

How to submit Matlab batch jobs to Sun Grid Engine

Written by Administrator

Thursday, 10 March 2011 18:03 - Last Updated Thursday, 10 March 2011 20:15

When we started porting our automatic fMRI processing to the cluster at work, it took a large amount of time to figure out exactly how to get Matlab, and SPM to run in batch mode on the cluster. So many variables and so many steps... so many places things could go wrong. In this article I describe the steps necessary to run Matlab batch jobs through SGE and cron.

Our setup consisted of running a Matlab batch program to process SPM. We use something we called 'center_scripts' which allows automatic processing of fMRI data through SPM without user input. Once a 'preferences' file is created, the automatic processing can take off on its own and go do its thing. This is easy to call from Perl or bash or even Matlab. I will only describe how to batch Matlab, not how we batched all of the fMRI/SPM processing.

Step 1 - Create a Matlab batch file

The first step is to create a matlab batch file. This .m file will contain all of your matlab processing. Open files, process, close files, etc. We'll call our example thebatch.m.

```
matlabbatch.m
addpath('path/to/data'); disp('Adding 1 + 1'); 1+1 disp('Above result should be 2');
```

Step 2 - Create an SGE batch file

The next step is to create an SGE batch file. This file is specific to SGE and contains all the parameters under which the SGE job should be run.

```
sgebatch.sh
#!/bin/sh ## -N taskname ## -S /bin/sh ## -j y ## -o outputfile.log ## -u username /path/to/./matlab
```

How to submit Matlab batch jobs to Sun Grid Engine

Written by Administrator

Thursday, 10 March 2011 18:03 - Last Updated Thursday, 10 March 2011 20:15

The # \$ options are only read by SGE during submission. The -j option combines the error and output into a single log file. There are many more [options](#) available when submitting jobs to the SGE. This SGE batch file is basically a shell script, written in your favorite interpreter, and can contain any commands or logic you would normally put into a shell script. However, this script is basically run in an variable-less environment. Your path to matlab may not be setup when running this script. You may need to add a full path to the executable. You may also need to use full paths when specifying the location of the matlab script.

Since Matlab is run in batch mode, you'll need to remove any vestiges of it's UI, so add the -nodisplay -nodesktop -nojvm options. Matlab also won't be able to display any windows unless you use the -display option. That -r option executes the entire quoted string following it, so it will try to execute the matlab script, but if it fails, it will exit. Using the try/catch will basically cause Matlab to exit when an error occurs instead of displaying the error message and waiting for user input (which will never come).

Step 3 - Run the SGE batch file

Normally, you would simply type the following to submit the SGE batch to the cluster:

```
[user@localhost testing]$ qsub sgebatch.sh
```

However, you can automate this further by creating a cron job to submit SGE batches on a schedule. This involves creating a perl script (or other language) to be executed via cron which submits a bunch of SGE jobs when run.

```
auto.pl
```

```
#!/usr/bin/perl ... # write out the SGE batch file(s) # submit the batch file(s) print `path/to/./qsub -
```

Now you need to setup this perl script to be run through cron. Remember.... cron is a variable-less environment, so you'll need to specify the SGE variables before you run the perl. That can be done in the following cron command

How to submit Matlab batch jobs to Sun Grid Engine

Written by Administrator

Thursday, 10 March 2011 18:03 - Last Updated Thursday, 10 March 2011 20:15

```
[user@localhost testing]$ crontab -l * * * * SGE_ROOT=/path/to/sgc; export SGE_ROOT;  
SGE_CELL=thecell; ↵ export SGE_CELL; perl /path/to/auto.pl > output.log
```

And now you have a scheduled cron job that submits SGE jobs to run matlab