Parallel MATLAB on HPC

Avalon Johnson, Jimi Chui, Cesar Sul and Erin Shaw

USC Center for High-Performance Computing (HPC)
HPC Research and Education Facilitation

USC HPC

- Foundation for computational research at USC
  - Shared resource that is freely available to faculty, researchers and graduate students
  - World class supercomputing center
    - Currently ranked 12th fastest academic cluster

- Research & Education Facilitation
  - Drop-in to office hours or request a consultation
  - Attend a new user meeting or participate in a workshop
    - See https://hpc.usc.edu/education
    - See https://hpc.usc.edu/officehours
    - Email hpc@usc.edu
Outline

- **Introduction**
  - Prerequisites, setup workspace and launch MATLAB
  - Introduction to parallel MATLAB
- **Run MATLAB on single node w/ Parallel Computing Toolbox (PCT)**
  - Using graphical user interface (GUI)
  - Using command line interface (CLI)
  - Using CLI with batch parallel job execution
- **Run MATLAB on multiple nodes w/ Distributed Computing Server (DCS)**
  - Using command line interface (CLI)
- **Beyond parfor**
  - labIndex, Composite, and Broadcast&Receive
  - Practice using PBS
Prerequisites

- You will need *X11 Display Forwarding* to use the GUI
  - On Windows: Set ‘Enable X11 forwarding’ on your client (this is usually the default)
  - On Macs: Download and install Xquartz (www.xquartz.org)

- Login
  
  ```
  $ ssh -X {USCNetId}@hpc-login3.usc.edu
  ```

- Run ‘xeyes’ program to verify that display is working
  
  ```
  $ xeyes #now move cursor
  $ ^C #control-c to quit
  ```

Workspace setup

- Always work in your project directory!
  
  ```
  $ cd /home/rcf-proj/{project}/{user}
  ```

- Create a matlab workshop directory & copy workshop files to it
  
  ```
  $ mkdir matlab
  $ cd matlab
  $ cp /home/rcf-proj/workshop/matlab/* .
  ```

- You should now have the following files
  
  ```
  $ ls
  broadcastReceive.m EstimatePi.m jobScript.m labIndex.m submit.pbs submitTasks.m
  ```
Request an interactive compute node

$ qsub -I -l walltime=02:00:00 -l procs=8 -l software=MPCT+1
   -A workshop -l 'advres=HPCWorkshop.xx'

- l procs=8
  Use "procs" instead of typical "nodes=x:ppn=x"
  Distributes MATLAB processes across nodes most efficiently
- l software=MPCT+1
  Checkout 1 MATLAB PCT license for interactive application (client)
  Each core needs a license – MATLAB will request more licenses per our configuration
- l advres=HPCWorkshop.xx -A workshop
  Use nodes reserved in advance for this workshop
  Omit these lines after workshop

NOTE: When you run real jobs, if you have access to a condo queue, be sure to include
the –q <queue_name> option
Launch MATLAB in second window

This is your original hpc-login shell
When ‘qsub’ succeeds, a compute node shell will open in this window
Do not run MATLAB with GUI in this shell, it will be very slow; run in new terminal window instead
Use this window to edit your program and monitor your job

Open a second terminal window and login to head node (with –X)
$ ssh –X hpc-login3.usc.edu
Then login to compute node and run MATLAB from here
$ ssh –X hpcxxxx
$ source /usr/usc/matlab/R2016a/setup.sh
$ matlab

Outline

- Introduction
  - Prerequisites, setup workspace and launch MATLAB
  - Introduction to parallel MATLAB
- Run MATLAB on single node w/ Parallel Computing Toolbox (PCT)
  - Using graphical user interface (GUI)
  - Using command line interface (CLI)
  - Using CLI with batch parallel job execution
- Run MATLAB on multiple nodes w/ Distributed Computing Server (DCS)
  - Using command line interface (CLI)
- Beyond parfor
  - labIndex, Composite, and Broadcast&Receive
  - Practice using PBS
How MATLAB supports parallelization

- **Cluster objects**
  - `parallel.cluster.Local` (interact with cluster on local machine)
  - `parallel.cluster.Torque` (interact with cluster running Torque)
  - `parallel.cluster.Mpiexec` (interact with cluster using mpiexec)

- **Parallel Computing Toolbox (USC licensed)**
  - `parpool` + `parfor`

- **Distributed Computing Server (USC licensed)**
  - `batch parfor`, `spmd` (single program, multiple data) jobs,
    `spmd` MPI communication

