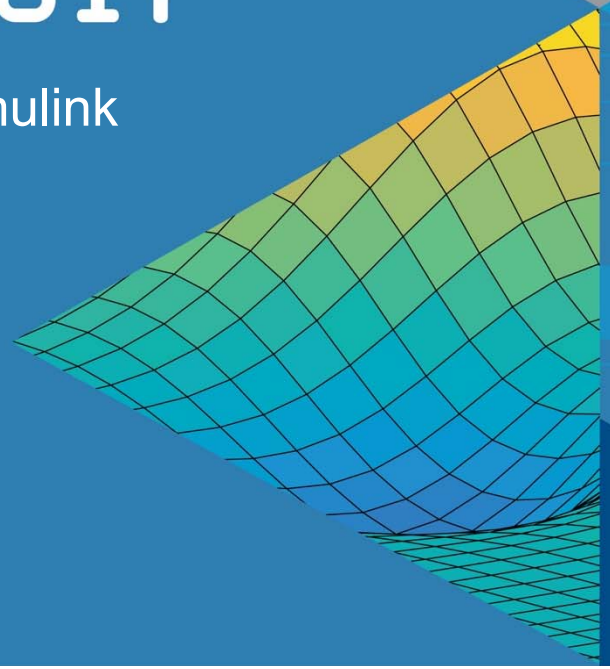
# MATLAB EXPO 2017

Parallel Computing with MATLAB and Simulink



# Overcome Challenges

## Challenges

- Need faster insight to bring competitive products to market quickly
- Complexity of analytical problems is growing across industries

## Solution

- Leverage computational power of multicore desktops, GPUs, clusters, and clouds
  - Without being an expert in parallel computing
  - Easily scale from desktop to clusters/cloud

# Agenda

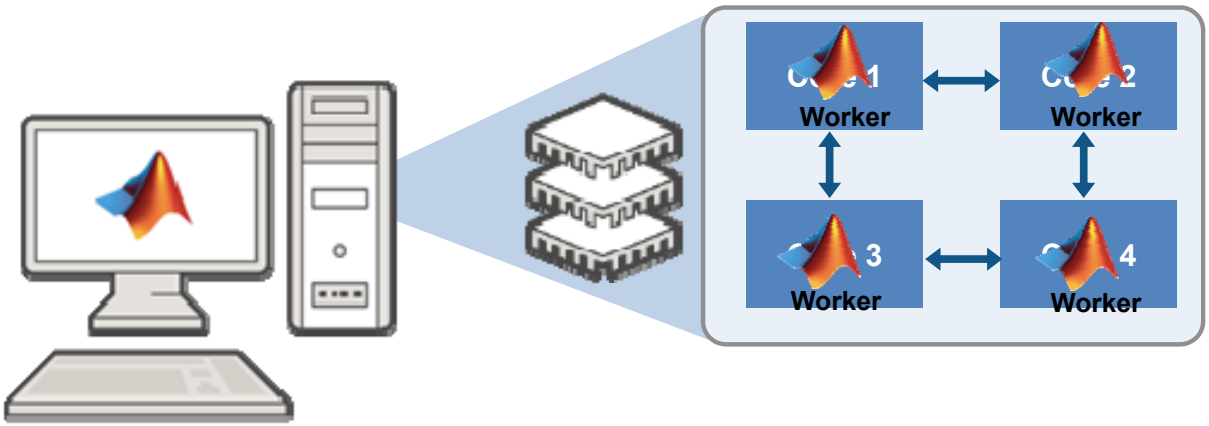
- Accelerate MATLAB and Simulink applications in your desktop
- Accelerate with NVIDIA™ GPUs
- Handle Big Data
- Scale to clusters and the cloud

# Agenda

- Accelerate MATLAB and Simulink applications in your desktop
- Accelerate with NVIDIA™ GPUs
- Handle Big Data
- Scale to clusters and the cloud

