How big is big? What characterises “big” data?
“Any collection of data sets so large and complex that it becomes difficult to process using … traditional data processing applications.”

Wikipedia
MATLAB Application Development Landscape

Prototyping  Programming  Deployment
MATLAB Application Development Landscape

Prototyping  Programming  Deployment
Data Analytics with MATLAB

Tackling the Challenges of Big Data

How big is big? What characterises “big” data?

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

Wikipedia
Data-Driven Decisions and Data-Driven Design

Measurement Devices

Big Data

Compute Power
Need is Across Many Application Areas

- **System Design**
  - Hybrid and electric vehicles

- **Signal Processing**
  - Sound quality analysis

- **Image Processing**
  - Advanced driver assistance system

- **Model-Based Design**
  - Engine Calibration

- **Data Analysis**
  - Portfolio risk optimization
Data Analytics in MATLAB
Moving up the Information Hierarchy

Physical Sensors
Data
Information
Knowledge
Action

Source: Information Warfare
Edward Waltz – 1998
Data Analytics in MATLAB
Moving up the Information Hierarchy

- Physical Sensors
  - Sensing
  - Collecting
  - Measurement
  - Data Acquisition

- Imaging devices

- Data acquisition Instruments

- Flat files, Excel, Web

- Databases
  - Data warehouses
  - HDFS (Hadoop)
Data Analytics in MATLAB
Moving up the Information Hierarchy

- Physical Sensors

Data
- Sensing
- Collecting
- Measurement
- Data Acquisition

Data Processing
- Preprocessing
- Calibration
- Filtering
- Data Reduction

Information
- ORGANIZATION

Knowledge
- Action

Exploratory Analysis

Filtering
Data Analytics in MATLAB
Moving up the Information Hierarchy

- **Physical Sensors**
- **Data**
- **Information**
- **Knowledge**
- **Action**

**UNDERSTANDING**

**Machine Learning**

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

**Categories of Algorithms**
Data Analytics in MATLAB
Moving up the Information Hierarchy

Physical Sensors

Data

Information

Knowledge

Action

- Reporting
- Apps
- Scalable Deployment
- Integration

- Analysis
- Visualization
- Modeling
- Prediction

- Preprocessing
- Calibration
- Filtering
- Data Reduction

- Sensing
- Collecting
- Measurement
- Data Acquisition

MATLAB Applications

Reports

Integration into Existing Systems

Excel

Feedback for Design and Operations

APPLICATION

Integration

Excel

JavaScript

.NET

C/C++

Java

MATLAB
Large Data Analytics

Work on the desktop

Scale capacity as needed
Large Data Analytics on the Desktop

Access big data from your desktop

<table>
<thead>
<tr>
<th>Collections of Text Files</th>
<th>datastore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases</td>
<td>Database Toolbox</td>
</tr>
<tr>
<td>Binary Files</td>
<td>memmapfile</td>
</tr>
</tbody>
</table>
Example: Airline Flight Distance

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

- **Task**
  - Find the maximum distance travelled by commercial airlines based upon flight operations performance data

CSV Data
- 22 files
- 12GB
Standard Workflow (up to R2014a)


fmtspec = ['%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q'; ...
              '%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q'; ...
              '%*q%*q %*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q%*q'];

maxdist = -Inf;
for i = 1 : numfiles
    filei = fopen(files{i});
    data = textscan(filei, fmtspec, ...);
    fclose(filei);
    maxi = max(data{:});
    maxdist = max(maxdist, maxi);
end
New Workflow with datastore (in R2014b)

```matlab
airdata = datastore('*.csv');

airdata.SelectedVariableNames = {'Distance'};
airdata.SelectedFormats = {'%f'};

maxdist = -Inf;
while hasdata(airdata)
    data = read(airdata);
    maxi = max(data.Distance);
    maxdist = max(maxdist, maxi);
end
```

Location
Format
Read data
Compute
Combine
Easily specify data set
- Single text file (or collection of text files)
- Database (using Database Toolbox)

Preview data structure and format

Select data to import using column names

Incrementally read subsets of the data

```matlab
airdata = datastore('*.*');
airdata.SelectedVariables = {'Distance', 'ArrDelay'};
data = read(airdata);
```
Large Data Analytics 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 – use with multicore/multi-process machines
  - GPU Arrays
Scaled Large Data Analytics

Load, Analyze, Discard
datastore, parfor

Distributed Memory
SPMD

Complexity

Embarrassingly Parallel
Non-Partitionable

out-of-memory
in-memory

Prototype
for k=true
x = fft(data)
y = 20*log1

Access
Explore
Share/Deploy

Scale

Parallel
Non-Partitionable
Example: Airline Delay Analysis

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

- **Tasks**
  - Calculate delay patterns
  - Visualize summaries
  - Estimate & evaluate predictive models

- **Resources**
  - Amazon S3 data store
  - Amazon EC2 cluster
Airline Delay Analysis: Framework
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* – distributed processing

- Hadoop clusters
  - *MapReduce* – analyze data stored in the Hadoop Distributed File System
Scaled Large Data Analytics

Load, Analyze, Discard

MapReduce

Distributed Memory

Embarrassingly Parallel

Non-Partitionable

out-of-memory

in-memory
mapreduce (in R2014b)

