Introduction to MATLAB in HPC Environment

Sagnik Singha
High Performance Computing Center
(on behalf of the HPCC staff)

Nov 9th, 2022

Outline

- MATLAB in the TTU HPCC environment
- Job submission methods
- Batch Submission
- Graphical Client based Submission
- Containerized MATLAB (MathWorks account)

Logging on to TTU HPCC

TEXAS TECH UNIVERSITY
Information Technology Division

https://www.depts.ttu.edu/hpcc/userguides/general_guides/login_general.php

Connecting to RedRaider – On Campus

ssh <eraider>@login.hpcc.ttu.edu

MATLAB in the TTU HPCC - Environment setup

TEXAS TECH UNIVERSITY
Information Technology Division

Setup MATLAB environment
`module load matlab`

Default version loaded (nocona): `matlab/R2021b`

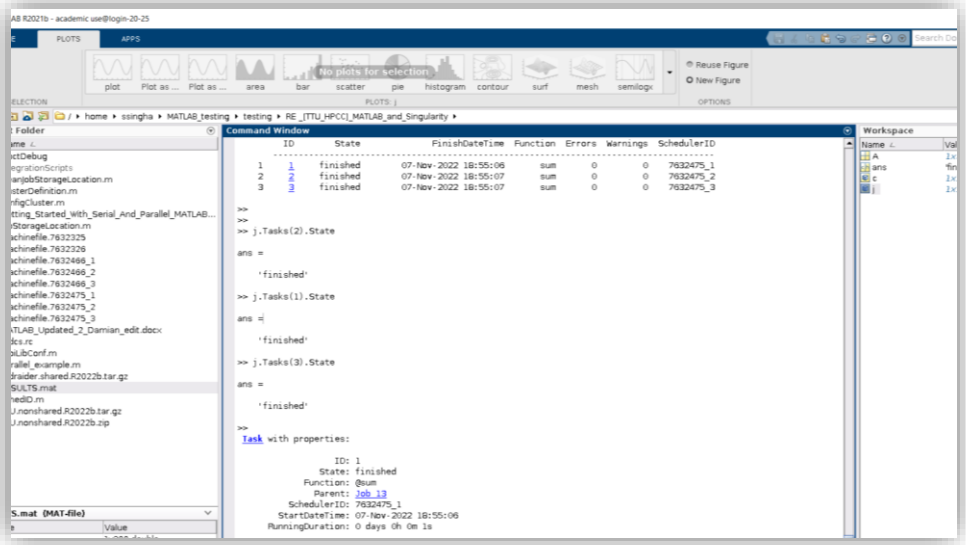
check if MATLAB is currently loaded in your environment, please type:
`module list`

```
cpu-26-10:/MATLAB_testing/nocona/matlabPar$ ml matlab
cpu-26-10:/MATLAB_testing/nocona/matlabPar$ ml list

Currently Loaded Modules:
  1) nocona/0.15.4   2) matlab/R2021b
cpu-26-10:/MATLAB_testing/nocona/matlabPar$
```

```
quanah:$ ml matlab
quanah:$ ml list

Currently Loaded Modules:
  1) matlab/R2020a
```

Job submission methods

- Graphical User Interface –
  - If code is in developmental stages
  - If direct interaction with output necessary
  - Worker node



- BATCH submission –
  - If code is in production stages
  - If multiple runs needed without direct interaction with output
  - Login node

```
#SBATCH --job-name=matlab-test
#SBATCH --output=%x.o%j
#SBATCH --error=%x.e%j
#SBATCH --partition nocona
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=10:00:00      ##10 hour runtime, you may change or remove this op
#SBATCH --cpus-per-task=1                # 1 core per instance

####SBATCH --mem=512000MB


# Add MATLAB to system path
module load matlab

# Run code
matlab -batch calc_pi_multi_node
~
~
~
~
~
```

Batch submission

Example file's location:
`/lustre/work/examples/nocona/matlab`

Copy files to own workspace
`cp -r /lustre/work/examples/nocona/matlab /home/<eraider_ID>/.`

Example 1:

Matrix multiplication
Files:
`A_matrix.dat`
`B_matrix.dat`
`atimesb.txt`
`matlab.sh`

Batch submission

TEXAS TECH UNIVERSITY
Information Technology Division

https://www.depts.ttu.edu/hpcc/userguides/Job_User_Guide.pdf

matlab.sh

```
#!/bin/bash
#SBATCH --job-name=matlab-test
#SBATCH --output=%x.o%j
#SBATCH --error=%x.e%j
#SBATCH --partition nocona
#SBATCH --nodes=1
#SBATCH --ntasks=1
module load matlab
matlab -batch multiplyMatrix
```

matmulfile.m

```
load A.dat;
load B.dat;
atimesb = A*B
save -ascii atimesb.txt atimesb
quit
```

Batch submission and Monitoring

TEXAS TECH UNIVERSITY
Information Technology Division

Submission

```
sbatch matlab.sh
```

Monitoring

```
squeue -u <username>
```

Status 1:

```
cpu-26-10:/MATLAB_testing/nocona/matlab/Example_1$ sbatch matlab.sh
Submitted batch job 7599523
cpu-26-10:/MATLAB_testing/nocona/matlab/Example_1$ squeue -u ssingha
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           7599523    nocona matlab-t  ssingha PD       0:00      1 (None)
```

Status 2:

```
cpu-23-38:/MATLAB_testing/nocona/matlab/Example_1$ squeue -u ssingha
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           7628458    nocona INTERACT  ssingha  R    2:13:20      1 cpu-23-38
           7628776    nocona matlab-t  ssingha  R       0:11      1 cpu-23-23
```

Status 3:

```
cpu-23-38:/MATLAB_testing/nocona/matlab/Example_1$ squeue -u ssingha
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           7628458    nocona INTERACT  ssingha  R    2:13:22      1 cpu-23-38
cpu-23-38:/MATLAB_testing/nocona/matlab/Example_1$
```

https://www.depts.ttu.edu/hpcc/userguides/Job_User_Guide.pdf

Batch submission - Monitoring

Monitoring

```
$squeue -u <username>
```

```
$ssh cpu-23-23
$top | grep MATLAB
```

Status 2:

```
cpu-23-38:/MATLAB_testing/nocona/matlab/Example_1$ squeue -u ssingha
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           7628458    nocona INTERACT  ssingha  R    2:13:20      1 cpu-23-38
           7628776    nocona matlab-t  ssingha  R       0:11      1 cpu-23-23
```

```
cpu-23-1:$ top | grep MATLAB
  5318 ssingha   20   0   21.3g 475432 217068 S 133.3   0.1   0:03.12 MATLAB
  4541 ssingha   20   0   21.3g 466260 214360 S 105.6   0.1   0:03.11 MATLAB
  4537 ssingha   20   0   21.3g 482448 221040 S 100.0   0.1   0:03.14 MATLAB
  4544 ssingha   20   0   21.3g 486596 220236 S  94.4   0.1   0:03.15 MATLAB
  4539 ssingha   20   0   21.3g 489100 220840 S  88.9   0.1   0:03.19 MATLAB
  5074 ssingha   20   0   21.3g 480076 216208 S  83.3   0.1   0:03.08 MATLAB
  4534 ssingha   20   0   21.3g 476296 215080 S  77.8   0.1   0:03.11 MATLAB
  4531 ssingha   20   0   21.7g 736408 230976 S  55.6   0.1   0:03.75 MATLAB
  4534 ssingha   20   0   22.1g 951032 262028 S 105.0   0.2   0:06.29 MATLAB
  5074 ssingha   20   0   22.1g 953908 262056 S 103.6   0.2   0:06.22 MATLAB
  4539 ssingha   20   0   22.1g 952092 263620 S 101.0   0.2   0:06.25 MATLAB
  4537 ssingha   20   0   22.0g 936720 262352 S  97.7   0.2   0:06.10 MATLAB
  4544 ssingha   20   0   22.1g 923068 261824 S  97.7   0.2   0:06.11 MATLAB
  4541 ssingha   20   0   22.0g 896352 257788 S  93.4   0.2   0:05.94 MATLAB
  4531 ssingha   20   0   22.2g 976064 272068 S  89.1   0.2   0:06.45 MATLAB
  5318 ssingha   20   0   22.0g 895616 256076 S  87.8   0.2   0:05.78 MATLAB
  3698 ssingha   20   0   23.2g   1.2g 312528 S   0.3   0.2   0:13.07 MATLAB
  5074 ssingha   20   0   22.8g   1.1g 290956 S  99.3   0.2   0:09.23 MATLAB
  4544 ssingha   20   0   22.5g   1.1g 292092 S  99.0   0.2   0:09.11 MATLAB
```