# Parallel Computing Paradigm

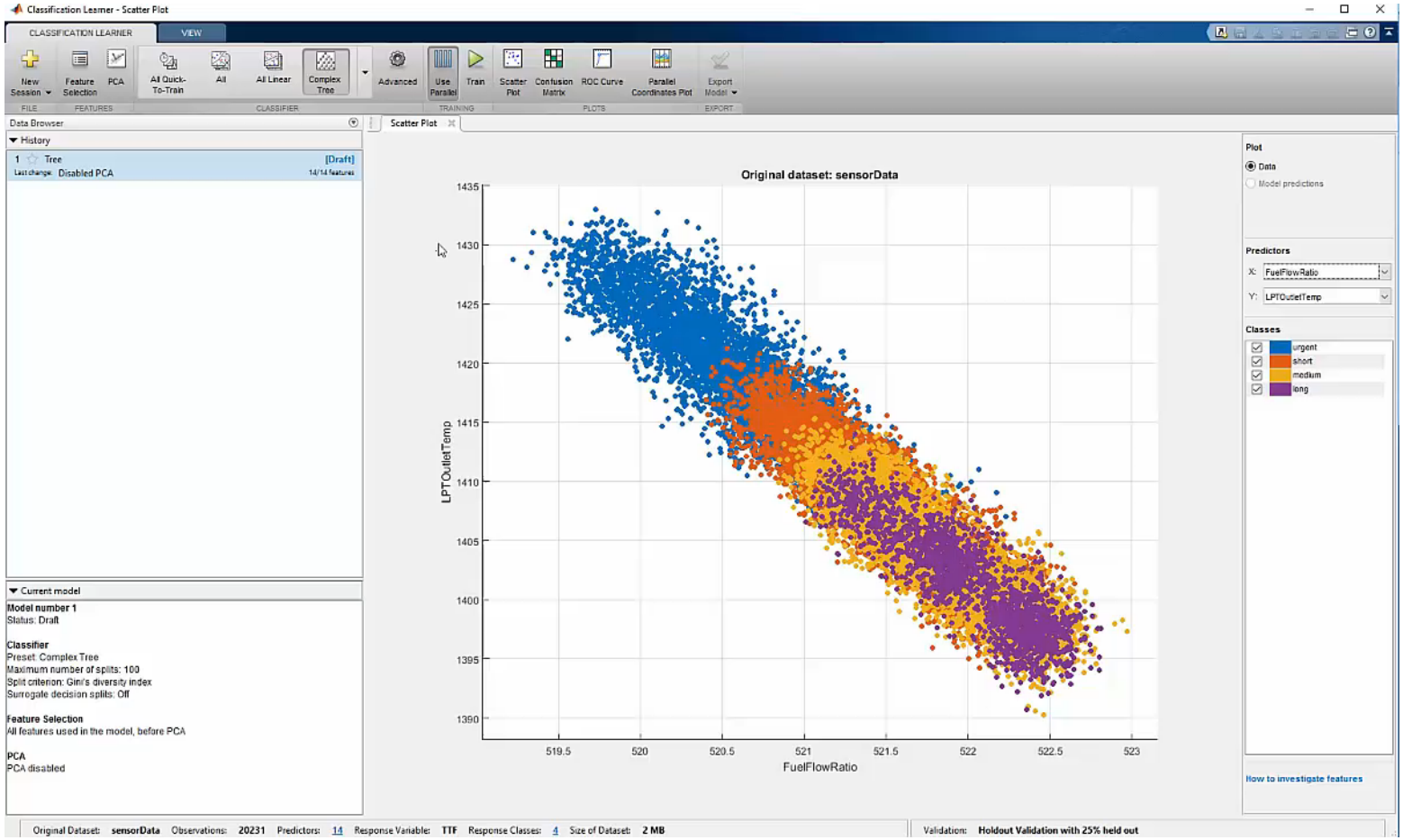
## Multicore Desktops



# Accelerating MATLAB and Simulink Applications

Parallel-enabled toolboxes

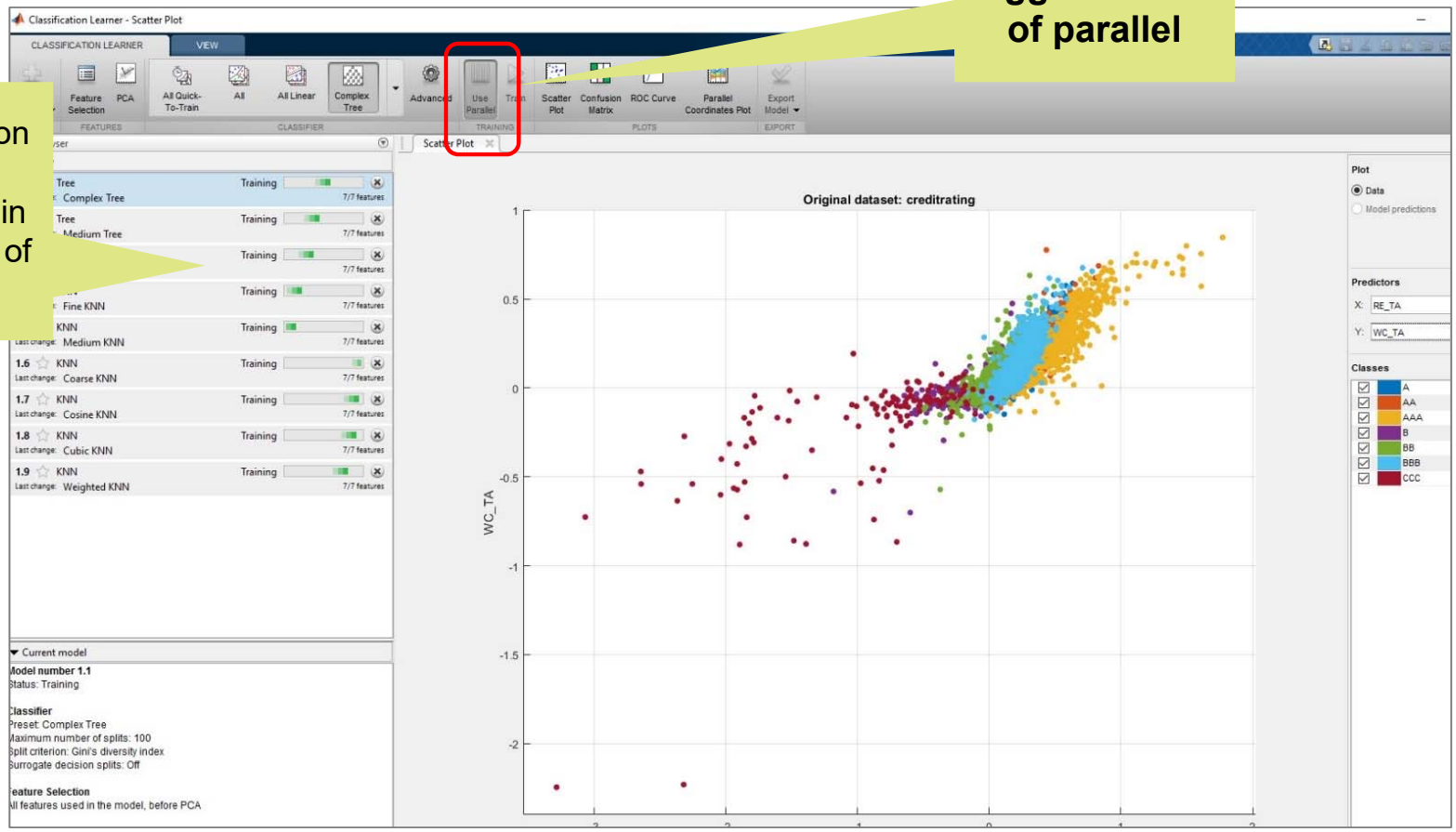
# Classification Learner



# Classification learner demo

Run classification learner quick to train classifiers in parallel instead of one by one

One click to toggle the use of parallel



# Parallel-enabled Toolboxes (MATLAB® Product Family)

## Enable acceleration by setting a flag or preference

### Image Processing

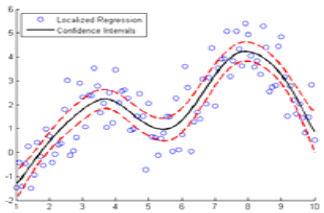
Batch Image Processor, Block Processing, GPU-enabled functions

Original Image of Peppers

