MATLAB EXPO 2015
KOREA

2015년 5월 21일 목요일
인터넷컨티넨탈 코엑스, 서울
빅 데이터 및 다양한 데이터 처리 위한 MATLAB의 인터페이스 환경 및 새로운 기능

엄준상 대리
Application Engineer
MathWorks
Challenges of Data

“Any collection of data sets so large and complex that it becomes difficult to process using … traditional data processing applications.” (Wikipedia)

- Various Data Sources
- Rapid data exploration
- Development of scalable algorithms
- Ease of deployment
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
MATLAB Connects to Your Hardware Devices

Data Acquisition Toolbox
Plug in data acquisition boards

Instrument Control Toolbox
Electronic and scientific instrumentation

Image Acquisition Toolbox™
Video devices

MATLAB
Interfaces for communicating with everything
Considerations for Choosing an Approach

- Data characteristics
  - Size, type and location of your data

- Compute platform
  - Single desktop machine or cluster

- Analysis Characteristics
  - Embarrassingly Parallel
  - Analyze sub-segments of data and aggregate results
  - Operate on entire dataset
Techniques for Big Data in MATLAB

Load, Analyze, Discard
parfor, datastore,

MapReduce

Distributed Memory
SPMD and distributed arrays

Complexity

Embarrassingly Parallel

Non-Partitionable

out-of-memory
in-memory
Demo: Determining Land Use

*Using Parallel for-loops (parfor)*

- **Data**
  - Arial images of agriculture land
  - 24 TIF files

- **Analysis**
  - Find and measure irrigation fields
  - Determine which irrigation circles are in use (by color)
  - Calculate area under irrigation
Access Big Data
dataset

- Easily specify data set
  - Single text file (or collection of text files)

- Preview data structure and format

- Select data to import using column names

- Incrementally read subsets of the data

```matlab
airdata = datastore('*.csv');
airdata.SelectedVariables = {'Distance', 'ArrDelay'};
data = read(airdata);
```
Demo: Vehicle Registry Analysis

**Using a DataStore**

- **Data**
  - Massachusetts Vehicle Registration Data from 2008-2011
  - 16M records, 45 fields

- **Analysis**
  - Examine hybrid adoptions
  - Calculate % of hybrids registered by quarter
  - Fit growth to predict further adoption
When to Use datastore

- **Data Characteristics**
  - Text data in files, databases or stored in the Hadoop Distributed File System (HDFS)

- **Compute Platform**
  - Desktop

- **Analysis Characteristics**
  - Supports Load, Analyze, Discard workflows
  - Incrementally read chunks of data, process within a `while` loop
Analyze Big Data
mapreduce

- Use the powerful MapReduce programming technique to analyze big data
  - mapreduce uses a datastore to process data in small chunks that individually fit into memory
  - Useful for processing multiple keys, or when Intermediate results do not fit in memory

- mapreduce on the desktop
  - Increase compute capacity (Parallel Computing Toolbox)
  - Access data on HDFS to develop algorithms for use on Hadoop

- mapreduce with Hadoop
  - Run on Hadoop using MATLAB Distributed Computing Server
  - Deploy applications and libraries for Hadoop using MATLAB Compiler
### Data Store

<table>
<thead>
<tr>
<th>Veh_typ</th>
<th>Q3_08</th>
<th>Q4_08</th>
<th>Q1_09</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SUV</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SUV</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Car</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Map

#### Hybrid

- **Key: Q3_08**
  - 0
  - 1

- **Key: Q4_08**
  - 0
  - 1

- **Key: Q1_09**
  - 0
  - 1

### Shuffle and Sort

#### Hybrid

- **Key: Q3_08**
  - 0
  - 1

- **Key: Q4_08**
  - 0
  - 1

- **Key: Q1_09**
  - 0
  - 1

### Reduce

<table>
<thead>
<tr>
<th>Key</th>
<th>% Hybrid (Value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q3_08</td>
<td>0.4</td>
</tr>
<tr>
<td>Q4_08</td>
<td>0.67</td>
</tr>
<tr>
<td>Q1_09</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Demo: Vehicle Registry Analysis
Using MapReduce

- **Data**
  - Massachusetts Vehicle Registration Data from 2008-2011
  - 16M records, 45 fields

- **Analysis**
  - Examine hybrid adoptions
  - Calculate % of hybrids registered
    - By Quarter
    - By Regional Area
  - Create map of results
Explore and Analyze Data on Hadoop

MATLAB Distributed Computing Server

Datastore

HDFS

Node

Data

Map

Reduce

Node

Data

Map

Reduce

Node

Data

Map

Reduce

MATLAB MapReduce Code

MATLAB EXPO 2015
Deployed Applications with Hadoop

MATLAB runtime

Datastore

HDFS

Node

Data

Map

Reduce

Node

Data

Map

Reduce

Node

Data

Map

Reduce

MATLAB MapReduce Code

MATLAB EXPO 2015
When to Use mapreduce

- **Data Characteristics**
  - Text data in files, databases or stored in the Hadoop Distributed File System (HDFS)
  - Dataset will not fit into memory

- **Compute Platform**
  - Desktop
  - Scales to run within Hadoop MapReduce on data in HDFS

- **Analysis Characteristics**
  - Must be able to be Partitioned into two phases
    1. Map: filter or process sub-segments of data
    2. Reduce: aggregate interim results and calculate final answer
Example: Airline Delay Analysis

- **Data**
  - BTS/RITA Airline On-Time Statistics
  - 123.5M records, 29 fields

- **Analysis**
  - Calculate delay patterns
  - Visualize summaries
  - Estimate & evaluate predictive models
When to Use Distributed Memory

- **Data Characteristics**
  - Data must be fit in collective memory across machines

- **Compute Platform**
  - Prototype (subset of data) on desktop
  - Run on a cluster or cloud

- **Analysis Characteristics**
  - Consists of:
    - Parts that can be run on data in memory (spmd)
    - Supported functions for distributed arrays
Big Data on the Desktop

- Expand workspace
  - 64 bit processor support – increased in-memory data set handling

- Access portions of data too big to fit into memory
  - Memory mapped variables – huge binary file
  - Datastore – huge text file or collections of text files
  - Database – query portion of a big database table

- Variety of programming constructs
  - System Objects – analyze streaming data
  - MapReduce – process text files that won’t fit into memory

- Increase analysis speed
  - Parallel for-loops with multicore/multi-process machines
  - GPU Arrays
Further Scaling Big Data Capacity

MATLAB supports a number of programming constructs for use with clusters

- General compute clusters
  - Parallel for-loops – embarrassingly parallel algorithms
  - SPMD and distributed arrays – distributed memory

- Hadoop clusters
  - MapReduce – analyze data stored in the Hadoop Distributed File System

Use these constructs on the desktop to develop your algorithms

Migrate to a cluster without algorithm changes
Learn More

- MATLAB Documentation
  - Strategies for Efficient Use of Memory
  - Resolving "Out of Memory" Errors

- Big Data with MATLAB

- MATLAB MapReduce and Hadoop
  - www.mathworks.com/discovery/matlab-mapreduce-hadoop.html
Questions?