---

MATLAB’s parallel machinery

![Diagram showing the flow of jobs from client to worker nodes through cluster nodes and job statuses (Pending, Submitted, Queued, Running, Finished).]
MATLAB program: EstimatePi.m

```matlab
% Calculate the value of Pi using a Monte Carlo simulation
max=1e9;
tic;
n=0;
parfor i = 1:max
    x=rand;
y=rand;
    if (x^2 + y^2 < 1.0)
        n=n+1;
    end
end
elapsedTime = toc;
pi = (4.0 * n / max);
```

MATLAB’s parfor loop

The Mechanics of parfor Loops
Outline

- Introduction
  - Prerequisites, setup workspace and launch MATLAB
  - Introduction to parallel MATLAB
- Run MATLAB on single node w/ Parallel Computing Toolbox (PCT)
  - Using graphical user interface (GUI)
  - Using command line interface (CLI)
  - Using CLI with batch parallel job execution
- Run MATLAB on multiple nodes w/ Distributed Computing Server (DCS)
  - Using command line interface (CLI)
- Beyond parfor
  - labIndex, Composite, and Broadcast&Receive
  - Practice using PBS

Run on single node with PCT (GUI)
MATLAB’s Parallel Computing Toolbox (MPCT)

- In second terminal window
  ```
  $ ssh -X hpcxxxx
  $ source /usr/usc/matlab/R2016a/setup.sh #add matlab to path
  $ matlab
  ```
- There are three steps for configuring MATLAB’s PCT
  1. Set cluster profile (SingleNodeProfile_GUI)
     
     *Tell MATLAB which PBS resources are available*
  2. Set parallel pool preferences
     
     *Tell MATLAB which cluster profile and how many workers to use for this run*
  3. Start parallel pool and execute program
- When MATLAB ready, set path to your matlab directory
1. Set cluster profile (GUI)
Set “Description” (optional)
Set “NumWorkers”
#workers = #cores (procs)
Set “JobStorageLocation”
where results will go
Close window

2. Set parallel pool preferences (GUI)

- Access Preferences panel from top tab or lower-left button
- Set “Default Cluster”
  select a profile to use
- Set “Preferred number of workers in a parallel pool”
  #workers = #cores – 1 (i.e., leave one core for MATLAB)
- Close window
3. Start parallel pool (GUI)

- Select “Start parallel pool”
  - Parallel bars will blink until ready
  - Hover over for run status
- In “Current Folder” window
  - Select script (estimatePI.m)
  - Opens in Editor window
- Run program (green triangle)
  - Type $ top to see running processes
- Exit MATLAB

Outline

- Introduction
  - Prerequisites, setup workspace and launch MATLAB
  - Introduction to parallel MATLAB
- Run MATLAB on single node w/ Parallel Computing Toolbox (PCT)
  - Using graphical user interface (GUI)
  - Using command line interface (CLI)
  - Using CLI with batch parallel job execution
- Run MATLAB on multiple nodes w/ Distributed Computing Server (DCS)
  - Using command line interface (CLI)
- Beyond parfor
  - labIndex, Composite, and Broadcast&Receive
  - Practice using PBS
CLI: Run on single node with PCT

- Next, we’ll take the same three steps using command line
  - Remember, we are still running interactively on a compute node
- Start MATLAB without GUI

```
$ source /usr/usc/matlab/R2016a/setup.sh
$ matlab --nodesktop --nodisplay
```

MATLAB is selecting SOFTWARE OPENGL rendering.

```
< M A T L A B (R) >
Copyright 1984-2015 The MathWorks, Inc.
R2015a (8.5.0.197613) 64-bit (glnxa64)
February 12, 2015
```

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

```
   Academic License
```

1. Set cluster profile (CLI)

- Create cluster profile for single-node parallel job
  - Tells PCT which PBS resources are available

```
>> MPCTprofile = parallel.cluster.Local
>> MPCTprofile.JobStorageLocation = 
    '/home/rcf-proj/{project}/{user}/{matlab_dir}'
>> MPCTprofile.NumWorkers = 8
```
Set cluster profile

- You can save your profile and reuse it

  ```matlab
  >> MPCTprofile.saveAsProfile('SingleNodeProfile')
  • This profile is saved in .matlab, and is re-usable
  • If you have already created this, next time you run, instead of all the former commands, just say:

