MATLAB

Batch Script and Compiled Array Job

Scripts and Function

- A script is a file with Matlab commands to run one task. It sets up variables and calls functions. The script is terminated by a **quit** command.

- A function begins with a prototype line and comments, which a used by the Matlab help system. The function is terminated by a **end** command.

Matlab function

```
function maxe = maxEig(sd,dim)

%  maxEig  maximum real eigenvalue of a normally distributed random matrix
%     Input parameters
%       sd - seed for random generator
%       dim - size of the square matrix
%     Output value
%       maxe - maximum real eigenvalue

  rng(sd);
  ev = eig( randn(dim) );
  maxe = max( ev(imag(ev)==0) )

end
```

Saved in maxEig.m file

Matlab Script

```
% script to run maxEig function 200 times and print average.

count = 200;
dim = 5001;
sumMaxe = 0;

for i=1:count;
  sumMaxe = sumMaxe + maxEig(i,dim);
end;

avgMaxEig = sumMaxe/count

quit
```

Saved in script.m file

Queue Script

```
#$ -N script
#$ -m eas
#$ -M traine@gmail.com
#$ -l exclusive=1

vpkg_require matlab/r2014b
matlab -nodisplay -nojvm -r 'script'
```

Saved in batch.qs file

Submit steps

- Copy files **maxEig.m**, **script.m** and **batch.qs** to a scratch directory in /lustre/scratch/traine
  - **scp -r Documents/MATLAB farber:/lustre/scratch/traine**
- Logon the head node
  - **ssh farber**
- start a workgroup shell
  - **workgroup -g it_css**
- change to the directory
  - **cd /lustre/scratch/traine/MATLAB**
- Submit job
  - **qsub batch.qs**

email message

After 2 1/2 hours traine gets the message:

```
Job 2362 (script.m) Complete
User             = traine
Queue            = it_css.q@n038
Host             = n038.farber.hpc.udel.edu
Start Time       = 10/21/2014 14:45:42.100
End Time         = 10/21/2014 17:09:24.782
User Time        = 12:41:56
System Time      = 00:11:31
Wallclock Time   = 02:23:42
CPU              = 12:53:27
Max vmem         = 3.924G
Exit Status      = 0
```

```
>> disp((12+53/60)/(2+23/60))
    5.4056
```

script.o2362

```
[CGROUPS] No /cgroup/memory/UGE/2362.1 exists for this job
[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting none bytes (vmem none bytes) on n038 (master)
[CGROUPS]   with 20 cores = 0-19
[CGROUPS] done.

Adding package `matlab/r2014b` to your environment

                            < M A T L A B (R) >
                  Copyright 1984-2014 The MathWorks, Inc.
                     R2014b (8.4.0.150421) 64-bit (glnxa64)
                               September 15, 2014


To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.


avgMaxEig =

   69.5131
```

Answer is in the **JOBNAME**.o**JOBID** file

email message

A run with -pe threads 5 -l m_mem_free=1G

```
Job 64501 (script) Complete
User           = dnairn
Queue          = it_css.q@n038
Host           = n038.farber.hpc.udel.edu
Start Time     = 04/28/2015 11:00:16.153
End Time       = 04/28/2015 13:31:45.073
User Time      = 05:13:18
System Time    = 00:03:08
Wallclock Time = 02:31:28
CPU            = 05:16:26
Max vmem       = 1.673G
Exit Status    = 0
```

A -singleCompThread run with -l m_mem_free=5G

```
Job 64503 (script) Complete
User           = train
Queue          = it_css.q@n036
Host           = n036.farber.hpc.udel.edu
Start Time     = 04/28/2015 11:55:16.107
End Time       = 04/28/2015 15:13:16.715
User Time      = 03:17:21
System Time    = 00:00:43
Wallclock Time = 03:18:00
CPU            = 03:18:05
Max vmem       = 1.336G
Exit Status    = 0
```

Deploy Function

```
function maxeDisplay(sd,dim)
%  evaluate and display value of maxEig
%     Output line - maxEig(sd,dim) = value

  if (isdeployed)
    sd = str2num(sd);
    dim = str2num(dim);
  end

  maxe = maxEig(sd,dim);

  fprintf('maxEig(%d,%d) = %f\n’,sd,dim,maxe);

end
```

