Manual de uso del cluster Toko - Versión 2024.04.01

FCEN - UNCuyo

Solicitud de cuenta de usuario

Para pedir una cuenta debe enviar un email a emmanueln_at_gmail_com completando el formulario http://toko.uncu.edu.ar/data/uploads/FORMULARIO_SOLICITUD_TOKO.docx. El reglamento de uso se encuentra en http://toko.uncu.edu.ar/data/uploads/RegCluster_2018.pdf .

Conexión remota al cluster

Para conectarse al cluster debe utilizar el programa ***ssh*** (en windows puede usar el programa putty) con los siguientes datos:

servidor: toko.uncu.edu.ar
puerto: 22
usuario
contraseña

Ejemplo en Linux:
 # ssh wbishop@toko.uncu.edu.ar

Creación de llave SSH

Si no desea ingresar los datos de usuario y contraseña cada vez que se conecta o transfiere archivos, puede generar una 'llave'. Para esto, seguir los siguientes pasos (Linux y MacOS):

Paso 1: Crea la llave en la PC local

```
wbishop@local-host$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/hone/wbishop/.ssh/id_rsa): [press Enter]
Enter passphrase (empty for no passphrase): [press enter]
Enter same passphrase again: [press enter]
Your identification has been saved in /home/wbishop/.ssh/id_rsa.
Your public key has been saved in /home/wbishop/.ssh/id_rsa.pub.
The key fingerprint is:
33:b3:fe:af:95:95:18:11:31:d5:de:96:2f:f2:35:f9 wbishop@local-host
```

Paso 2: Copia la llave a la PC remota

```
wbishop@local-host$ ssh-copy-id -i ~/.ssh/id_rsa.pub wbishop@toko.uncu.edu.ar
wbishop@toko.uncu.edu.ar password:
```

Paso 3: Ingresa a la PC remota vía ssh

```
wbishop@local-host$ ssh wbishop@toko.uncu.edu.ar
Last login: Sun Nov 16 17:22:33 2008 from 192.168.1.2
[Nota: no debería pedir contraseña]
```

Transferencia de archivos utilizando rsync

Los archivos deberán almacenarse dentro de la carpeta scratch (ahí dentro podrá crear todos los directorios que desee). Para transferir archivos de la PC local a la PC remota puede utilizar rsync:

```
rsync -r --progress /Users/wbishop/PATH_al_archivo_o_carpeta_a_transferir
wbishop@toko.uncu.edu.ar:/home/wbishop/scratch/PATH_a_la_carpeta_donde_desea_recibir_los_archivos
```

```
-r permite transferir carpetas, --progress muestra el progreso de la
transferencia y la velocidad de la misma
```

Para realizar la transferencia inversa, de la PC remota a la PC local invierta el comando:

```
rsync -r --progress
wbishop@toko.uncu.edu.ar:/home/wbishop/scratch/PATH_al_archivo_o_carpeta_a_transferir  /Users/wbishop/PATH_a_la_carpeta_donde_desea_recibir_los_archivos
```

Puede enviar múltiples archivos indicando uno detrás del otro (con todo su PATH) separados por un espacio. También puede usar expresiones regulares.

PRECAUCIÓN: se recomienda leer detenidamente la ayuda de ***rsync*** antes de su uso. Colocar el orden del destino/origen de forma invertida o utilizar algun parametro que borre los archivos que no estan en el destino puede provocar la pérdida de datos.