Recolored Image of Peppers

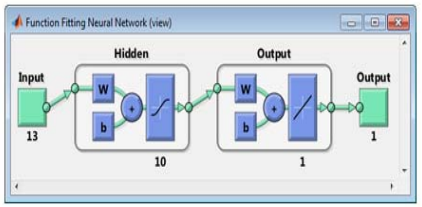
### Statistics and Machine Learning

Resampling Methods, k-Means clustering, GPU-enabled functions



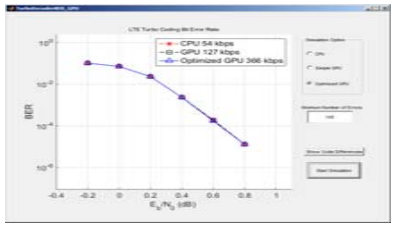

### Neural Networks

Deep Learning, Neural Network training and simulation



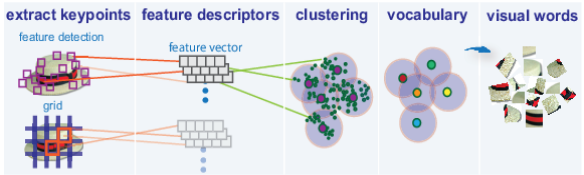

### Signal Processing and Communications

GPU-enabled FFT filtering, cross correlation, BER simulations



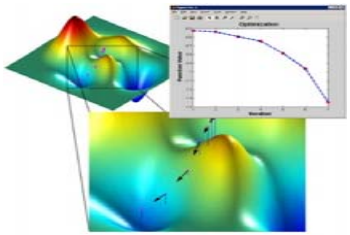
### Computer Vision

Bag-of-words workflow




### Optimization

Estimation of gradients



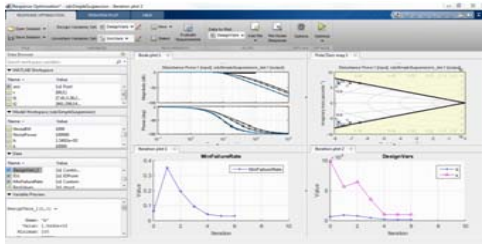
[Other Parallel-enabled Toolboxes]

