

Matlab on a Supercomputer

Shelley L. Knuth
Research Computing
April 9, 2015

Outline

- Description of Matlab and supercomputing
- Interactive Matlab jobs
- Non-interactive Matlab jobs
- Parallel Computing

Slides available at

http://researchcomputing.github.io/meetup_spring_2015/

What Is Matlab?

- A high performance tool for technical computing
 - Integrates computation, visualization, and programming
 - Analyze data, develop algorithms, create applications
- www.mathworks.com
- Uses many specialized toolboxes to make our lives easier
 - Used by many in the sciences, engineering, math
 - Academia and industry

What Is a Supercomputer?

- One large computer made up of many smaller computers and processors
- Each different computer is called a **node**
- Each node has processors/cores
 - Carry out the instructions of the computer
- With a supercomputer, all these different computers talk to each other through a communications network

Janus Supercomputer

- 1368 nodes, 16,000+ cores
- Resource available to all of CU
- RC environment has login, compile, and compute nodes
- Login nodes
 - Virtual machine
 - Presents a development environment
 - Where you land when you log in
 - Used for:
 - Editing scripts
 - Job submission

Janus Supercomputer

- Compile nodes
 - Used for:
 - Compiling/installing software
 - Compiling large programs
 - Nodes on Janus that have the same setup as computational nodes
 - Hardware and software
- Compute nodes
 - Where jobs are run
 - Jobs are scheduled to run on the compute nodes via a scheduler

Setting up to run Matlab

- Different than running Matlab on your local machine
- Must login, set environment to run

- Login:

```
ssh -X knuths@login.rc.colorado.edu
```

Enter password: 4-digit PIN + OTP number

- Setting environment up to run Matlab:

```
module load matlab
```

How to Run Matlab

- Once you have your environment set up to run Matlab, there are two main ways you can run Matlab
 - Interactively
 - In the background
- We will examine both methods, and their pros and cons

Interactive Matlab

- “Interactive” means working with software where you input commands and the software gives a response
- One common mistake people make is to login, be on a login node, and then type “matlab”
 - Similar to how they would run it on their laptop
 - This is a great example of **WHAT NOT TO DO**
 - Login nodes are a shared resource
 - Running software that requires a lot of resources can impact other’s usability of the node

Batch job submission

- Matlab can also be run interactively through a batch job
 - Achieves the same effect, more efficient
- Batch jobs are scheduled to run on a supercomputer once the required resources are available and after it is your turn in the queue
- RC uses SLURM to submit/schedule jobs
- Common commands:

`sbatch`

`sinteractive`

`salloc`

Interactive Batch Jobs - Matlab

- To submit a batch job to run Matlab interactively:
 - Request CPU time on the compute nodes
 - Wait for this the resources to become available
 - Open Matlab
 - Do your thing!
- Interactive batch jobs are mainly used for
 - Debugging
 - Short job runs

Submission – Interactive Batch Jobs

- To submit an interactive batch job on Janus, do the following:

```
module load matlab (described earlier)
module load slurm
```

Then to request compute time, either use:

```
salloc
sinteractive
```

Salloc vs. Sinteractive

- Both are SLURM commands
- `salloc`
 - Use this if you don't need X11 forwarding
 - If you don't need the GUI
- `sinteractive`
 - Use this if you do need X11 forwarding

```
salloc --qos=janus-debug
```

```
sinteractive --qos=janus-debug
```

Length – Interactive Batch Jobs

- NEVER run for a long time
- Your job will start when resources become available
- This could be at 2 am
- Want job to start asap
- Ask for as few resources as possible
 - Small wall-time
- Good for the janus-debug queue
 - Keeps you to under 1 hour

Interactive Batch Jobs

- Once `salloc` or `sinteractive` runs

```
salloc --qos=janus-debug
```

- And resources become available you are granted a job allocation on a node

```
salloc: Granted job allocation 602576 (output)
```

- Once this comes on the screen your compute allocation is being used

- If you type `hostname` you will be told which node you are on

```
bash-4.1$ hostname
```

```
node0211
```

Interactive Matlab

- Now you can run Matlab

```
matlab
```

- If you use `salloc` you will get Matlab without the GUI
- If you use `sinteractive` you will get Matlab with the GUI
- Let's try it!