  ```matlab
  >> MPCTprofile = parcluster('SingleNodeProfile')
  ```

- At this point we can either
  • Run EstimatePi.m script interactively in the foreground or run it in the background as a batch job

2. Set parallel pool and execute program

%Start parpool (specify cluster profile and workers)
>> parpool(MPCTprofile,7)
    Starting parpool ...connected to 7 workers.
%Run program
>> estimatePi
ans =
    Elapsed time = ........
%Check values
>> pi
    pi =
        3.1416
%Always clean up
>> delete(gcp('nocreate'))
Outline

- Introduction
  - Prerequisites, setup workspace and launch MATLAB
  - Introduction to parallel MATLAB
- Run MATLAB on single node w/ Parallel Computing Toolbox (PCT)
  - Using graphical user interface (GUI)
  - Using command line interface (CLI)
  - Using CLI with batch parallel job execution
- Run MATLAB on multiple nodes w/ Distributed Computing Server (DCS)
  - Using command line interface (CLI)
- Beyond parfor
  - labIndex, Composite, and Broadcast&Receive
  - Practice using PBS

MATLAB batch parallel job execution

- You can spawn jobs from within MATLAB
  - A batch job will run in the background
- MATLAB has its own scheduler for batch jobs
  - For example, if you had two independent tasks
- Run a batch job using PCT

```
>> job = batch(MPCTprofile, 'EstimatePi', 'pool', 7)
```
MATLAB batch parallel job execution

- To view job state
  ```
  >> job %view state of job
  ```

- To view the results
  ```
  >> load(job)
  ```

- Type variable names to see results
  ```
  >> pi
  pi = 3.1416
  >> elapsedTime
  elapsedTime = 1.9440
  ```

Outline

- Introduction
  - Prerequisites, setup workspace and launch MATLAB
  - Introduction to parallel MATLAB

- Run MATLAB on single node w/ Parallel Computing Toolbox (PCT)
  - Using graphical user interface (GUI)
  - Using command line interface (CLI)
  - Using CLI with batch parallel job execution

- Run MATLAB on multiple nodes w/Distributed Computing Server (DCS)
  - Using command line interface (CLI)

- Beyond parfor
  - labIndex, Composite, and Broadcast&Receive
  - Practice using PBS
Run on multiple nodes (GUI)  
MATLAB’s Distributed Computing Server (MDCS)

- Request two nodes

```
qsub -l walltime=01:00:00 -l nodes=2:ppn=8 -l software=MPCT+1
-A workshop -l ‘advres=HPCWorkshop.xx’
```

- Open new window

```
$ ssh -X {user_id}:hpc-login3.usc.edu
$ ssh -X hpcxxxx
$ matlab
```

- There are three steps for MATLAB MDCS configuration
  1. Set cluster profile (MultiNodeProfile_GUI)
  
     *Setup resource template: Tell MATLAB how to pass resources to PBS*
  2. Set parallel pool preferences
  
     *Tell MATLAB which cluster profile and how many workers to use for this run*
  3. Start parallel pool and execute program

1. Set cluster profile

- Open Main Menu=>Environment=>Parallel=>Manage Cluster Profiles
- Choose Torque
- Rename to “MultiNodeProfile_GUI”
- Edit
Set resource template

- Set JobStorageLocation
- Set NumWorkers
- Set ResourceTemplate
  
  -l procs=^N^ -software=MDCS+^N^ -m abe -M {user_id}@usc.edu

2. Set parallel pool preferences

- Access Preferences panel from top tab or lower-left button
- Set “Default Cluster”
- Set “Preferred number of workers in pool”
  
  #workers = #cores – 1

- Close window
3. Start parallel pool

- Select “Start parallel pool”
  - Parallel bars will blink until ready
  - Select script (EstimatePi.m) and run

- How to check your multinode process?
  - $ myqueue
    - One job is our stdin job (interactive shell)
    - One job is matlab’s which has requested 15 cores (and received two nodes) and is waiting
    - You can ssh to any one of these nodes and run $ top

- Exit MATLAB

Outline

- Introduction
  - Prerequisites, setup workspace and launch MATLAB
  - Introduction to parallel MATLAB

- Run MATLAB on single node w/ Parallel Computing Toolbox (PCT)
  - Using graphical user interface (GUI)
  - Using command line interface (CLI)
  - Using CLI with batch parallel job execution

- Run MATLAB on multiple nodes w/ Distributed Computing Server (DCS)
  - Using command line interface (CLI)

- Beyond parfor
  - labIndex, Composite, and Broadcast&Receive
  - Practice using PBS
Run on multiple nodes (CLI)

- Next, we’ll take these steps using the command line
- Start MATLAB without GUI

