

Sesión 0 – Conexión, cuentas y passwords

El servidor

Utilizaremos el mismo servidor que usan en PAR (Paralelismo): boada. Este servidor empezó con 4 nodos y en cada nodo 2 procesadores Intel Xeon E5645 de 6 cores con 24GB de memoria DDR3 1333 ECC. El servidor ha ido evolucionando, se ha actualizado, ha crecido, ... pero el nombre se mantiene. Actualmente boada tiene 15 nodos.

En este servidor hemos añadido un nodo adicional (boada-10) con 2 procesadores Intel Xeon 43134 2.4 GHz de 16 cores con 128 GB de memoria DDR4. Además, en este nodo hemos instalado 4 tarjetas NVIDIA RTX 3080. Cada RTX 3080 dispone de 8.704 cores de CUDA con 10 GB de memoria GDDR6X (ancho de banda de 760,3 GB/s) y tiene una potencia de cálculo de 29,77 TFLOPs en FP32 (465,1 GFLOPS en FP64).

Normalmente trabajaréis en uno de los nodos interactivos de boada (edición, compilación) y cuando necesitéis trabajar con las GPUs, lo haréis utilizando el sistema de colas. **No se podrá trabajar de forma interactiva con las GPUs.**

Conexión, Cuentas y Passwords

Para trabajar con el servidor necesitáis trabajar desde Linux. Para establecer la conexión con el servidor hay que usar *secure shell*:

```
ssh -X username@boada.ac.upc.edu
```

La opción `-X` será necesaria cuando necesitéis abrir una ventana remota en vuestro escritorio. El username y password os lo proporcionaremos en la primera sesión de laboratorio. Si utilizáis un Mac hay que usar la opción `-Y`, e instalar previamente el programa Xquartz.

Una de las primeras cosas que tendréis que hacer es cambiar el password usando el siguiente comando:

```
ssh -t username@boada.ac.upc.edu passwd
```

El cambio de password es algo lioso, pero tiene su lógica. Supongamos que tenéis como username/password: cudaXX/**123456**, y queréis que el nuevo password sea **abcdef**. Hay que hacer lo siguiente:

```
> ssh -t cudaXX@boada.ac.upc.edu passwd
cudaXX@boada.ac.upc.edu's password: 123456
(current) UNIX password: 123456
Enter new UNIX password: abcdef
Retype new UNIX password: abcdef
```

Hemos puesto en **ROJO** lo que tenéis que escribir vosotros. Fijaos que hay que poner el password viejo 2 veces, y el nuevo otras 2. Repito, lioso, pero tiene lógica (el primer password 12345 es para entrar en boada, el segundo password 12345 es el que os piden por el cambio de password, ...).

Cuando entréis por primera vez, vuestra zona está prácticamente vacía. Podéis organizarla como más os convenga. Para empezar sólo nos queda por saber cómo transferir ficheros entre nuestra máquina y boada. Para copiar datos entre las dos máquinas usaremos **scp**:

```
scp file username@boada.ac.upc.edu:.
scp username@boada.ac.upc.edu:file .
```

El comando **scp** lo hemos de ejecutar **siempre** en nuestra máquina local y **nunca** en boada.

En todas las sesiones colgaremos un fichero tal como éste: **Sesion01.tar**. Antes de empezar hay que desempaquetar el fichero con el siguiente comando:

```
tar xvf Sesion01.tar
```

Uso de las colas

El comando a utilizar para mandar un trabajo a la cola de ejecución es:

```
sbatch job.sh
```

El contenido del fichero `job.sh`¹ podría ser el siguiente:

```
#!/bin/bash

### Directivas para el gestor de colas
#SBATCH --job-name=deviceQuery
#SBATCH -D .
#SBATCH --output=submit-deviceQuery.o%j
#SBATCH --error=submit-deviceQuery.e%j
#SBATCH -A cuda
#SBATCH -p cuda
#SBATCH --gres=gpu:1

export PATH=/Soft/cuda/11.2.1/bin:$PATH
### Este es nuestro programa
./kernel.exe
```

Al acabar la ejecución se generan 2 ficheros: `submit-deviceQuery.oXXXX` y `submit-deviceQuery.eXXXX`. En el primero está la salida de `kernel.exe`, en el segundo los posibles errores y la información de profiling cuando la pidáis. `XXXX` es un contador que varía en cada ejecución.

Otros comandos útiles para tratar con las colas son:

- `squeue`, permite comprobar el estado de las colas
- `scancel <job_id>`, permite eliminar un trabajo del sistema de colas
- `scontrol show node boada-10`, permite ver el estado del nodo boada-10
- `sinfo`, permite ver el estado general todo el cluster

Para obtener más detalles de estos comandos utilizad el `man`.

¹ En todas las sesiones os daremos una versión operativa inicial del `job.sh`