MATLAB EXPO 2016
KOREA
4월 28일 (목)
등록 하기 matlabexpo.co.kr
Integrating MATLAB Analytics into Business-Critical Applications
Problem statement

- **Democratization**: Is it possible to make the results and insights from these analytics available to all stakeholders in an organization?

- **Agility**: Is it possible to accelerate the development of data analytics?

- **Production**: Is it possible to build data analytics algorithms in a scalable manner that is suitable for production usage and rigor?
Nothing new…
In 1454 Gutenberg put his press to commercial use…

The printing press was an important step towards the democratization of knowledge […]

Now that more people had access to knowledge both new and old, more people could discuss these works […]

On the other hand, the printing press was criticized for allowing the dissemination of information which may have been incorrect.

Taking MATLAB analytics into production

- Lightweight processes
- Visualization
- Access to data
- Expressive language
- Integration with best-in-class tools
Taking MATLAB analytics into production

- Lightweight processes
- Visualization
- Access to data
- Expressive language
- Integration with best-in-class tools

- Reliability
- Maintainability
- Usability
- Extendibility
- Portability
- Readability
Taking MATLAB analytics into production
Taking MATLAB analytics into production
Example: code preparation

**Task:** Make analytics developed in MATLAB available to Excel users

**Input:**
- Time series from race track
  - Time stamp
  - Latitude
  - Longitude
  - RPM
  - Forward acceleration
  - Lateral acceleration

**Output:** Lap statistics
- Lap number, start and duration
- Statistics per lap: max RPM, max forward acceleration, max lateral acceleration
Example: code preparation

Analyse lap data

Contents

- Load data
- Can we detect periodicity?
- Plot trajectory
- Start line: use data tip, and then export to workspace
- Find points where we cross the line
- Loop through laps to construct lap number, start and duration
- Compute Lap Statistics using FINDGROUPS/SPLITAPPLY

What needs to be deployed?

- Access to initial data: NO
- Exploratory calculations: NO
- Exploratory visualisation: NO
- Interactive parameter tuning: NO
- Computation of lap stats: YES
Example: code preparation

```matlab
for n = 1:N
    LapStart(n,1) = data.FrameTime(laps(n));
    LapDuration(n,1) = data.FrameTime(laps(n+1)-1) - data.FrameTime(laps(n));
    lapno(laps(n):laps(n+1)) = n;
    LapName{n} = ['Lap ' num2str(n)];
end
LapStats = table( LapName, LapStart, LapDuration )
```

Still to address:

- Interleaved data access, processing and visualisation
- What should be the input/output of the deployed functions?
- What are the code dependencies?
- Performance optimisation
Code preparation step 1: decouple code layers

- Presentation (Client)
- Logic (Algorithm)
- Data
Code preparation step 2: fit into production context

Presentation (Client)

Logic (Algorithm)

Data
Code preparation step 2: fit into production context

- **Layers**
  - tools, scale, performance

- **Interfaces**
  - types, scale, performance

- **Dependencies**
  - state, configuration
Taking MATLAB analytics into production

Development → Code preparation → Testing → Production
Testing: is it just a stage?

- Check whether results “look” correct
- Write a script to check if results are within predicted bands

- Write code to check whether results stay the same after code optimization

- Write a formal suite of tests covering your code
- Run tests automatically
- User acceptance

- Operations
Example: testing

- Test results
- Coverage
- Comments
- …
Taking MATLAB analytics into production

- Prototyping
- Code preparation
- Testing
- Production
**MATLAB deployment targets**

**MATLAB Compiler** enables sharing MATLAB programs without integration programming.

**MATLAB Compiler SDK** provides implementation and platform flexibility for software developers.

**MATLAB Production Server** provides the most efficient development path for secure and scalable web and enterprise applications.
The range of application platforms
Production Deployment of MATLAB Programs

- **Algorithm developers**
  - MATLAB to design, test, and refine their algorithms

