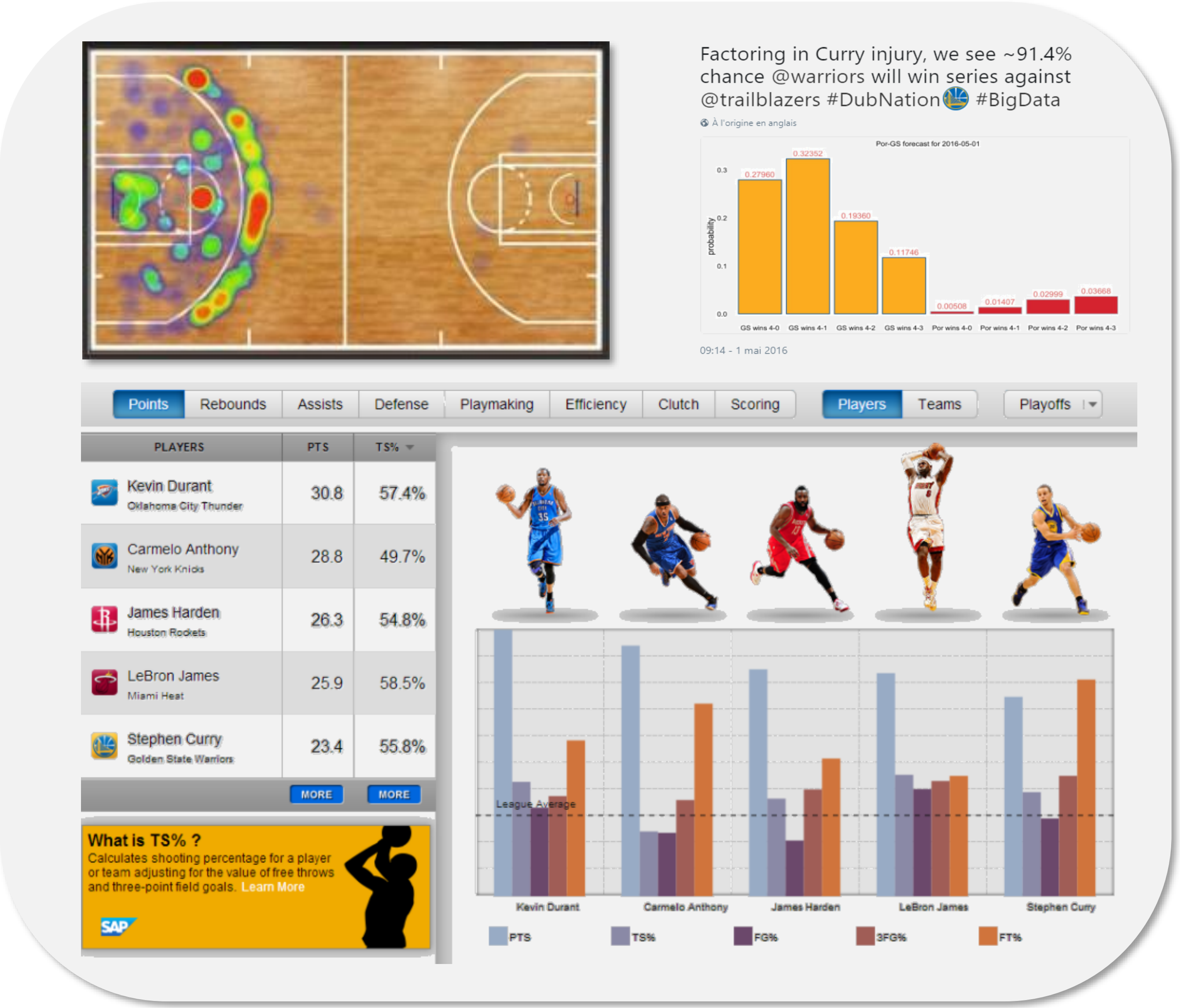
MATLAB EXPO 2018

Big Data
with MATLAB and Spark

Pierre Harouimi

Real-World Example: Sports Analytics



Factoring in Curry injury, we see ~91.4% chance @warriors will win series against @trailblazers #DubNation #BigData

