Building a Big Engineering Data Analytics System using MATLAB

성호현 차장
Senior Application Engineer
The MathWorks Korea
Trend: Data Economy

“Information is the oil of the 21st century, and analytics is the combustion engine”

Peter Sondergaard, Gartner Research
Problem Statement

- Can MATLAB® scale up and meet increasingly demanding fleet test data analytics requirements?

- Is it possible to build data analytics algorithms in a flexible, scalable manner and yet satisfy production requirements?
The Fleet Data Analysis System

http://goo.gl/EeE5JE
A quick recap of our connected cars

- OBD2
- Bluetooth
- 4G LTE

LAMR Stack
- RAILS
- Ubuntu

Deployed MATLAB

Hadoop Ecosystem
- MapReduce
- Hadoop
- HDFS

Amazon™ EC2™

Apache, the Apache feather logo, Hadoop are registered trademarks or trademarks of the Apache Software Foundation
Fleet Data Analysis System (Test Bed Summary)

- 8-25 Mb per day
- Non-telemetric data files
- Up to 25 operators
- 660Mb of data and counting
Insights (Engine Fuel Consumption and Efficiency)
Insights (8 mile traffic)

- Traffic Patterns (the case for roundabouts)

- 0.0351 Gal/car at the intersection
- 12 cars a minute on the average
- A saving of 121.3 gallons of gasoline per day if the traffic lights were replaced with a round-about.
- A rough saving of 4.5 million pounds of CO2 per year.
Study of Driving Patterns

“By the way, this is pretty neat data to look at in MATLAB. Even without the GPS, I can pinpoint (1) the moment I nearly clobbered a deer this morning and (2) the merging onto and off of route 9…”
Performance

```matlab
% Fetch the data
% Read the data into MATLAB
trackData = xlsread(dataFile);
% Isolate area of interest
idx = timeVec>1403714100426 & timeVec<1403717061373;
% Gather statistics
topSpeed = max(trackData(idx,7)); in kmph.
% topSpeed = 167.4000
% Compute fuel usage
ccMin = trackData(:,17);
fuelUsedInCC = trapz(ccMin(idx))/(60);
fuelUsedInUSGal = 0.000264172052*fuelUsedInCC;
fuelCostInUSD = fuelUsedInUSGal*3.91;
% fuelCostInUSD = 6.1048
```
Discovering problems in veracity of collected data

- Software modifications to give higher peak ft-lbs of torque
- Increased boost pressure and optimized ignition timing
- Aftermarket software was conditioning the data to ensure proper operation
Operationalization of Analytics
Product Changes (1) - Big Data Capabilities in MATLAB

Memory and Data Access
- 64-bit processors
- Memory Mapped Variables
- Disk Variables
- Databases
- Datastores

Programming Constructs
- Streaming
- Block Processing
- Parallel-for loops
- GPU Arrays
- SPMD and Distributed Arrays
- MapReduce

Platforms
- Desktop (Multicore, GPU)
- Clusters
- Cloud Computing (MDCS on EC2)
- Hadoop
Product Changes (2) – Datastores and MapReduce

**Datastore**
- Incremental data processing feature for reading collections of data containing *tabulartext* or *keyvalue* pairs.

```
Example: 'file1.csv'
Example: './dir/data/file1.csv'
Example: ['C:\dir\data\file1.csv', 'C:\dir\data\file2.csv']
Example: 'C:\dir\data\data.mat'
Example: 'hdfs://myserver:8027/data/file1.txt'
```
- Enables access to relational databases using Database Toolbox™

**MATLAB® MapReduce**
- Allows analysis of out-of-memory data.
- Deployment of mapreduce algorithms to:
  - Serial Mapreduce using local workers
  - Parallel Computing Toolbox™
  - MATLAB® Distributed Computing Server™
  - Hadoop® using the MATLAB Compiler
Out of memory processing using Datastores