# Parallel-enabled Toolboxes (Simulink® Product Family)

## Enable parallel computing support by setting a flag or preference

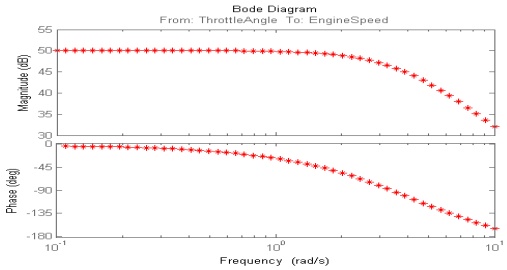
### Simulink Design Optimization

Response optimization, sensitivity analysis, parameter estimation



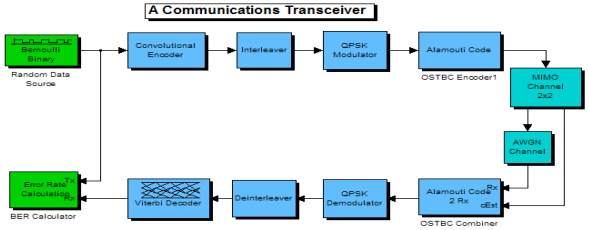
### Simulink Control Design

Frequency response estimation



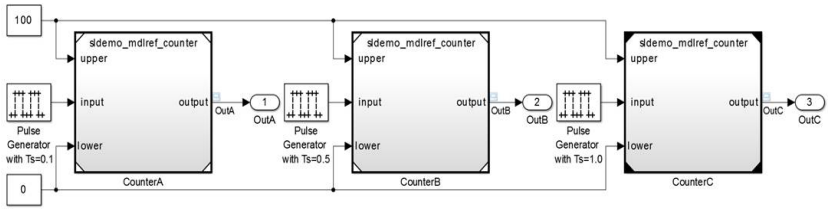

### Communication Systems Toolbox

GPU-based System objects for Simulation Acceleration




### Simulink/Embedded Coder

Generating and building code





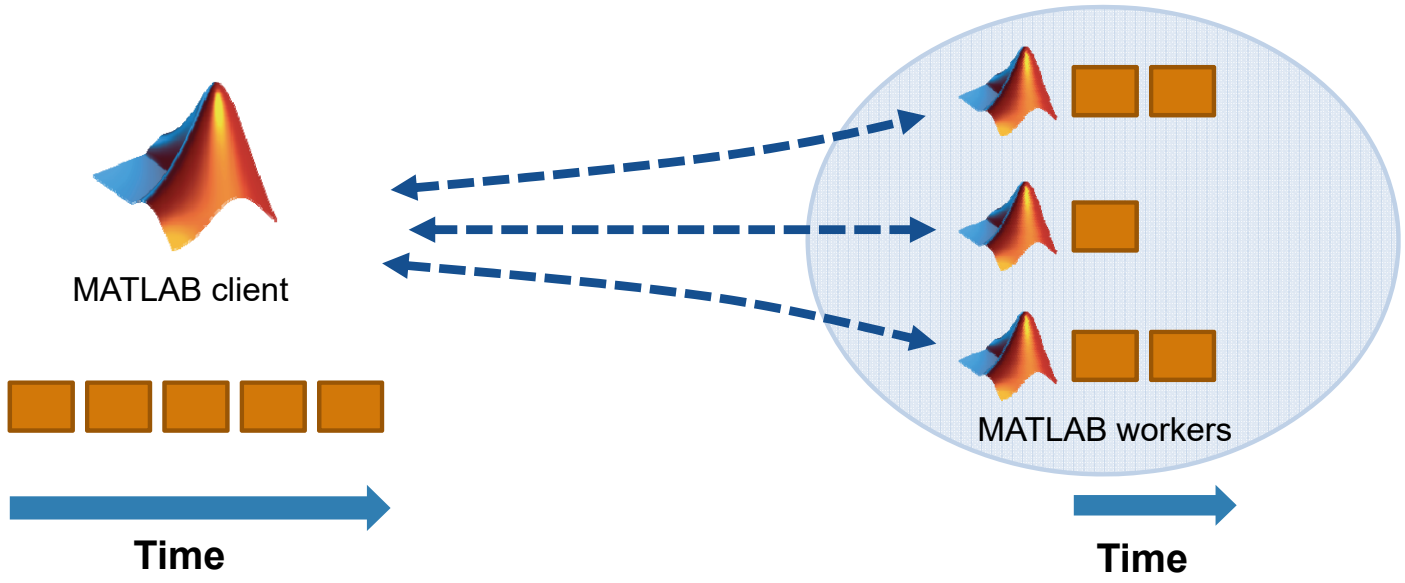

Other Parallel-enabled Toolboxes

# Accelerating MATLAB and Simulink Applications

Simple programming constructs

# Explicit Parallelism: Independent Tasks or Iterations

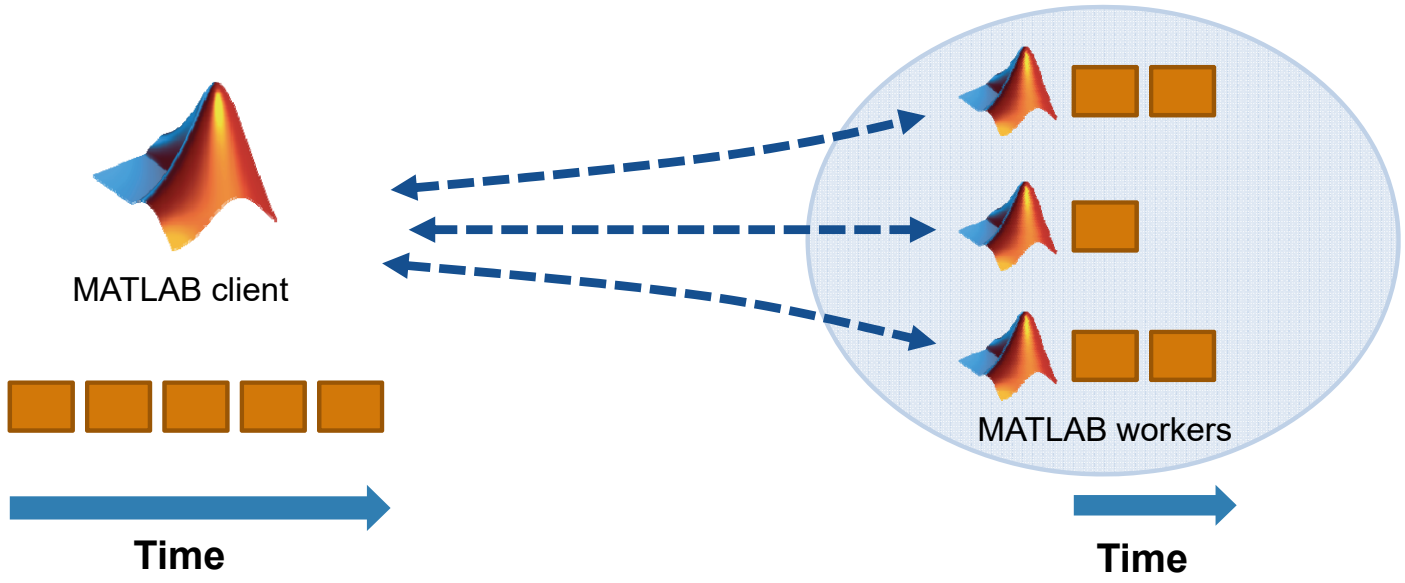
- Examples: parameter sweeps, Monte Carlo simulations
- No dependencies or communications between tasks



