Scaling up MATLAB Analytics

Marta Wilczkowiak, PhD
Senior Applications Engineer
MathWorks
Agenda

- Giving access to your analytics to more users
- Handling larger problems
When Scalability is a Challenge

- You have great functionality, yet people do not have access to it
- You have purchased more computers, yet you cannot execute much more functions
- You hired more people, yet software gets developed at the same scale
Examples

- Increase number of Monte Carlo simulations
- Share portfolio, risk or trading apps with a few traders
- Deploy pricing functions into an enterprise pricing platform serving front, middle and back offices
- Update prices nightly, or intraday, in an enterprise database
- Add strategies to a trading platform

More computational power $\rightarrow f(x) \rightarrow$ More access
Multi-tier Architecture

Presentation (Client)

Logic (Algorithm)

Data
What We Are Looking at in This Session

More access

More computational power
Enabling Access to MATLAB Algorithms (Deployment)
What Is Application Deployment?

- Sharing MATLAB apps or native files with other MATLAB users
- Sharing MATLAB programs with people who do not have MATLAB
Tools
Deployment options

- Live MATLAB
  - Apps

- MATLAB Compiler
  - Standalone applications
  - Software modules

- MATLAB Production Server
Deployment options

- Live MATLAB
  - Apps

- MATLAB Compiler
  - Standalone applications
  - Software modules

- MATLAB Production Server
MATLAB Apps

MATLAB Apps: Custom interactive applications running in MATLAB

- Easy to package, install and find
- Support full MATLAB language
- Require MATLAB to run
Deployment options

- Live MATLAB
  - Apps

- MATLAB Compiler
  - Standalone applications
  - Software modules

- MATLAB Production Server
What is MATLAB Compiler?

Software that let’s you to:

- Package your MATLAB code so that it does not require MATLAB license
- Encrypt your intellectual property
MATLAB Compiler Runtime

MATLAB Source Code

MATLAB Compiler

MATLAB Encrypted Code

Execute

Single threaded access

MATLAB Compiler Runtime (MCR)
Typical deployment process for desktop applications

1. Create MATLAB code
2. Package the code
3. Share the installer
4. Install the application and MCR
Desktop Applications

Potential scalability challenges

- Distributing software updates to multiple users
- Installing application in uncontrolled IT setup (admin rights, different OS, etc.)
- Slow data transfer from a remote location
- Hardware resources on the client machines inadequate to application demands
- Distributing MCR updates to multiple users
Scaling Up: Demand for Server Analytics

Client
- Less need for processing power and memory

Server
- Simplified code management
- Optimal access to data and hardware resources
- Security

Challenge:
Giving access to MATLAB to multiple requesters (single threaded MCR)
Deployment options

- Live MATLAB
  - Apps

- MATLAB Compiler
  - Standalone applications
  - Software modules

- MATLAB Production Server
What is MATLAB Production Server?

Enterprise class framework for running packaged MATLAB programs

Component repository

Stores deployed components

Runtime libraries

Execute algorithms

Server software

Manages algorithms & worker pool

Lightweight client library

Requests execution of algorithms

Request Broker & Program Manager

MPS Archive
Integrating with IT Systems

- Web Applications
- Desktop Applications
- Web Server
- Application Server
- MATLAB Production Server
  - Portfolio Optimization
  - Pricing
  - Risk Analytics
- Database Server
- Excel®
MATLAB Production Server
Scalable and reliable way for giving access to MATLAB algorithms

Centralised library of algorithms

Simultaneous access to multiple users

Simple management
- Let MPS manage the worker pool, algorithms and MCRs
- Update the algorithms automatically
- Separate the development and execution of application and algorithmic code

Execution efficiency
- Access immediately
- Execute with optimised hardware and access to the data

Ease of access
- Use with different clients: web, database & application servers
- Integrate easily with application software using the lightweight client library
- Call using native data types
Deploying Enterprise Applications
Deploying Your Code To a Server

Considerations
Server Deployment: Main Considerations

- Will Clients Call the Algorithm Simultaneously?
- What Interfaces the Client Can Use?
- What Should the Input / Output Be?
- What are Other Dependencies of the Algorithm?
- How to Keep the Algorithm Available?
- What is the Response Time?
Will Clients Call the Algorithm Simultaneously?