### Data Store

<table>
<thead>
<tr>
<th>Muni_ID</th>
<th>Airline ID</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1503</td>
<td>UA</td>
<td>LAX -5</td>
</tr>
<tr>
<td>540</td>
<td>PS</td>
<td>BUR 13</td>
</tr>
<tr>
<td>1920</td>
<td>DL</td>
<td>BOS 10</td>
</tr>
<tr>
<td>1840</td>
<td>DL</td>
<td>SFO 0</td>
</tr>
<tr>
<td>272</td>
<td>US</td>
<td>BWI 4</td>
</tr>
<tr>
<td>784</td>
<td>PS</td>
<td>SEA 7</td>
</tr>
<tr>
<td>796</td>
<td>PS</td>
<td>LAX -2</td>
</tr>
<tr>
<td>1525</td>
<td>UA</td>
<td>SFO 3</td>
</tr>
<tr>
<td>632</td>
<td>US</td>
<td>SJC 2</td>
</tr>
<tr>
<td>1610</td>
<td>UA</td>
<td>MIA 60</td>
</tr>
<tr>
<td>2032</td>
<td>DL</td>
<td>EWR 10</td>
</tr>
<tr>
<td>2134</td>
<td>DL</td>
<td>DFW -2</td>
</tr>
</tbody>
</table>

### Map

- UA 2356
- PS 186
- DL 1876

### Reduce

- US 359
- PS 237
- DL 1876
- US 245
- UA 1365
- DL 914
- US 359
% Specify and format the data
indata = datastore('*csv');
indata.SelectedVariables = 'Distance';
indata.SelectedFormats = '%f';

function mapfun(data,~,intermed)
    % Compute and save intermediate result
    maxi = max(data.Distance);
    add(intermed,'maxi',maxi);
end

function reducefun(~,intermed,output)
    maxdist = -Inf;
    while hasnext(intermed)
        maxi = getnext(intermed);
        % Combine intermediate results
        maxdist = max(maxdist,maxi);
    end
    add(output,'maxdist',maxdist);
end

outdata = mapreduce(indata,@mapfun,@reducefun)
mapreduce

- Use the powerful MapReduce programming technique to analyze big data
  - Multiple items (keys) to organize and process
  - Intermediate results do not fit in memory

- On the desktop
  - Analyze big database tables (Database Toolbox)
  - Increase compute capacity (Parallel Computing Toolbox)
  - Access data on HDFS to develop algorithms for use on Hadoop

- With Hadoop
  - Run on Hadoop using MATLAB Distributed Computing Server
  - Deploy applications and libraries for Hadoop using MATLAB Compiler
Data Analytics Landscape

**COMPLEX**
- iterative
- all data needed in memory at once

**SIMPLE**
- easily partitioned; independent tasks

**Algorithm complexity**
- built-in numerical & statistical algorithms
- vectorisation

**Increasing Data Size**
- SMALL
- Increasing Data Size

**More programming effort required**
- spmd distributed arrays
- mapreduce
- R2014b
- parfor
- gpuarray
# Strengths of MATLAB for Large Data Analytics

<table>
<thead>
<tr>
<th>Challenge</th>
<th>MATLAB Solution</th>
</tr>
</thead>
</table>
| Getting started                   | Easy access to data from your desktop  
Tools for accessing typical big data sets consisting of text or binary files, contained in database tables or stored on Hadoop |
| Rapid data exploration            | All the tools to explore and visualize data  
• Easy to try different methods  
• Ideal environment for developing your own methods |
| Development of scalable algorithms | Work on the desktop and scale to clusters  
Tools for use in analyzing big data on your desktop, which scale for use on clusters, including Hadoop, if needed |
| Use within business systems       | Ease of deployment and leveraging enterprise  
Push-button deployment into production including support for Hadoop |
Strengths of MATLAB for Large Data Analytics

<table>
<thead>
<tr>
<th>Challenge</th>
<th>MATLAB Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting started</td>
<td>Easy access to data from your desktop</td>
</tr>
<tr>
<td></td>
<td>• Tools for accessing typical data sets consisting of text or binary files, Excel files, contained in database tables.</td>
</tr>
<tr>
<td></td>
<td>• Data import from instruments</td>
</tr>
<tr>
<td>Rapid data exploration</td>
<td>All the tools to explore and visualize data</td>
</tr>
<tr>
<td></td>
<td>• Easy to try different methods</td>
</tr>
<tr>
<td></td>
<td>• Ideal environment for developing your own methods</td>
</tr>
<tr>
<td>Development of scalable algorithms</td>
<td>Work on the desktop and scale to clusters</td>
</tr>
<tr>
<td></td>
<td>Tools for use in analyzing big data on your desktop, which scale for use on clusters, including cloud, if needed</td>
</tr>
<tr>
<td>Use within business systems</td>
<td>Ease of deployment and leveraging enterprise</td>
</tr>
<tr>
<td></td>
<td>Push-button deployment into production framework</td>
</tr>
</tbody>
</table>
MATLAB Application Development Landscape

Prototyping  Programming  Deployment
MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders. © 2014 The MathWorks, Inc.