À l'origine en anglais

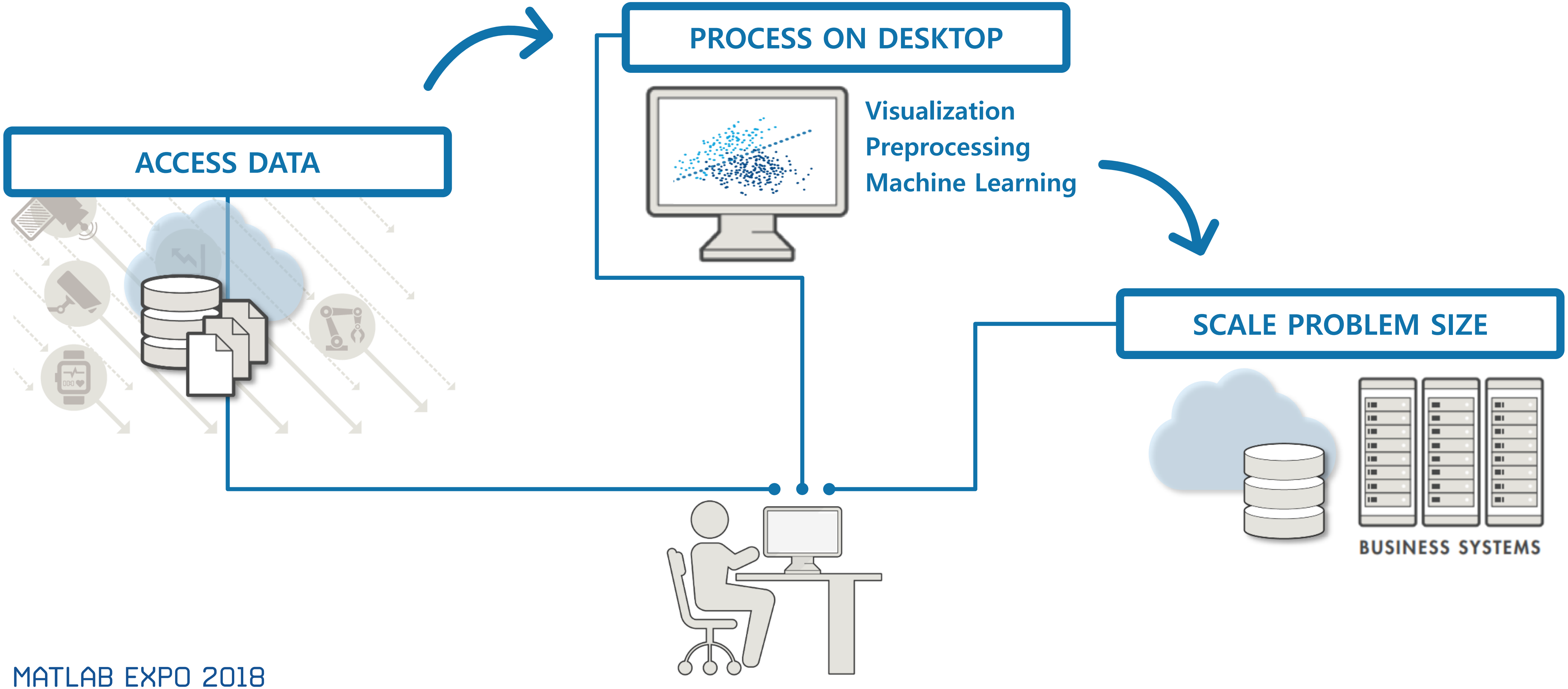
09:14 - 1 mai 2016

PLAYERS	PTS	TS%
Kevin Durant Oklahoma City Thunder	30.8	57.4%
Carmelo Anthony New York Knicks	28.8	49.7%
James Harden Houston Rockets	26.3	54.8%
LeBron James Miami Heat	25.9	58.5%
Stephen Curry Golden State Warriors	23.4	55.8%

What is TS% ?
Calculates shooting percentage for a player or team adjusting for the value of free throws and three-point field goals. Learn More

- Too much data to handle and capture it
- Difficult to predict
- Real-Time dependence

Big data workflow: from desktop to production



ACCESS DATA

PROCESS ON DESKTOP

Visualization
Preprocessing
Machine Learning

SCALE PROBLEM SIZE

So, what’s the big (data) challenges?

- Standard tools won’t work
- Time-consuming
- Need to learn new tools & rewrite algorithms

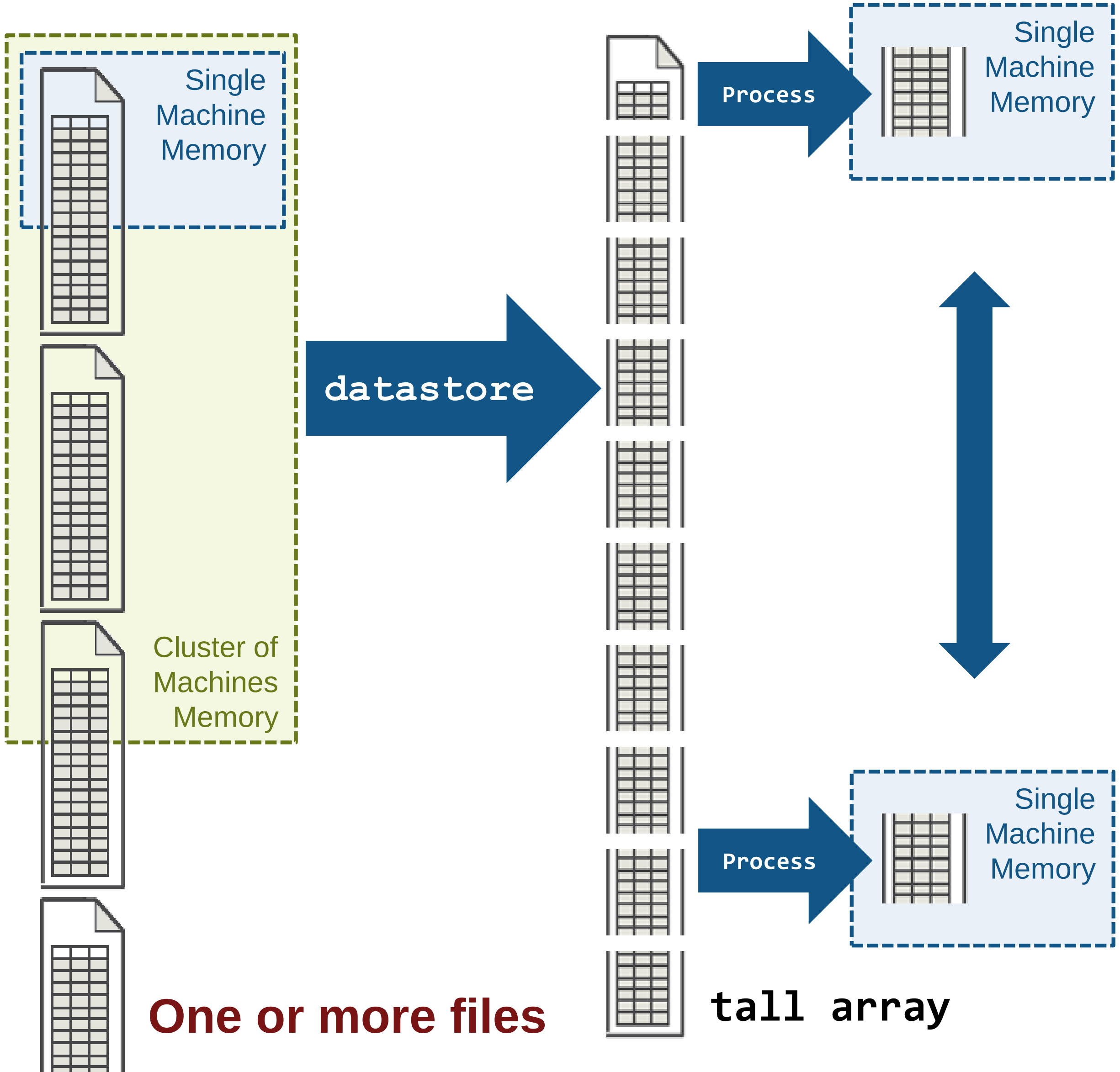
Solution!

