How Do I Ensure My Job Has Enough Memory To Run Using SLURM?

Introduction

With all nodes on the cluster being shared among all users, it is possible that a single process owned by a single user might be consuming all of the memory on the node. If this happens, new jobs assigned to that node will likely not have enough memory to run. If your job is memory intensive, it would be best to request a node with enough available memory to run your job.

Determining Memory Requirements

In order to request a node with an appropriate amount of memory available, it might first be necessary to determine how much memory your job will actually need if this is not known ahead of time. To do this, it is recommended that you submit a test job and request exclusive access to a node. This means that only your job will be running on the node and will not interfere with or be disrupted by other jobs. Exclusive access will provide the best opportunity for you to measure the amount of memory that your job can consume without interference from other jobs.

There are a variety of methods that can be used to determine memory usage of a job. A relatively straightforward way to do this is to enable job profiling using job submission parameters ***#SBATCH --profile=ALL*** as follows:

```
#!/bin/bash
#SBATCH --job-name=YourJobNameHere
#SBATCH --ntasks=1
#SBATCH --time=00:30:00
#SBATCH --export=ALL
#SBATCH --profile=ALL

echo "My job ran on:"
echo $SLURM_NODELIST
if [[ -d $SHARED_SCRATCH/$USER && -w $SHARED_SCRATCH/$USER ]]
then
    cd $SHARED_SCRATCH/$USER
    /path/to/myprogram
fi
```

A *jobID* will be provided so that job usage data can be displayed.

Show job usage data

You can use ***sacct*** for your test job to report, among other things, the memory utilization of your job. Default output does not contain memory utilization so will need to use the ***--format*** parameter to ***sacct*** to provide memory information.

```
# sacct -j jobID
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
jobID               TEST    commons    commons         16  COMPLETED      0:0
jobID.batch        batch               commons          1  COMPLETED      0:0
jobID.0           memhog               commons         16  COMPLETED      0:0
jobID.1            sleep               commons         16  COMPLETED      0:0



# sacct -j jobID
--format=JobID,JobName,Partition,Account,AllocCPUS,State,ExitCode,MaxRSS,MaxVMSize
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
MaxRSS  MaxVMSize
------------ ---------- ---------- ---------- ---------- ---------- --------
---------- ----------
jobID               TEST    commons    commons         16  COMPLETED      0:0
jobID.batch       batch               commons          1  COMPLETED      0:0
5944K    190800K
jobID.0           memhog               commons         16  COMPLETED      0:0
0          0
jobID.1            sleep               commons         16  COMPLETED      0:0
524K     58932K
```

In this example, the *MaxVMSize* represents the maximum amount of physical memory used by **all processes** in the job. This number can be inaccurate because it is polled from the operating system at regular intervals. It does not represent the memory usage at an instant in time. Fluctuations during the polling interval will be missed. To ensure the best possible accuracy you must request exclusive access to the node when your test job runs so that the job will have the maximum amount of memory available to it.

Use *sacct* on a Running Job

It is also possible to see the above values while a job is running with the following command:

```
# sacct -j jobID
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
jobID               TEST    commons    commons         16    RUNNING      0:0
jobID.batch        batch               commons          1    RUNNING      0:0
jobID.0           memhog               commons         16  COMPLETED      0:0
jobID.1            sleep               commons         16    RUNNING      0:0
```

Requesting a Specific Amount of Memory

Once you have determined how much memory your job requires, you should request access to a node with enough memory available to satisfy the job for all future jobs of its type. There is a SLURM option that can be used to find a node with a specific amount of memory available. Use the *--mem* option in your SLURM script similar to the following:

```
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=1
#SBATCH --mem=2048MB
```

This combination of options will give you four nodes, only one task per node, and will assign the job to nodes with *at least* 2GB of physical memory available. The *--mem* option means the amount of physical memory that is needed by each **task** in the job. In this example the unit is

megabytes, so 2GB is 2048MB. This example requires 8GB total memory or 2GB per **process** with one process per node. If your job requires more than one process per node, then the amount of free memory available on the node must be *--ntasks-per-node X --mem*. For example, if each process needs 2GB (*--mem*) and you will be using two tasks per node (*--ntasks-per-node*), then the node must have 4GB of free memory for the job to run.

--mem is enforced
This memory limit is enforced, meaning if your job exceeds this 2GB limit it will be killed.

Accuracy is important
It is important to be accurate with your request. If you request an amount that is higher than you need, the scheduler might have a hard time finding nodes with enough memory available to run your job. Thus, you might unnecessarily delay execution of your job.

Use less memory than is available on the node
If your job uses memory near the amount of the maximum physical memory installed on the node, performance of your job might suffer and there is a chance that the job will crash. Keep in mind that the operating system resides in physical memory along side your job. A good rule of thumb is to allow 1GB of memory for the operating system.

Exclusive access jobs do not need pmem
If you are requesting exclusive access to a node to run your job, you do not need the *--mem* and *--ntasks-per-node* options as your job will have access to all available memory.

Do not request more memory than is available on a node
If you pick a *--mem* value that is high, such that *--ntasks-pernode X --mem* is greater than the amount of memory on a single node, your job will never run because it needs more memory than is available on a node.