# Explicit Parallelism: Independent Tasks or Iterations

```
for i = 1:5
    y(i) = myFunc(myVar(i));
end
```

```
parfor i = 1:5
    y(i) = myFunc(myVar(i));
end
```

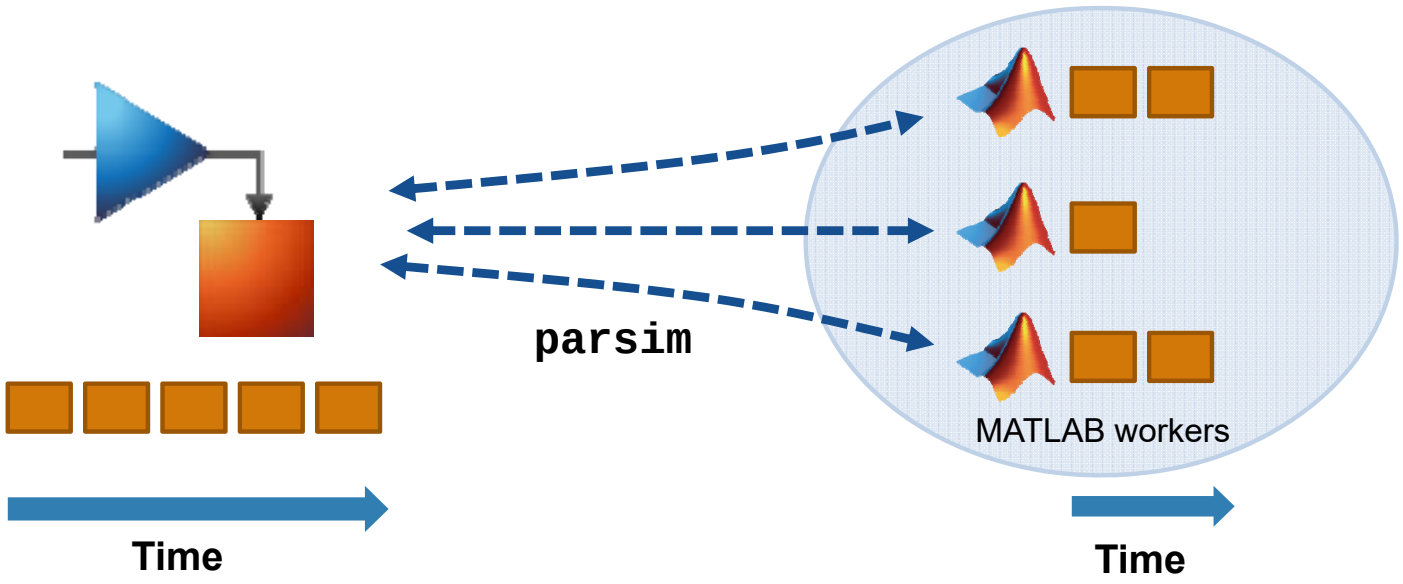


## parsim R2017a

- Run independent Simulink simulations in parallel

**Find out more:**
**Speeding Up Simulink Applications**

Murali Yeddanapudi
MathWorks

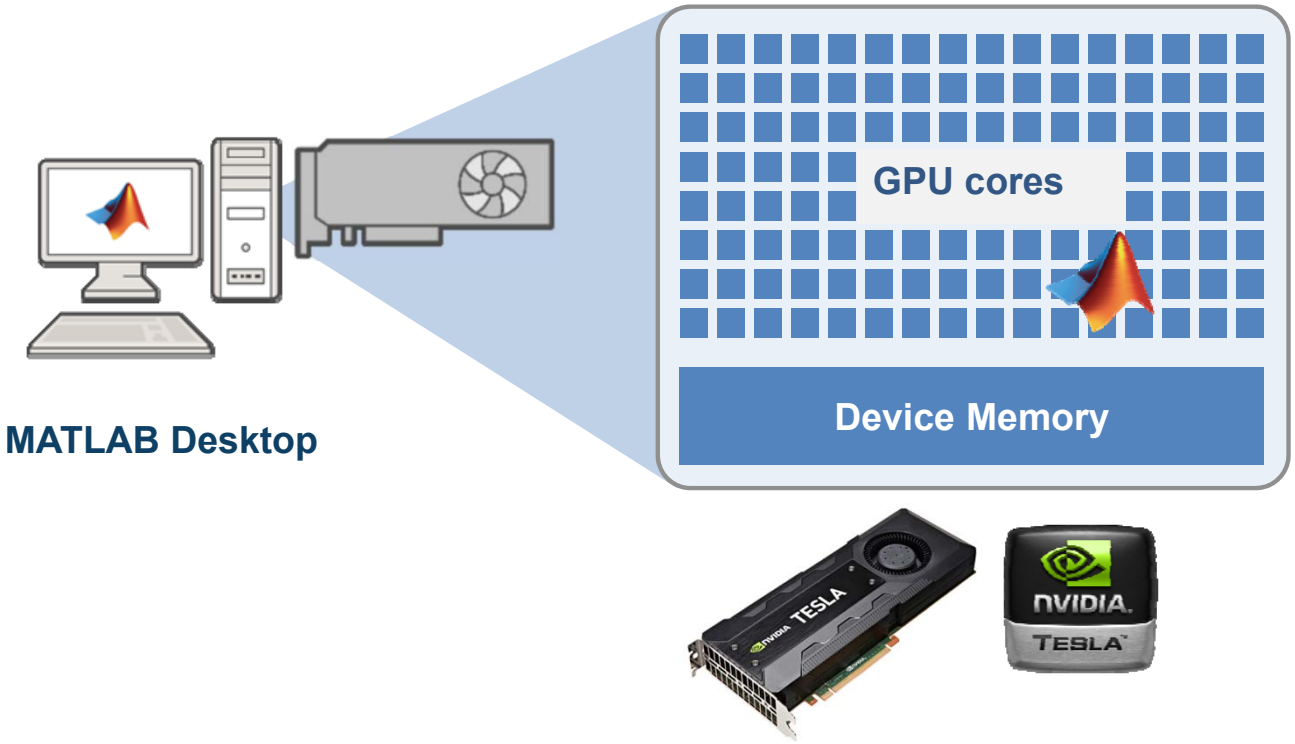


# Agenda

- Accelerate MATLAB and Simulink applications in your desktop
- Accelerate with NVIDIA™ GPUs
- Handle Big Data
- Scale to clusters and the cloud

