Parallel and Distributed Computing with MATLAB
Practical Application of Parallel Computing

- Why parallel computing?
  - Need faster insight on more complex problems with larger datasets
  - Computing infrastructure is broadly available (multicore desktops, GPUs, clusters)

- Why parallel computing with MATLAB
  - Leverage computational power of more hardware
  - Accelerate workflows with minimal to no code changes to your original code
  - Focus on your engineering and research, not the computation
Steps for Improving Performance

- First get code working
- Speed up code with core MATLAB
- Include compiled languages and additional hardware

Webinar: Optimizing and Accelerating Your MATLAB Code
Programming Parallel Applications

▪ Built-in multithreading
  – Automatically enabled in MATLAB since R2008a
  – Multiple threads in a single MATLAB computation engine

▪ Parallel computing using explicit techniques
  – Multiple computation engines controlled by a single session
  – High-level constructs to let you parallelize MATLAB applications
  – Perform MATLAB computations on GPUs
Parallel Computing
Agenda

- Utilizing multiple cores on a desktop computer
- Scaling up to cluster and cloud resources
- Tackling data-intensive problems on desktops and clusters
- Accelerating applications with NVIDIA GPUs
- Summary and resources
Programming Parallel Applications

- Built in support

    - ..., 'UseParallel', true)
Parallel-enabled Toolboxes

**Image Processing**
Batch Image Processor, Block Processing, GPU-enabled functions

**Statistics and Machine Learning**
Resampling Methods, k-Means clustering, GPU-enabled functions

**Neural Networks**
Deep Learning, Neural Network training and simulation

**Signal Processing and Communications**
GPU-enabled FFT filtering, cross correlation, BER

**Computer Vision**
Parallel-enabled functions in bag-of-words workflow

**Optimization**
Parallel estimation of gradients

**Other Parallel-enabled Toolboxes**
Programming Parallel Applications

- Built in support
  - ..., ‘UseParallel’, true)

- Simple programming constructs
  - parfor, batch
Embarrassingly Parallel: Independent Tasks or Iterations

- No dependencies or communication between tasks

- Examples:
  - Monte Carlo simulations
  - Parameter sweeps
  - Same operation on many files
Mechanics of `parfor` Loops

```matlab
a = zeros(10, 1);
parfor i = 1:10
    a(i) = i;
end
a
```
Tips for Leveraging PARFOR

- Consider creating smaller arrays on each worker versus one large array prior to the parfor loop.

- Take advantage of `parallel.pool.Constant` to establish variables on pool workers prior to the loop.

- Encapsulate blocks as functions when needed.
Programming Parallel Applications

- Built in support
  - ..., ‘UseParallel’, true)

- Simple programming constructs
  - parfor, batch

- Full control of parallelization
  - spmd, parfeval
Utilizing multiple cores on a desktop computer

Scaling up to cluster and cloud resources

Tackling data-intensive problems on desktops and clusters

Accelerating applications with NVIDIA GPUs

Summary and resources
Offloading Computations
Offloading Serial Computations

- `job = batch(...);`
Offloading and Scaling Computations

- \( \text{job} = \text{batch}(\ldots, \text{'Pool'}, n); \)
Migrate to Cluster / Cloud

- Use MATLAB Distributed Computing Server
- Change hardware without changing algorithm
Use MATLAB Distributed Computing Server

1. Prototype code
Use MATLAB Distributed Computing Server

1. Prototype code

2. Get access to an enabled cluster
Use MATLAB Distributed Computing Server

1. Prototype code
2. Get access to an enabled cluster
3. Switch cluster profile to run on cluster resources
Take Advantage of Cluster Hardware

- Offload computation:
  - Free up desktop
  - Access better computers

- Scale speed-up:
  - Use more cores
  - Go from hours to minutes

- Scale memory:
  - Utilize tall arrays and distributed arrays
  - Solve larger problems without re-coding algorithms
Scale your applications beyond the desktop