Visualize Torque Speed Data from Morning Commute

This script will visualize the torque speed scatter using the datastore.

```matlab
%% Auth/Revision: Arvind Bosegrahara
%% Copyright 2014 The MathWorks Consulting Group
%% 
%% Load our logged data
fleetDS = datastore('tracking.csv');

%% Select columns of interest and chunk the data
fleetDS.SelectedVariableNames = {'Torque_Nm', 'Engine_RPM_rpm', 'FuelFlow_Race_minute_gal_min'};
fleetDS.RowsPerRead = 1000;

%% Preview the data to inspect for correctness
preview(fleetDS);

%% Read the data and prep for plotting
plotData = read(fleetDS);
plotArray = table2array(plotData);
plotArray{1:end,1} = plotArray{1:end,1} * 1000;

%% Visualize the data
scatter(cellfun(@(x)str2num,plotArray(:,2)),cellfun(@(x)str2num,plotArray(:,1)),',');
xlabel('Speed (rpm)');
ylabel('Torque (Nm)');
title('Torque-Speed Distribution');
grid on;
```
function SimpleMapReduceDemo varargin

% Auth/Revision: Arvind Honnasasara
% Copyright 2014 The MathWorks Consulting Group
% $Id$

% Load our logged data
fleetDS = dataset('tracklog-2014-Mar-11-9-12-32.csv');
% fleetDS = dataset('/ec2-54-166-203-200.us-west-2.s3.amazonaws.com/tracklog-2014-Mar-11-9-12-32.csv');

% Select columns of interest and chunk the data
fleets.SelectVariablesNames = {'0_x', '0_y', '0_z'};
fleetDS.RowsPerRead = 1000;

% Run the MapReduce on serial MATLAB
rideDS = mapreduce(fleetDS, @RideMapFun, @RideReduceFun, 'GroupingVariables', 'id', 'NumberOfReducers', 5);
rideTable = readall(rideDS); %$\theta<\iota>\alpha\bogus$end

Analyzing Fleet Test Data Using MATLAB

4:00-5:00 p.m. Seth DeLand, MathWorks

Trial>> mapreduce(parpool)
Starting parallel pool (parpool) using the 'local' profile
Connected to the parallel pool (number of workers: 4).
Trial>> SimpleMapReduceDemo
Parallel mapreduce execution on the local cluster:
Map Stage 0% complete
Map Stage 100% complete
Reduce Stage 0% complete
Reduce Stage 100% complete
rideTable =

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Ride 0 Sensor Mean'</td>
<td>[3.7030]</td>
</tr>
</tbody>
</table>

Trial>>

Mapper Function

function RideMapFun(data, ~, inputKVStore)
% Assemble the ride data and compute ride metric based on 6 sensors
rideData = {data.0_x_data, data.0_y_data, data.0_z_data};
rideLenValue = sum(rms(rideData, 2)); size(rideData, 1);

% Store the key value for the reducer
put(inputKVStore, 'rides', rideLenValue);

end

Reducer Function

function RideReduceFun(inputKVStore, ~, outKVStore)
% Reduce the data
rideLenSum = 0;
% Get next key
key = getnext(inputKVStore);
% Get the ride length
rideLen = key.rideLen;
% Add the ride length to the sum
rideLenSum = rideLenSum + rideLen;
% Put the result
put(outKVStore, key.rideLen, rideLenSum);

end

Comparing BSFC (g/kWh)
Conclusion

- Single stack, open and extensible analytics toolset that plays well with other technologies
- New features support clean workflows for use in production systems and with Big Data
- A capacity for complexity
Analyzing Fleet Test Data
Topics

- Reasons for analyzing fleet data
- MATLAB datatypes for improved productivity
- Scaling up analysis with mapreduce and parallel computing
- Machine Learning
- Moving to production
Why Analyze Fleet Data?

Big-picture view that was previously unavailable

- Design Decisions
  - Real-world fuel economy
  - Emissions
  - Vehicle dynamics
  - Ride and handling
  - Prognostics
  - Durability

- Algorithm Validation
  - Engine
  - Prognostics
  - ADAS
  - Hybrid/EV
Accessing Fleet Data

Database Access
- ODBC
- JDBC
- HDFS (Hadoop)

File I/O
- Text
- Spreadsheet
- XML
- CDF/HDF
- Image
- Audio
- Video
- Geospatial
- Web content

Hardware Access
- Data acquisition
- Image capture
- GPU
- Lab instruments

Communication Protocols
- CAN (Controller Area Network)
- DDS (Data Distribution Service)
- OPC (OLE for Process Control)
- XCP (eXplicit Control Protocol)
New MATLAB Datatypes

- **Table**
  - Mixed-type tabular data
  - Flexible indexing
  - Built-in functionality (merge, sort, join, etc.)