Saved in maxeDisplay.m file

Compile Script

```
prog=maxeDisplay
opt='-nojvm,-nodisplay,-singleCompThread'
version='r2015a'

vpkg_require matlab/$version
mcc -R "$opt" -mv $prog.m

[ -d $WORKDIR/sw/bin ] && mv "$prog" $WORKDIR/sw/bin

vpkg_rollback 1
```

Saved in compile.sh file

Compile steps

- qlogin to a compute node (should not run the MATLAB compiler on head node)
  - **qlogin**
- Source the compile source script.
  - **source compile.sh**
- Check to see where it is installed (should not be on lustre)
  - **which maxeDisplay**
- Test to make sure it works (optional)
  - **vpkg_require mcr/$version-nojvm**
  - **maxeDisplay 1 100**
- Finish
  - **exit**

Queue Script

```
#$ -N maxEig
#$ -t 1-200
#$ -l m_mem_free=5G
#
# Parameter sweep array job to run the maxEig compiled MATLAB function with
#    lambda = 1,2. ... 200
#
date "+Start %s"
echo "Host $HOSTNAME"

vpkg_require mcr/r2015a-nojvm
export MCR_CACHE_ROOT=$TMPDIR

maxeDisplay $SGE_TASK_ID 5001

date "+Finish %s"
```

Saved in batch.qs file

Submit steps

- In your MATLAB directory create a place to store the 200 output files:
  - **mkdir run1**
  - **cp batch.qs run1**
  - **cd run**1
- Submit the batch job source script:
  - **qsub batch.qs**
- Wait until done (checkout the -sync qsub options)
- List all 200 displayed lines
  - **grep -h '^maxE\S* = [0-9-][0-9.]*' *.o*.***
- Compute the average
  - **grep -h '^maxE\S* = [0-9-][0-9.]*' *.o*.* l**
  - **awk '{sum+=$3} END { print "Average = ",sum/NR}'**

Answers are in the ***JOBNAME*.o*JOBID*.*TASKID*** files

Gather Examples

One line commands:

```
$ grep -h '^maxE\S* = [0-9-][0-9.]*' *.o*.* | head -3
maxEig(1,5001) = 70.021958
maxEig(10,5001) = 62.005743
maxEig(100,5001) = 71.048502

$ grep -h '^maxE\S* = [0-9-][0-9.]*' *.o*.* |
> awk '{sum+=$3} END { print "Average = ",sum/NR}'
Average =  69.5131
```

You and use any script language, e.g. perl and bash, to read all the files and produce a report.

You could also write a MATLAB post processing script to combine all results to one file.

Holding Jobs

- Each array jobs

```
name='SA7_SL'
options='-l standby=1,m_mem_free=15G’
qsub -N ${name}a $options SAShell_SL.qs
qsub -N ${name}b -hold_jid_ad ${name}a $options SAShell_SL.qs
qsub -N ${name}c -hold_jid_ad ${name}b $options SAShell_SL.qs
```

- Post processing job

```
name='SA7_SL'
qsub -N ${name} SAShell_SL.qs
qsub -hold_jid ${name} postprocess.qs
```

Load / Save

```
function T1Combine(maxID, result_matfile)

if (isdeployed)
   maxID=str2num(maxID);
end

for taskID=1:maxID
  load(sprintf(result_matfile,taskID));
  if taskID==1
    mc = cast(T1Count,'int32');
    T1pass = cast(T1passCount,'int32');
  else
    mc = mc+T1Count;
    T1pass = T1pass+T1passCount;
  end
end

T1perpass=cast(T1pass,'double')/mc;

save('T1perpass.mat','T1perpass');
```

MatfileSaveFormat should be sv7.3 to save files bigger than 2G

```
$ grep MatfileSaveFormat $HOME/.matlab/R2015a/*
/home/1005/.matlab/R2015a/matlab.prf:MatfileSaveFormat=Sv7.3
```

Matfile Save Format

Sv7.3 is needed to save more than 2G and it is a hierarchal file format, which allows selectively loading part of the file.

Here is MATLAB code to set the MatfileSaveFormat for large cluster:

```
pref = regexp( fileread(fullfile(prefdir,'matlab.prf')), '\n', 'split');
loc = find(~cellfun(@isempty,regexp(pref,'MatfileSaveFormat=')));

if isempty(loc); pref{end+1} = 'MatfileSaveFormat=Sv7.3';
else;            pref{loc} = 'MatfileSaveFormat=Sv7.3';
end

fid = fopen(fullfile(prefdir,'matlab.prf'), 'w');
fprintf(fid, '%s\n', A{:});
fclose(fid);
```