Parallel Computing with MATLAB

Jamie Winter, The MathWorks Account Manager
Sarah Wait Zaranek Ph.D, Senior Application Engineer
Solving Big Technical Problems

**Challenges**

- Long running
- Computationally intensive
- Large data set

**You could...**

- Wait
- Reduce size of problem

**Solutions**

- Run similar *tasks* on independent processors in *parallel*
- Load *data* onto multiple machines that work together in *parallel*
Parallel Computing with MATLAB

MATLAB
SIMULINK
TOOLBOXES
BLOCKSETS

Worker
Worker
Worker
Worker
Agenda

- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets
- Schedule your programs to run
- Tips on scheduling parallel code
Parallel Computing with MATLAB

No code changes
- Implicit Multithreaded MATLAB
- Toolbox Support:
  - Optimization Toolbox™
  - Genetic Algorithm and Direct Search™
  - Bioinformatics Toolbox™
  - Model Calibration Toolbox™
  - SystemTest™
  - Simulink Response Optimization™

Trivial changes
- Task Parallel
  - parfor
  - job and tasks

Data Parallel
- spmd
- codistributed
- MATLAB and MPI

Extensive changes
Agenda

- Develop parallel code interactively
  - Task parallel applications for faster processing
    - Data parallel applications for handling large data sets
- Schedule your programs to run
- Tips on scheduling parallel code
Distributing Tasks (Task Parallel)
Demo: Monte Carlo Simulation of Coin Tossing

10 Simulations of Flipping 20 Coins at a Time

Number of Heads Out of 20
Parallel `for`-Loops

```
parfor i = 1 : n
    % do something with i
end
```

- Mix task-parallel and serial code in the same function
- Run loops on a pool of MATLAB resources
- Iterations must be order-independent
- M-Lint analysis helps in converting existing `for`-loops into `parfor`-loops
Agenda

- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets
- Schedule your programs to run
- Tips on scheduling parallel code
Large Data Sets (Data Parallel)
\textbf{spmd}

\textbf{spmd} : single program, multiple data

- Primarily used for data parallel applications
  - distributed arrays
  - mpi functionality
- Runs using a matlabpool
- Data stays on workers between blocks
- Allows users to interleave serial and data parallel MATLAB code

\begin{verbatim}
x = 1
spmd
  y = x+1
end
y
\end{verbatim}
Distributed Arrays and Parallel Algorithms

- **Distributed arrays**
  - Store segments of data across participating workers
  - Create from any built-in class in MATLAB
    - Examples: doubles, sparse, logicals, cell arrays, and arrays of structs

- **Parallel algorithms for distributed arrays**
  - Matrix manipulation operations
    - Examples: indexing, data type conversion, and transpose
  - Parallel linear algebra functions, such as `svd` and `lu`
  - Data distribution
    - Automatic, specify your own, or change at any time
### Enhanced MATLAB Functions That Operate on Distributed Arrays