- Standard tools won’t work → Prototype algorithms quickly

- Time-consuming → Run directly from MATLAB with **tall arrays**

- Need to learn new tools & rewrite algorithms → Use the **same MATLAB code**

Datastore & tall arrays



1. Use datastore to define file-list

```
>> ds = datastore('*.csv')
```

2. Create tall table from datastore

```
>> tt = tall(ds)
```

3. Act like ordinary table in parallel

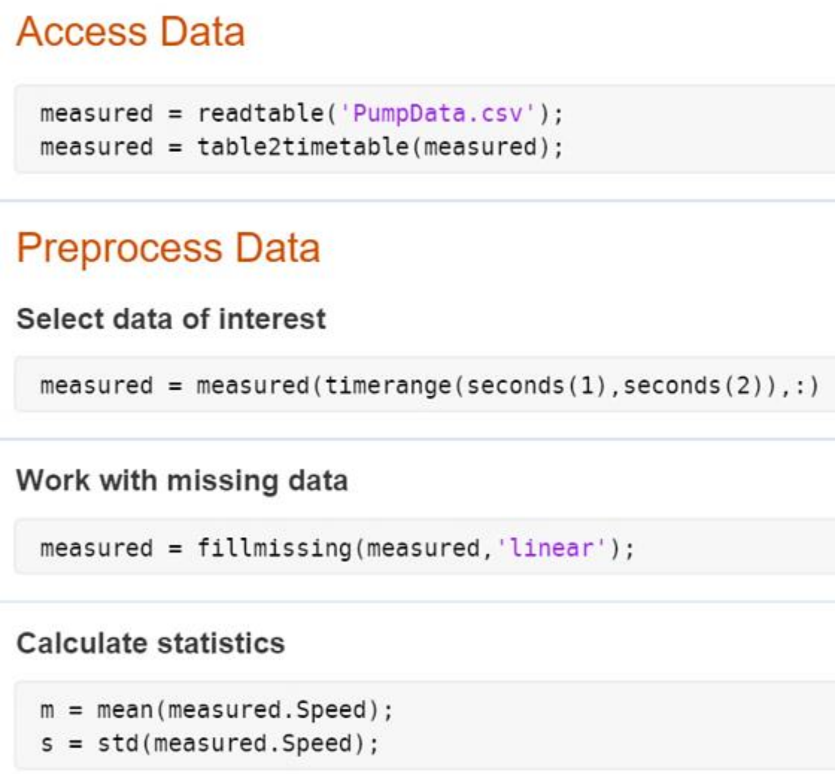
```
>> model = fitlm(tt.Temp=...)
```

4. Request on local machine

```
>> result = gather(tt.result)
```

Tall arrays: very small changes

1 file



Access Data

```
measured = readtable('PumpData.csv');
measured = table2timetable(measured);
```

Preprocess Data

Select data of interest

```
measured = measured(timerange(seconds(1),seconds(2)),:)
```

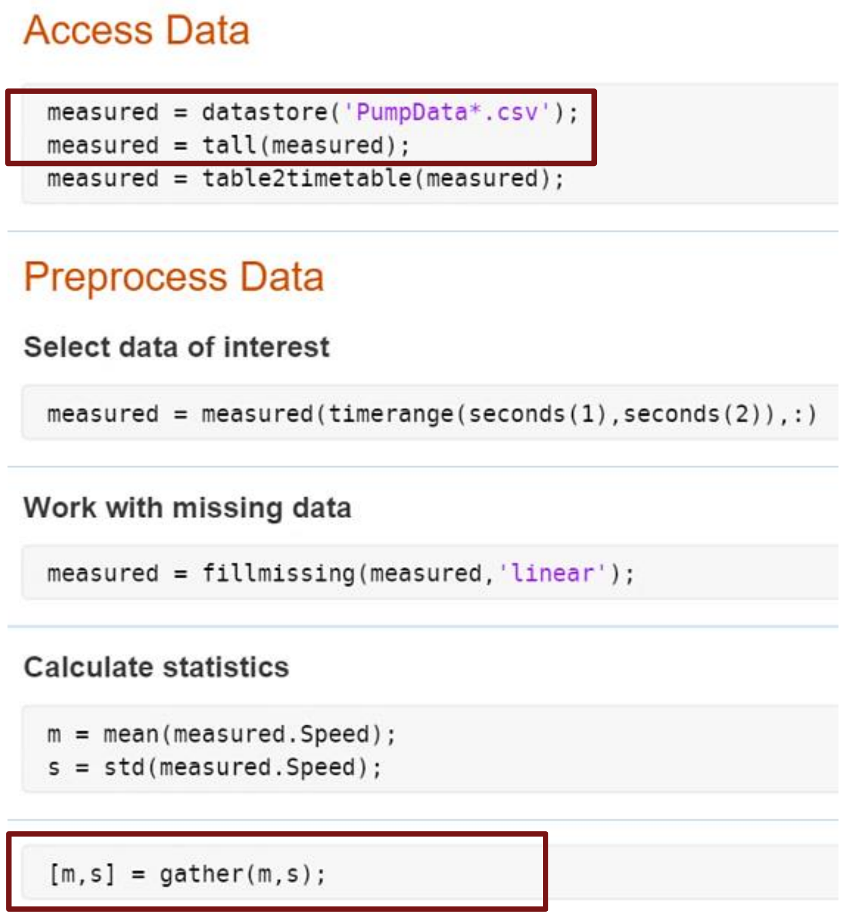
Work with missing data

```
measured = fillmissing(measured,'linear');
```

