Distributed Computing with MATLAB

Mandar Gujrathi
Applications Engineer,
MathWorks.
Mandar.Gujrathi@mathworks.com.au
Agenda

- Methods to improve productivity
- Why Distributed Computing
- Methods to distribute the computations
- Using remote resources distribute computations
Improving Productivity

<table>
<thead>
<tr>
<th>Problem</th>
<th>Single Session</th>
<th>Multiple Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Optimize Code</td>
<td>Distribute Work</td>
</tr>
</tbody>
</table>

Large Data Set → Process Subsets of Data → Distribute Data
How can we speed up?

Distributing Jobs.
Computing Jobs.
Why?
Why Distributed Computing?

- Reduce computation time by
  - Using more cores
  - Accessing Graphical Processing Units
- Solve large data problems by distributing data to available hardware
- Offload applications to a cluster
Some terminologies

- **Job**?
  - *Large Operation to be performed in MATLAB session*

- **Task**?
  - *A job is divided into several tasks. A task could be independent or dependent.*

- **Client**?
  - *Your computer, which has MATLAB, PCT & toolboxes.*

- **Worker**?
  - *Additional Instance of MATLAB.*

  **Worker could be on your machine* or on remote machine**

- **Cluster**?
  - *Machines configured (where MDCS is installed) to run Jobs*

*To start multiple workers on your machine (LOCAL) you need PCT only.*
What are the Components?

- There are two separately licensed components of MATLAB Distributing Computing:
  - **Parallel Computing Toolbox (PCT):**
    - Unlimited local workers (R2014a and higher)
    - Required for MDCS – sends jobs to the scheduler
  - **MATLAB Distributed Computing Server (MDCS):**
    - Allows you to utilize a large number of machines or one machine with a large number of cores
    - Allows you to send jobs to other machines
Distributed Computing - PCT

- As of MATLAB Parallel Computing allows “local scheduling” of jobs, i.e., making use of multi-core functionalities of the hardware.
  - Speed up jobs by dividing them among local cores
  - Each additional task spawns an extra worker
  - PCT doesn’t require any configuration of schedulers
Ways to speed up (PCT) : parfor

- Loop Variable MUST be an INTEGER
- The loop variable MUST be consecutive & increasing
- The loop iterations MUST be independent
- Nested parfor WILL NOT work.
- You could call a function using parfor within a parfor
- All for loops CANNOT be converted to parfor
Distributed Computing – MDGS

- If the local scheduler is not enough, a user can submit to a computing cluster that is running MDGS.
- Jobs that are submitted to the scheduler are distributed among workers.
- Licensing and physical resources define how many workers a cluster may have.
- PCT must be installed on the client’s MATLAB.
- MDGS will be generally installed on a cluster of powerful machines.
MATLAB Distributed Computing Workflow

1. Prototype code
MATLAB Distributed Computing Workflow

1. Prototype code
2. Get access to an enabled cluster
MATLAB Distributed Computing Server Workflow

1. Prototype code
2. Get access to an enabled cluster
3. Switch cluster profile and run on cluster resources
How to distribute jobs

- There are two methods of distributing jobs:
  - Using the multi core functionalities of your machine (PCT)
  - Get access to the resources of a remote machine/s. (PCT+MDCS)
Some more terminologies

- **Distributed job?**
  - *Multiple tasks running independently on multiple workers with no information passed among them.*

- **Distributed Array?**
  - *Splitting large dataset among several MATLAB workers.*

- **Communicating Job?**
  - *Task running concurrently on multiple workers that may communicate with each other.*
Agenda

- Methods of improving performance
- Why Distributed Computing
- Methods to distribute the computations locally
- Using remote resources distribute computations
Methods of Distributed Computing

- Using Parallel For Loops (parfor)
- Create an Independent Job
- Create a Communicating Job
- Create a Batch Job
- Distribute the data between sessions.
## Independent and Communicating Jobs Workflow