Batch submission - Monitoring

Monitoring

`squeue -u <username>`

Files

- matlab-test.e<JOB_ID>
- matlab-test.o<JOB_ID>
- atimesb.txt

Status 1:

```
cpu-26-10:/MATLAB_testing/nocona/matlab/Example_1$ sbatch matlab.sh
Submitted batch job 7599523
cpu-26-10:/MATLAB_testing/nocona/matlab/Example_1$ squeue -u ssingha
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           7599523    nocona matlab-t  ssingha PD       0:00      1 (None)
```

Status 2:

```
cpu-23-38:/MATLAB_testing/nocona/matlab/Example_1$ squeue -u ssingha
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           7628458    nocona INTERACT  ssingha  R    2:13:20      1 cpu-23-38
           7628776    nocona matlab-t  ssingha  R       0:11      1 cpu-23-23
```

Status 3:

```
cpu-23-38:/MATLAB_testing/nocona/matlab/Example_1$ squeue -u ssingha
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           7628458    nocona INTERACT  ssingha  R    2:13:22      1 cpu-23-38
cpu-23-38:/MATLAB_testing/nocona/matlab/Example_1$
```

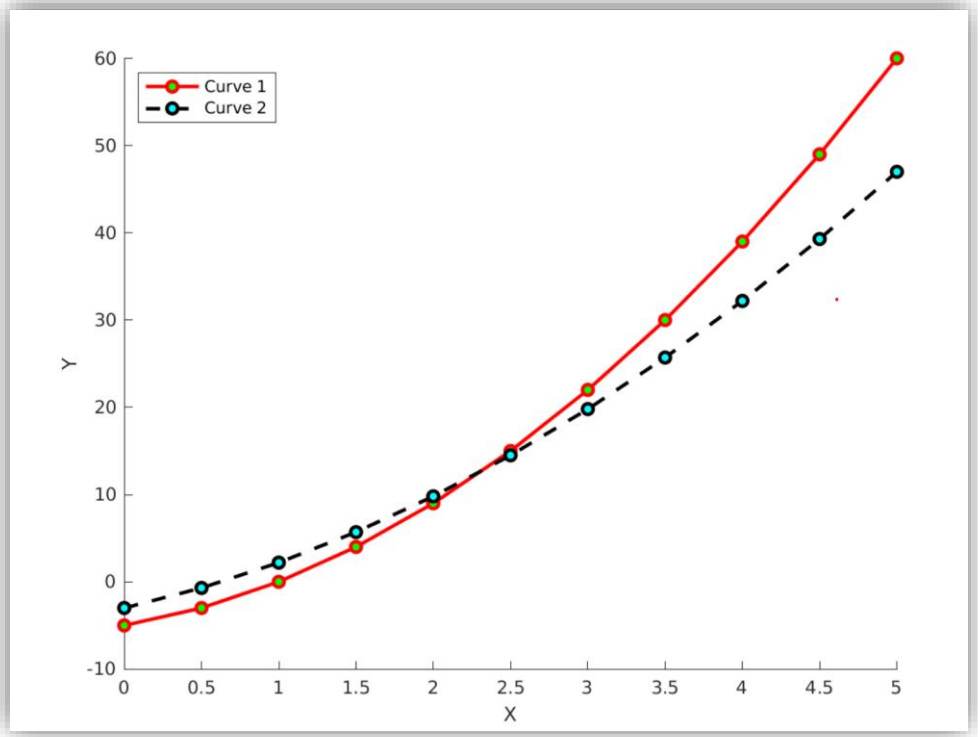
Batch submission

Example 2:

Matrix multiplication
Files:
`example_2.m`
`matlab.sh`

Files

- matlab-test.e<JOB_ID>
- matlab-test.o<JOB_ID>
- func_plot.png



Batch submission -

Parallel implementation, Maximum Eigenvalue of a matrix

TEXAS TECH UNIVERSITY
Information Technology Division

Serial version

Code:

```
%Start timing
tic

%Calculates spectral radius of each matrix and
displays results
n = 10;
A = 500;
a = zeros(n);
for i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;


%Display the time of computation
fprintf('The parallel method is executed in %5.2f
seconds.  \n', time);
```

Uniform distribution of random numbers

Parallel version

Parallel implementation – Maximum Eigenvalue of a matrix

TEXAS TECH UNIVERSITY
Information Technology Division

Serial version

Code:

```
%Start timing
tic

%Calculates spectral radius of each matrix and
displays results
n = 10;
A = 500;
a = zeros(n);
for i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f
seconds.  \n', time);
```

P 1

P 2

Parallel version

Code:

```
c = parcluster('local'); % local is a cluster profile name
sz = str2num([getenv('SLURM_CPUS_PER_TASK]);
if isempty(sz), sz = maxNumCompThreads; end
if isempty(gcp('nocreate')), c.parpool(sz); end

%Start timing
tic

%Calculates spectral radius of each matrix and displays
results
n = 10;
A = 500;
a = zeros(n);
parfor i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f seconds.
\n', time);
```

Parallel implementation – Components

TEXAS TECH UNIVERSITY
Information Technology Division

P1-

- Creates a cluster object representing the cluster identified by the cluster profile name *local*

P 1

Parallel version

Code:

```
c = parcluster('local'); % local is a cluster profile name
sz = str2num([getenv('SLURM_CPUS_PER_TASK]);
if isempty(sz), sz = maxNumCompThreads; end
if isempty(gcp('nocreate')), c.parpool(sz); end

%Start timing
tic

%Calculates spectral radius of each matrix and displays
results
n = 10;
A = 500;
a = zeros(n);
parfor i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f seconds.
\n', time);
```

Parallel implementation – Components

TEXAS TECH UNIVERSITY
Information Technology Division

P1-

- Picks up --*cpus-per-task* from batch submit script
- Assigns to the variable *sz*
- String to number check to ensure correct format

P 1

Parallel version

Code:

```
c = parcluster('local'); % local is a cluster profile name
sz = str2num([getenv('SLURM_CPUS_PER_TASK]);
if isempty(sz), sz = maxNumCompThreads; end
if isempty(gcp('nocreate')), c.parpool(sz); end

%Start timing
tic

%Calculates spectral radius of each matrix and displays
results
n = 10;
A = 500;
a = zeros(n);
parfor i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f seconds.
\n', time);
```

Parallel implementation – Components

TEXAS TECH UNIVERSITY
Information Technology Division

P1-

- If *--cpus-per-task* not defined, maximum number of cores are allotted

P 1

Parallel version

Code:

```
c = parcluster('local'); % local is a cluster profile name
sz = str2num([getenv('SLURM_CPUS_PER_TASK]);
if isempty(sz), sz = maxNumCompThreads; end
if isempty(gcp('nocreate')), c.parpool(sz); end

%Start timing
tic

%Calculates spectral radius of each matrix and displays
results
n = 10;
A = 500;
a = zeros(n);
parfor i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f seconds.
\n', time);
```

Parallel implementation – Components

TEXAS TECH UNIVERSITY
Information Technology Division

P1-

- If a parallel pool is not created, creates one and takes the size from *--cpus-per-task*

P 1

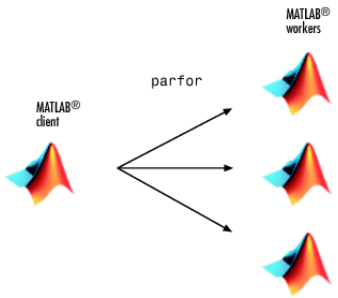
Parallel version

Code:

```
c = parcluster('local'); % local is a cluster profile name
sz = str2num([getenv('SLURM_CPUS_PER_TASK]);
if isempty(sz), sz = maxNumCompThreads; end
if isempty(gcp('nocreate')), c.parpool(sz); end

%Start timing
tic

%Calculates spectral radius of each matrix and displays
results
n = 10;
A = 500;
a = zeros(n);
parfor i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f seconds.
\n', time);
```