Calculate statistics

```
m = mean(measured.Speed);
s = std(measured.Speed);
```

1000+ files



Access Data

```
measured = datastore('PumpData*.csv');
measured = tall(measured);
measured = table2timetable(measured);
```

Preprocess Data

Select data of interest

```
measured = measured(timerange(seconds(1),seconds(2)),:)
```

Work with missing data

```
measured = fillmissing(measured,'linear');
```

Calculate statistics

```
m = mean(measured.Speed);
s = std(measured.Speed);
```

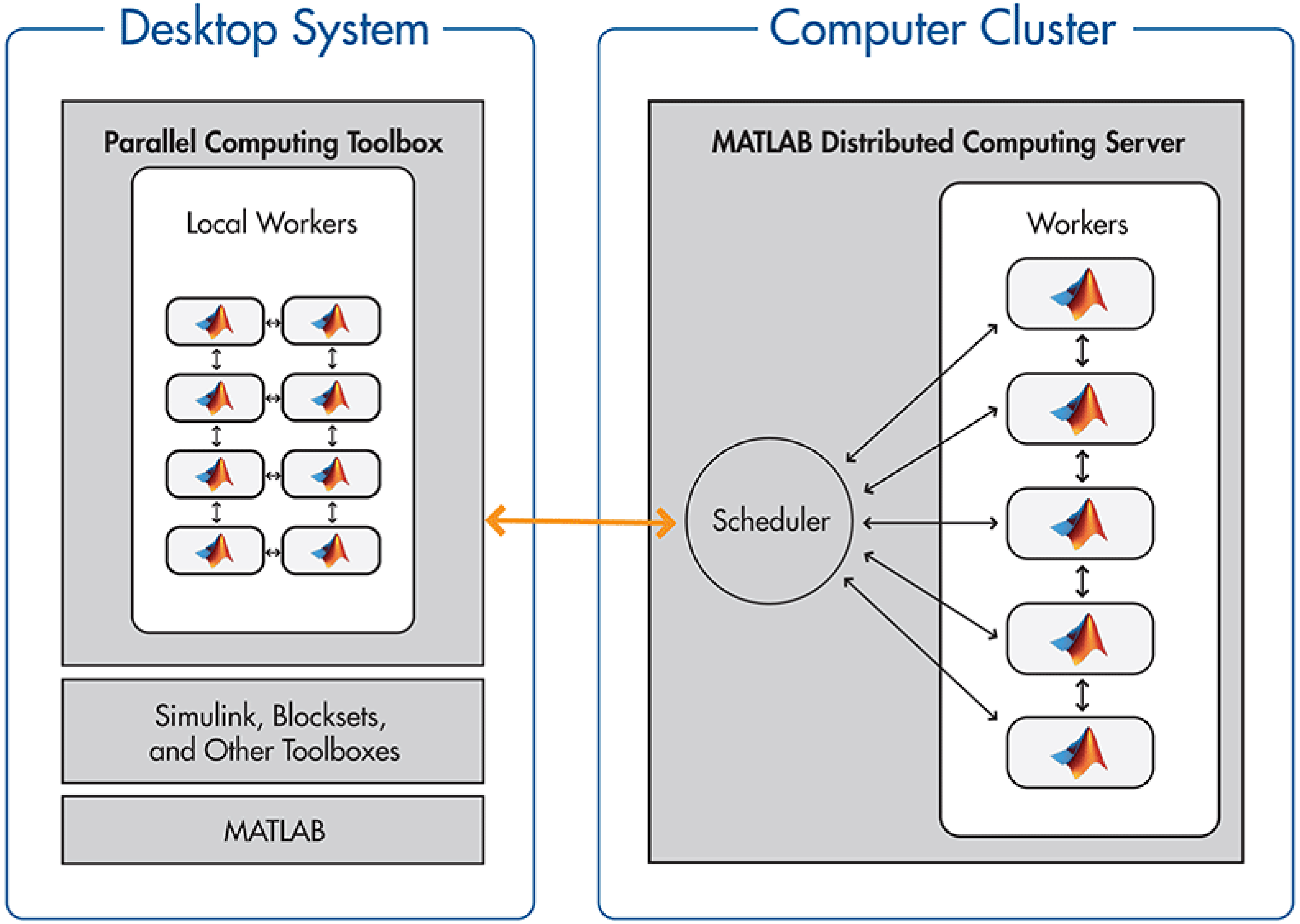
```
[m,s] = gather(m,s);
```

Workflow Pattern

Access out of memory data	`datastore & tall`
Work with subsets of your data	`findgroups, splitapply`
Develop functions for event detection and calculation	`Normal MATLAB code`
Apply functions to all of your data	`cellfun`
Aggregate, summarize, & visualize	`table, histogram, heatmap, boxplot, binScatterPlot`

