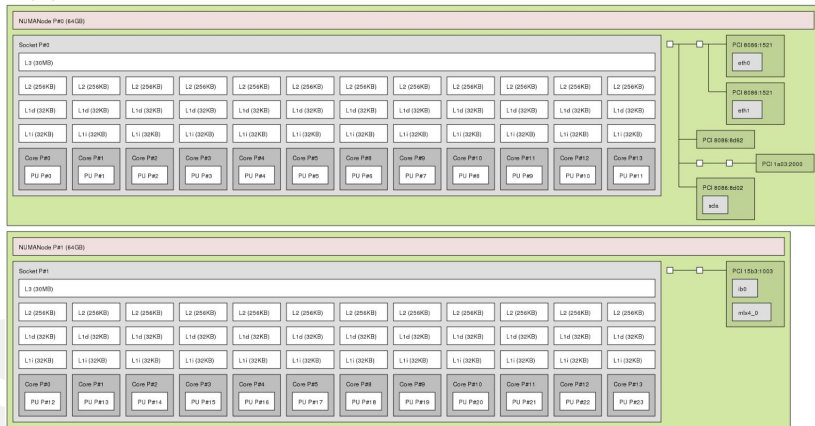
# Descripción del sistema (I)

## 306 nodos de cómputo general

- Dos procesadores Intel Haswell 2680v3.
  - Doce núcleos cada uno.
- 128GB de memoria.
- Disco de 1TB.
- Conexión InfiniBand FDR de alto ancho de banda y baja latencia.

# Descripción del sistema (y II)

Machine (128GB)



# Datos del script

- `#SBATCH -N`. Número de nodos.
- `#SBATCH -n`. Número de tareas.
- `#SBATCH -c`. Número de núcleos por tarea.
- `#SBATCH --ntasks-per-node=`. Número de tareas por nodo.
- `#SBATCH -t`. Tiempo requerido en hh:mm:ss.



# Compiladores de MPI

## Código abierto

- OpenMPI

## Comerciales

- Intel MPI
- Bull MPI

- Cada compilador tiene su módulo asociado.

# Módulos (I)

- Conjuntos de variables de entorno creadas por el equipo del CESGA para ayudar en la compilación y ejecución de programas y aplicaciones sin necesidad de conocer donde se encuentran ciertos compiladores y bibliotecas.
- Divididos en secciones.
- Muy útil para ciertos compiladores (MPI) o aplicaciones (bioinformática).





# Módulos (y II)

## Comandos de trabajo con módulos

- `module avail`. Lista los módulos disponibles.
  - Si no hay ninguno cargado, lista los de la sección principal.
  - Al cargar algún módulo se pueden habilitar otros de secciones relacionadas.
- `module load`. Carga un cierto módulo.
- `module unload`. Descarga un módulo previamente cargado.
- `module list`. Lista los módulos cargados actualmente.
- `module spider`. Lista todas las aplicaciones disponibles.
  - `module spider app`. Muestra si app está disponible.
- `module help`. Ayuda de todas las opciones de module.
  - `module help app`. Ayuda del módulo de la app.

# Compilación y ejecución con MPI

## OpenMPI

- 1 `module load gcc`
- 2 `module load openmpi`

## Intel MPI

- 1 `module load intel`
- 2 `module load impi`

```
mpicc -o HolaMundoMPI HolaMundoMPI.c &&run  
HolaMundoMPI
```

# Particiones en el Finis Terrae II

- Conjuntos lógicos de los nodos que se pueden utilizar para lanzar los trabajos.
- Se especifican en el script con `#SBATCH -p`.
- `thin-shared`
  - Por defecto.
  - Solo para trabajos dentro de un nodo (hasta 24 núcleos).
- `thinnodes`
  - Para ejecuciones que requieren más de un nodo.
- `gpu-shared`.
  - Para uso de hasta cuatro GPUs.
- `iphinodes`.
  - Para obtener los nodos con Intel Xeon Phi.
- Especiales para el curso:
  - `-p thinnodes`
  - `--reservation=CAPB_15Feb`

# Sistemas de colas en el Finis Terrae II

- Asociación de límite de recursos a un usuario.
- Por defecto la cola default.
- Se puede cambiar de cola con `#SBATCH --qos=nombre_cola`.
- Todos los usuarios pueden acceder a default, interactive y urgent.
- Comando `batchlim` para conocer los límites.



- 1 Trabajo con varios nodos del FTII
- 2 Ejercicio



# Objetivos

- Aprendizaje del uso de la cola thinnodes.
- Compilación, ejecución y análisis de escalabilidad de una aplicación bioinformática paralelizada para sistemas de memoria distribuida.



# Aplicación bioinformática de estudio

## MSAProbs-MPI

- Mismos resultados que MSAProbs pero paralelizado con MPI/PThreads para trabajar en sistemas híbridos.
- Disponible en Moodle (con Makefile adaptado para el FTII y OpenMPI) y públicamente en <https://sourceforge.net/projects/msaprobs/>
- Información y uso en: <http://msaprobs.sourceforge.net/homepage.htm>
- Puede usar los mismo conjuntos de entrada que MSAProbs.

# Tareas

- ❶ Transferir la aplicación al FTII.
- ❷ Descomprimir la aplicación (`tar -xvf`).
- ❸ Cargar los módulos
  - `module load gcc`
  - `module load openmpi`
- ❹ Compilar la herramienta (`make`).
- ❺ Ejecutar la herramienta lanzando a colas (`./msaprobs-mpi entrada -num_threads X -o salida`).
  - `#SBATCH -n` debería ser igual al número de procesos MPI.
  - `#SBATCH -c` debería ser igual al número de hilos por proceso (obligatorio indicarlo también al programa como parámetro).
  - Ejemplo para todos los conjuntos de entrada y cuatro hilos por proceso: `srun ./msaprobs-mpi BB1101*.tfa -num_threads 4 -o out.txt`
- ❻ Analizar la escalabilidad de la herramienta usando diferentes configuraciones de procesos e hilos.