Running MATLAB on the HPCC

Dirk Colbry
colbrydi@msu.edu
Research Specialist
Institute for Cyber-Enabled Research

Agenda

- MATLAB Licenses
- Running MATLAB Interactively
- Running MATLAB non-interactively
- Using MATLAB Distributed Computing Service

MATLAB Licenses

- 35 MATLAB licenses on our system
- Many toolboxes
- View available toolboxes and Current license usage:

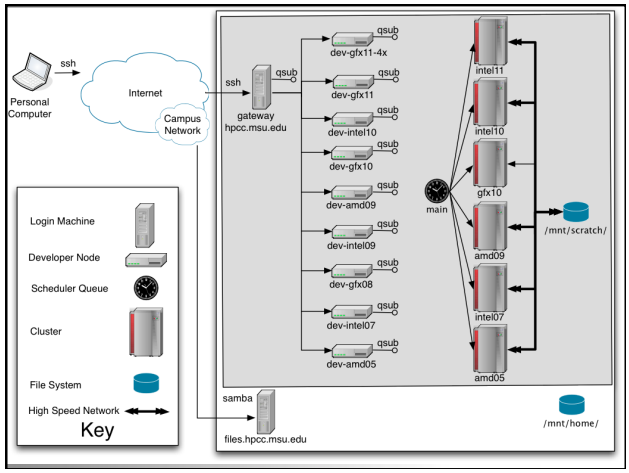
```
module load powertools
licensecheck matlab
```

Running MATLAB Interactively

- Must be run from a developer node.
  - ssh dev-intel09
- Once on a developer node just type:

```
matlab &
matlab –nodisplay &
```

WARNING: Interactive MATLAB jobs running on the developer nodes are limited to a maximum of two hours. Jobs over that limit will automatically be killed.



Multi-Threaded MATLAB

- Some MATLAB functions can take advantage of more than one thread. Use the following to run MATLAB in a multi-threaded mode on our system:
  - matlab-mt

WARNING: Multi-thread MATLAB may take up all of the resources on the node. There is no easy way to control the number of CPU cores MATLAB is using. Your job will be kicked off if it is using more than it's fair share.

Do not use this option inside of a job script.

Interactively for more than 2 hours

- If you need to run a job longer than 2 hours you can do this though the job scheduler by typing the following:

```
qsub -I -Z -l walltime=04:00:00,nodes=1:ppn=1,mem=2gb -W x=gres:MATLAB
```

WARNING: If the resources are not immediately available you may have to wait for this job to start.

Running MATLAB non-interactively

- Write a submission script.
  - Next page
- Submit the job to the queue using qsub:
  - qsub myjob.qsub

MATLAB Submission script

```
#!/bin/bash
#PBS -l walltime=04:00:00,mem=2gb
#PBS -l nodes=1:ppn=1,feature=gbe
#PBS -W x=gres:MATLAB

cd ${PBS_O_WORKID}

./matlab -nodispay -r "myscript()"

qstat -f ${PBS_JOBID}
```


Using MATLAB Distributed Computing Service

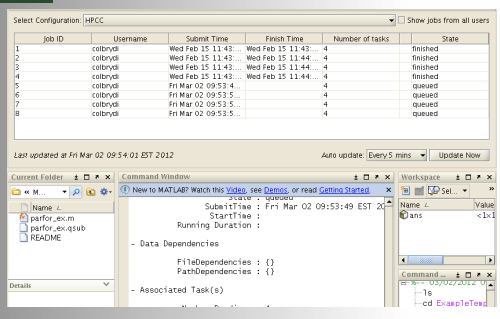
Step 1: Set up your account

```
module load powertools
setup_parallel_matlab
```

Step 2: Start Matlab

```
>> testCluster
```


Parallel->Job Monitor



Using the dev nodes

- You can develop with the parallel toolbox using a developer node. Start MATLAB and use the following commands:

```
>> matlabpool('local', 4)
>> matlabpool(close)
```

Parallel jobs using parfor

- The parfor command can be used to parallelize for loops in MATLAB.
- Lets get an example:

```
module load powertools
getexample
getexample MATLAB_parfor
```

MICHIGAN STATE UNIVERSITY

ICER

ClusterInfo class

ClusterInfo.state

- Arch
- ClusterHost
- EmailAddress
- MemUsage
- ProcsPerNode
- ProjectName
- QueueName
- Reservation
- UseGpu
- UserDefinedOptions
- UserNameOnCluster
- WallTime

ClusterInfo.clear

MICHIGAN STATE UNIVERSITY

ICER

Using ClusterInfo

```
>> ClusterInfo.setWallTime('04:00:00');
>> ClusterInfo.setUserGpu('false');
>> ClusterInfo.setMemUsage(15)
```

– Amount of memory needed in GB

MICHIGAN STATE UNIVERSITY

ICER

Running a job from MATLAB

```
>> job = batch (@yourscript,1,
{input},'Matlabpool',24,'CaptureDiary', true,
'PathDependancies', {'/path/to/code'});
```

- @yourscript is the funciton to submit.
- 1 is the number of output arguments
- {input} is an array of input arguments
- The job will se 24 processing cores
- Stores the output to a diary
- Specify location of code

MICHIGAN STATE UNIVERSITY

ICER