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Explicit desktop scaling</td>
<td>Single-user, basic scaling to cloud</td>
<td>Scale to EC2 with some customization</td>
<td>Scale to custom cloud</td>
<td>Scale to clusters</td>
</tr>
<tr>
<td>Maximum workers</td>
<td>No limit</td>
<td>16</td>
<td>256</td>
<td>No limit</td>
<td>No limit</td>
</tr>
<tr>
<td>Hardware</td>
<td>Desktop</td>
<td>MathWorks Compute Cloud</td>
<td>Amazon EC2</td>
<td>Amazon EC2, Others</td>
<td>Any</td>
</tr>
<tr>
<td>Availability</td>
<td>Worldwide</td>
<td>United States and Canada</td>
<td>United States, Canada and other select countries in Europe</td>
<td>Worldwide</td>
<td>Worldwide</td>
</tr>
</tbody>
</table>

Learn More: [Parallel Computing on the Cloud](#)
Agenda

- Utilizing multiple cores on a desktop computer
- Scaling up to cluster and cloud resources

- Tackling data-intensive problems on desktops and clusters
- Accelerating applications with NVIDIA GPUs
- Summary and resources
Tall and Distributed Data

- **Tall Data**
  - Columnar data that does not fit in memory of a desktop or cluster

- **Common Actions**
  - Data manipulation, math, statistics
  - Summary visualizations
  - Machine learning

- **Distributed Data**
  - Large matrices using the combined memory of a cluster

- **Common Actions**
  - Matrix Manipulation
  - Linear Algebra and Signal Processing
Tall Arrays

- New data type in MATLAB R2016b

- Applicable when:
  - Data is **columnar** – with **many** rows
  - Overall data size is **too big to fit into memory**
  - Operations are mathematical/statistical in nature

- Statistical and machine learning applications
  - Hundreds of functions supported in MATLAB and Statistics and Machine Learning Toolbox
Execution Environments for Tall Arrays

- Process out-of-memory data on your **Desktop** to explore, analyze, gain insights and to develop analytics.
- Use **Parallel Computing Toolbox** for increased performance.
- Run on **Compute Clusters**, or **Spark** if your data is stored in **HDFS**, for large scale analysis.

Local disk
Shared folders
Databases

Spark+Hadoop
Distributed Arrays

- Distributed Arrays hold data remotely on workers running on a cluster
- Manipulate directly from client MATLAB (desktop)
- 200+ MATLAB functions overloaded for distributed arrays
Agenda

▪ Utilizing multiple cores on a desktop computer
▪ Scaling up to cluster and cloud resources
▪ Tackling data-intensive problems on desktops and clusters
  ▪ Accelerating applications with NVIDIA GPUs
▪ Summary and resources
Graphics Processing Units (GPUs)

- For graphics acceleration and scientific computing
- Many parallel processors
- Dedicated high speed memory
GPU Requirements

- Parallel Computing Toolbox requires NVIDIA GPUs

<table>
<thead>
<tr>
<th>MATLAB Release</th>
<th>Required Compute Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLAB R2014b and newer releases</td>
<td>2.0 or greater</td>
</tr>
<tr>
<td>MATLAB R2014a and earlier releases</td>
<td>1.3 or greater</td>
</tr>
</tbody>
</table>
Programming with GPUs

- Built in toolbox support
- Simple programming constructs
  - `gpuArray`, `gather`
Programming with GPUs

- Built in toolbox support
- Simple programming constructs
  - `gpuArray`, `gather`
- Advanced programming constructs
  - `spmd`, `arrayfun`
- Interface for experts
  - `CUDAKernel`, `mex`
Speed-up using NVIDIA GPUs

- **Ideal Problems**
  - Massively Parallel and/or Vectorized operations
  - Computationally Intensive
  - Algorithm consists of supported functions

- **300+ GPU-enabled MATLAB functions**

- **Additional GPU-enabled Toolboxes**
  - Algorithm consists of supported functions
  - Neural Networks
  - Image Processing
  - Communications
  - Signal Processing

..... Learn more
Agenda

- Utilizing multiple cores on a desktop computer
- Scaling up to cluster and cloud resources
- Tackling data-intensive problems on desktops and clusters
- Accelerating applications with NVIDIA GPUs

- Summary and resources
Summary

- Easily develop parallel MATLAB applications without being a parallel programming expert

- Speed up the execution of your MATLAB applications using additional hardware

- Develop parallel applications on your desktop and easily scale to a cluster when needed
Some Other Valuable Resources

- MATLAB Documentation
  - MATLAB → Advanced Software Development → Performance and Memory
  - Parallel Computing Toolbox

- Parallel and GPU Computing Tutorials

- Parallel Computing on the Cloud with MATLAB