Parallel implementation – Components

P2- *parfor* splits the execution of for-loop iterations over the workers in a parallel pool



P 2

Parallel version

Code:

```
c = parcluster('local'); % local is a cluster profile name
% Query for available cores (assume either Slurm or PBS)
sz = str2num([getenv('SLURM_CPUS_PER_TASK]);
if isempty(sz), sz = maxNumCompThreads; end
if isempty(gcp('nocreate')), c.parpool(sz); end

%Start timing
tic

%Calculates spectral radius of each matrix and displays
results
n = 10;
A = 500;
a = zeros(n);
parfor i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f seconds.
\n', time);
```

Parallel implementation- Benefits

TEXAS TECH UNIVERSITY
Information Technology Division

1st Run

Serial version	Parallel version
1 core	**10 cores**
1304.93 seconds(~21.75 mins)	133.62 seconds (~2.3 mins)

Next Run

Serial version	Parallel version
1 core	**10 cores**
10.31 seconds	1.89 seconds

Parallel implementation-
Multi core parallel implementation

TEXAS TECH UNIVERSITY
Information Technology Division

Nocona – 128 cores/node

https://www.depts.ttu.edu/hpcc/operations/equipment.php

Partition:	Nocona	Quanah / XLQuanah	Matador	Toreador	Ivy
Type	CPU	CPU	GPU	GPU	Auxiliary CPU*
Total Nodes	240	467 / 16	20	11	50 / 2
Theoretical Max	983 TFLOPS	565 TFLOPS /19 TFLOPS	280 TFLOPS	287 TFLOPS	40 TFLOPS
Benchmarked	804 TFLOPS	485 TFLOPS / (N/A)	226 TFLOPS		N/A
OS	CentOS 8.1	CentOS 7.4 /CentOS 8.1	CentOS 8.1	CentOS 8.1	Rocky Linux 8.5 /CentOS 8.1
Manufacturer	Dell	Dell	Dell	Dell	Dell
Node Model	PowerEdge C6525	PowerEdge C6320	PowerEdge R740	Poweredge R7525	PowerEdge C6220 II
Cooling	Liquid Cooled	Air Cooled	Air Cooled	Air Cooled	Air Cooled
Processor Make and Model	AMD EPYC™ 7702	Intel Xeon E5-2695 v4	Intel Xeon Gold 6242	AMD EPYC™ 7262	Intel Xeon E5-2670v2
Family	Rome	Broadwell	Cascade Lake	Rome	Ivy Bridge
Cores/Processor	64	18	20	8	10
Cores/Node	128	36	40 cpu + 1280 tensor + 10,240 CUDA	16 cpu + 1,296 tensor + 20,736 CUDA	20
Total Cores In Partition	30,720	16,812 / 576	800 cpu + 25,600 tensor +	528 cpu + 14,256 tensor+	1,000 / 40

MATLAB Graphical Interface

Interactive session

$ interactive -p nocona -c 8

$ interactive -h

Number of cores

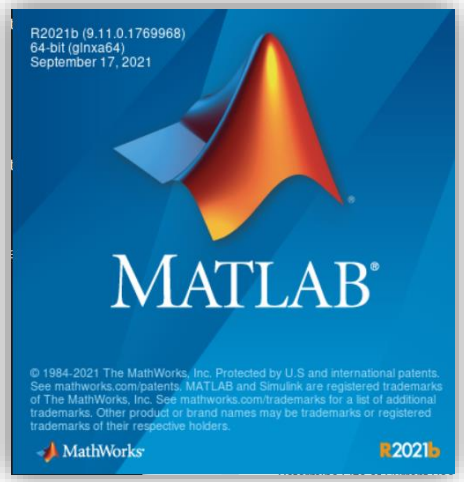
Setting up environment

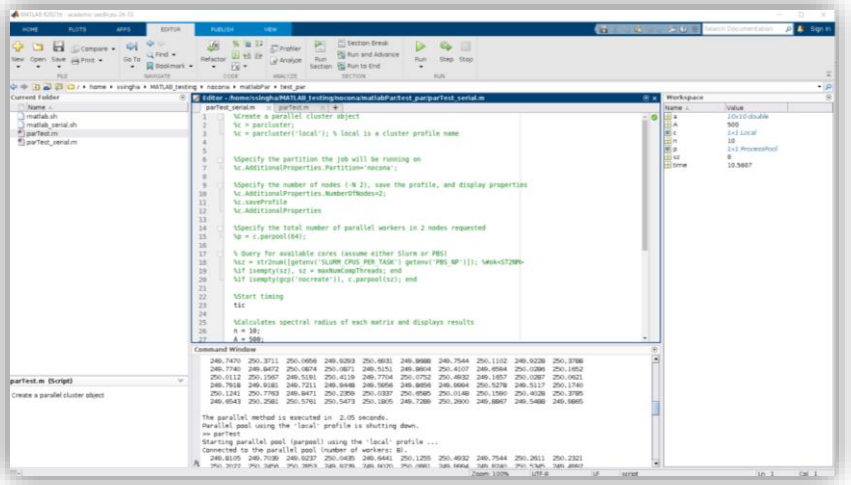
$module load matlab
$ml list

Launching Client

$matlab

```
cpu-24-33:/MATLAB_testing/nocona/matlabPar$ ml matlab
cpu-24-33:/MATLAB_testing/nocona/matlabPar$ ml list

Currently Loaded Modules:
  1) nocona/0.15.4    2) matlab/R2021b
cpu-24-33:/MATLAB_testing/nocona/matlabPar$
```





Parallel implementation – Maximum Eigenvalue of a matrix

Link: /matlab/matlabPar/parTest_interactive

Serial version

parTest_serial.m

Parallel version

parTest.m

Parallel implementation – Maximum Eigenvalue of a matrix

TEXAS TECH UNIVERSITY
Information Technology Division

Serial version

Code:

```
%Start timing
tic

%Calculates spectral radius of each matrix and
displays results
n = 10;
A = 500;
a = zeros(n);
for i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f
seconds.  \n', time);
```

Parallel version

Code:

```
c = parcluster('local'); % local is a cluster profile name

p = c.parpool(10);

%Start timing
tic

%Calculates spectral radius of each matrix and displays
results
n = 10;
A = 500;
a = zeros(n);
parfor i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f seconds.
\n', time);
```

Parallel implementation – Maximum Eigenvalue of a matrix

TEXAS TECH UNIVERSITY
Information Technology Division

Object representing cluster identified by the cluster profile name *local*

Starts a parallel pool of 10 workers using the cluster object *c*

- Splits the execution of for-loop iterations over the workers in a parallel pool
- Say n*n=10, 10 workers available, MATLAB assigns it to each core

Parallel version

Code:

```
c = parcluster('local'); % local is a cluster profile name


p = c.parpool(10);

%Start timing
tic

%Calculates spectral radius of each matrix and displays
results
n = 10;
A = 500;
a = zeros(n);
parfor i = 1:n*n
    a(i) = max(abs(eig(rand(A))));
end
disp(a)

%End timing
time = toc;

%Display the time of computation
fprintf('The parallel method is executed in %5.2f seconds.
\n', time);
```