```bash
$ source /usr/usc/matlab/R2016a/setup.sh
$ matlab -nodesktop -nodisplay
```

MATLAB is selecting SOFTWARE OPENGL rendering.

![MATLAB](https://www.mathworks.com/help/pdf_doc/matlab/)

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

```
>> Academic License

Setting MDCS profiles

- Create cluster profile for MATLAB’s DCS
  - Tells MATLAB how to talk to multiple nodes
```

```
>> MDCSprofile = parallel.cluster.Torque
>> MDCSprofile.JobStorageLocation = '/home/rcf-proj/{project}/{user}/{matlab_dir}'
```

- Save profile, and reuse later
```
>> MDCSprofile.saveAsProfile('MultiNodeProfile')
>> MPCTprofile = parcluster('MultiNodeProfile')
```
Setting MDCS profiles

- **Set Resource Template**
  - MATLAB also needs to know
  - How to log into compute nodes (ssh)
  - How to transfer files (scp)

```matlab
>> MDCSprofile.RshCommand='ssh
>> MDCSprofile.RcpCommand='scp'
```

- **Note:** There is no walltime specified
- **N** will be determined by us later

```matlab
>> MDCSprofile.ResourceTemplate=' -l procs="N" -l software=MDCS+"N" -m abe -M {user_id}@usc.edu'
```

- **Save your profile**

```matlab
>> MDCSprofile.saveAsProfile('MultiNodeProfile')
```
Using MDCS profiles

- Using your previously saved profile
  >> MDCSprofile = parcluster('MultiNodeProfile')

- Set maximum walltime
  >> MDCSprofile.SubmitArguments='-- 1 walltime=1:00:00'

- If you have a walltime you use often you can save and restore the profile
  >> MDCSprofile.saveAsProfile('OneHr_MultiNodeProfile')
  >> MPCTprofile = parcluster('OneHr_MultiNodeProfile')

Launching an MDCS job

- How many cores to request?
  - Most nodes in main queue have either 8 or 16 cores per node so multiples of 8 (8, 16, 24, 32...) or 16 (16, 32, 48...) are best
  - Makes it easier for job scheduler to find your resources, job starts sooner

- Launch a batch job
  >> job = batch(MDCSprofile,'EstimatePi', 'parpool', 15)

  - 16 cores, 1 for MATLAB plus 15 workers
  - MATLAB handles qsub details
  - NOTE: You can close your MATLAB session and interactive session at this point if you are saving results to file (but don’t do for tutorial)
  - Check $myqueue
### Multinode job: Checking status

```matlab
>> jobs=findJob(MDCSprofile)
```

```matlab
jobs=

8x1 Job array:

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>State</th>
<th>FinishTime</th>
<th>Username</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>running</td>
<td>Apr 18 11:01:49</td>
<td>csul 1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>finished</td>
<td>Apr 22 15:05:07</td>
<td>csul 7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>finished</td>
<td>May 27 10:04:36</td>
<td>csul 3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>finished</td>
<td>May 27 10:04:27</td>
<td>csul 32</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>finished</td>
<td>May 27 10:35:18</td>
<td>csul 32</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>finished</td>
<td>May 27 10:36:23</td>
<td>csul 32</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>finished</td>
<td>May 27 11:39:00</td>
<td>csul 32</td>
<td>12</td>
</tr>
</tbody>
</table>
```

### Multinode job: Retrieving results

```matlab
>> load(jobs(12))
```

```matlab
>> pi
pi =
    3.1416
```

```matlab
>> elapsedTime
elapsedTime =
    42.3374"
Outline

- Introduction
  - Prerequisites, setup workspace and launch MATLAB
  - Introduction to parallel MATLAB
- Run MATLAB on single node w/ Parallel Computing Toolbox (PCT)
  - Using graphical user interface (GUI)
  - Using command line interface (CLI)
  - Using CLI with batch parallel job execution
- Run MATLAB on multiple nodes w/ Distributed Computing Server (DCS)
  - Using command line interface (CLI)
- Beyond parfor
  - labIndex, Composite, and Broadcast&Receive
  - Practice using PBS