# Parallel Computing Paradigm

## NVIDIA GPUs



# Speed-up using NVIDIA GPUs

Transfer Data To GPU From Computer Memory

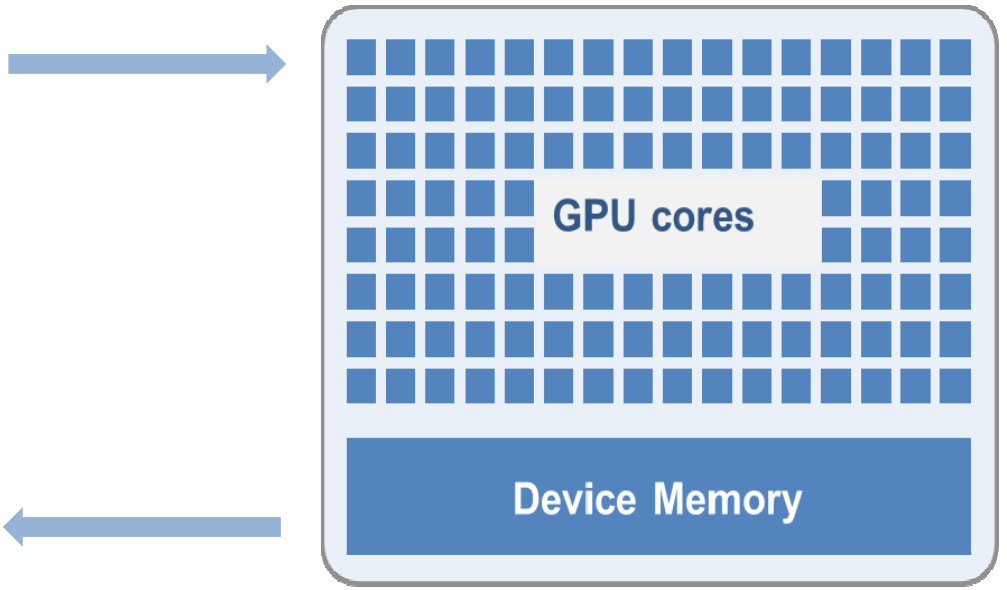
```
A=gpuArray(A);
```

Perform Calculation on GPU

```
X=eig(A);
```

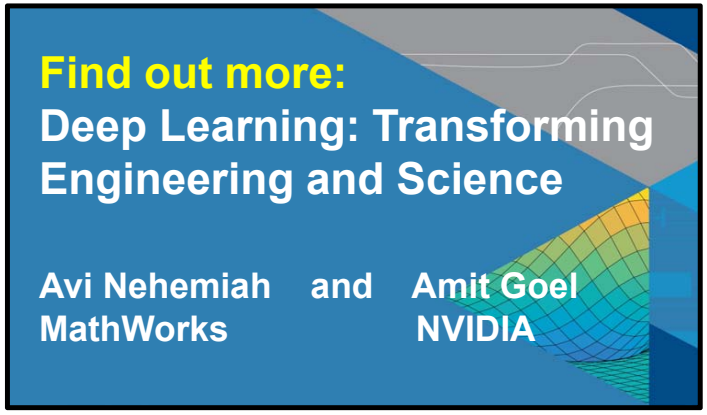
Gather Data or Plot

```
X=gather(X)
```



# Speed-up using NVIDIA GPUs

- 300+ GPU-enabled MATLAB functions
- Additional GPU-enabled Toolboxes
  - Neural Networks
  - Image Processing
  - Communications
  - Signal Processing
- Achieve even higher performance with GPU Coder™