<table>
<thead>
<tr>
<th>Independent Job</th>
<th>Communicating Job</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Workers perform tasks without communication.</strong> Ability to choose workers.</td>
<td>Communication between the workers happen while running the job</td>
</tr>
<tr>
<td>&gt;&gt;createJob</td>
<td>&gt;&gt;createCommunicatingJob</td>
</tr>
<tr>
<td><strong>User defines independent tasks within the job.</strong></td>
<td>User defines communicating task within the job (SPMD)</td>
</tr>
<tr>
<td>&gt;&gt;createTask</td>
<td>&gt;&gt;createTask</td>
</tr>
<tr>
<td><strong>Tasks are distributed to workers as their availability</strong></td>
<td>Tasks run simultaneously. Job may be delayed as workers become available.</td>
</tr>
<tr>
<td>&gt;&gt;submit</td>
<td>&gt;&gt;submit</td>
</tr>
<tr>
<td><strong>Get the results from the workers.</strong></td>
<td>Get results from workers</td>
</tr>
<tr>
<td>&gt;&gt;load (Scripts)</td>
<td>&gt;&gt;load (Scripts)</td>
</tr>
<tr>
<td>&gt;&gt;fetchOutputs (Functions)</td>
<td>&gt;&gt;fetchOutputs (Functions)</td>
</tr>
<tr>
<td><strong>Example:</strong> birthdayjobs.m</td>
<td><strong>Worker communications:</strong></td>
</tr>
<tr>
<td></td>
<td>&gt;&gt;labSend</td>
</tr>
<tr>
<td></td>
<td>&gt;&gt;labReceive</td>
</tr>
<tr>
<td></td>
<td>&gt;&gt;labSendReceive</td>
</tr>
</tbody>
</table>
Batch Job  `>>batch birthdayBatchWithFile.m`

- Useful to offload work from MATLAB session to another session (cluster)
- Job runs in the background without blocking the input.
- Allows to submit job, close current session and return later
- Allows to attach files, provide additional paths, etc.
Overcoming Memory Limitations

Distribute Memory

```matlab
A = distributed.rand(5e5,2);
```

MATLAB functions on distributed and codistributed arrays (doc)
Example: Overcoming Memory Limitations

\[
\frac{\partial U}{\partial t} = c^2 \nabla^2 U
\]

\[
u_{t+1}(i,j) = u_t(i,j) + \lambda (u_t(i-1,j) + u_t(i+1,j) - 4u_t(i,j) + u_t(i,j+1) + u_t(i,j-1))
\]
Agenda

- Methods of improving performance
- Why Distributed Computing
- Methods to distribute the computations locally
- Using remote resources distribute computations
Scaling your parallel applications to the cloud

**Benefit:**
Easy replication and anywhere access.

MATLAB + Parallel Computing Toolbox

MDCS on Amazon EC2

MDCS on Private Cloud
MATLAB on Amazon EC2

- Requires Amazon Web Services account

- Supports both MATLAB and MDCS

- MathWorks’ Cloud Center web application allows you to customize MDCS cluster size, machine type, and storage options.
MDCS on Private Cloud

- Private cloud, other cloud services, on premise and ad-hoc clusters, and grids.
- MDCS can be integrated with a variety of cluster schedulers.
- Preinstalled options:
  - Amazon EC2
  - Penguin Computing
- MHLM provides added convenience with cloud-based licensing.
- Limited support for custom cloud. Recommend using pre-installed options.
MathWorks Hosting Providers in Australia

- Licensed for BYOL
- Pre-installed with MATLAB and MDCS

Australia:
- NCI (Active)
- QCIF (Active)
Summary

- For computationally expensive operations,
  - Parallel Computing Toolbox (PCT)
  - MATLAB Distributed Computing Server (MDCS)

- PCT allows distributing the computations locally on your machine

- MDCS along with PCT allows distributing the computations to a remote machine or a cloud.

- Resources: [Distributed Computing](#), [Training](#)