- One thing to remember – you are burning compute time
- If finish earlier than time asked for type `exit`

Non-Interactive Batch Jobs

- Submit job that will be executed when resources are available
- Create a text file containing information about the job
- Or submit on the command line
- Submit the job file to a queue

```
sbatch
```

Batch Job example

- **Contents of scripts**

- Matlab_tutorial_general.sh
 - Wrapper script loads the slurm commands
 - Changes to the appropriate directory
 - Calls the matlab .m files
 - Matlab_tutorial_general_code.m

- **Run matlab program as a batch job**

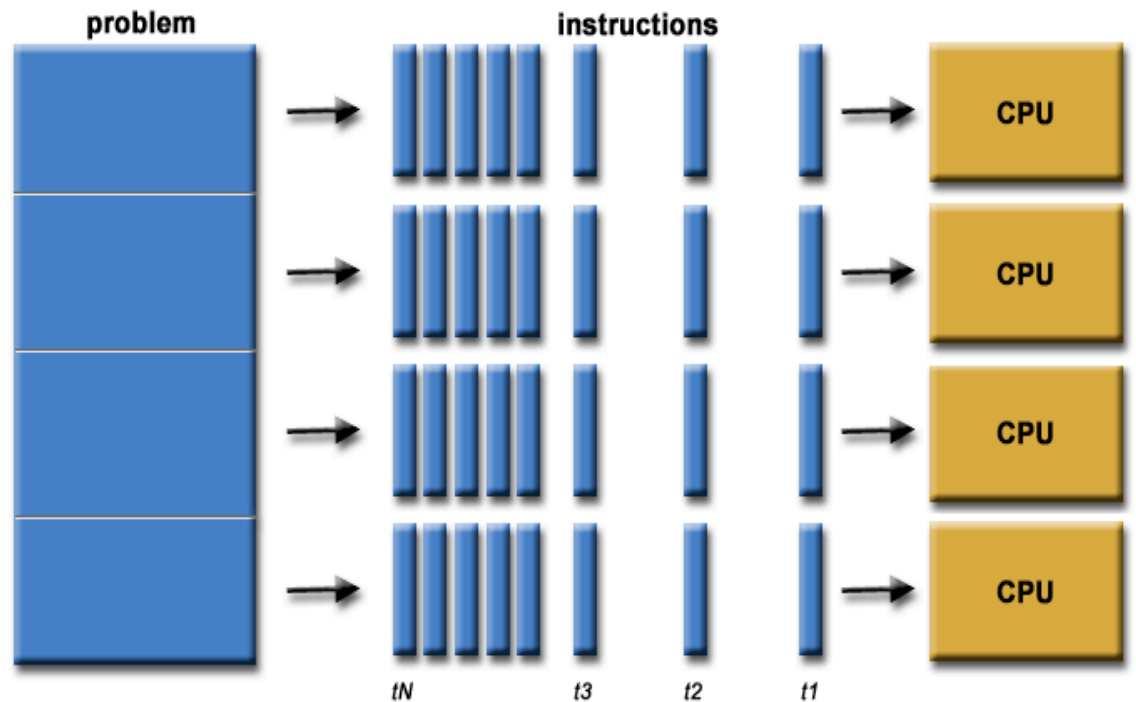
- `sbatch matlab_tutorial_general.sh`

- **Check job status:**

- `squeue -q janus-debug`
- `cat SERIAL.out`

Parallel Processing

- In parallel processing we use several CPUs to solve one problem
- One node with several cores
- Several nodes with many cores
- Embarrassingly parallel



Source: https://computing.llnl.gov/tutorials/parallel_comp/

Parallel Computing Toolbox (PCT)

- Additional toolbox as part of Matlab
- Perform parallel computations on multicore computers, GPUs, and computer clusters
- Allows you to parallelize Matlab code without MPI programming
- Many Matlab functions work in concert with the PCT
- Simple to utilize with just the use of certain commands

Parallel and Not Parallel

Not Parallel:

