

Sesión de Prácticas 1

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

Curso 2021/2022



- 1 Descripción del Finis Terrae II
- 2 Trabajo en un nodo del Finis Terrae II
- 3 Ejercicio



- 1 Descripción del Finis Terrae II
- 2 Trabajo en un nodo del Finis Terrae II
- 3 Ejercicio



Información general

Localización

- Instalado en el Centro de Supercomputación de Galicia (CESGA), en Santiago de Compostela.
- Información vía web:
<https://www.cesga.es/es/infraestructuras/computacion/FinisTerra2>
 - Información de estado.
 - Manual de usuario.
 - Direcciones de contacto.

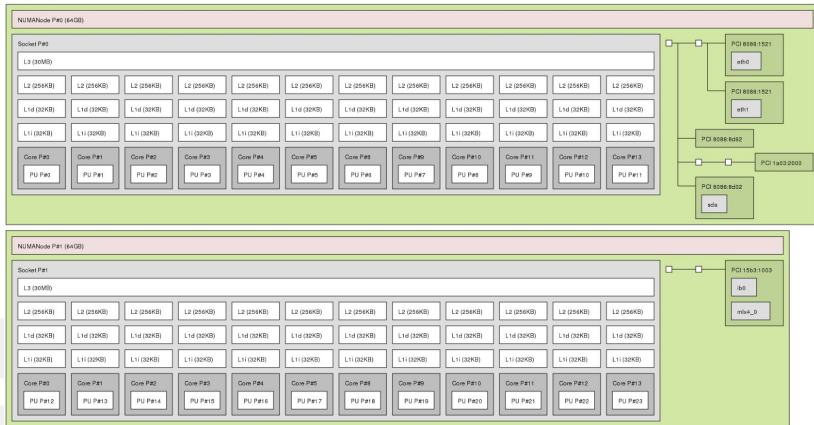
Descripción del sistema (I)

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 (II)

Machine (128GB)



Descripción del sistema (y III)

Nodos especiales

- Cuatro nodos adicionales con GPUs.
 - Cuatro GPUs K80 por nodo.
 - Otros con GPUs más modernas.
- Cuatro nodos de login.
 - No destinados a ejecución de trabajos.
 - Para compilar, mover ficheros y carpetas, enviar trabajos a colas, etc.
 - Límite de ocho horas y 8GB de memoria.

- 1 Descripción del Finis Terrae II
- 2 Trabajo en un nodo del Finis Terrae II
- 3 Ejercicio



Conexión al sistema

- Uso de un cliente de ssh para conectarse a máquinas remotas.
- Dirección `ft2.cesga.es`.
- Usuario entra en nodo de login.

Prácticas y trabajo tutelado

- Cuenta que será válida a lo largo del cuatrimestre.
- Para acceder desde fuera de la universidad darse de alta en el portal de usuarios del CESGA y solicitar acceso VPN.
- `ssh usuario@ft2.cesga.es`.

Transferencia de ficheros y datos

- Uso de un cliente scp o sftp.
- `scp fichero usuario@ft2.cesga.es:/path/`.

Sistemas de ficheros

- `$HOME`. Sistema por defecto útil para almacenar ficheros de código y pequeñas entradas/salidas. Baja velocidad de acceso y soporte de backup. 10GB por usuario.
- `$STORE`. Sistema de gran almacenamiento útil para resultados finales de simulaciones. 100GB por usuario sin soporte de backup.
- `$LUSTRE`. Sistema de almacenamiento paralelo de alta velocidad para ficheros temporales de gran tamaño. Útil para resultados temporales de simulaciones. 3TB por usuario y sin soporte de backup.
- Información de espacio libre con comando `myquota`.

Uso de sistema de colas

- Sistema basado en SLURM para gestión de los recursos.
- Aconsejable crear un script para lanzar los trabajos.
- `sbatch script.sh`

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.