- **System administrators**
  - manage the deployment of MATLAB analytics within a production environment using MPS

- **Application developers**
  - integrate the lightweight client libraries included with MPS into their enterprise applications
Integration with Enterprise Applications
Develop and Test Framework – for MATLAB Production Server
Reference Architecture

MATLAB analytics with Data Visualization tools
Reference diagram for Plotly
MATLAB analytics with TIBCO Spotfire
Reference diagram for TIBCO Spotfire

Mobile  Web  Desktop

TIBCO Spotfire Web Player

MathWorks. MPSExtension

TIBCO Spotfire Server

MathWorks. MPSExtension

MATLAB Production Server

MATLAB Analytics

HTTP(s)
Reference Architecture

MATLAB analytics with SQL / NoSQL
MATLAB analytics with Microsoft SQL Server

Off-Database
- ODBC/JDBC Driver
  - ODBC
  - JDBC
  - Database Toolbox

On-Database
- "Processes"
- "Transactions"
  - C# Common Language Runtime (CLR)
    - .NET Assembly
      - MATLAB Component Runtime
    - MPS Client Library

MATLAB Production Server

MATLAB Compiler SDK
MATLAB analytics with MongoDB

Web Server
- Web App
- Client Library

Application Server
- Batch Job
- Client Library

Users
- Business Application
- Client Library

MATLAB Production Server
MATLAB Analytics

MongoDB Application Driver

MongoDB
- Router (Mongos)
- Data
  - Shard
  - Shard
  - Shard
- Metadata
  - Config Server
  - Config Server
  - Config Server
MATLAB analytics with Microsoft SQL Server

1. `sp_configure 'clr enabled', 1;
   GO
   RECONFIGURE;
   GO

2. `CREATE ASSEMBLY MATLABClient
   AUTHORIZATION db0
   FROM 'C:\Program Files\MATLAB\MATLAB Production Server
   Local\R2015a\client\dotnet\MathWorks.MATLAB.ProductionServer.Client.dll'
   WITH PERMISSION_SET = UNSAFE;
   GO`

3. `using MathWorks.MATLAB.ProductionServer.Client;
   MHTTPClient client = new MHTTPClient();
   try {
     // Attempt to connect to MPS and call the MATLAB functionality
     // Create the proxy interface for the MPS component
     KSDensityDouble mSq = client.CreateProxy<KSDensityDouble>(new
     Uri("http://localhost:9910/KSDensity"));
     ksdensity = mSq.KSDensityDemo(); // Call the method
     // Marshal and return the results
   } catch (Exception e) {
     Console.WriteLine(e.Message);
   }
`
Integration with production systems: benefits

<table>
<thead>
<tr>
<th>Rapid development and deployment of MATLAB analytics</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analytics expressed in MATLAB with nearly all available toolbox functionality</td>
<td>MATLAB Compiler SDK, MATLAB Production Server</td>
</tr>
<tr>
<td>Robust, scalable analytics available to entire organization</td>
<td>Microsoft Excel, Plotly, Spotfire, SQL server, MongoDB, etc.</td>
</tr>
</tbody>
</table>
Conclusions: MATLAB analytics in production

- Democratization: Yes.
  - Integrate analytics with enterprise systems

- Agility: Yes.
  - Access and explore data from within MATLAB during prototype development
  - Establish process allowing rapid iterations between ideas and production system

- Production Quality: Yes.
  - MATLAB products and services provide a single-stack solution when used with supporting technologies to address production data analytics demands
Taking MATLAB analytics into production

- Agility
- Lightweight processes
- Visualization
- Access to data
- Expressive high-level language
- Integration with best-in-class tools

- Architecture
- Code optimization
- Data scalability
- Class system
- Error handling

- Verification and validation
  - Correctness
  - Performance
- Test-driven development

- Reliability
- Maintainability
- Usability
- Extendibility
- Portability
- Readability