Useful Slurm commands

Slurm proporciona una variedad de herramientas que permiten al usuario administrar y comprender sus trabajos. Este tutorial presentará estas herramientas y proporcionará detalles sobre cómo usarlas.

Encontrar información en la cola de trabajo con squeue

El comando `squeue` es una herramienta que usamos para obtener información sobre los trabajos que están en la cola. De manera predeterminada, el comando squeue imprimirá el ID del trabajo, la partición, el nombre del trabajo, el usuario del trabajo, el estado del trabajo, el tiempo que lleva en ejecución, la cantidad de nodos y la lista de nodos asignados:

```
squeue
JOBID  PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
111111     batch   my_job   myuser  R    1:21:59      1 node0101-1
```

Podemos generar información no abreviada con la marca `--long`. Esta bandera imprimirá la información predeterminada no abreviada con la adición de un campo de límite de tiempo:

```
squeue -l
squeue --long
```

El comando squeue también brinda a los usuarios un medio para calcular la hora de inicio estimada de un trabajo agregando el indicador `--start` a nuestro comando. Esto agregará la hora de inicio estimada de Slurm para cada trabajo en nuestra información de salida.

```
squeue --user=username --start
```

> **Note**
>
> la hora de inicio proporcionada por este comando puede ser inexacta. Esto se debe a que la hora calculada se basa en los trabajos en cola o en ejecución en el sistema. Si un trabajo con una prioridad más alta se pone en cola después de ejecutar el comando, su trabajo puede ser demorado.

Al verificar el estado de un trabajo, es posible que desee llamar repetidamente al comando squeue para buscar actualizaciones. Podemos lograr esto agregando el indicador `--iterate` a nuestro comando squeue. Esto ejecutará squeue cada n segundos, lo que permite una actualización frecuente y continua de la información de la cola sin necesidad de llamar repetidamente a squeue:

```
squeue --start --iterate=n_seconds
```

Presione `ctrl-c` para detener el bucle del comando y regresar a la terminal.

Estado de los trabajos

Una vez que se ha enviado un trabajo a una cola de trabajos, la ejecución seguirá estos estados:

- `PENDING` o `PD`: El trabajo ha entrado en la cola pero aún no están disponibles los recursos solicitados para que empieze a trabajar, es decir, que no haya nodos libres.
- `RUNNING` o `R`: El trabajo se está ejecutando en la cola con los recursos que se han solicitado.
- `COMPLETED` o `CD`: El trabajo se ha ejecutado correctamente, o al menos, lo que se ha especificado en el script de lanzamiento.
- `COMPLETING` o `CG`: El trabajo está en proceso de terminar en un estado correcto.
- `SUSPEND` o `S`: la ejecución del trabajo se ha suspendido y los recursos utilizados se han liberado para otros trabajos.
- `CANCELLED` o `CA`: el trabajo ha sido cancelado o por el usuario o por los administradores de sistemas.
- `FAILED` o `F`: Ha fallado la ejecución del trabajo.
- `NODE_FAIL` o `NF`: Ocurrió un error con el nodo y el trabajo no pudo lanzarse. Por defecto, Slurm relanza el trabajo de nuevo.

Razones de que un trabajo esté en PENDING

Cuando un trabajo se encuntra en el estado de `PENDING`, se añade la razón de porqué está pendiente para ejecutar y ésta puede ser:

- **(Resources)**: el trabajo está esperando hasta que estén disponibles los recursos solicitados.
- **(Dependency)**: el trabajo es dependiente de otro y, por tanto, no entrará a ejecutar hasta que se cumpla la condición establecida para la dependencia.

- **(DependencyNeverSatisfied)**: El trabajo está esperando por una dependencia que no ha sido cumplida. El trabajo se quedara en este estado para siempre, por tanto, hay que cancelar el trabajo.
- **(AssocGrpCpuLimit)**: El trabajo no se puede ejecutar porque se ha consumido la cuota asignada de CPU.
- **(AssocGrpJobsLimit)**: El trabajo no se puede ejecutar porque se ha alcanzado el límite de trabajos simultáneos que tiene permitido ejecutar el usuario o la cuenta.
- **(ReqNodeNotAvail)**: El nodo especificado no está disponible. Puede ser que esté en uso, que esté resevado o puede que esté marcado como "fuera de servicio".