MATLAB Graphical Interface – Parallel implementation via parpool

TEXAS TECH UNIVERSITY
Information Technology Division

Client outputs

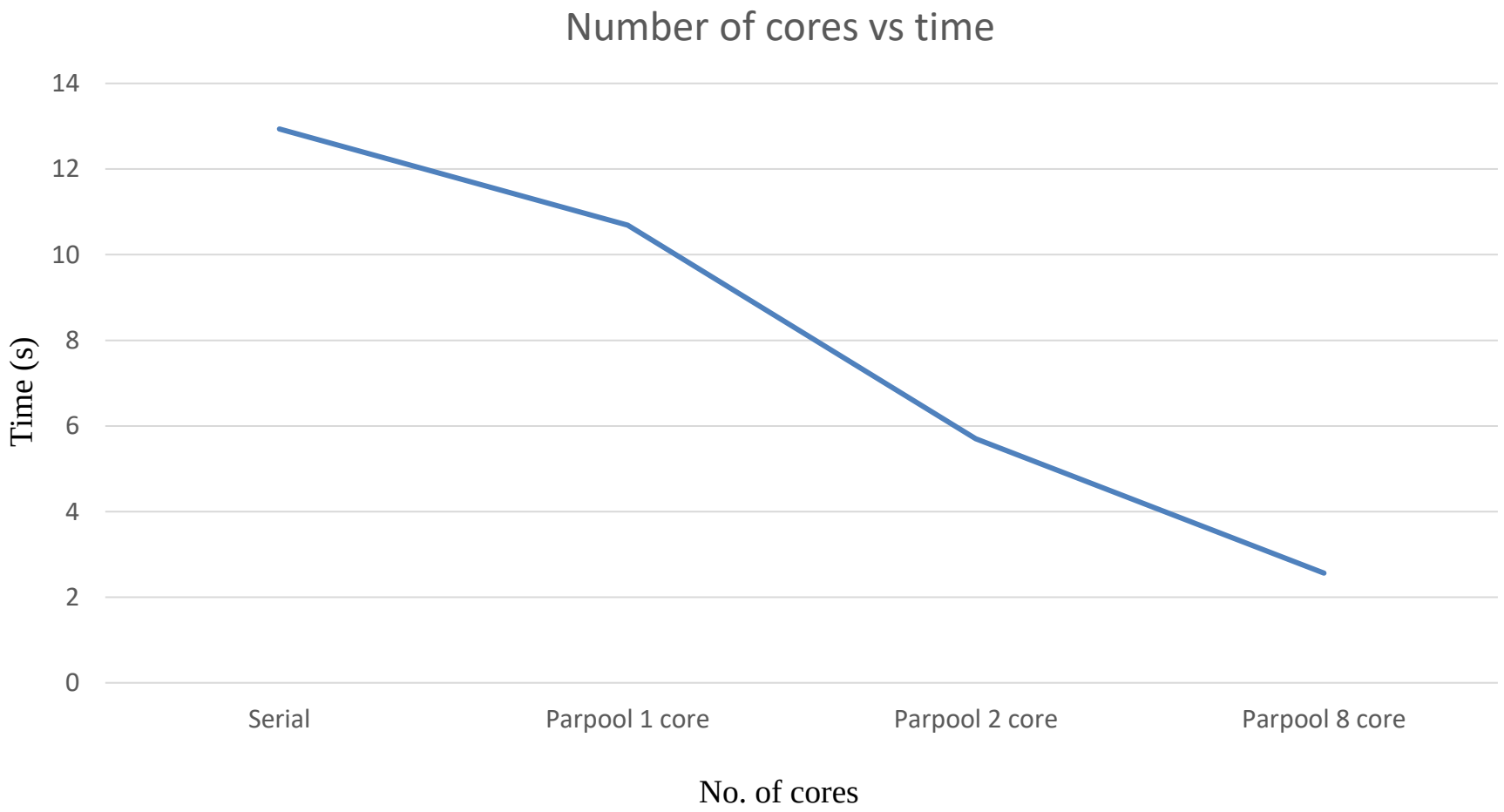
```
>> parTest_serial
…
The parallel method is executed in 12.94 seconds.
```

```
>> parTest
Starting parallel pool (parpool) using the 'local' profile ...
Connected to the parallel pool (number of workers: 1).
…
The parallel method is executed in 10.69 seconds.
Parallel pool using the 'local' profile is shutting down.
```

```
>>… Connected to the parallel pool (number of workers: 2).
The parallel method is executed in 5.70 seconds …
```

```
>>… Connected to the parallel pool (number of workers: 8).
The parallel method is executed in 2.56 seconds …
```

MATLAB Graphical Interface – Parallel implementation via parpool



Parallel implementation-
Multi core parallel implementation

TEXAS TECH UNIVERSITY
Information Technology Division

Nocona – 128 cores/node

https://www.depts.ttu.edu/hpcc/operations/equipment.php

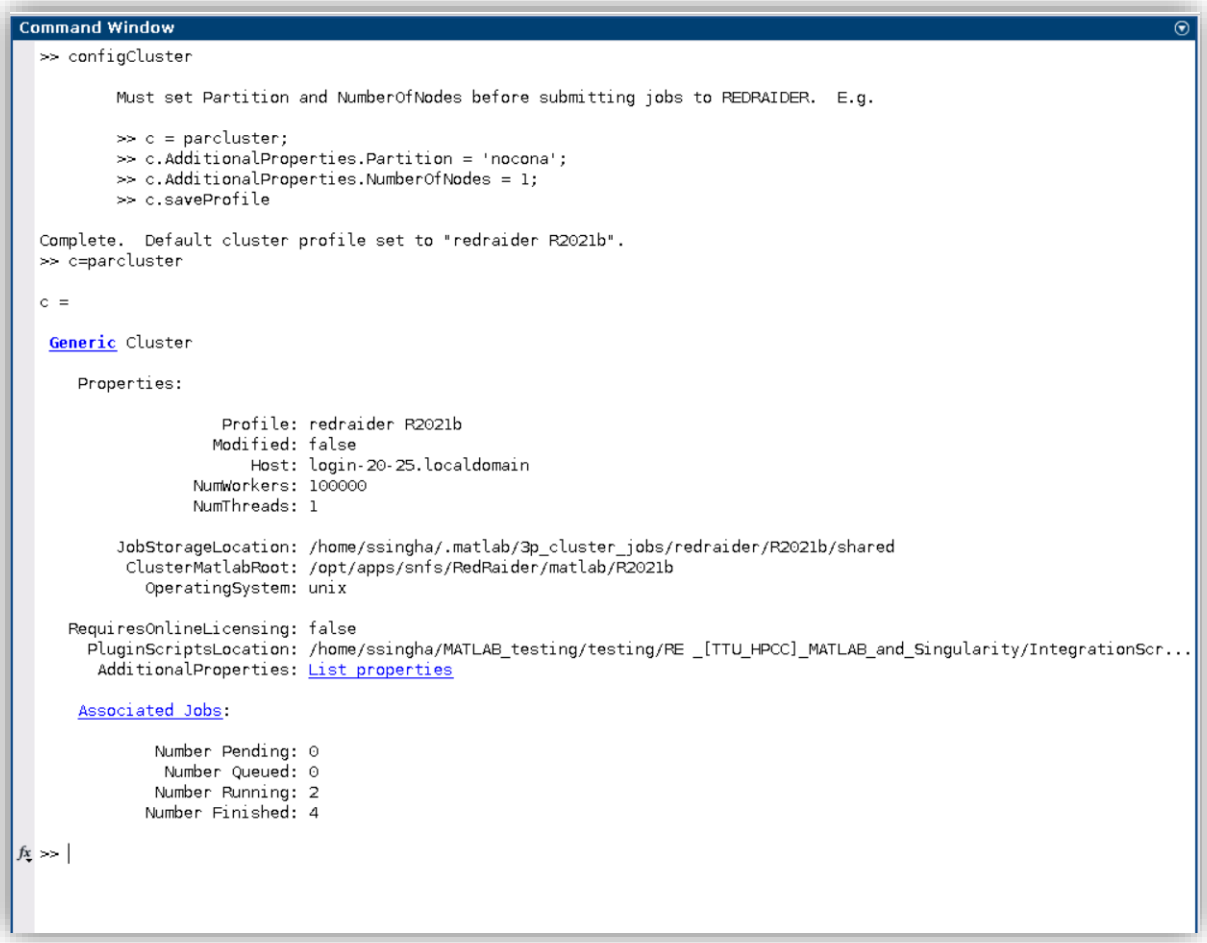
Partition:	Nocona	Quanah / XLQuanah	Matador	Toreador	Ivy
Type	CPU	CPU	GPU	GPU	Auxiliary CPU*
Total Nodes	240	467 / 16	20	11	50 / 2
Theoretical Max	983 TFLOPS	565 TFLOPS /19 TFLOPS	280 TFLOPS	287 TFLOPS	40 TFLOPS
Benchmarked	804 TFLOPS	485 TFLOPS / (N/A)	226 TFLOPS		N/A
OS	CentOS 8.1	CentOS 7.4 /CentOS 8.1	CentOS 8.1	CentOS 8.1	Rocky Linux 8.5 /CentOS 8.1
Manufacturer	Dell	Dell	Dell	Dell	Dell
Node Model	PowerEdge C6525	PowerEdge C6320	PowerEdge R740	Poweredge R7525	PowerEdge C6220 II
Cooling	Liquid Cooled	Air Cooled	Air Cooled	Air Cooled	Air Cooled
Processor Make and Model	AMD EPYC™ 7702	Intel Xeon E5-2695 v4	Intel Xeon Gold 6242	AMD EPYC™ 7262	Intel Xeon E5-2670v2
Family	Rome	Broadwell	Cascade Lake	Rome	Ivy Bridge
Cores/Processor	64	18	20	8	10
Cores/Node	128	36	40 cpu + 1280 tensor + 10,240 CUDA	16 cpu + 1,296 tensor + 20,736 CUDA	20
Total Cores In Partition	30,720	16,812 / 576	800 cpu + 25,600 tensor +	528 cpu + 14,256 tensor+	1,000 / 40