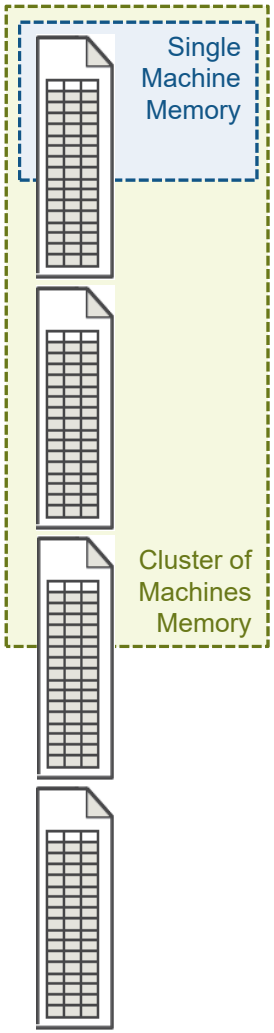


# Agenda

- Accelerate MATLAB and Simulink applications in your desktop
- Accelerate with NVIDIA™ GPUs
- Handle Big Data
- Scale to clusters and the cloud

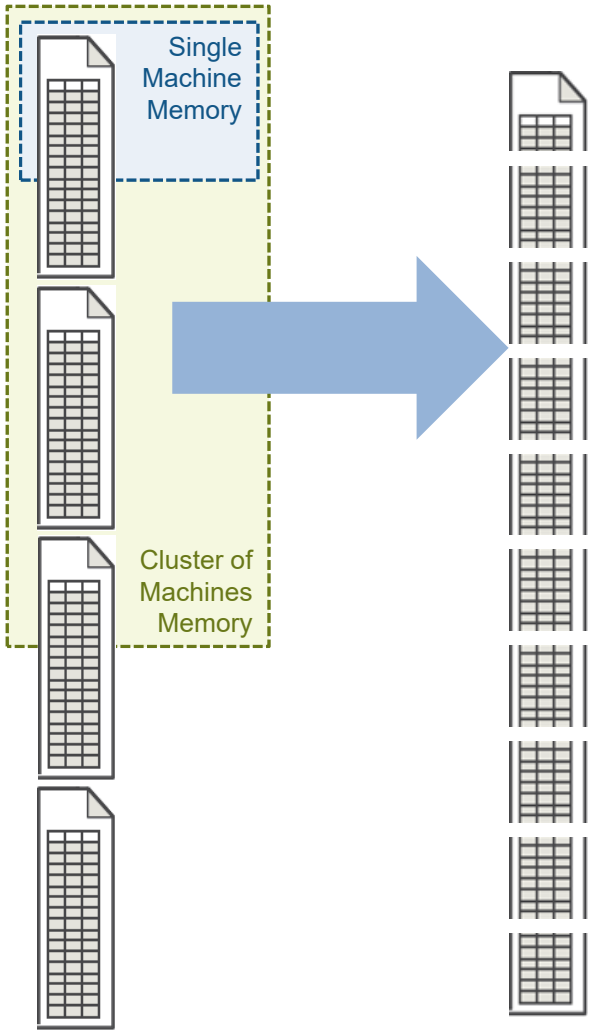
# `tall` arrays

- Data doesn't fit into memory (even cluster memory)