Info

Para obtener más información sobre squeue, visite la página de Slurm en squeue

Detener trabajos con scancel

En ocasiones, es posible que deba detener un trabajo por completo mientras se está ejecutando. La mejor manera de lograr esto es con el comando `scancel`. Este comando permite cancelar los trabajos que está ejecutando en los recursos de Research Computing utilizando la ID del trabajo. El comando se ve así:

```
scancel your_job-id
```

Para cancelar varios trabajos, puede usar una lista de ID de trabajos separados por comas:

```
scancel your_job-id1, your_job-id2, your_jobiid3
```

Info

Para obtener más información, visite el manual de Slurm en scancel

Información de estado con sstat

El comando `sstat` permite a los usuarios obtener fácilmente información sobre el estado de sus trabajos actualmente en ejecución. Esto incluye información sobre

el uso de la CPU, información de la tarea, información del nodo, tamaño del conjunto residente (RSS) y memoria virtual (VM). Podemos invocar el comando sstat como tal:

```
sstat --jobs=your_job-id
```

Formatear la salida del sstat

De forma predeterminada, `sstat` extraerá mucha más información de la que se necesitaría en la salida predeterminada de los comandos. Para remediar esto, podemos usar el indicador `--format` para elegir lo que queremos en nuestra salida. El indicador de formato toma una lista de variables separadas por comas que especifican los datos de salida:

```
sstat --jobs=your_job-id --format=var_1,var_2, ... , var_N
```

Algunas de estas variables se enumeran en la siguiente tabla:

Variable	**Descripción**
avecpu	Promedio de tiempo de CPU de todas las tareas en el trabajo.
averss	Tamaño medio del conjunto residente de todas las tareas.
avevmsize	Memoria virtual promedio de todas las tareas en un trabajo.
jobid	ID del trabajo.
maxrss	Número máximo de bytes leídos por todas las tareas del trabajo.
maxvsize	Número máximo de bytes escritos por todas las tareas en el trabajo.
ntasks	Número de tareas en un trabajo.

Por ejemplo, imprima la identificación de trabajo promedio de un trabajo, el tiempo de CPU, el máximo de rss y la cantidad de tareas. Podemos hacer esto escribiendo el comando:

```
sstat --jobs=your_job-id --format=jobid,cputime,maxrss,ntasks
```

Info

Se puede encontrar una lista completa de variables que especifican datos manejados por sstat con el indicador `--helpformat` o visitando la página de slurm en sstat.

Analizar trabajos terminados con sacct

El comando `sacct` permite a los usuarios obtener información sobre el estado de los trabajos terminados. Este comando es muy similar a sstat, pero se usa en trabajos que se ejecutaron previamente en el sistema en lugar de los trabajos que se están ejecutando actualmente. Podemos usar la identificación de un trabajo.

- Para todos los trabajos ejecutados:

```
sacct
```

- Para un único trabajo, identificado por su ID:

```
sacct --jobs=your_job_id
```

De forma predeterminada, `sacct` solo extraerá los trabajos que han ejecutado en **el día en curso**. Podemos usar el indicador `--starttime` para decirle al comando que mire más allá de su caché de trabajos a corto plazo.

```
sacct –jobs=your_job-id –-starttime=YYYY-MM-DD
```

Para ver una versión no abreviada de la salida de sacct, use el indicador `--long`:

```
sacct –jobs=your_job-id –-starttime=YYYY-MM-DD --long
```

Formatear la salida del sacct

Al igual que `sstat`, es posible que la salida estándar no proporcione la información que queremos. Para remediar esto, podemos usar el indicador `--format` para elegir lo que queremos en nuestra salida. De manera similar, el indicador de formato es manejado por una lista de variables separadas por comas que especifican los datos de salida:

```
sacct --user=your_rc-username --format=var_1,var_2, ... ,var_N
```