Transferencia de archivos con SSHFS en PC cliente Linux

sshfs permite montar un directorio de Toko en su PC linux y accederlo como si fuera un directorio local. Consulte su distribución de Linux para ver como se instala sshfs, para usarlo debe ejecutar lo siguiente en la consola de su equipo (no en el cluster), modificando su usuario y directorio destino de su PC local:

```
$ sshfs -o reconnect USUARIO@toko.uncu.edu.ar:/home/USUARIO/   /mnt/toko
```

Para más información vea:
https://wiki.archlinux.org/index.php/SSHFS_(Espa%C3%B1ol)

https://www.digitalocean.com/community/tutorials/how-to-use-sshfs-to-mount-remote-file-systems-over-ssh

Transferencia de archivos con Winscp para Windows

Desde una PC cliente Windows puede utilizar la aplicación Winscp. https://winscp.net/

Compilación de códigos

El cluster tiene instalado el compilador GCC 4.8.2 y varias versiones de OpenMPI. Para utilizar MPI es necesario cargar el módulo:

```
$ module list      (lista los módulos actualmente cargados en el entorno del usuario)
$ module avail     (lista los módulos instalados en el sistema)
$ module load openmpi-1.8.8          (carga el módulo para ser utilizado)
$ module unload openmpi-1.8.8        (descarga el módulo)
```

Para poder ejecutar el código compilado es necesario cargar los módulos utilizados en el script desde el que se encolan los trabajos, como se ve en la siguiente sección.

En general, cada usuario es responsable de compilar las librerías y software que necesiten, en caso de necesitar asistencia puede contactar al administrador.

Envío de trabajos

El cluster utiliza el administrador de colas **Slurm**. Los comandos necesarios para utilizarlo son los siguientes:

- **sinfo**: muestra información general sobre las colas (o particiones) y el estado de los nodos del cluster.
- **squeue**: muestra los trabajos encolados en el cluster y el estado en el que se encuentra cada uno. Se lista primero el id del trabajo, la partición en la cual está encolado, el nombre, el usuario que envió el trabajo, el estado que puede ser: R (running), PD (pending), CA (cancelled), CF(configuring), CG (completing), CD (completed), F (failed), TO (timeout), y NF (node failure). Luego el tiempo de ejecución (en caso de que esté running), el número de nodos que esta ocupando y finalmente el listado de nodos que está utilizando.
- **squeue-cpus**: muestra los trabajos encolados informando cuantos CPUs se están utilizando por trabajo.
- **sbatch**: comando para encolar trabajos. Se pueden pasar varios parámetros por línea de comando o utilizar un archivo de submit con la información necesaria para que el trabajo se ejecute en el cluster. En su directorio home hay un archivo de ejemplo (submit.job) para encolar trabajos con sbatch.

- **scancel:** comando para cancelar un trabajo encolado. Debe averiguar el ID del trabajo con *squeue* y luego ejecutar *scancel JOBID.*

En el directorio home de cada usuario encontrará un archivo de ejemplo para enviar trabajos al administrador de colas (Slurm). A continuación se muestra un ejemplo con una breve explicación de cada parámetro:

```
#!/bin/bash

## Dos numerales para comentar.
## Un numeral para pasarle parámetros a SLURM con la palabra SBATCH
## RESPETAR mayúsculas y minúsculas en el nombre de los parámetros
## y directorios

##Nombre del trabajo para identificarlo en la cola de trabajos
#SBATCH --job-name=MI_TRABAJO

## archivo donde guardar el output generado por el comando
#SBATCH --output=/home/wbishop/scratch/MI_TRABAJO.out
#SBATCH --mail-type=ALL

## mail para recibir notificaciones de inicio y fin de los trabajos encolados
#SBATCH --mail-user=wbishop@gmail.com
## Listado de nodos en donde se quiere ejecutar la simulación, puede estar
vacío para que Slurm los elija solo.
#SBATCH --nodelist=toko01

## Cantidad de nodos que se quieren utilizar
#SBATCH --nodes=1

## Cantidad de procesos que se ejecutarán por nodo,
#SBATCH --ntasks-per-node=64

## Número de procesos en los cuales se quiere ejecutar el trabajo
## ntask-per-node * nodes = ntasks.
#SBATCH --ntasks=64

##Partición o cola en la cual se quiere enviar el trabajo.
#SBATCH --partition=XL

#Tiempo estimado de ejecución
#SBATCH --time=1:00:00

##NO BORRAR la ejecución de /etc/profile
. /etc/profile
```

```
## MÓDULOS a cargar
module load openmpi-1.10.0
module load gcc-4.8.2

## Comandos para ejecutar, ej. para MPI
mpirun -np 64 ./lmp_ompi_g++_ompi1.10 -in in.lj > salida64cores.log
```

