

Yale

Yale Center for Research Computing

160 St. Ronan Street

research.computing@yale.edu

<https://research.computing.yale.edu>

<https://software.yale.edu>

<https://research.computing.yale.edu/support/hpc/user-guide/matlab>

Connecting to HPC Clusters

- To Grace

```
ssh <user-id>@grace.hpc.yale.edu
```

- To Farnam

```
ssh <user-id>@farnam.hpc.yale.edu
```

Load MATLAB on System Path

- On Grace

```
module load Apps/Matlab  
module load Apps/Matlab/R2017a
```

- On Farnam

```
module load MATLAB  
module load MATLAB/2017a
```

Configure MATLAB

```
module load Apps/Matlab           # GRACE
module load MATLAB                 # FARNAM
```

- Instruct MATLAB how to submit jobs to the HPC clusters

```
configCluster.sh
```

Sample MATLAB Code

my_parallel_app.m

```
function my_parallel_app

% Get a handle to the cluster
c = parcluster('ycrc');

% What the ideal pool size
sz = suggestedPoolSize();

% Open a pool of workers
p = c.parpool(sz);

% PARFOR
t0 = tic;
parfor idx = 1:sz*3
    pause(2)
end
t = toc(t0) %#ok<*NASGU,*NOPRT>

% SPMD
spmd
    o = gop(@plus,labindex);
end
o{end}
% Compare our results to a cumulative summation
o = cumsum(1:sz);
o(end)

% Close the pool
p.delete
```

Jobscript – matlab.parallel.jobscript

```
#!/bin/bash

#SBATCH --mem-per-cpu=2000                # In MB
#SBATCH --time=01:00:00                  # Default: 1 hr
#SBATCH --ntasks=32                      # REQUIRED

##SBATCH --gres=gpu[:<GPUS-PER-NODE>]
##SBATCH --gres-flags=enforce-binding    # If >1 GPU/node

# =====

# Be sure to not move to a directory where MATLAB can't find your code.
echo "Working directory: `pwd`"

# Load MATLAB on system path
module load Apps/Matlab

# Run MATLAB
# Use the Slurm output/error file, rather than the MATLAB -logfile option
matlab -nodisplay -r my_parallel_app
```

Submit the Jobscript

- `sbatch matlab.parallel.jobscript`
Submitted batch job 11704017
- `queue -j <job-id>`
- `cat slurm-<job-id>.out`