Job submission – Asynchronous Batch GUI

- Run once
- Re-runs deletes older profile, creates a new profile
- All saved data deleted

- *>>c=parcluster;*
- Default cluster profile is set as **redraider**
- Always makes a new secondary job submission to Slurm

- *>>c=parcluster('local');*
- Opens cluster profile to utilize present allocation
- Uses just resources on present node, no submission to scheduler

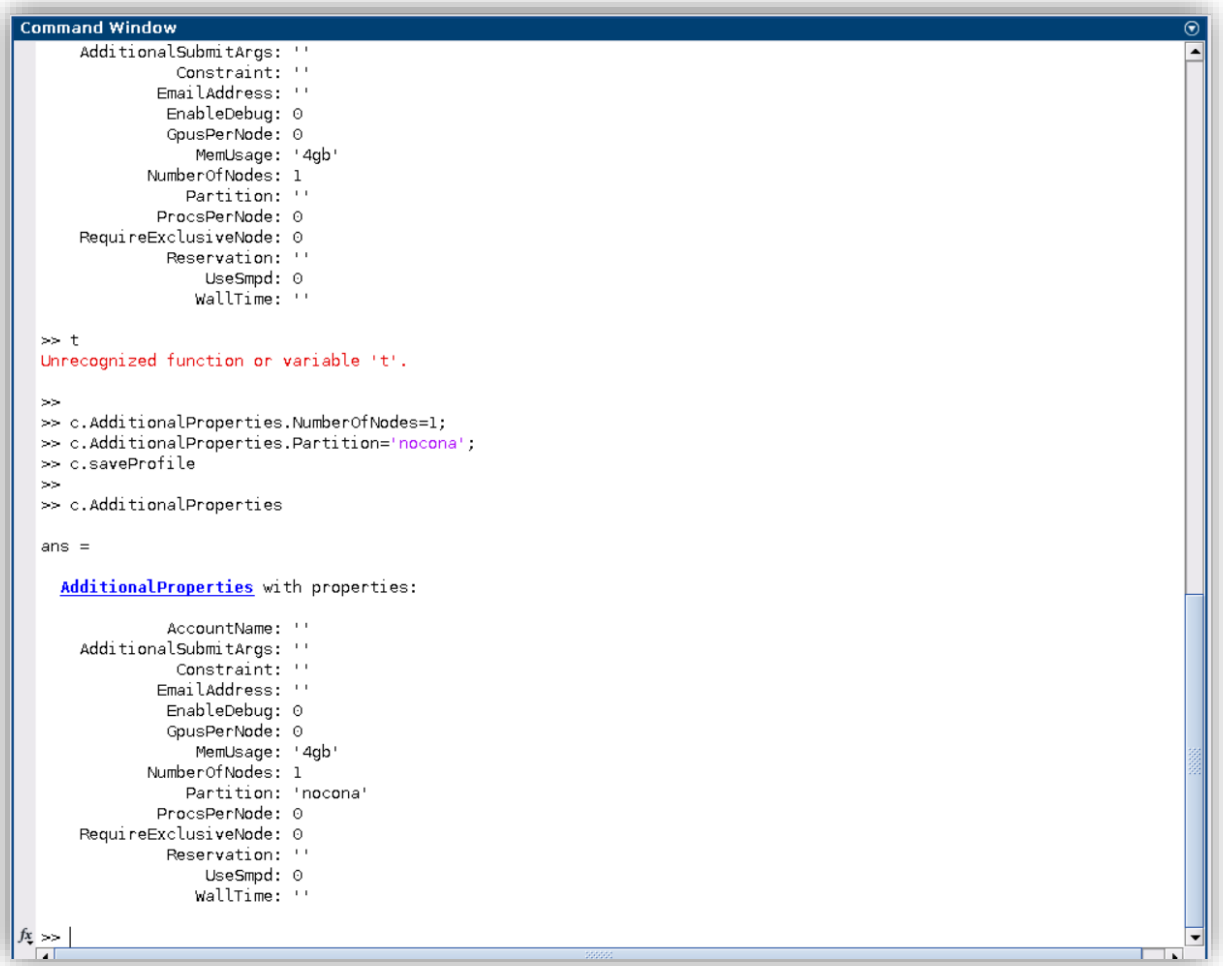


Job submission – Asynchronous Batch GUI

TEXAS TECH UNIVERSITY
Information Technology Division

Profile Properties

- Slurm properties that can be modified
- *AdditionalSubmitArgs* – Manually allows user to pass arguments, flags say constrains of the cluster
- c.Additional.Properties.NumberofNodes = 1;
- c.AdditionalProperties.Partitions=‘nocona’;
- c.saveProfile
- Creates a submission script and performs the submission for the user



