

Clase 0

Familiarización con ClusterUY

Como conectarse:

Linux:

1) dar permiso al archivo de claves: `chmod 600 *`

`ssh -i ruta/usuario_rsa usuario@cluster.uy`

Tambien se puede agregar como clave para no tener que ponerlo todas las veces que vayamos a ingresar:

`ssh-add ~/.ssh/modnummesoescala1_rsa`

`ssh -X usuario@cluster.uy -X` para poder obtener interfaz gráfica

Windows por medio de la aplicación MobaXterm, luego de instalada se inicia igual que en linux

Cluster Uy:

Solamente podrá conectarse desde la máquina que contenga la clave privada correspondiente.

Al generar la clave ssh (rsa) se tendrán dos archivos:

id_rsa - que corresponde a la clave privada, el cual puede copiarse al directorio .ssh del homedir de otra máquina para tener acceso a cluster.uy sin necesidad de generar otro par de claves público-privada

id_rsa.pub - que corresponde a la clave pública.

Cluster Uy:

Las ejecuciones en login están PROHIBIDAS.

<https://cluster.uy/ayuda/tips/>

Al ingresar al clusterUY ingresamos en la maquina Login, esta es na máquina con muy poco poder de procesamiento y dado que es la que realiza la gestión de los usuarios que se conectan con cluster, no está preparada (y no es su función) para soportar tareas que requieran gran cantidad de procesamiento o de memoria por eso siempre debemos trabajar mediante trabajos interactivos, ya sea para descomprimir archivos, copiar, editar, instalar o compilar.

Para iniciar un trabajo interactivo se debe decir el tiempo que voy a necesitar:

```
srun --time=00:30:00 --pty bash -l
```

Tengan cuidado que como siempre conecta en nodos diferentes no les queda guardado el historial de comandos utilizados, es preferible llevar apuntes.

Copiado de archivos grandes: Cuando se necesita copiar un gran volumen de datos y con el fin de ser solidario y no consumir ancho de banda en login se debe utilizar el puerto 10022 de cluster.uy.

Se puede utilizar ssh para copiar directamente de la consola en bash:

```
scp -P 10022 [archivo origen] cluster.uy:[carpeta destino]
```

Repaso comandos linux:

ls listar archivos, **ls -lt** los ordena, me da información de la última fecha de modificación.

pwd ¿donde estoy trabajando?

mkdir crear carpeta

rm -r borrar carpeta con todos los archivos que estén dentro

cp copiar dentro de mi usuario, **scp** copiar a otros usuarios/equipos

cat ver archivo sin entrar en el archivo

mv mover

grep buscar

wget -c descargar reenganchando si se corta

...

Editores:

vi: - esc i para comenzar a escribir

- esc wq para salir y guardar,

- esc q! para salir sin guardar

- %s/acambiar/nuevo/g en todo el texto

vim: En general funciona igual que el vi

nano: Ctrl+x guardar y salir

Ctrl+o guardar

Script para la ejecución de un trabajo:

Los trabajos que son ejecutados fuera de línea son iniciados con el comando `sbatch`. Este comando recibe como argumento un script que contiene la especificación del trabajo a ejecutar y los comandos necesarios para su ejecución. Por ejemplo:

```
#!/bin/bash

#SBATCH --job-name=mitrabajo      #SBATCH --ntasks=1
#SBATCH --mem=2048                #SBATCH --time=6:00:00
#SBATCH --tmp=9G                  #SBATCH --partition=normal
#SBATCH --qos=normal              #SBATCH --mail-type=ALL
#SBATCH --mail-user=mi@correo
```

Más opciones en `man sbatch`.

Una vez finalizada su ejecución, SLURM dejará disponibles el archivo `slurm-<ID>.out` con el contenido de la salida a pantalla de la ejecución donde `<ID>` es el ID asignado al trabajo. Este archivo incluye la salida estándar y la salida de error.

¿Como esta mi trabajo?

Para ver un listado de los trabajos del usuario <usuario>:

```
squeue -u <usuario>
```

Para cancelar la ejecución del trabajo <ID>:

```
scancel <ID>
```

También se puede enviar un trabajo de la siguiente forma:

```
srun --job-name=mitrabajo --time=00:20:00 --ntasks=1 --partition=normal --qos=normal --mem=512 --pty bash -l
```

Acceso a Matlab:

```
/clusteruy/apps/matlab/R2018b/bin/matlab -nodisplay
```

Acceso a Octave:

```
octave
```