Distributed and Parallel Computing with MATLAB®
Example: Land Classification

- National Land Cover Dataset from U.S. Geological Survey (~30GB)
- “Where are wetlands, forests, etc concentrated?”
- “How does the distribution compare with other datasets?”
Solving Big Technical Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>You could...</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long-running &amp; computationally intensive</td>
<td>Wait</td>
<td>Run similar tasks on independent processors in parallel</td>
</tr>
<tr>
<td>Large data set</td>
<td>Reduce size of problem</td>
<td>Load data onto multiple machines that work together in parallel</td>
</tr>
</tbody>
</table>
From Single Processor to Grids

- Single processor
- Multi-core
- Multi-processor
- Clusters
- Grids
Distributed Computing with MATLAB

Solves two types of problems:

1. Problems that take too much time
2. Problems that involve too much data
Parallel and Distributed Computing with MATLAB
Agenda

- Task-parallel applications (too much time)
  - Data-parallel applications (too much data)
  - Operational Requirements
Multiple independent problems
Agenda

- Task-parallel applications (too much time)
- Data-parallel applications (too much data)
- Operational Requirements
Large Data Sets (Data Parallel)
function dsMap = main(category)

numSections = 4;

for i = 1 : numSections
    out{i} = downsample(i, numSections, category);
end

postProcessing(out)
function dsMap = main(category)

jm = findResource('scheduler', 'configuration', 'jobmanager');

job = createJob(jm, ...
    'FileDependencies', {'downsample.m'}, ...,
    'PathDependencies', {'D:\DistCompDemos\WWLCI'}});

numSections = 4;

for i = 1 : numSections
    createTask(job, @downsample, 1, {i, numSections, category});
end

postProcessing(out)
Parallel for loops

\begin{verbatim}
parfor (i = 1 : n)
    % do something with i
end
\end{verbatim}

- Run loops on a pool of MATLAB resources
- Iterations must be order-independent
- M-Lint analysis helps to identify if existing for loops can be changed to `parfor`
Transposing a Distributed Matrix

Using FORTRAN and MPI

Using MATLAB and MPI

Using Distributed Arrays

\[ \mathbf{P} \gg \mathbf{E} = \mathbf{D}' \]
Agenda

- Task-parallel applications (too much time)
- Data-parallel applications (too much data)

Operational Requirements
Run four *local* workers with a DCT license

- Experiment with explicit parallelism on multi-core machines
- Develop parallel applications on local computer
- Take full advantage of desktop power
- No separate computing cluster required
Supported on all MATLAB platforms
Flexible Hardware

- Homogeneous cluster
- Heterogeneous cluster
Third Party Schedulers

Client Machine

MATLAB
SIMULINK
Toolboxes
Blocksets

Distributed Computing Toolbox

Computer cluster

MATLAB Distributed Computing Engine

Third-Party Scheduler
Dynamic Licensing
Dynamic Licensing
Dynamic Licensing

[Diagram of computer cluster with MATLAB and Simulink icons and a scheduler connecting to multiple workers]
Dynamic Licensing

Computer cluster

Scheduler

Worker

Worker

Worker

Worker
Research Engineers Advance Design of the International Linear Collider with MathWorks Tools

The Challenge
To design a control system for ensuring the precise alignment of particle beams in the International Linear Collider

The Solution
Use MATLAB, Simulink, the Distributed Computing Toolbox, and the Instrument Control Toolbox to design, model, and simulate the accelerator and alignment control system

The Results
- Simulation time reduced by an order of magnitude
- Development integrated
- Existing work leveraged

"With the Distributed Computing Toolbox, we saw a linear improvement in speed. MathWorks tools have enabled us to accomplish work that was once impossible."

Dr. Glen White, University of London
The Challenge
To design a control system for ensuring the precise alignment of particle beams in the International Linear Collider

The Solution
Use MATLAB, Simulink, the Distributed Computing Toolbox, and the Instrument Control Toolbox to design, model, and simulate the accelerator and alignment control system

The Results
- Simulation time reduced by an order of magnitude
- Development integrated
- Existing work leveraged

“Using the Distributed Computing Toolbox, we simply deployed our simulation on a large group cluster. We saw a linear improvement in speed, and we could run 100 simulations at once. MathWorks tools have enabled us to accomplish work that was once impossible.”

Dr. Glen White,
Queen Mary, University of London
University of Geneva Develops Advanced Portfolio Optimization Techniques with MathWorks Tools

The Challenge
To develop and implement a general-purpose, data-driven heuristic for portfolio optimization

The Solution
Use MATLAB® and Distributed Computing Toolbox to develop algorithms, visualize results, and rapidly compute solutions

The Results
- Solutions found in minutes
- Efficiency improved through visualization
- Students gain practical experience

“For years I have said that people who use Excel or C to do advanced financial analysis are wasting their time. For any kind of numerical computation, I can’t see using anything other than MATLAB.”

Professor Manfred Gilli, University of Geneva

Objective function for value-at-risk minimization for a portfolio of three assets.
EIM Group Develops Quantitative Tools for Hedge Fund Portfolio Management

The Challenge
To develop quantitative tools for hedge fund portfolio optimization and analysis

The Solution
Use MathWorks tools to develop distributed algorithms and models, accelerate simulation, and streamline user interface development and deployment

The Results
- Development time cut in half
- Simulation time reduced by 80%
- Development and deployment streamlined

“With MathWorks tools, we provide answers to complex portfolio management questions rapidly. Responding quickly to our clients with quantitative analysis is a competitive advantage for EIM.”

Dr. Stéphane Daul, EIM Group
Massachusetts Institute of Technology Integrates Cancer Research in the Lab and Classroom with MathWorks Tools

The Challenge
To improve diagnostic techniques for cancer by identifying proteins and analyzing their interactions

The Solution
Use MathWorks tools to enable students and researchers to analyze mass spectrometry data, model complex protein interactions, and visualize results

The Results
- Education integrated with research
- Computation time shortened by an order of magnitude
- Research grant won

"Using a distributed approach with MATLAB® code, we ran our analysis on a computer cluster and reduced computation time by an order of magnitude—from about a week to much less than a day."

-- Dr. Gil Alterovitz, MIT and Harvard University

3-D visualization of the human massome of protein interactions.