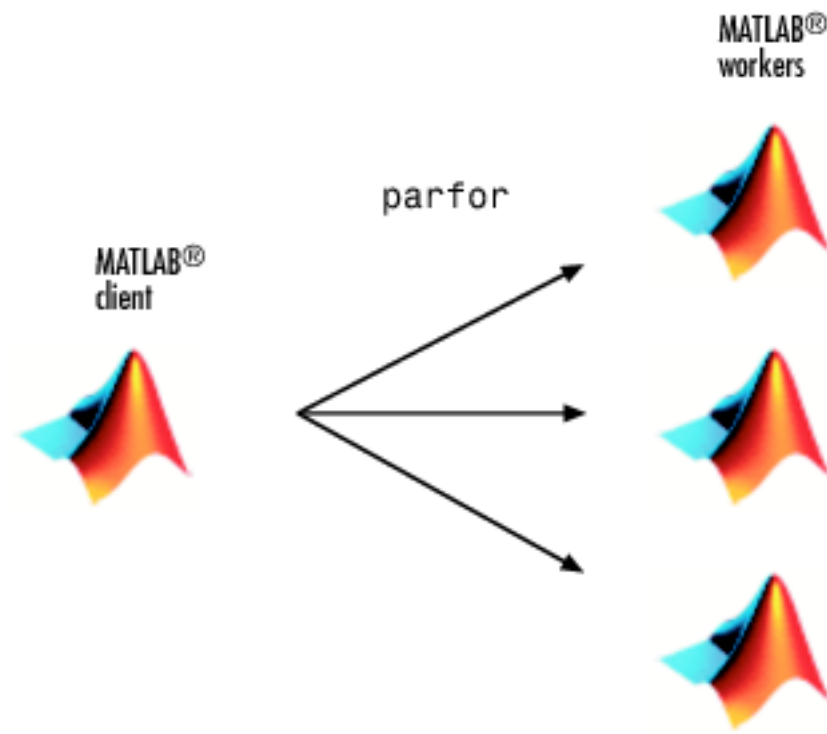
```
for i=1:10  
    x=x(i)+1;  
end
```

Parallel:

```
matlabpool open 3  
    parfor i=1:10  
        x=x(i)+1;  
    end  
matlabpool close
```

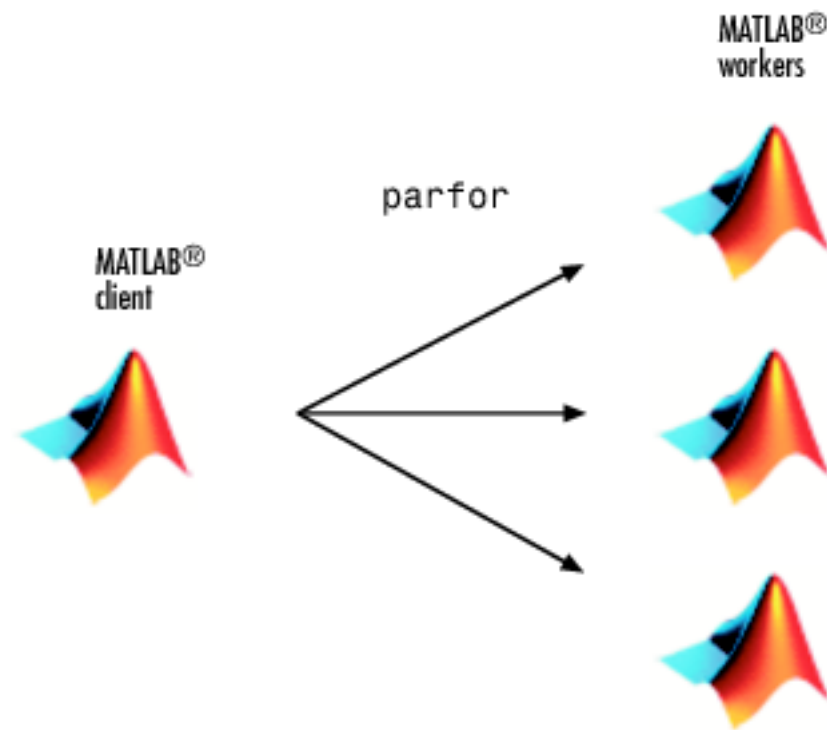
Running Matlab in Parallel

- **Workers:** copies of the original client created to assist in computation



Running Matlab in Parallel

- On Janus at CU can run up to 12 workers on one node
 - Used to be able to only run one Matlab job at a time
 - Now can run as many as you want
 - Can also run as many workers on the high memory nodes



parfor

- Easy to use
- Allows parallelism in terms of loops
- When client reaches a parfor loop iterations of loop are automatically divided up among workers
- Parfor requires results be completely independent
- Cannot determine how loops are divided

Running Matlab in Parallel On Lots of Cores

- Typically see a significant speed up when using parfor vs. when not
 - If code is parallelizable
- However, this might not always be the case
- Might spend more time in overhead
 - If code isn't parallelizable
 - If code isn't that complicated

Spmv Command

- Single process, multiple data
- The spmv command ensures more control
- Can parallelize much more than just loops
- Like a very simplified version of MPI
- More flexibility than parfor
- However, need to know what you're doing

Distributed Computing Toolbox

- PCT allows you to run programs in parallel across many processors
- DCT allows you to run across nodes
 - Allows you to run easily on clusters
 - Supports resource managers
 - Not an option for Janus

Questions?

shelley.knuth@colorado.edu