HOW TO RESERVE EXTRA MEMORY FOR YOUR CORLEONE (SUN GRID ENGINE) JOB

LAST UPDATE: 25 JULY 2014 AUTHOR: HENK VAN STEENBERGEN

THE PROBLEM

If your job consumes a lot of memory which is not available on the cluster, either your job or the host may crash (check http://jeetworks.org/node/93 to learn more about this). You may check the current memory use of the hosts by using the `qhost` command.

In the example below the output of `qhost` shows there is only ~1GB of memory used (`MEMUSE`) and only a few MBs swap used (`SWAPUS`). So no problems here. Swap is going to be used if the jobs demands more than the total memory available (~ 47 GB per host, maximum of 22 slots per host, so say 2 GB per slot/job). This should always be avoided because it leads to a slowing down (or even a crash) of the cluster. So if you observe that `SWAPUS` is more than say 1 GB, you should consider stopping those big jobs running on that particular host, and then resubmitting them after you first have reserved memory for that job. The scheduler will then decide when there is enough memory available to run the job.

```
HOSTNAME                ARCH         NCPU  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------
global                  -               -     -       -       -       -       -
corleone02              lx24-amd64     24  4.00   47.1G    1.1G   49.3G    8.9M
corleone05              lx24-amd64     24  2.00   47.1G  963.1M   49.3G     0.0
corleone06              lx24-amd64     24  4.00   47.1G    1.0G   49.3G    7.8M
```

The good news is that we have recently (July 2014) changed the cluster configuration so it uses now a consumable resource of 45 GB per host. Each job that is submitted to the cluster will automatically reserve 2GB to be used by that job. If the job requires more than 2 GB of working memory it will be killed by the cluster and you need to submit it again and change the reservation of the memory for that job.

THE SOLUTION: RESERVE MORE MEMORY FOR YOUR JOB

So if you have a memory-intensive job, it is advisable to reserve a few GBs more of memory for that job. The current maximum is 45GB per host for extreme cases (not advisable to reserve so much memory, you will block the cluster access of other users and you can only run 4 jobs in parallel).

There are several ways to reserve memory for jobs:

1. DURING SUBMISSION IF YOU USE QSUB

Add the following parameters to your `qsub` job: `-l h_vmem=<MYMEM> -l mem_free=<MYMEM>` and replace <MYMEM> with the value of the memory you need.
For example, if you want to reserve 4GB for a job submitted with qsub do this:

```
qsub -q veryshort.q -l h_vmem=4G -l mem_free=4G $SGE_ROOT/examples/jobs/simple.sh
```

2. ALTER A JOB THAT IS NOT RUNNING YET

You can alter the memory reservation as long as a job is not running yet (check with `qstat -s r`) using `qalter`

The example below shows how to use `qalter` to change the memory reservation for job number 1012.

```
qalter -l h_vmem=4G -l mem_free=4G 1012
```

or if you want to apply this to all the jobs you have currently submitted define your user name with `-u`, like I did below

```
qalter -u henk -l h_vmem=4G -l mem_free=4G
```

3. TRICK FOR FSL / FSL_SUB USERS

If you are submitting jobs using `fsl_sub` you can't pass the `-l` parameters because `fsl_sub` reserves –l for defining the location of log files. So you need to use the `qalter` procedure described above. This can be tricky when you cannot call `fsl_sub` because fsl submits jobs automatically for you and they start running immediately on the cluster.

For jobs that already run, you can't change the memory reservation anymore. So we need a trick that holds the jobs so eager to start running immediately ☺ This is the trick:

a) Open a new ssh session (i.e. next to the existing session) and run the following command:

```
watch -n -0.1 qhold -u henk
```

replace `henk` with your user name. This command set all the jobs already submitted to "hold" state. The `watch` statement runs the `qhold` command each 0.1 second in the background.

b) Now go to the earlier session and start the scripts/FSL commands that submit jobs.

Because the `qhold` job is running in the background, any new jobs submitted by the scripts/FSL commands should get in "hold" state immediately, that is before they can start running on the cluster.

c) Wait until all jobs are submitted to the cluster.

d) Now use the `qalter` statement to change the memory requirements of your job. For example:

```
qalter -u henk -l h_vmem=4G -l mem_free=4G
```

replace `henk` with your user name. Or use `qstat` to get the correct job numbers and alter only the jobs that need more memory (e.g. job 1012,1014,1016 in the example below):

```
qalter -l h_vmem=4G -l mem_free=4G 1012,1014,1016
```

You can check whether this was succesful by checking whether there is a "`hard resource_list`" property in the output of the following command showing the values `h_vmem=4G,mem_free=4G`:

```
qstat -j <job number>
```

for example: `qstat -j 1012`

e) if the memory reservation for the jobs are succesful it is time to quit the `qhold` statement. Go to the other ssh session you started in (a) and use the Ctrl-C key combination to stop the `watch` command.

f) finally, release all your jobs so that they can be scheduled by the cluster and start running as soon as there are resources (memory and slots) available:

```
qrls -u henk
```

replace `henk` with your user name.

Check the result with qstat. Releasing jobs should change the status of `qwh` to `qw` (at least for jobs that are not depending on other jobs being finished first, those will stay in `qwh` mode) and if resources become available, jobs will start running and status will change to `r`.