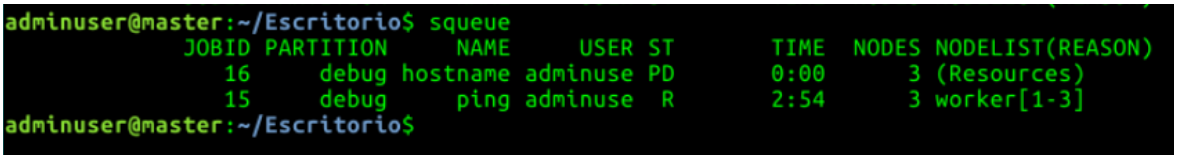
Sistemas Distribuidos

Slurm

Carlos Gómez Gómez -1460840
Julio Velásquez Cárdenas - 1397896

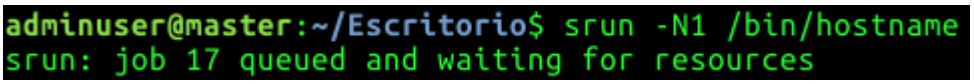
PREGUNTA 1: ¿En qué estado se encuentra el job?

El job se encuentra en estado pending, está esperando a que haya recursos disponibles para poder ejecutarse. Como el job 15, está utilizando los 3 nodos, el job 16 debe esperar a que haya algún recurso disponible.



Comando squeue *ejecutado en el master.*

PREGUNTA 2: ¿Devuelve el intérprete de comandos el control al usuario que lanza el comando srun? ¿Por qué?

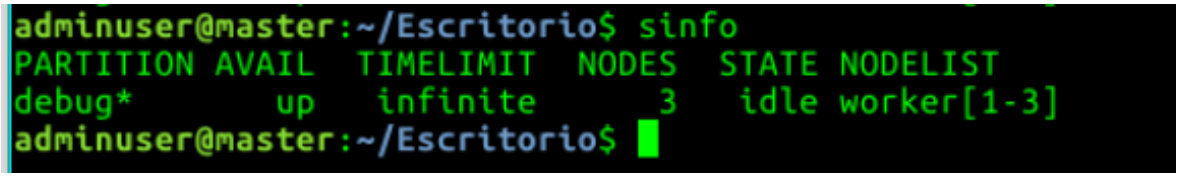


Comando srun *-N1 /bin/hostname*

No devuelve el control al usuario. El comando srun ejecuta directamente sobre la terminal la orden indicada y devuelve el control al usuario una vez esta ha finalizado

Una vez se realiza el scancel 15, para abortar la ejecución del comando ping, los 2 trabajos que están en pending ya se pueden ir ejecutando porque los recursos quedan disponibles

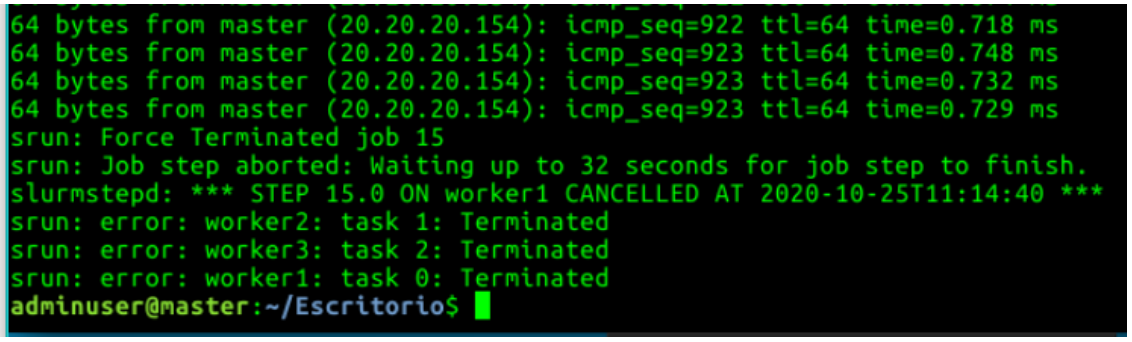
PREGUNTA 3: En este punto ¿En qué estado quedan los nodos del cluster y por qué?



Comando sinfo *ejecutado después de acabar los 3 trabajos.*

Los nodos se vuelven a encontrar en estado idle ya que se han realizado todos los trabajos pendientes al cancelar el trabajo ping.

PREGUNTA 4: ¿Cuántos procesos ping se han ejecutado en nuestro ejemplo?



Comando ping una vez ejecutado el comando scancel *en otra terminal*

Como podemos ver en esta captura de pantalla la orden srun ha ejecutado 3 procesos ping, uno en cada nodo worker.