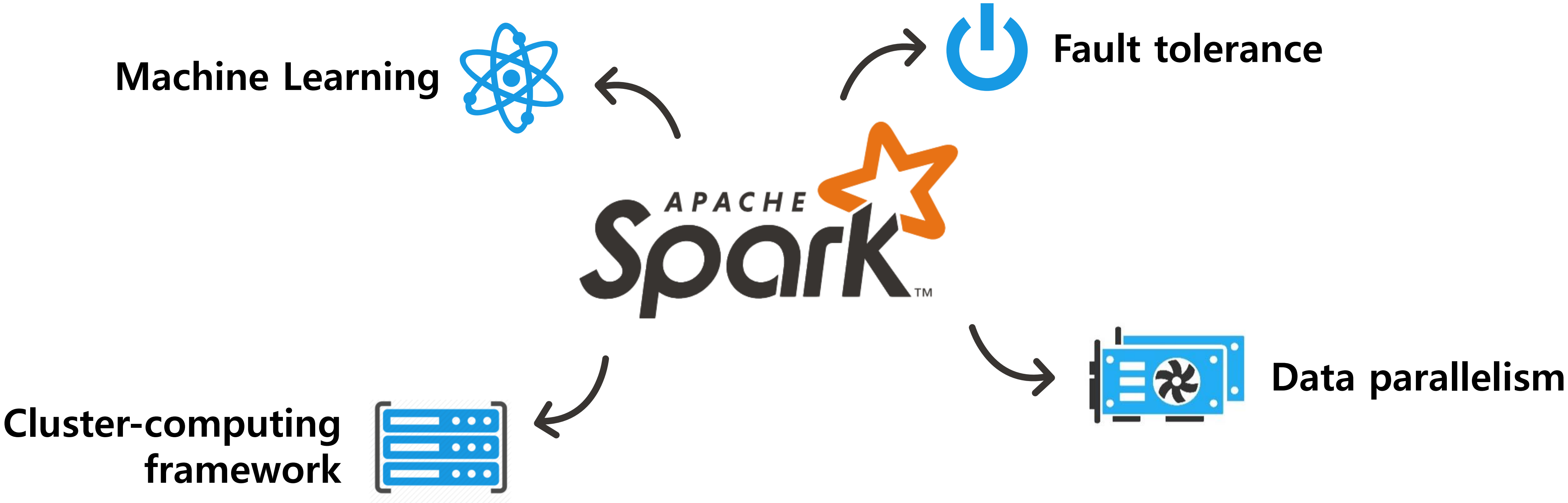
MATLAB Distributed Computing Server (MDCS)

Local Parallel Computing


Desktop System
Parallel Computing Toolbox
Local Workers
Simulink, Blocksets, and Other Toolboxes
MATLAB
Computer Cluster
MATLAB Distributed Computing Server
Workers
Scheduler

Deployed Parallel Computing

What is Hadoop/Spark?



Machine Learning

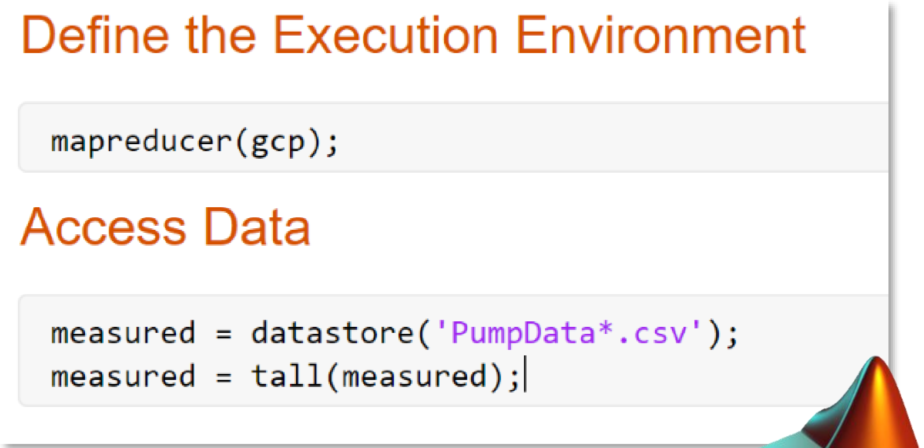
Fault tolerance

Cluster-computing framework

Data parallelism

Scaling with Spark: Very small changes too!

Desktop Code → Spark + Hadoop Code

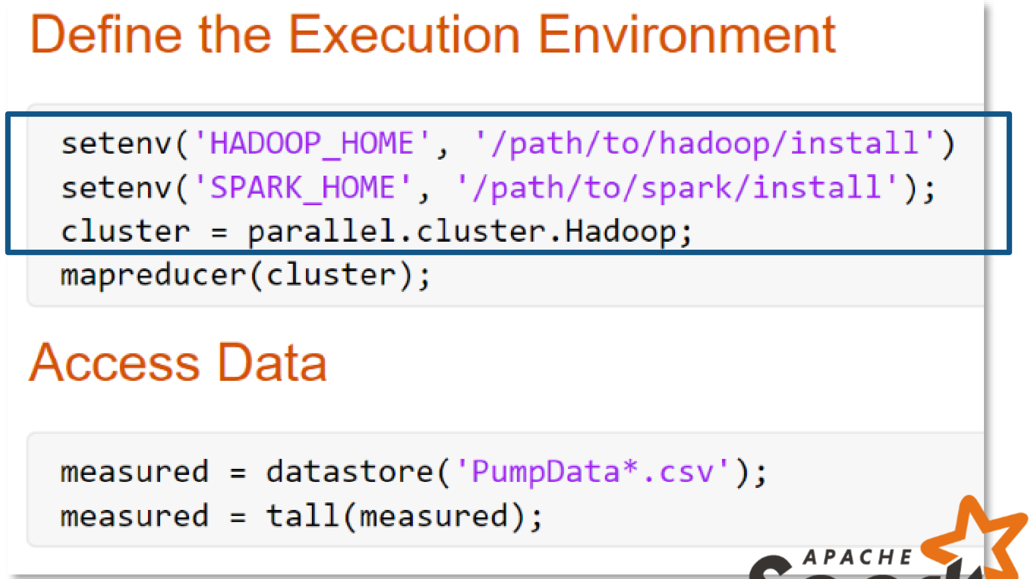


Define the Execution Environment

```
mapreducer(gcp);
```

Access Data

```
measured = datastore('PumpData*.csv');
measured = tall(measured);
```