Desktop application | Web application | Database/web/application server

Providing access to multiple MCRs
- Using MPS
- Using 3rd party or proprietary solutions (watch MCR start up and development time)
What Interfaces the Client Can Use

What “language” does the client speak?
Bottom line: MATLAB components can be called from any application that has external interfaces

- Desktop application
- Web application
- Database/web/application server

What “language” does the client speak?
Bottom line: MATLAB components can be called from any application that has external interfaces
What Should Be the Input/Output?

\[ I \rightarrow f(x) \rightarrow O \]
What Should Be the Input/Output?  
Example 1

Deploy pricing functions into an enterprise pricing platform

Client sends simple parameters

MATLAB pulls data from sources

Client sends simple parameters

Client

MATLAB

f(x)
What Should Be the Input/Output? Example 2

Add algorithms to a trading platform

Client pushes all data to MATLAB

Client

f(x)

MSMQ, JMS, IBM MQ

Database

Chart
What Should Be the Input/Output? Example 3

Update prices nightly, or intraday, in an enterprise database

Client pushes all data to MATLAB

Bring algorithm to data
Examples

TERADATA
NETEZZA
HADOOP
What Should Be the Input/Output?

Summary

Main considerations
- What is needed
- Data types
- Is data on the client anyway?
- Need/want to use of existing enterprise I/O functions
- Marshalling time
What are Other Dependencies of the Algorithm?

State/persistent data

```
function out = getNextNumber
    persistent n;
    if isempty(n)
        n = 0;
    end
    n = n + 1;
    out = n;
```

0 → 1 → 2
What are Other Dependencies of the Algorithm?

Removing state/persistent data
- Refactor
- Store state in
  - Database
  - MAT files
  - In-memory storage

Config files
- Make sure they will be present in the production environment

```matlab
function out = getNextNumber
    persistent n;
    if isempty(n)
        n = 0;
    end
    n = n+1;
    out = n;
end
```

```matlab
function out = getNextNumber(n)
    out = n+1;
end
```
How to Keep the Algorithm Available

Imagine

Monitor and maintain the worker(s)
What is the Response Time

1 min

1 min is fast for daily risk report

1 min is slow for interactive pricing

If the algorithm is not fast enough for the client
- Accelerate your code
- Warn the client
- Minimize call overhead
  - Reduce data marshalling
  - MCR start-up time
  - Algorithm closer to data
## Off the Shelf Product Features for Server-Side Deployment

<table>
<thead>
<tr>
<th>Feature</th>
<th>MATLAB Compiler +Builders</th>
<th>MATLAB Production Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code locked</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Logic out of process</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Reduced MCR start up time</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Simultaneous access</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Monitor and maintain the worker(s)</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Creating Production Ready MATLAB Algorithms

Prototype
Algorithm Developer

Production Code
Software Engineer

Internal collaboration
MathWorks Consulting Services
Using More Computational Power (Parallel Computing)
Parallel Computing Tools Address…

Long computations

- Multiple independent iterations

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

- Series of tasks

Task 1  Task 2  Task 3  Task 4

Large data problems
Parallel Computing Toolbox for the Desktop

- Speed up parallel applications
- Take advantage of GPUs
- Prototype code for your cluster
Scale Up to Clusters and Clouds

Desktop Computer

Local

MATLAB Desktop (Client)

Computer Cluster

Cluster

Scheduler
Main Considerations for Scalability

- Data transfer
- Resource contention
- Other overheads
Resource Contention

HT  HT   HT  HT   HT  HT
Core Core Core Core
Cache Memory (L3) Cache Memory (L3)
Core Core Core Core
HT  HT   HT  HT   HT  HT

IO Hub

Disk
Network
Memory
Speedup vs. num. Concurrent Processes

\[ a = \text{bigMatrix} \]

\[ a \times a \]

\[ \text{fft}(a) \]

\[ \text{sum}(a) \]
Speedup vs. num. Concurrent Processes

\[ a = \text{bigMatrix} \]

\[ a^*a \]

\[ \text{fft}(a) \]

\[ \text{sum}(a) \]
What We Have Seen

More access

More computational power

More Data
Scaling Up MATLAB Analytics

- Rapid prototyping of ideas
- Scalable algorithms
  - using more computational power
  - with wide access