A continuación se proporciona una lista de algunas variables:

Variable	Description
account	Cuenta con la que se ejecutó el trabajo
avecpu	Tiempo promedio de CPU de todas las tareas en el trabajo.
averss	Average resident set size of all tasks in the job.
cputime	Tiempo transcurrido de CPU usado por un job o paso
elapsed	Tiempo transcurrido de los trabajos con formato DD-HH:MM:SS
exitcode	El código de salida devuelto por el script de trabajo o salloc.
jobid	ID del trabajo.
jobname	Nombre del trabajo.
maxdiskread	Número máximo de bytes leidos por todas las tareas.
maxdiskwrite	Número máximo de bytes leidos por todas las tareas.
maxrss	El código de salida devuelto por el script de trabajo o salloc.
ncpus	Cantidad de CPU asignadas.
nnodes	Número de nodos usados.
ntasks	Número de tareas en un job.
priority	Prioridad Slurm.
qos	Calidad de servicio.
reqcpu	Número de CPUs solicitados
reqmem	Cantidad de memoria necesaria para un trabajo
user	Nombre de usuario de la persona que ejecutó el trabajo.

Como ejemplo, suponga que desea buscar información sobre los trabajos que se ejecutaron el 12 de marzo de 2018. Desea mostrar información sobre el nombre del trabajo, la cantidad de nodos utilizados en el trabajo, la cantidad de CPU, el maxrss y el tiempo transcurrido. Su comando se vería así:

```
sacct --starttime=2018-03-12 --format=jobname,nnodes,ncpus,maxrss,elapsed
```

Como otro ejemplo, suponga que desea obtener información sobre los trabajos que se ejecutaron el 21 de febrero de 2018. Le gustaría obtener información sobre la identificación del trabajo, el nombre del trabajo, la calidad del servicio, la cantidad de nodos utilizados, la cantidad de CPU utilizadas, el RSS máximo y la CPU. tiempo, tiempo promedio de CPU y tiempo transcurrido. Su comando se vería así:

```
sacct –-starttime=2018-02-21 --
format=jobid,jobname,qos,nnodes,ncpu,maxrss,cputime,avecpu,elapsed
```

Info

Se puede encontrar una lista completa de variables que especifican datos manejados por sacct con el indicador `--helpformat` o visitando la página de slurm en sacct.

Control de trabajos en cola y en ejecución mediante scontrol

El comando `scontrol` proporciona a los usuarios un mayor control de sus trabajos ejecutados a través de Slurm. Esto incluye acciones como suspender un trabajo, detener la ejecución de un trabajo o extraer información detallada sobre el estado de los trabajos.

Para suspender un trabajo que se está ejecutando actualmente en el sistema, podemos usar scontrol con el comando suspender. Esto detendrá un trabajo en ejecución en su paso actual que se puede reanudar en un momento posterior. Podemos suspender un trabajo escribiendo el comando:

```
scontrol suspend job_id
```

Para reanudar un trabajo en pausa, usamos scontrol con el comando reanudar:

```
scontrol resume job_id
```

Slurm también proporciona una utilidad para retener trabajos que están en cola en el sistema. Retener un trabajo colocará el trabajo en la prioridad más baja, efectivamente "reteniendo" el trabajo para que no se ejecute. Un trabajo solo se puede retener si está esperando que el sistema se ejecute. Usamos el comando de espera para poner un trabajo en estado de espera:

```
scontrol hold job_id
```

Luego podemos liberar un trabajo retenido usando el comando de `release`:

```
scontrol release job_id
```

scontrol también puede proporcionar información sobre trabajos mediante el comando show job. La información proporcionada por este comando es bastante extensa y detallada, así que asegúrese de borrar la ventana de su terminal, recopilar cierta información del comando o canalizar la salida a un archivo de texto separado:

```
scontrol show job job_id
```

Streaming de salida a un archivo de texto

```
scontrol show job job_id > outputfile.txt
```

Canalizar la salida a Grep y encontrar líneas que contengan la palabra "Tiempo"

```
scontrol show job job_id | grep Time
```