Job submission – Asynchronous Batch GUI

Job object name Function name Number of outputs

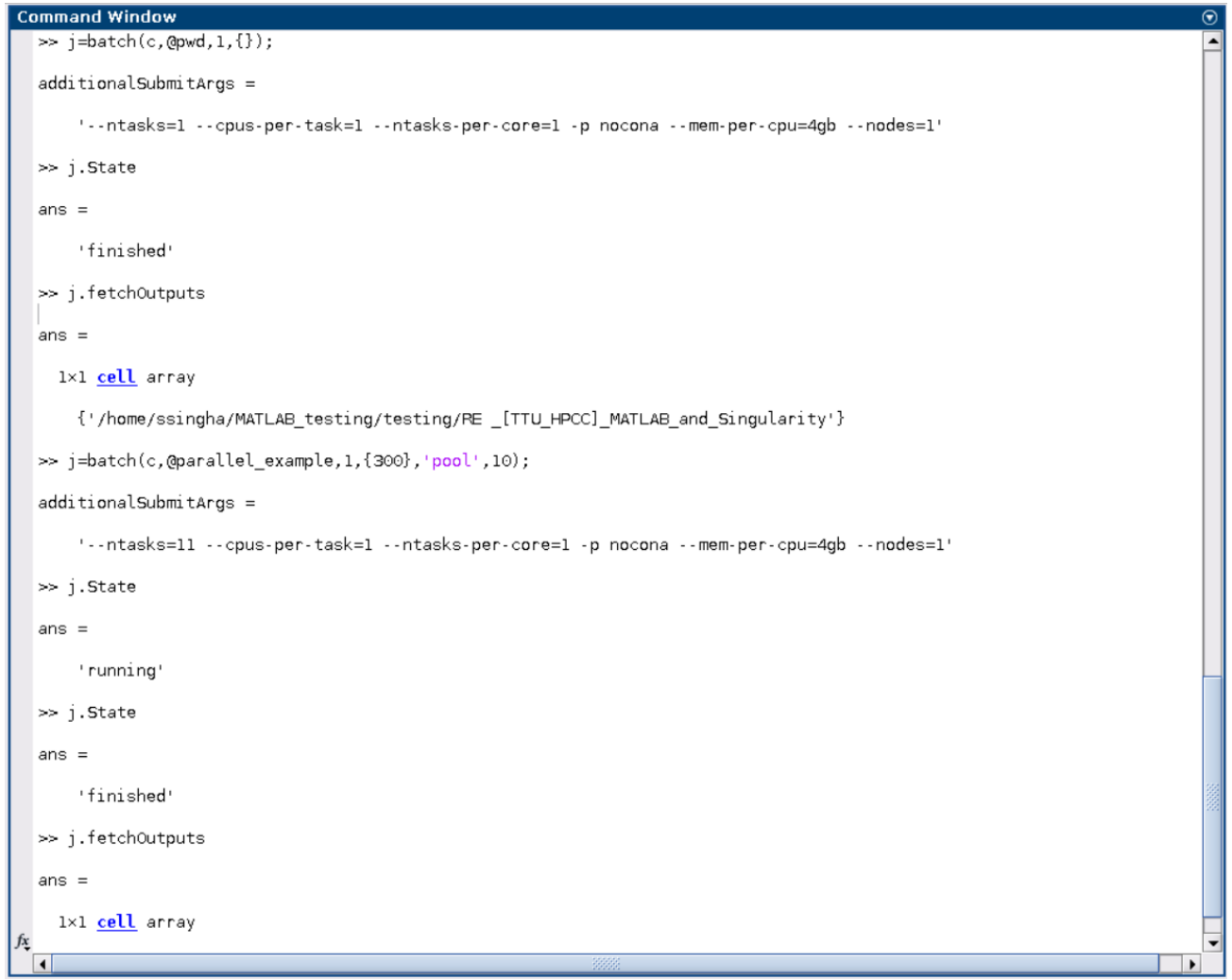
```
>>j=batch(c,@pwd,1,{});
```

Inputs to function

>>j.State – returns status of job

>>j.fetchOutputs – returns output of job

>>j.diary – returns command line outputs



Job submission – Asynchronous Batch GUI

TEXAS TECH UNIVERSITY
Information Technology Division

Serial Example

```
function [t, A] = parallel_example(iter)
if nargin==0
    iter = 8;
end
disp('Start sim')
t0 = tic;
for idx = 1:iter
    A(idx) = idx;
    pause(2)
    idx
end
t = toc(t0);
disp('Sim completed')
save RESULTS A
end
```

```
>> j=batch(c,@serial_example,1,{8});

>> j.State

>> j.fetchOutputs

ans =
  1×1 cell array
    {[16.0098]}
```

Job submission – Asynchronous Batch GUI

Parallel Example

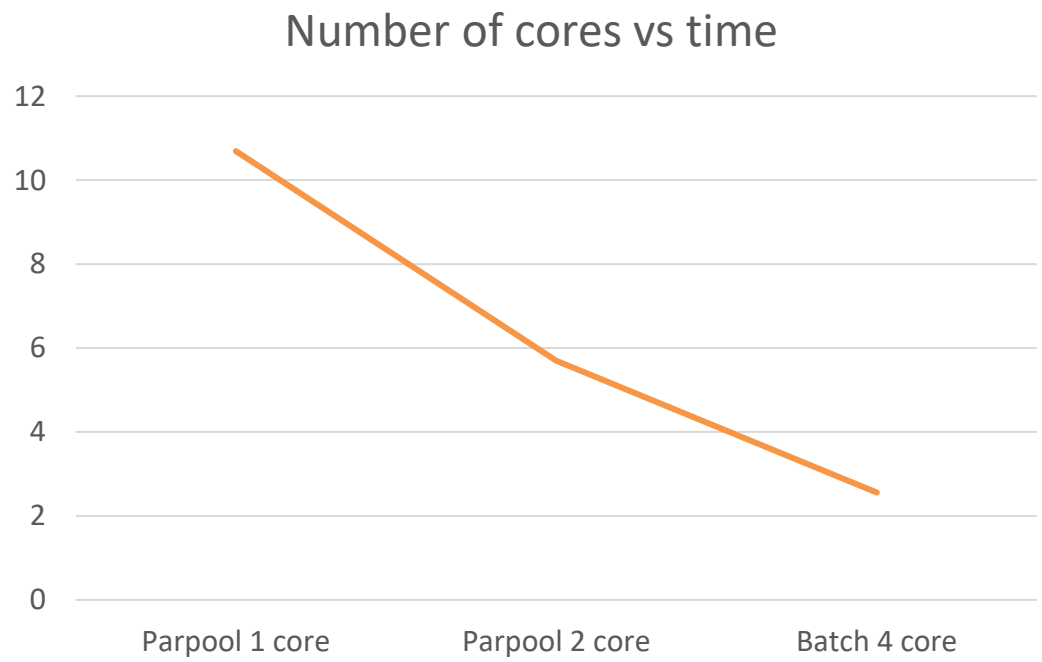
```
function [t, A] = parallel_example(iter)
if nargin==0
    iter = 8;
end
disp('Start sim')
t0 = tic;
parfor idx = 1:iter
    A(idx) = idx;
    pause(2)
    idx
end
t = toc(t0);
disp('Sim completed')
save RESULTS A
end
```

```
>> j=batch(c,@serial_example,1,{8});

{[16.0098]}

>>j=batch(c,@parallel_example,1,{8},'pool',2);
{[8.3045]}

>> =batch(c,@parallel_example,1,{8},'pool',4);
{[4.3124]}
```

Job submission – Asynchronous Batch GUI

Parallel Example

```
function [t, A] = parallel_example(iter)
if nargin==0
    iter = 8;
end
disp('Start sim')
t0 = tic;
parfor idx = 1:iter
    A(idx) = idx;
    pause(2)
    idx
end
t = toc(t0);
disp('Sim completed')
save RESULTS A
end
```



Job submission – Asynchronous Batch GUI

Job Arrays

```
>> j=batch(c,@sum,1,{[1 1]});
```

```
>> j=createJob(c);
createTask(j,@sum,1,{[1 1]});
createTask(j,@sum,1,{[1 1]});
createTask(j,@sum,1,{[1 1]});
submit(j);
```

```
>>
>> j=createJob(c);
createTask(j,@sum,1,{[1 1]});
createTask(j,@sum,1,{[1 1]});
createTask(j,@sum,1,{[1 1]});
submit(j);

additionalSubmitArgs =

    '--ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 -p nocona --mem-per-cpu=4gb --nodes=1'

>> j.Tasks

ans =

 3x1 Task array:

        ID       State              FinishDateTime  Function  Errors  Warnings  SchedulerID
    ------------------------------------------------------------------------------------------
    1    1      running                                 sum       0         0    7632475_1
    2    2      pending                                 sum       0         0    7632475_2
    3    3      pending                                 sum       0         0    7632475_3

>>
>>
>>
```

Job submission – Asynchronous Batch GUI

TEXAS TECH UNIVERSITY
Information Technology Division

Job Arrays

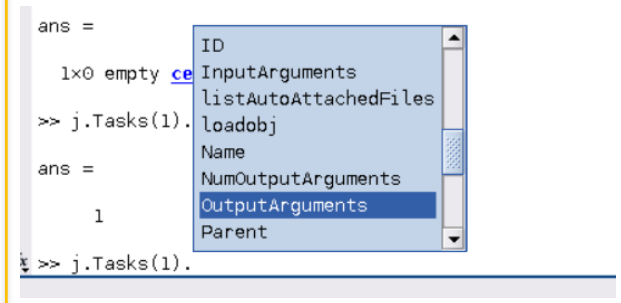
>> j.Tasks

```
>> j.Tasks

ans =

 3x1 Task array:

        ID       State             FinishDateTime  Function  Errors  Warnings
      ------------------------------------------------------------------------
    1    1      pending                                sum       0         0
    2    2      pending                                sum       0         0
    3    3      pending                                sum       0         0
```

>> j.Tasks(2).cancel
>> j.Tasks(1).State

```
>> j.Tasks(2).cancel
>> j.Tasks

ans =

 3x1 Task array:

        ID       State             FinishDateTime  Function  Errors  Warnings
      ------------------------------------------------------------------------
    1    1      pending                                sum       0         0
    2    2     finished       09-Nov-2022 00:11:14      sum       1         0
    3    3      pending                                sum       0         0
```