Luego para encolar el trabajo se ejecuta en una consola
```
$ sbatch trabajo.submit
```

Colas o particiones disponibles

El cluster está configurado con 7 particiones:

- **Small**: hasta 72 horas de ejecución en 4 núcleos de un nodo Toko.
- **Large**: hasta 72 horas de ejecución en 32 núcleos en un nodo Toko.
- **XL**: hasta 72 horas de ejecución en 64 núcleos en un nodo Toko.
- **XXL**: hasta 6 horas de ejecución en 128 núcleos utilizando dos nodos Toko.
- **gpu**: hasta 72 horas de ejecución en 8 núcleos sobre una sola GPU Titan Xp.
- **mini**: hasta 72 horas de ejecución en 8 núcleos / 16 hilos en un Ryzen 2700/3700x con 64GB de RAM.

Hay dos tipos de nodos para ejecutar: *toko0X* son 7 nodos AMD Opteron/Epyc de 64 núcleos por nodo con entre 64GB (toko02,03,04), 128GB(toko01,05,06) y 256GB (toko00) por nodo; Se solicita no hacer ejecuciones en el nodo maestro sin encolar trabajos a través de Slurm.

Hasta ahora, el sistema de colas del cluster permite la ejecución de trabajos de hasta 3 días corridos, sin cuota, sin prioridad (Criterio FIFO, First In - First Out), sin límite de uso en cantidad de trabajos por usuario.

Como toda comunidad que crece con recursos limitados, se impone la necesidad de ir estableciendo y modificando reglas de convivencia/uso de los recursos. Es por esta razón que nos vemos obligados a tomar algunas medidas en estos momentos.

Primeramente, hemos reducido la duración del tiempo máximo de ejecución por trabajo, de 4 días (96 horas) a 3 días (72 horas), de manera que deberán ajustar sus scripts de envío de trabajos en consecuencia, caso contrario, los trabajos no entrarán. Les solicitamos encolar sus trabajos utilizando hasta 2 nodos al mismo tiempo (dos nodos de toko01 a 07 o dos nodos de mini), como máximo, de la siguiente manera, que explicamos con el ejemplo a continuación:

Supongamos que el cluster tiene 5 nodos libres (64 tareas por nodo), donde puedo encolar 10 trabajos de 32 tareas y 3 días c/u, y preciso encolar 20 trabajos de 32 tareas y 3 días c/u. El encolado de los 20 trabajos significa tomar la totalidad del cluster durante 6 días corridos para un solo usuario. En tal caso, solicitamos tomar hasta 2 nodos, enviando 10 trabajos a uno de los nodos, ejemplo toko01, y otros 10 trabajos a otro nodo, ejemplo toko02, cosa que pueden especificar incluyendo la siguiente línea en sus scripts de envío

#SBATCH --nodelist=toko02 (por ejemplo)

De esta forma, estarían tomando 2 nodos completos por 15 días. Dejando los restantes 3 nodos libres para otros usuarios. Casos excepcionales de necesidad de alta carga dedicada son, por supuesto, conversables/planificables.

En resumen, lo que pedimos es que hagan un uso razonable del cluster considerando posibles necesidades de otros usuarios, a efectos de evitar inconvenientes.

Algunos comandos que, en este sentido, les pueden resultar útiles a efectos de elegir los recursos de una forma más eficiente:

Comando: sinfo

Entrega lo siguiente en este momento

```
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
Small up 3-00:00:00 4 mix toko[00,02,04-05]
Small up 3-00:00:00 1 alloc toko06
Small up 3-00:00:00 2 idle toko[01,03]
```

Esto indica que toko01 y 03 están completamente disponibles, podría tomarme completamente uno de ellos sin ningún problema, encolando todo con #SBATCH --nodelist=toko03 (por ejemplo).