Practice using PBS

- There are practice scripts in /home/rcf-proj/workshop/matlab
  - Copy them to your matlab workshop directory
- Normally you would do the following
  1. Create a PBS script
  2. Submit a batch job from head node (e.g., `qsub submit.pbs`)
- You can try this now
  - Exit compute node (you should be on head node)
  - The scripts `submit.pbs` and `jobScript.m` show how to submit EstimatePi to the queue
Submit EstimatePi.m to queue

```bash
#!/bin/bash
#PBS -l nodes=1:ppn=1
# Launch matlab, workers will run in another job
#PBS -N matlab_launcher
source /usr/usc/matlab/R2016a/setup.sh
cd $PBS_O_WORKDIR
matlab -nodisplay -nosplash -nodesktop -r "jobScript"
```

```matlab
parpool('MultiNodeProfile_GUI',7)
% Increase workers as desired
% Replace with other examples
EstimatePi
% Increase size of problem as desired
pi elapsedTime
delete(gcp('nocreate'))
```

Beyond parfor

- parfor is the most basic way that MATLAB can parallelize your code
- Sometimes you'll need to have more control over what gets run on each worker
- Here are some examples to give you an idea of the capabilities of MATLAB’s Parallel Computing Toolbox and Distributed Computing Server
Beyond parfor

- labIndex
- Composite
- Broadcast & receive
- Look at the scripts you copied from /home/rcf-proj/workshop/matlab


```
% Demo task submission
% This script show you how to create job and task
% and submit them

% load the default cluster profile
C = parcluster;

% You can specify a different profile if you like
% C = parcluster('MultiNodeProfile_CLI')

% create submit arguments if you want to run this
% as a multinode job
% C.SubmitArguments='-l walltime=00:30:00 -E';

% create job
j = c.createjob;
a = rand(4);

% create two tasks
j.createTask(@sin,1,{a});
j.createTask(@cos,1,{a});

% run both tasks at the same time
j.submit;

% wait till job finish before fetching results
j.wait('finished');
pause(20);

% fetch output
result = j.fetchOutputs;
f = [result{:}];

% save data
save('submitTasks.out','f','-ascii');
```
labIndex.m

% Demo of parpool, Composite and labindex
% This script demos parpool and Composite, labindex
% it must be run as a multi-node job
% load the default cluster profile
%c = parcluster;

% You can specify a different profile if you like
c = parcluster('MultNodeProfile_CLI');

% create submit arguments
c.SubmitArguments=-l walltime=00:30:00 -E ;

% request 3 workers
c.parpool(3)

% compute process
a = Composite();
spmd
temp=labindex*ones(10);
a = temp*temp;
end

% save worker results
d=[a(:)];
Save('labindex.out', 'd', '-ascii');

% cleanup
delete(gcp('nocreate'))

broadcastReceive.m

% Demo labindex and mpi communication
% demo labindex and mpi communication, send, receive, broadcast
% This script must be run as a multi-node job

% load the default cluster profile
%c = parcluster;

% You can specify a different profile if you like
c = parcluster('DefaultMultNodeProfile')

% create submit arguments
c.SubmitArguments=-l walltime=00:30:00 -E ;
c.parpool(3)

% Set up a 'single program, multiple data' environment
spmd
switch labindex
% If labindex is 1, 1:5 to screen
% If labindex is 2, receive data from 1 on what to print to screen
% otherwise print 11:15 to screen
  case 1
    A=1:5;
    labSend(A,2,199); % send to worker 2
  case 2
    A = labReceive(1,199); % receive from worker 1
  otherwise
    A = 11:15; % create his own
end
end
A{;}

1/3

2/3
broadcastReceive.m

spmd
% if labindex is 1, print 6:10 to screen
% otherwise receive data from 1 on what to
print to screen
if labindex == 1
A=labBroadcast(1,6:10); % broadcast to all
else
A=labBroadcast(1); %receive broadcast
end
end
A{:}

delete(gcp('nocreate'))

End Notes

- MATLAB documentation
  http://www.mathworks.com/help/

- MATLAB licenses
  - To view availability, type on command line
    /auto/usc/matlab/R2016a/etc/glnxa64/lnutil lmunstat -a -c 27000@hpc-licenses.usc.edu | grep MATLAB
  - To view graphically
    https://hpc-monitoring.usc.edu/dashboards/queues.html