Monitoring job parameters



Job submission – Interactive pool submission

TEXAS TECH UNIVERSITY
Information Technology Division

Opening an interactive parpool (local profile)

```
>> c=parcluster('local');
>> p=c.parpool(n)
```

n is the number of cores across the cluster requested

```
Command Window
>>
>> p=c.parpool(10)
Starting parallel pool (parpool) using the 'redraider R2021b' profile ...

additionalSubmitArgs =

    '--ntasks=10 --cpus-per-task=1 --ntasks-per-core=1 -p nocona --mem-per-cpu=4gb --nodes=1'

Connected to the parallel pool (number of workers: 10).

p =

 ClusterPool with properties:

           Connected: true
          NumWorkers: 10
             Cluster: redraider R2021b
       AttachedFiles: {}
    AutoAddClientPath: true
         IdleTimeout: 30 minutes (30 minutes remaining)
         SpmdEnabled: true
 EnvironmentVariables: {}
```

Running a parpool job (local profile)

```
>> parallel_example
```

Job submission – Interactive pool submission

TEXAS TECH UNIVERSITY
Information Technology Division

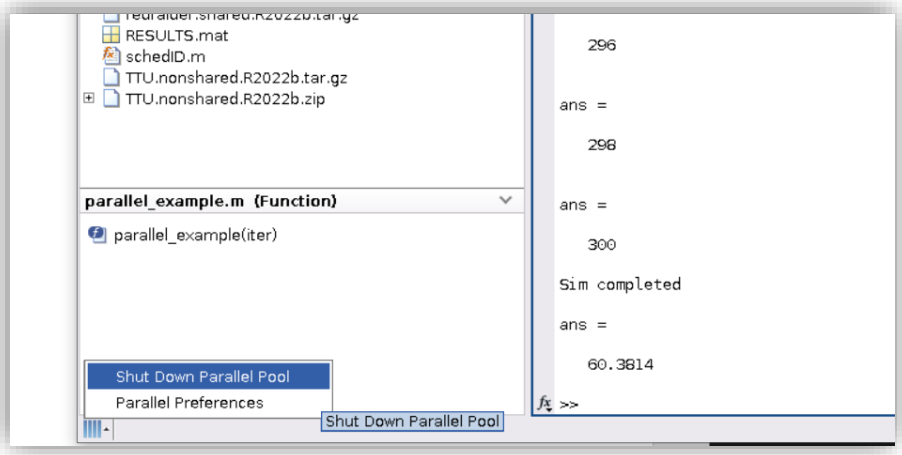
Another parallel construct like *parfor*

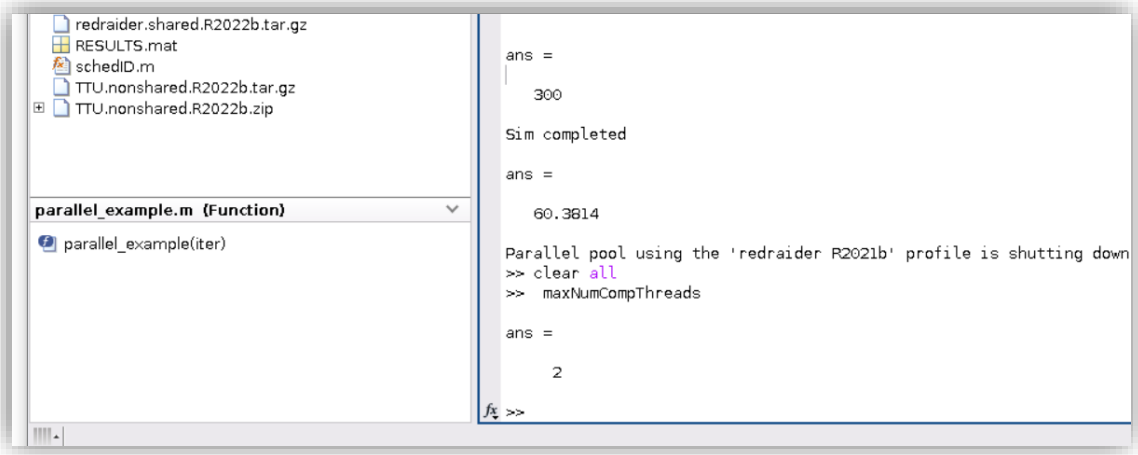
```
>> spmd,getCurrentWorker,end
Worker  1:
  ans =
   Worker
                    Host: cpu-26-56.localdomain
            ComputerType: GLNXA64
                 ProcessId: 95462
  Worker  2:
    ans =
     Worker
                    Host: cpu-26-56.localdomain
            ComputerType: GLNXA64
                 ProcessId: 95463
...
```