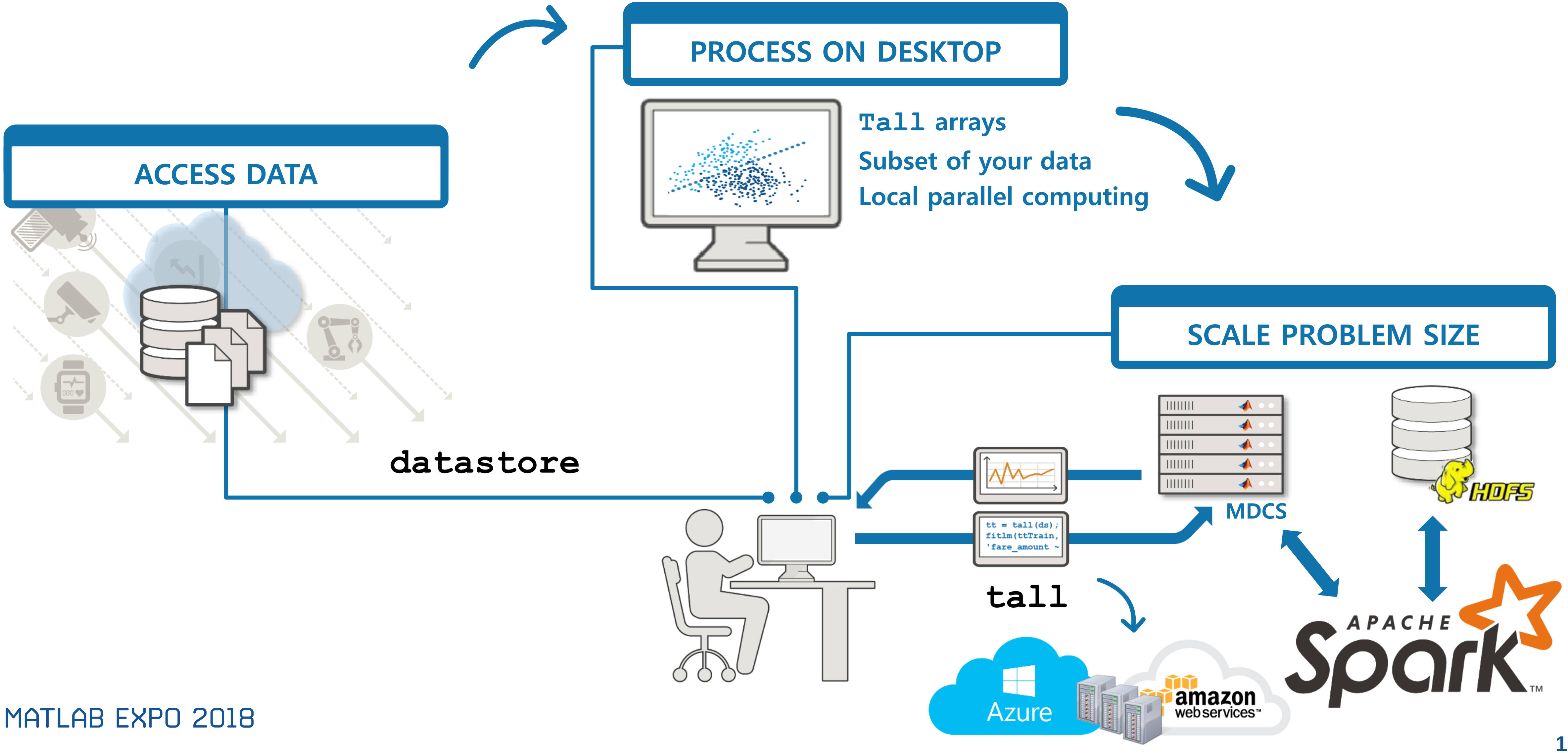
Define the Execution Environment

```
setenv('HADOOP_HOME', '/path/to/hadoop/install')
setenv('SPARK_HOME', '/path/to/spark/install');
cluster = parallel.cluster.Hadoop;
mapreducer(cluster);
```

Access Data

```
measured = datastore('PumpData*.csv');
measured = tall(measured);
```