<table>
<thead>
<tr>
<th>Type of Function</th>
<th>Function Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data functions</td>
<td><code>cumprod, cumsum, fft, max, min, prod, sum</code></td>
</tr>
<tr>
<td>Data type functions</td>
<td><code>cast, cell2mat, cell2struct, celldisp, cellfun, char, double, fieldnames, int16, int32, int64, int8, logical, num2cell, rmfield, single, struct2cell, swapbytes, typecast, uint16, uint32, uint64, uint8</code></td>
</tr>
<tr>
<td>Elementary and trigonometric functions</td>
<td><code>abs, acos, acosd, acosh, acot, acotd, acoth, acsc, acscd, acsch, angle, asec, asecd, asech, asin, asind, asinh, atan, atan2, atand, atan2d, ceil, complex, conj, cos, cosd, cosh, cot, cotd, coth, csc, cscd, csch, exp, expm1, fix, floor, hypot, imag, isreal, log, log10, log1p, log2, mod, nextpow2, nthroot, pow2, real, reallog, realpow, realsqrt, rem, round, sec, secd, sech, sign, sin, sind, sinh, sqrt, tan, tand, tanh</code></td>
</tr>
<tr>
<td>Elementary matrices</td>
<td><code>cat, diag, eps, find, isempty, isequal, isequalwithequalnans, isfinite, isinf, isnan, length, ndims, size, tril, triu</code></td>
</tr>
<tr>
<td>Matrix functions</td>
<td><code>chol, eig, lu, norm, normest, svd</code></td>
</tr>
<tr>
<td>Array operations</td>
<td><code>all, and, any, bitand, bitor, bitxor, ctranspose, end, eg, ge, gt, horzcat, ldivide, le, lt, minus, mldivide, mrdivide, mtimes, ne, not, or, plus, power, rdivide, subsasgn, subsindex, subsref, times, transpose, uminus, uplus, vertcat, xor</code></td>
</tr>
<tr>
<td>Sparse matrix functions</td>
<td><code>full, issparse, nnz, nonzeros, nzmax, sparse, spfun, spones</code></td>
</tr>
<tr>
<td>Special functions</td>
<td><code>dot</code></td>
</tr>
</tbody>
</table>
MPI-Based Functions in Parallel Computing Toolbox™

Use when a high degree of control over parallel algorithm is required

- High-level abstractions of MPI functions
  - labSendReceive, labBroadcast, and others
  - Send, receive, and broadcast any data type in MATLAB

- Automatic bookkeeping
  - Setup: communication, ranks, etc.
  - Error detection: deadlocks and miscommunications

- Pluggable
  - Use any MPI implementation that is binary-compatible with MPICH2
Using the matlabpool

- **SPMD … END**
  - A MATLAB block
  - Worker communication
  - Mostly Data Parallel

- **PARFOR … END**
  - A MATLAB block
  - Independent iterations (Task Parallel)
  - MATLAB turns user code into parallel code.
Agenda

- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets

Schedule your programs to run

- Tips on scheduling parallel code
Third-Party Scheduler (Condor)

Computer Cluster
MATLAB Distributed Computing Server

Third-Party Scheduler

Client Machine

MATLAB
SIMULINK
Toolboxes
Blocksets

Parallel Computing Toolbox
Demo: Scheduled Monte Carlo Coin

>> batch
Distributed Applications

Client Machine

Parallel Computing Toolbox

Compute Cluster

MATLAB® Distributed Computing Server™

Task

Result

Job

Result

Task

Result

Task

Result

Task

Result

Task

Result

Worker

Worker

Worker

Worker
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\WWLCD'});

numSections = 4;

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

postProcessing(out)
Demo: Scheduled Monte Carlo Coin

```matlab
>> createJob
>> createTask
```
Dependencies

- **job – FileDependencies**
  - Files are copied from client to each worker machine
  - Zip compressed
  - Uncompressed and added to the MATLAB path
  - Convenient for .m files, but can be slow for large data files

- **job – PathDependencies**
  - Shared directories are added to the MATLAB path
  - Mixing of Windows® and UNIX® paths allowed
  - Reduces the amount of data transfer from client to cluster
Configurations

- Save environment-specific parameters for your cluster

- Benefits
  - Enter cluster information only once
  - Modify configurations without changing MATLAB code
  - Apply multiple configurations when running within same session
Agenda

- Develop parallel code interactively
  - Task parallel applications for faster processing
  - Data parallel applications for handling large data sets
- Schedule your programs to run

Tips on scheduling parallel code
Factors to Consider for Scheduling

- Share code and data with workers efficiently using `FileDependencies` or `PathDependencies`.

- There is always an overhead to distribution
  - Don’t make a task too small
  - Combine small repetitive function calls into one larger one
  - Vectorize task creation

- Minimize IO
  - Use JobData if jobs share data
  - Use Workspace for batch

- Use Diary or `CaptureCommandWindowOutput`
Development and Debugging Process

Run serial code normally on local machine

Task or Data Parallel?

Task Parallel

_parfor_ / Jobs and Tasks

Run on local workers

Data Parallel

_spmd_, _pmode_ distributed Arrays

Run on N cluster nodes