- **Datetime**
  - Time stamps, Durations
  - Add, subtract, sort, compare, plot
  - Customize display formats
  - Nanosecond precision
Working with Fleet Data

Goals

- Import data from different sources
- Clean-up and preprocess data
- Merge data
- Share analysis with others
Scaling up analysis

- **Parallel Computing Toolbox**
  - MATLAB worker on each core of your desktop
  - Loops: for -> parfor
  - Built-in support in many functions
  - MATLAB Distributed Computing Server

- **Map-Reduce**
  - `Datastore` for accessing data
  - `Mapreduce` for processing data
  - Run on Hadoop, or on MATLAB Distributed Computing Server
Event Detection

- Parse data, find sudden deceleration

- Map Reduce workflow
  - Serial on Desktop
  - Parallel on Desktop
  - Point to HDFS
  - Deploy to Hadoop
Machine Learning

Machine learning uses **data** and produces a **program** to perform a **task**

**Task:** Human Activity Detection

**Standard Approach**

- Hand Written Program
  - If $X_{acc} > 0.5$ then “SITTING”
  - If $Y_{acc} < 4$ and $Z_{acc} > 5$ then “STANDING”
  - ...

- Formula or Equation
  - $Y_{activity} = \beta_1 X_{acc} + \beta_2 Y_{acc} + \beta_3 Z_{acc} + ...$

**Machine Learning Approach**

- **model:** Inputs $\rightarrow$ Outputs

- $\text{model} = <\text{Machine Learning } \text{Algorithm}>(\text{sensor_data, activity})$
Overview – Machine Learning

- **Type of Learning**
  - Supervised Learning
    - Develop **predictive model** based on both input and output data
  - Unsupervised Learning
    - Group and interpret data based only on input data

- **Categories of Algorithms**
  - Classification
  - Regression
  - Clustering
Clustering Example
Gear Selection

Goal

- Given data for engine speed and vehicle speed, identify clusters of data using k-means

- Other applications
  - Separating different operating conditions
  - Clustering driver/buyer types
Classification Example
Human Activity Learning Using Mobile Phone Data

Data:
- 3-axial Accelerometer data
- 3-axial Gyroscope data

Other applications:
- ADAS: Ground-truth
- Predictive Maintenance
Overview – Machine Learning

- **Type of Learning**
  - Supervised Learning
    - Develop predictive model based on both input and output data
  - Unsupervised Learning
    - Group and interpret data based only on input data

- **Categories of Algorithms**
  - Classification
  - Regression
  - Clustering

MathWorks
Analysis Domains

**Statistics**
- Summary Statistics
- Regression, ANOVA, Machine Learning

**Signal Processing**
- Sound quality analysis
- LIDAR analysis

**Image Processing**
- Active Safety

**Location/Mapping**
- Analyzing GPS Data
- Custom Visualizations
Taking MATLAB to Production

Code Generation

Report Generation

Application Packaging

Embedded System

Reports

Applications

Business Systems
Additional Resources

Machine Learning

mathworks.com/machine-learning

Parallel Computing


MapReduce

MathWorks Services

- Consulting
  - Integration
  - Data analysis/visualization
  - Unify workflows, models, data
  
  [www.mathworks.com/services/consulting/](http://www.mathworks.com/services/consulting/)

- Training
  - Classroom, online, on-site
  - Data Processing, Visualization, Deployment, Parallel Computing, Machine Learning
  
  [www.mathworks.com/services/training/](http://www.mathworks.com/services/training/)
Summary

- Reasons for analyzing fleet data
- MATLAB datatypes for improved productivity
- Scaling up analysis with mapreduce and parallel computing
- Machine Learning
- Moving to production
Questions?