Big Data with MATLAB & Spark



ACCESS DATA

PROCESS ON DESKTOP

Tall arrays

Subset of your data

Local parallel computing

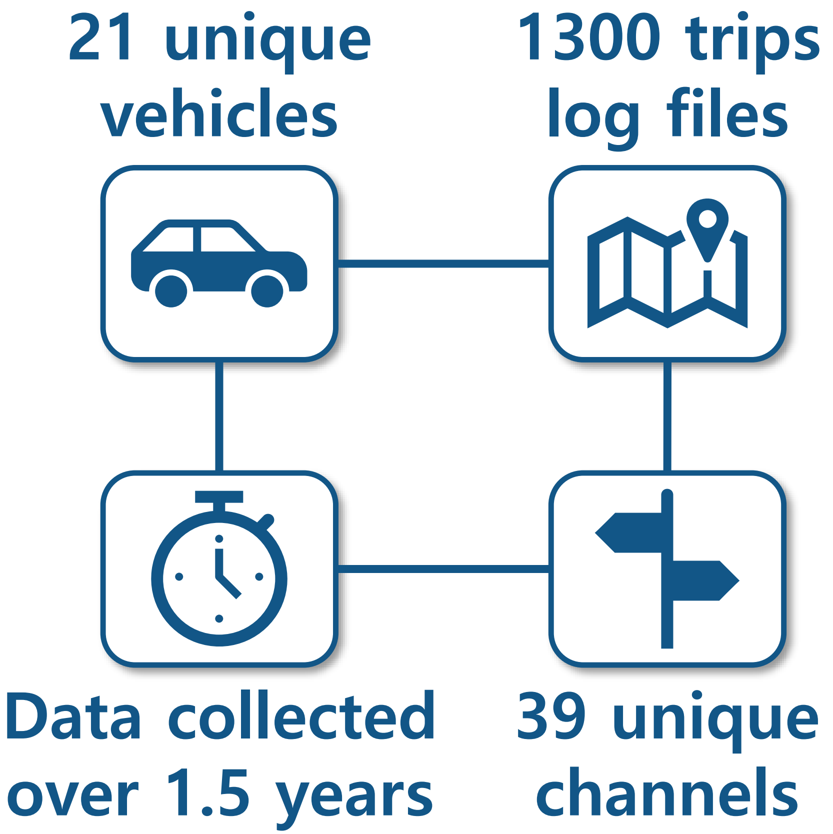
SCALE PROBLEM SIZE

datastore

tall

```
tt = tall(ds);
fitlm(ttTrain,
'fare_amount ~
```

The MathWorks Fleet Data



21 unique vehicles

1300 trips log files

Data collected over 1.5 years

39 unique channels

Challenges

Big Data

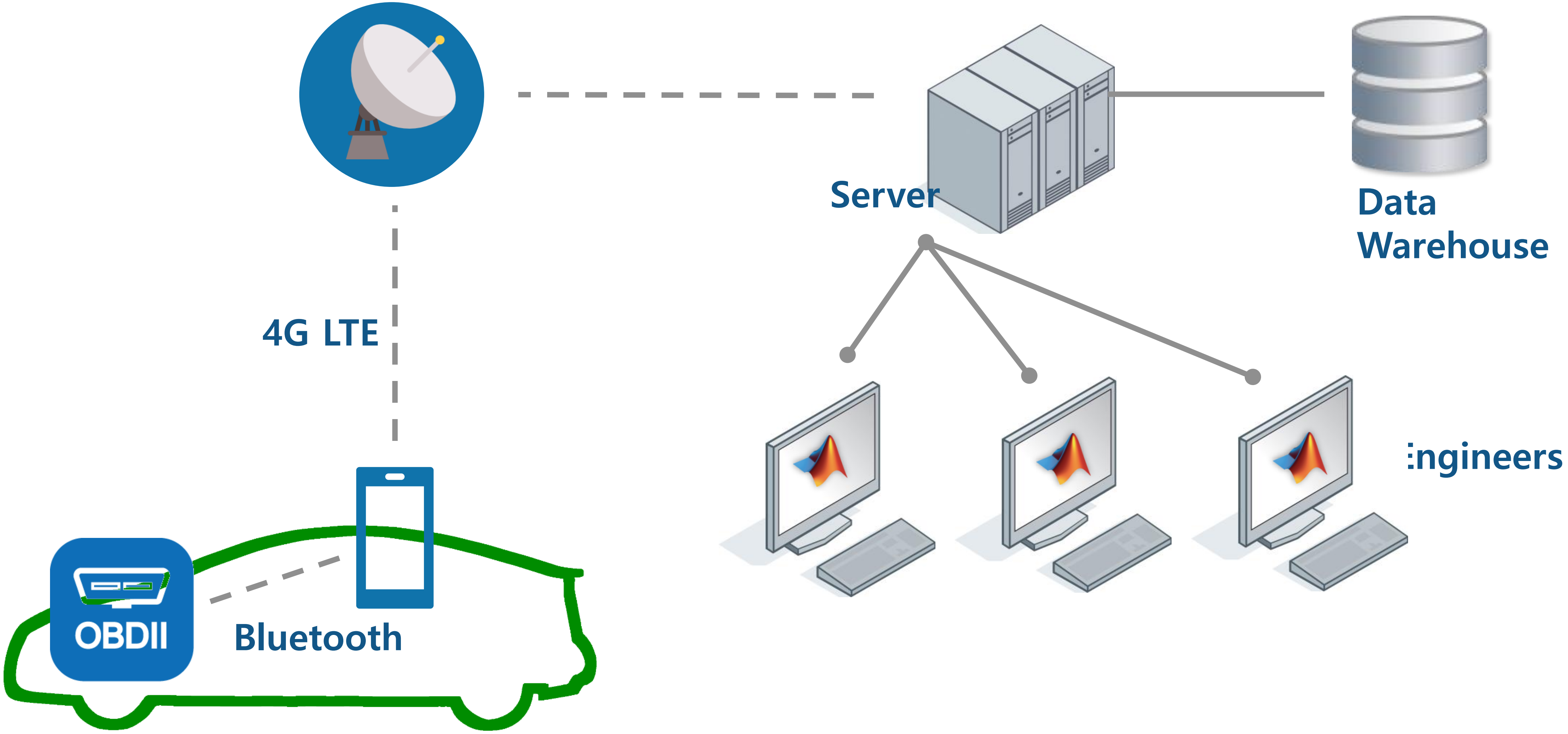
Testing Ideas

Needle in the Haystack

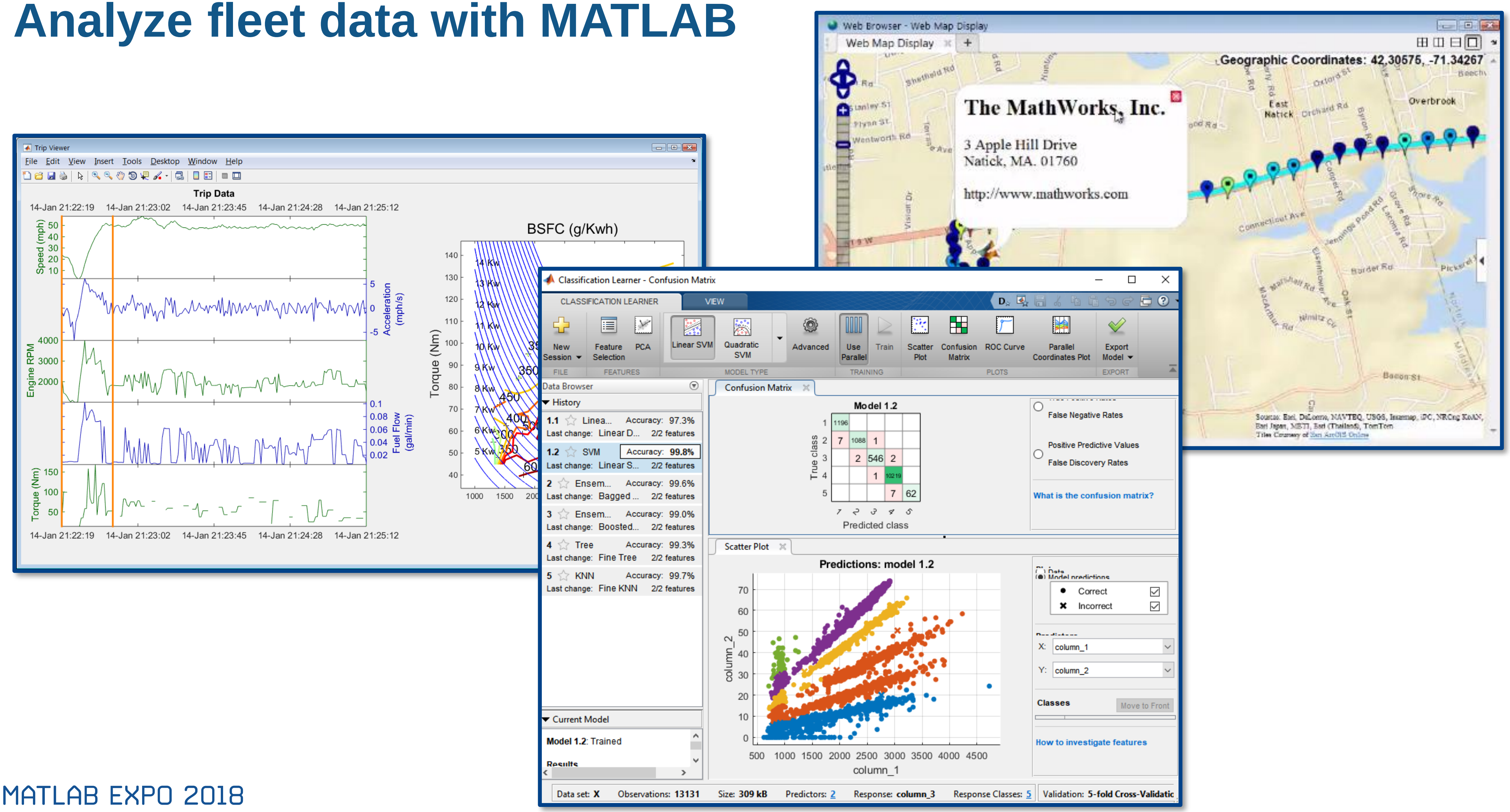
Event Detection

Objectives

Example Setup at MathWorks



Analyze fleet data with MATLAB



The MathWorks, Inc.

3 Apple Hill Drive
Natick, MA. 01760

http://www.mathworks.com

1.1	Linea...	Accuracy: 97.3%	
Last change:	Linear D...		2/2 features
1.2	SVM	Accuracy: 99.8%	
Last change:	Linear S...		2/2 features
2	Ensem...	Accuracy: 99.6%	
Last change:	Bagged ...		2/2 features
3	Ensem...	Accuracy: 99.0%	
Last change:	Boosted...		2/2 features
4	Tree	Accuracy: 99.3%	
Last change:	Fine Tree		2/2 features
5	KNN	Accuracy: 99.7%	
Last change:	Fine KNN		2/2 features

Model 1.2: Trained

Results

Data set: X Observations: 13131 Size: 309 kB Predictors: 2 Response: column_3 Response Classes: 5 Validation: 5-fold Cross-Validatic

