

HIGH PERFORMANCE RESEARCH COMPUTING

Using Matlab on the Grace Cluster

HPRC Training
February 17, 2023



High Performance
Research Computing
DIVISION OF RESEARCH

Outline

- Interactive Matlab Processing
- Single Node Batch Jobs
- Parallelism: Matlab Concepts
 - Cluster Profiles
 - Parallel Programming
 - GPUs
- Multi Node Batch Jobs

Running Matlab on login node

1. Run Matlab on command line (don't forget to load the Matlab module)
 - To start, type: matlab
 - if X-forwarding enabled, type: matlab -nosplash -nodesktop
 - Will start the Matlab interpreter.
 - Remember: one hour cpu time limit
2. Run Matlab GUI
 - to start, type: matlab
 - Will start the Matlab GUI
 - NOTE: need to have X-forwarding enabled
 - NOTE: GUI might respond slow
 - Remember: one hour cpu limit

REMINDER: Loading Matlab

We have different versions of Matlab installed on our clusters. Before we can use our preferred Matlab version we have to set up the environment. On HPRC clusters, we use the module system to achieve this.

```
module spider Matlab
```

```
# See all the available Matlab version
```

```
module load Matlab/R2021b
```

```
# Load Matlab version R2021b
```

```
module list
```

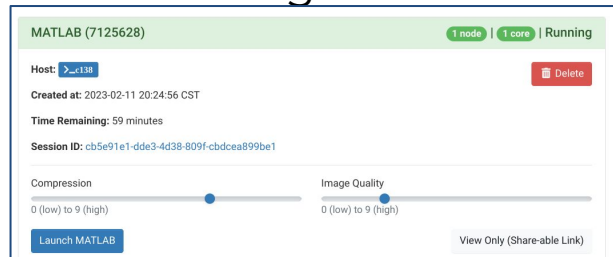
```
# Show all modules that have been loaded
```

Demo Time

(let's start the command line interpreter and the GUI)

Running Matlab on the Portal

1. visit portal-grace-hprc.tamu.edu
2. Login using TAMU credentials
 - Need VPN off campus
3. Click on “Interactive Apps”
4. Select “Matlab”
5. Fill out the Form
 - Use R2021B, set time to 3 hours, threads to 8, memory 50GB, enable gpu (We will use the GUI later)
6. Click Launch
7. Once Matlab Job is running click on “Launch Matlab”



MATLAB (7125628) 1 node | 1 core | Running

Host: >_c138 Delete

Created at: 2023-02-11 20:24:56 CST

Time Remaining: 59 minutes

Session ID: cb5e91e1-dde3-4d38-809f-cbdcea899be1

Compression 0 (low) to 9 (high) Image Quality 0 (low) to 9 (high)

Launch MATLAB View Only (Share-able Link)

MATLAB

This app will launch the [MATLAB GUI on Grace](#).

MATLAB version
Matlab/R2021b

Number of computational threads

(to utilize Matlab's automatic multi-threaded functionality)

Number of workers for parallel processing (max 47)

(only set if using parallel constructs; e.g. parfor)

Number of hours

Total amount of memory (GB) per compute node
360
The requested total memory (4 - 360GB).

Select if you are planning to utilize MATLAB GPU capabilities

Account

This field is optional.

I would like to receive an email when the session starts

Launch

* The MATLAB session data for this session can be accessed under the [data root directory](#).

Demo Time

(Let's visit the portal and start the Matlab GUI)

Running Matlab on a Single Compute Node

HPRC created a tool “**matlabsubmit**” to streamline the process of submitting jobs

- **matlabsubmit options**
- **flags to set Matlab parameters**
 - workers
 - threads
 - memory/walltime
- **creates boilerplate Matlab code**
 - set number of threads
 - creates parallel pool
- **creates batch job**
- **submits the job**

```
[pennings@grace1 ~]$ matlabsubmit
No matlab script (or function) provided
Usage: /sw/hprc/sw/Matlab/bin/matlabsubmit [options] SCRIPTNAME

This tools automates the process of running matlab codes on the compute nodes.

OPTIONS:
  -h Shows this message
  -m set the amount of requested memory in MEGA bytes(e.g. -m 20000)
  -t sets the walltime; form hh:mm (e.g. -t 03:27)
  -w sets the number of ADDITIONAL workers
  -g indicates script needs GPU (no value needed)
  -b sets the billing account to use
  -s set number of threads for multithreading (default: 8 ( 1 when -w > 0)
  -p set number of workers per node
  -f run function call instead of script
  -x add explicit batch scheduler option

DEFAULT VALUES:
memory   : 2500 per core
time     : 02:00
workers  : 0
gpu      : no gpu
threading: on, 8 threads
[pennings@grace1 ~]$
```


Running Matlab on a Single Compute Node

matlabsubmit will create a directory, named MatlabSubmitLOG<N> (<N> is the *matlabsubmit* job ID) where all generated files and redirected output will be stored

- **batch_job_id-XXX**
 - file containing the Slurm batch id (mostly for convenience)
- **matlabsubmit_wrapper.m**
 - boilerplate matlab code to set up environment and calls the matlab program provided by user
- **slurm.out**
 - redirected output from the script
- **submission_script**
 - the generated Slurm batch script

```
batch_job_id-7125904
matlabsubmit_wrapper.m
slurm.out
submission_script
```

NOTE: matlabsubmit will create different files when running on multiple nodes (will discuss later)

Demo Time

(Let's submit a dummy job using matlabsubmit)

Matlab Parallel processing on Grace

We know how to run Matlab on Grace, in the next part of the session we will do parallel programming on grace

1. Let's go back to the Interactive Matlab session we created before
 - if the session was closed, click “Launch Matlab” again
2. Open the live script “matlab_grace.mlx”
 - might need to upload file to grace
 - can also copy from directory /scratch/training/Matlab/matlab_grace.mlx
3. We will use the live script to explain and practise
 - parallel processing using multi threading
 - discover parallel profiles
 - local profile
 - cluster profile
 - parallel pools / parallel concepts
 - GPU
4. Continue here; demo multi node parallel jobs using matlabsubmit

Welcome back

**Now we know how to create parallel codes in Matlab,
let's use matlabsubmit to run these on Grace compute
nodes**

Running Multiple-node Matlab Jobs

For multi node jobs **matlabsubmit** will start an interactive Matlab session and runs a batch job using a TAMUClusterProperties profile

- Creates a Job<M> directory
 - Log/debug info
 - Redirected output from all workers
 - submission script
 - meta info
- Creates a MatlabSubmitLOG<N> directory
 - similar to single node jobs

```
matlab-batch-commands.log
matlabsubmit_driver.m
submission_script
```

```
Job2.log
Task2.out.mat
Task2.state.mat
Task3.out.mat
Task3.state.mat
Task1.out.mat
Task1.state.mat
Task1.diary.txt
Task3.diary.txt
Task3.common.mat
Task2.diary.txt
Task1.common.mat
Task2.common.mat
Task1.in.mat
Task2-3.in.mat
tp6a66bb81_e8bb_415a_a6b6_42b7af18c8d6
```

Final demo time

(let's create some parallel Matlab code and use
matlabsubmit to run them)



**HIGH PERFORMANCE
RESEARCH COMPUTING**
TEXAS A&M UNIVERSITY

Thank you.

Any questions?