PREGUNTA 5 ¿Dónde volcará la salida el comando sort del script? ¿Hacia dónde redirecciona esa salida este script Slurm ? ¿Y la salida de errores (si los hubiese)?

La salida del comando sort del script se vuelca en el fichero slurm.worker1.24.out. En este fichero aparece la salida del comando sort.

El comando sort debería salir por pantalla, que es la salida estándar y mediante el siguiente comando se redirecciona al fichero .out

#SBATCH -o slurm.%N.%j.out # STDOUT

La salida de los errores se redirecciona al fichero slurm.worker1.24.err, en nuestro caso el fichero de errores estaba vacío. Si se hubiesen cometido errores en la creación del script se habrían mostrado en este fichero.

Con el siguiente comando se redirecciona la salida de errores del script

#SBATCH -e slurm.%N.%j.err # STDERR

PREGUNTA 6 ¿Qué significan las directivas Slurm de %N y %j ? ¿Para qué se utilizan en este script?

El %N significa el nombre del nodo donde se ejecuta el trabajo, en este caso worker1, worker2 o worker3, y el %j la ID del trabajo enviado. Se utilizan para dar nombre a los archivos de salida y de error.

PREGUNTA 7 ¿Qué contienen los ficheros de salida generados?

Al ejecutarse el trabajo, se generan 3 ficheros de salida. El SomeRandomNumbers.txt contiene todos los números generados por el RANDOM, el archivo .out contiene los números ordenados por el comando sort y el .err contiene los errores en caso de que hubiese alguno.

PREGUNTA 8: Copia el parametric-random.bash a parametric-random-V2.bash, y modifica esta segunda versión según estas características:
• Pon el número de nodos (-N) a 3
• Cada job generará 10000 números aleatorios.
• Sustituye las dos veces que aparece SomeRandomNumbers.txt por SomeRandomNumbers.%j.txt Con esto, el nodo que ejecute el trabajo crea un fichero SomeRandomNumber por cada trabajo (ya que todos los trabajos comparten el home de slurmuser, y si no distinguimos de alguna manera se mezclarán los resultados de un trabajo con otros)

```
adminuser@master:~/Escritorio$ sbatch parametric-random-V2.bash
Submitted batch job 37
adminuser@master:~/Escritorio$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                37     debug parametr adminuse  R       0:00      3 worker[1-3]
adminuser@master:~/Escritorio$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
adminuser@master:~/Escritorio$
```

PREGUNTA 9 ¿Cuántos procesos se han ejecutado? ¿Cómo se explica esto en comparación con la respuesta a Pregunta 4? Parece contradictorio, pero no lo es. Observa: copia un parametric-random-V3.bash y déjalo así.

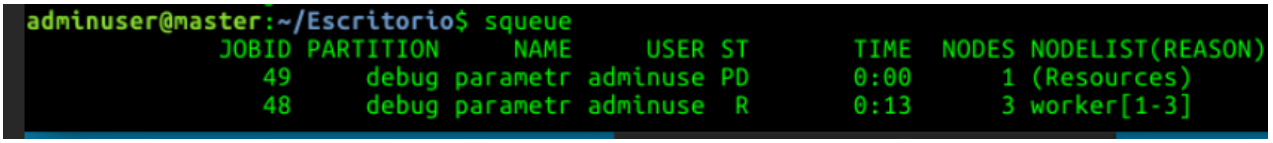
Se ejecuta un solo proceso en un nodo, el worker 1. Con el sbatch aunque se reserven los 3 nodos para la ejecución del job enviado, al poder ser realizado tan solo usando un nodo el sbatch no usa los dos nodos restantes.

PREGUNTA 10 A partir de los resultados del problema ¿Cuál es la diferencia en la asignación de jobs a nodos enviando con srun y enviando con sbatch?

```
adminuser@master:~/Escritorio$ srun -N3 /bin/hostname
worker1
worker2
worker3
adminuser@master:~/Escritorio$
```

Cuando se ejecuta el comando srun -N3 /bin/hostname, los 3 nodos realizan el proceso hostname, y devuelven el valor de hostname. En cambio al utilizar sbatch solo el worker 1 realiza el proceso en cuestión.

Se ha realizado una prueba sustituyendo el comando hostname de la versión parametric-random-V3.bash por el comando ping master y se observa lo siguiente



El job 48, el que realiza el comando ping master, habiendo reservado los 3 nodos, no permite que el job 49 se ejecute, porque está pendiente de recursos. Cuando se realiza el scancel 48, se observa en el fichero de salida que solamente el worker1 ha ejecutado el comando ping. Sbatch reserva los 3 nodos pero si no se usan implícitamente mediante algún mecanismo de paralelización, esos 3 nodos no se utilizan.