Parallel Computing with MATLAB

By: Alex Townsend
Who am I?

- **Alex Townsend**
  - Applications Specialist at RCC since December 2019.
  - 6th Year PhD student in the Department of Scientific Computing
  - B.S. in Environmental Science
  - B.S. in Computational Science
  - M.S. in Computational Science
Outline

- A Brief Introduction to MATLAB
- What is Parallel Computing?
- Applications of Parallel Computing
- Parallel Computing Options in MATLAB
- Multi-Core Programming in MATLAB
- GPU Programming in MATLAB
- RCC Resources for Parallel MATLAB
A Brief Introduction to MATLAB
A Brief Introduction to MATLAB

- MATLAB is a closed-source program created and maintained by MathWorks.
- FSU maintains a number of academic licenses for students, faculty and staff who need access to it. Otherwise, it can be quite expensive.
- RCC maintains its own licenses for several versions of MATLAB which are available for RCC users.
MATLAB is ubiquitous in research and industry, particularly in application-focused areas such as engineering.

It is very easy to use and takes a lot of the complexity and learning curve out of computer programming.

It has numerous toolboxes available to suit almost every application imaginable these days.
What is Parallel Computing?
What is Parallel Computing?

- Two major paradigms:
  - Multi-Core Computing
    - *One machine, multiple CPU Cores*
    - *Single Memory space*
  - Distributed Computing
    - *Many computers.*
    - *Separate memory spaces per process.*
What is Parallel Computing?

- **Multi-Core Computing**
  - OpenMP, Threads, Python Multiprocessing
  - MATLAB Parallel Computing Toolbox

- **Distributed Computing**
  - Primarily MPI in its various flavors (OpenMPI, MVAPICH2, etc)
  - MATLAB Parallel Server
What is Parallel Computing?

- **GPU Computing**
  - Can be thought of as a kind of multi-core computing but is generally treated as its own paradigm.

- **Hybrid Computing**
  - It is possible to mix Multi-Core and Distributed computing if you are running your code on a cluster of computers each of which has several processor cores available.
  - Common approaches include MPI/OpenMP hybrid, MPI and Threads, MPI/GPU and more.
What is Parallel Computing?

- What will we focus on in this talk?
  - Multi-Core Programming
  - GPU Programming
Applications of Parallel Computing
Applications of Parallel Computing

• Engineering
  – *Solving the Navier-Stokes equation for complex compressible flows in wind tunnel simulations.*

• Business
  – *Monte Carlo simulations for portfolio optimization and macroeconomic modeling.*

• Science
  – *Molecular Dynamics and Quantum Chemistry simulations for materials and chemical characterization.*
Applications of Parallel Computing

- **Science**

- **Social Sciences**
  - *Deep Learning for large and complex psychology, linguistics and education data sets.*

- **Arts**
  - *Deep Learning and Natural Language Processing for authorship identification of historical writings.*
Parallel Computing Options in MATLAB
Parallel Computing Options

- **Parallel Computing Toolbox**
  - Supports GPU Computing
  - Supports Multi-Core Programming
  - We will use this toolbox today.

- **Parallel Server**
  - Used for Distributed Computing
  - We won’t look at this today.
Multi-Core Programming in MATLAB
Multi-Core Programming

• Let’s look at a simple Serial Code. This code will run in about 27 seconds.

```matlab
% Serial
disp('====== SERIAL ======

n = 500;
A = 100;
tf = 10;
t = 0;
a = zeros(1,n);
tic;
while t < tf
    a_new = a;
    for i = 2:n
        a_new(t) = a(t-1) + max(abs(eig(rand(A))));
    end
    a = a_new;
t = t + 1;
end
toc;

====== SERIAL ======
Elapsed time is 26.963008 seconds.
```

• Code similar to this shows up all the time in solutions to PDE problems and time-dependent simulations.
Multi-Core Programming

- Can we do better?

- Yes. We can use the parallel computing toolbox to parallelize that inner FOR loop over as many cores as our computer has. The HPC node I’m on now has 12.
• How can we go about this?

• MATLAB’s PARFOR Loop.
  – MATLAB’s Parallel Computing Toolbox has a built-in function called `parfor` which takes a FOR loop, breaks it up onto as many cores as possible.
  – So can we just change FOR \(I = 1:N\) to PARFOR \(I = 1:N\)? … No. Not quite.
  – We need to eliminate any loop dependencies so that the ORDER the iteration occurs in is not important.
We need to make sure that when one process updates $T(n,i)$, the next iteration will still be able to access and use the OLD value of $T(n,i-1)$ and $T(n,i+1)$. 

We have a problem here. If EITHER $T(i)$ or $T(j)$ gets Overwritten before the Other process accesses it, We get the wrong answer!
• The easy and simple answer? Keep a copy of the old values and use that. That will decouple the “i” and “j” values from each other and make it so we can update the array in any order we want. Thus, processes “i” and “j” can run at different times and speeds and no problems occur!
Let’s re-code that program to use MATLAB’s PARFOR construct. Same setup as before. This runs in 4 seconds now. That’s a little more than 7 times faster!

```matlab
% Parallel
disp('====== PARALLEL ======
)n = 500;
A = 100;
tf = 10;
t = 0;
a = zeros(1,n);
tic;
while t < tf
    a_new = a;
    parfor i = 2:n
        a_new(i) = a(i-1) + max(abs(eig(rand(A))));
    end
    a = a_new;
t = t + 1;
end
toc;

====== PARALLEL ======
Elapsed time is 3.848416 seconds.
```
As we can see, for complex calculations with large data, parallel computing with multiple cores can really speed up our calculations!

This sort of thing shows up in programs all over the place, not just in MATLAB!
A Word on Distributed Computing in MATLAB
A Word on Distributed Computing

- Many of the same constructs will work for distributed computing as well. MATLAB’s integration of parallel computing across clusters is very good.
- This can be useful for scaling up these kinds of simulations to huge domains (kilometers of material and decades of time!)
- We will not discuss distributed computing today as RCC does not currently have a license for the Parallel Server.
GPU Programming in MATLAB
GPU Programming

- MATLAB’s Parallel Computing Toolbox also supports GPU Computing.
  - `gpuArray constructs allow you to move data to the GPU for computation there!`
  - `We can then call functions on that array and they will be run on the GPU!`
• By running big matrix calculations on the GPU, we can get even bigger speedups!
• This is particularly important for Image Processing and Deep Learning applications which have to use large numbers of computations on matrices that are often very large.
RCC Resources for Parallel Computing in MATLAB
RCC has compute nodes available with between 12 and 48 cores each.

Using our SLURM job scheduler, you can get access to however many cores you need to run on.

You can also access GPU nodes, each of which have 4 GPU cards available (GTX 1080Ti).
OpenOnDemand

- Visit: https://ood.rcc.fsu.edu
- If you have an RCC account, you can use RCC compute nodes as if they were your normal desktop! This is great for MATLAB and other compute-intensive graphical applications!
Thank You!

Questions?