MATLAB EXPO 2018

Access & Explore Data: *MATLAB & Spark*
MathWorks Vehicle Fleet

Challenge Develop and deploy Data Analytics to run on Spark against vehicle fleet data stored on Hadoop

Solution Use MATLAB **`tall`** arrays to develop analytics on the desktop and then scale out to the Spark cluster

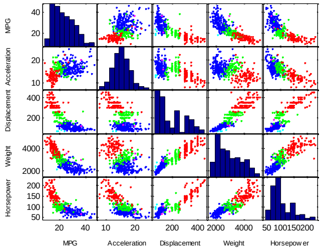
Results Developed insight and understanding of over 1300 vehicle trips

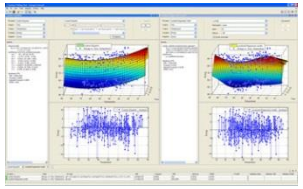
Fuel efficiency performance under real-world driving conditions

Analysis Domains

Statistics

- Summary Statistics
- Regression, ANOVA, Machine Learning





Signal Processing

- Sound quality analysis
- LIDAR analysis

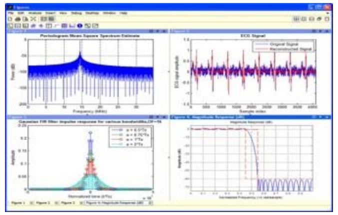
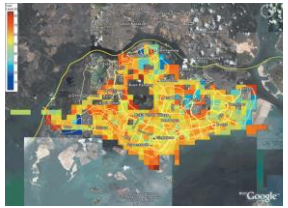


Image Processing

- Active Safety

Location/Mapping

- Analyzing GPS Data
- Custom Visualizations



Key Takeaways

- Use the **same MATLAB code**

- Use new MATLAB data types **`datastore`** & **`tall`** arrays for **out of memory** data sets

- **Scale** your work up with **Parallel Computing Toolbox** on the desktop or the **MATLAB Distributed Computing Server (MDCS)** on **Spark**

MathWorks®

Accelerating the pace of engineering and science