Toko06 está totalmente ocupado, puedo encolar ahí pero no entrará hasta que ese trabajo y todos los que estuvieran ya encolados en toko06 terminen.

Toko[00,02,04-05] indican estado mix, es decir, están parcialmente disponibles para que envíe trabajos a los mismos, siempre y cuando se ajusten a los cores disponibles.

Comando: squeue –start

Indicará por usuario y trabajo, cuando está previsto el ingreso de los trabajos a ejecución. Este comando es útil pues me permite prever que recurso se liberará primero, además de los trabajos que tengo delante mío. ***IMPORTANTE:*** *Intenten pedir el tiempo de cómputo que realmente necesitan con un pequeño margen de cobertura, si necesito encolar 10 trabajos que necesitan 8 horas reales, puedo pedir 9 o 10 hs por trabajo, pero si pido 96 hs por cada uno, cuando en realidad toman 8 hs, el sistema no lo sabe, y entonces la estimación que realiza este comando no refleja la realidad.*

Almacenamiento de datos

Actualmente el cluster cuenta con dos opciones para guardar los códigos y resultados de las ejecuciones: el directorio ***home*** de cada usuario y la partición ***/home/USUARIO/scratch***. En el directorio ***home*** del usuario encontrará un directorio que se llama ***scratch***, ese directorio se ve también desde los nodos, los datos que se escriban durante la ejecución del código se escribirán en el servidor maestro (toko). Se recomienda monitorear el espacio disponible en cada uno de

estos directorios con el comando "***df -h***", también puede utilizar el comando "***du -sh***" para ver el tamaño de su directorio home o sobre cualquier subdirectorio del mismo. Además, se solicita por favor que los usuarios retiren los archivos del cluster luego de que no sean necesarios.

Cada directorio ***home*** también tiene acceso a un directorio denominado "***nas***", dicho directorio se encuentra en otro equipo que tiene 36TB de almacenamiento. Se puede utilizar dicho almacenamiento para guardar resultados de ejecuciones luego de que han finalizado. El directorio ***nas*** no se puede ver desde los nodos de ejecución. Los archivos se deben mover desde el nodo maestro, el directorio ***nas*** no se puede utilizar desde scripts de slurm desde los nodos de ejecución.

En resumen:

- ***Directorio home***: para almacenar software, librerías y scripts; tiene poco espacio de almacenamiento (1.6TB compartido).
- ***Directorio scratch dentro de home***: para almacenar datos de entrada y resultados de las ejecuciones realizadas en el cluster, poco espacio de almacenamiento (2.7TB compartido)
- ***Directorio nas***: para guardar resultados de ejecuciones, almacenamiento a largo plazo y de mayor capacidad (36TB compartido).

Cómo citar el cluster en publicaciones

Se agradece que los usuarios del cluster agreguen una oración en los agradecimientos de publicaciones o tesis que hagan uso del cluster.

Español:
"Este trabajo utilizó el Cluster Toko de FCEN-UNCuyo que forma parte del SNCAD-MinCyT, Argentina."

En caso de haber utilizado la cola ***gpu*** o el nodo ***gpu0***, solicitamos agradecer de la siguiente manera: "Este trabajo utilizó el Cluster Toko de FCEN-UNCuyo que forma parte del SNCAD-MinCyT, Argentina. La GPU Titan Xp utilizada en este trabajo fue donada por NVIDIA Corporation"

English:
"This work used the Toko Cluster from FCEN-UNCuyo, which is part of the SNCAD-MinCyT, Argentina."

En caso de haber utilizado la cola ***gpu*** o el nodo ***gpu0***: "This work used the Toko Cluster from FCEN-UNCuyo, which is part of the SNCAD-MinCyT, Argentina. The Titan Xp GPU used for this research was donated by the NVIDIA Corporation.