Job submission – Interactive pool submission

TEXAS TECH UNIVERSITY
Information Technology Division

Shutting down Parallel pool



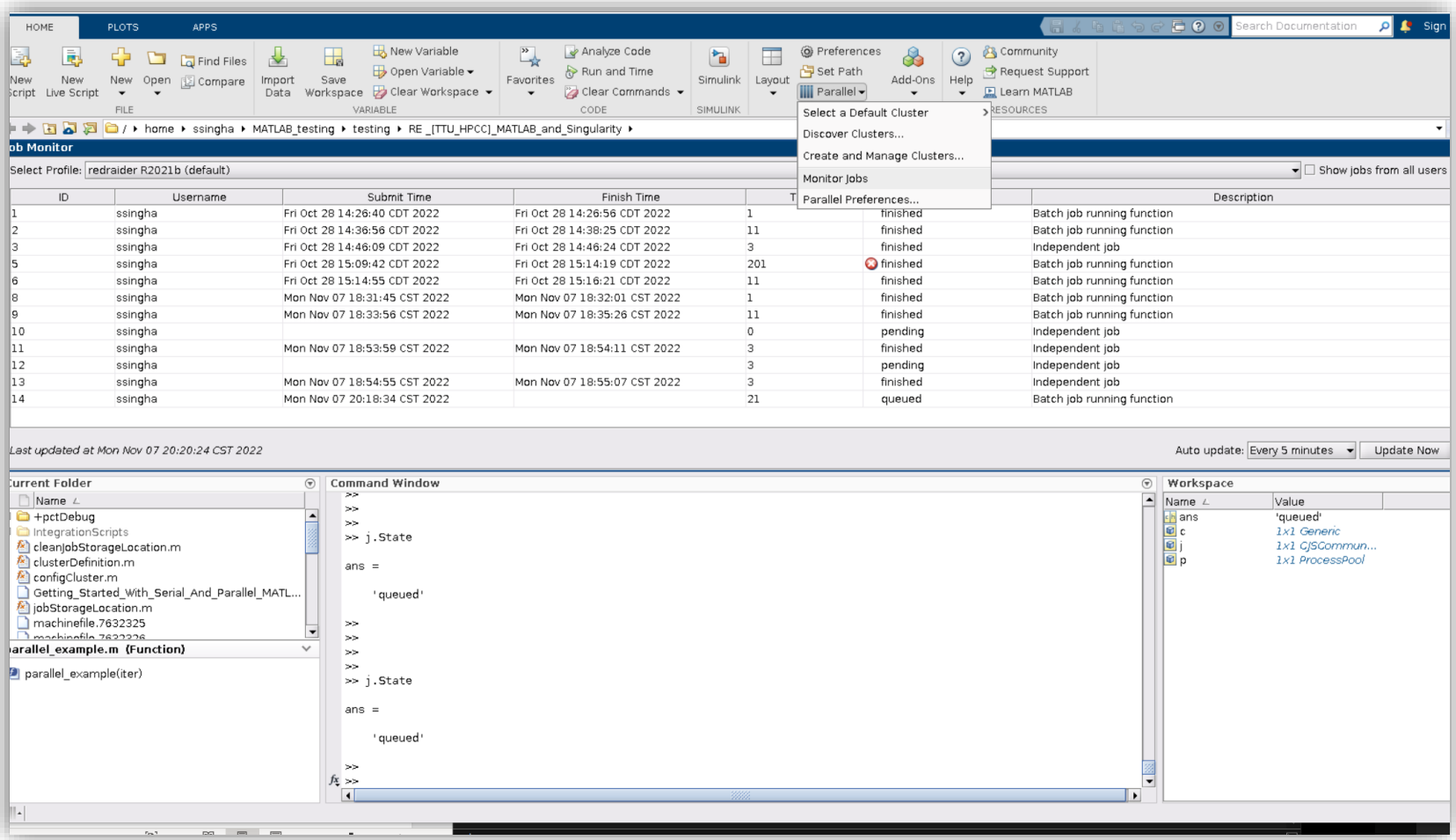



```
>>clear all
```

Job submission – Interactive pool submission

TEXAS TECH UNIVERSITY
Information Technology Division

Monitoring

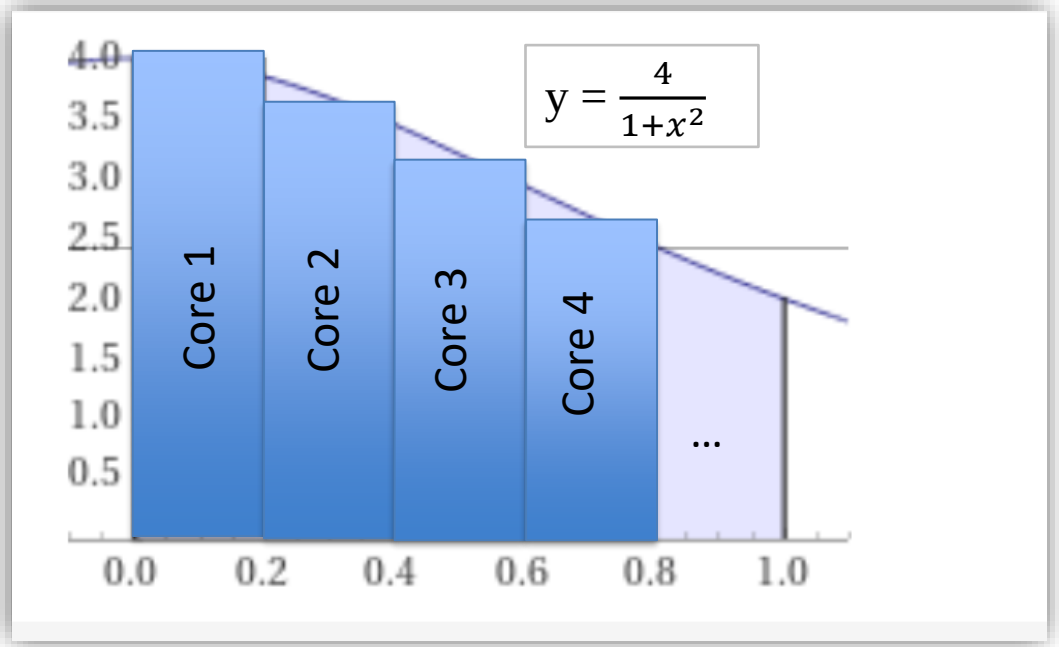


Parallel job implementation example

Calculation of π

(based on Leibniz formula)

$$\int_0^1 \frac{4}{1+x^2}\,dx = 4\arctan = \pi$$



Plotted in Wolfram Alpha

Parallel job implementation example

TEXAS TECH UNIVERSITY
Information Technology Division

Single node parallel

```
function calc_pi

c = parcluster('local');

%sz = str2num([getenv('SLURM_CPUS_PER_TASK')]); %#ok<ST2NM>
sz = str2num([getenv('SLURM_CPUS_PER_TASK')]); %#ok<ST2NM>
if isempty(sz), sz = maxNumCompThreads; end

if isempty(gcp('nocreate')), c.parpool(sz); end

spmd
    a = (labindex - 1)/numlabs;
    b = labindex/numlabs;
    fprintf('Subinterval: [%-4g, %-4g]\n', a, b)

    myIntegral = integral(@quadpi, a, b);
    fprintf('Subinterval: [%-4g, %-4g]   Integral: %4g\n', a, b, myIntegral)

    piApprox = gplus(myIntegral);
end

approx1 = piApprox{1};  % 1st element holds value on worker 1
fprintf('pi           : %.18f\n', pi)
fprintf('Approximation: %.18f\n', approx1)
fprintf('Error        : %g\n',    abs(pi - approx1))

function y = quadpi(x)
%QUADPI Return data to approximate pi.

% Derivative of 4*atan(x)
y = 4./(1 + x.^2);
```

Multi node parallel

```
function calc_pi_multi_node

c = parcluster;

% Required fields
c.AdditionalProperties.WallTime = '00:20:00';
c.AdditionalProperties.Partition = 'nocona';
c.AdditionalProperties.NumberOfNodes = 1;

if isempty(gcp('nocreate')), c.parpool(20); end

spmd
    a = (labindex - 1)/numlabs;
    b = labindex/numlabs;
    fprintf('Subinterval: [%-4g, %-4g]\n', a, b)

    myIntegral = integral(@quadpi, a, b);
    fprintf('Subinterval: [%-4g, %-4g]   Integral: %4g\n', a, b,
myIntegral)

    piApprox = gplus(myIntegral);
end

approx1 = piApprox{1};  % 1st element holds value on worker 1
fprintf('pi           : %.18f\n', pi)
fprintf('Approximation: %.18f\n', approx1)
fprintf('Error        : %g\n',    abs(pi - approx1))


function y = quadpi(x)
%QUADPI Return data to approximate pi.

% Derivative of 4*atan(x)
y = 4./(1 + x.^2);
```

Parallel job implementation example

TEXAS TECH UNIVERSITY
Information Technology Division

Accuracy of numerical calculation

```
>> calc_pi
Starting parallel pool (parpool) using the 'local' profile ...
Connected to the parallel pool (number of workers: 4).
Worker 1:
  Subinterval: [0   , 0.25]
  Subinterval: [0   , 0.25]   Integral: 0.979915
Worker 2:
  Subinterval: [0.25, 0.5 ]
  Subinterval: [0.25, 0.5 ]   Integral: 0.874676
Worker 3:
  Subinterval: [0.5 , 0.75]
  Subinterval: [0.5 , 0.75]   Integral: 0.719414
Worker 4:
  Subinterval: [0.75, 1   ]
  Subinterval: [0.75, 1   ]   Integral: 0.567588
pi            : 3.141592653589793116
Approximation: 3.141592653589793560
Error         : 4.44089e-16
```

Lower

```
>> calc_pi_multi_node
Starting parallel pool (parpool) using the 'teton R2022a' profile ...

additionalSubmitArgs =

    '--ntasks=20 --cpus-per-task=1 --ntasks-per-core=1

Connected to the parallel pool (number of workers: 20).
Worker  1:
  Subinterval: [0   , 0.025]
Worker  2:
  Subinterval: [0.025, 0.05]
Worker  3:
  Subinterval: [0.05, 0.075]

. . .

Worker 18:
  Subinterval: [0.425, 0.45]   Integral: 0.0839331
Worker 20:
  Subinterval: [0.525, 0.55]   Integral: 0.0775848
pi            : 3.141592653589793116
Approximation: 3.141592653589793116
Error         : 0
```

Lower

Containerized MATLAB

1. Login into login.hpcc.ttu.edu using your eRaider account and password.
2. Get an interactive session on a partition (say nocona)
   $ interactive -p nocona
3. Check singularity installation and version
   $ singularity --version
   singularity version 3.7.3-1.el8
4. Pull a sample singularity container to get started
   $ singularity pull shub://singularityhub/hello-world

```
cpu-25-27:/singularity_testing/nocona$ singularity pull shub://singularityhub/hello-world
INFO:    Downloading shub image
59.8MiB / 59.8MiB [==========================================================================================================================================] 100 % 44.5 MiB/s 0s
cpu-25-27:/singularity_testing/nocona$ ls
hello-world_latest.sif
```

Containerized MATLAB

1. In the event of errors, re-runs execute
   $ singularity cache list
   to list container files, present in the account memory
   $ singularity cache clean
   to remove such files before beginning a fresh pull
2. Execute the singularity image file (*.sif)
   $ ./hello-world_latest.sif
   Tacotacotaco
3. Alternate ways to explore yourself:
   Directly run a singularity container
   $ singularity run hello-world_latest.sif
4. Pull latest MATLAB docker container:
   $ singularity pull docker://mathworks/matlab:r2022b

Thoughts ?

Feedback