Opciones adicionales

Nodos en exclusiva

- Para que la ejecución de otros trabajos no influya en nuestras medidas de rendimiento.
- `#SBATCH -exclusive`

Salida y error

- Por defecto se combina la salida y el error en un único fichero con formato `slurm-id.out`.
- Se puede especificar la salida y el error con `#SBATCH --error=` y `#SBATCH --output=`

Emails de notificación

- `#SBATCH --mail-type=begin`
- `#SBATCH --mail-type=end`
- `#SBATCH --mail-user=xxx@udc.es`

Particiones

- Conjuntos lógicos de nodos a los que se pueden enviar trabajos.
- Partición por defecto: `thin-shared`.
 - Trabajos que solo requieren un nodo (24 núcleos o menos).

Particiones especiales para prácticas

- El CESGA nos ha reservado una serie de recursos para que podamos trabajar en estas sesiones.
- Solo existirán durante las horas de prácticas.
- `#SBATCH --reservation=CAPB_08Feb`
- `#SBATCH -p thinnodes`

Comandos de SLURM

- `sbatch script`: Envío de un trabajo.
- `scancel id`: Cancelación de un trabajo en ejecución.
- `sinfo`: Información sobre los nodos de SLURM y sus particiones.
- `squeue`: Información sobre los trabajos ejecutando y en cola.
 - `--start`: Hora y fecha estimada de entrada en ejecución.
 - `-t RUNNING/PENDING`: Ver solo trabajos en ejecución o espera.
 - `-u user`: Ver solo trabajos de un usuario.
 - `-p partición`: Ver solo trabajos en una partición.
- `sacct`: Consulta histórico de trabajos.

- 1 Descripción del Finis Terrae II
- 2 Trabajo en un nodo del Finis Terrae II
- 3 Ejercicio



Objetivos

- Familiarización con el entorno de trabajo del Finis Terrae II.
- Compilación, ejecución (tanto en nodo de login como con envío a colas) y análisis en clave de velocidad de ejecución de una aplicación bioinformática paralelizada para sistemas de memoria compartida.



Aplicación bioinformática de estudio (I)

- En las tres sesiones usaremos herramientas para el alineamiento múltiple de secuencias (MSA en inglés).
- Pueden ser secuencias de ADN o proteínas.
- Identifican aquellos grupos de secuencias (tres o más) del conjunto de entrada que presentan un mejor alineamiento.
- Más complejo computacionalmente que alineamiento de dos secuencias (o frente un genoma) porque hay muchas más posibles combinaciones.

Aplicación bioinformática de estudio (II)

MSAProbs

- Realiza el alineamiento usando un modelo de Markov.
- Paralelizado para uso de múltiples núcleos en un mismo nodo (PThreads).
- Estudios que indican que proporciona los mejores alineamientos múltiples.
- Disponible públicamente en Moodle (con modificación para que muestre tiempos de ejecución) y en <https://sourceforge.net/projects/msaprobs/>
- Información y uso en: <http://msaprobs.sourceforge.net/homepage.htm>

Aplicación bioinformática de estudio (y III)

IMPORTANTE: No es necesario conocer al dedillo el funcionamiento interno de la herramienta. Varios estudios ya han demostrado que los resultados de los alineamientos son precisos, así que nosotros analizaremos simplemente el rendimiento desde el punto de vista de la velocidad de ejecución.



Tareas

- ❶ Transferir la aplicación/conjuntos de entrada al nodo de login.
- ❷ Descomprimir la aplicación/conjuntos de entrada (`tar -xvf`).
- ❸ Compilar la herramienta en el nodo de login (`make`).
- ❹ Ejecutar la herramienta en interactivo en el nodo de login (`./msaprobs entrada -num_threads X -o salida`).
 - Ejemplo para todos los conjuntos de entrada y cuatro hilos:
`./msaprobs BB1101*.tfa -num_threads 4 -o out.txt`
- ❺ Ejecutar la herramienta lanzando a colas (mejor utilizar `--reservation=CAPB_4Feb`).
- ❻ Analizar el tiempo necesario para ejecutar esta herramienta para diferente número de conjuntos de entrada (tamaño del problema) usando diferente número de hilos.
 - Se puede utilizar un único script con varias ejecuciones o varios scripts para diferente número de núcleos.