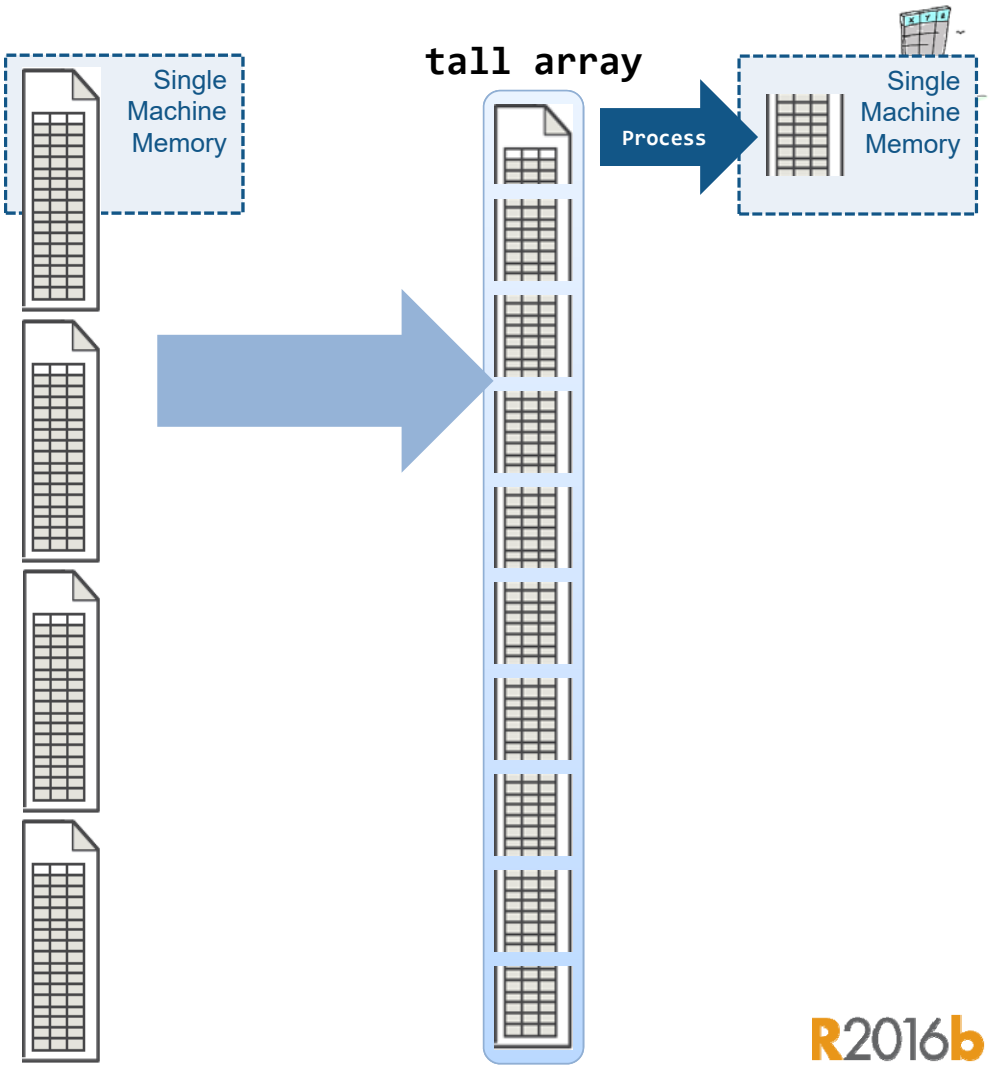
# `tall` arrays

- Automatically breaks data up into small “chunks” that fit in memory



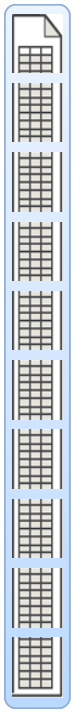
# `tall` arrays

- Automatically breaks data up into small “chunks” that fit in memory
- “Chunk” processing is handled automatically
- Processing code for tall arrays is the same as ordinary arrays



# `tall` arrays

- Data doesn't fit into memory



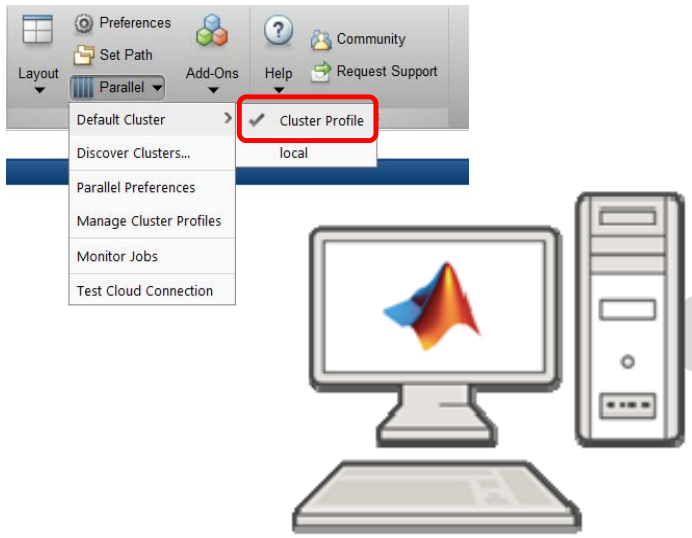
- Easy data access from files, databases or cloud storage systems
- Looks like a normal MATLAB array
  - Numeric types, tables, datetimes, strings, etc…
  - Basic math, stats, indexing, etc.
  - **Statistics and Machine Learning Toolbox**
    (regression, clustering, classification, etc.)

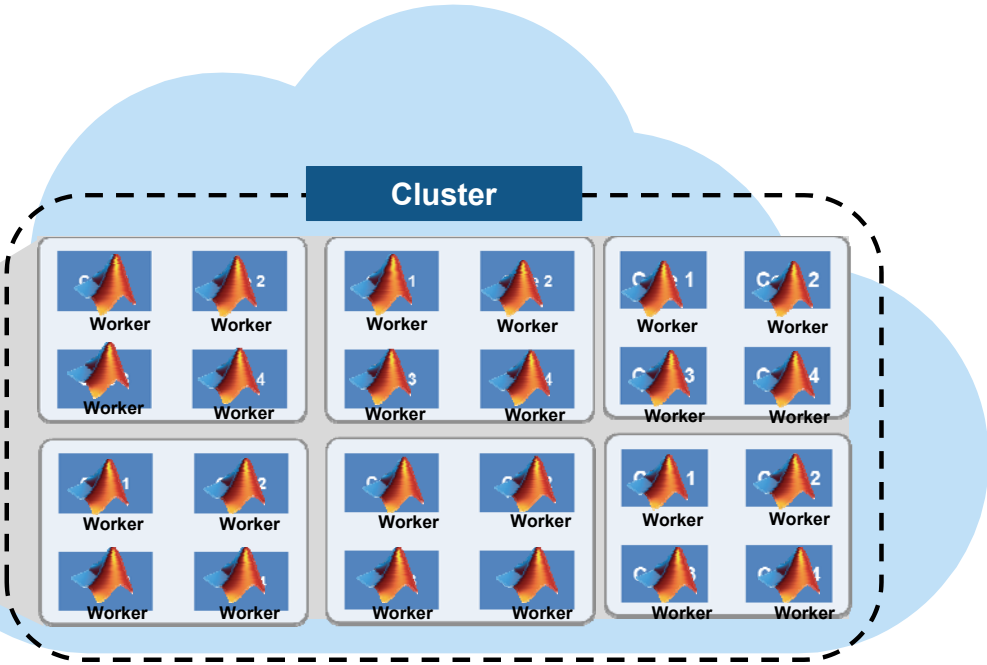
# Agenda

- Accelerate MATLAB and Simulink applications in your desktop
- Accelerate with NVIDIA™ GPUs
- Handle Big Data
- Scale to clusters and the cloud

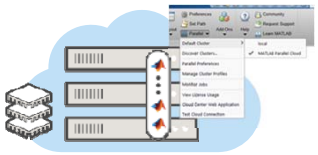
# Parallel Computing Paradigm

## Clusters/Cloud



# Options to scale beyond the desktop



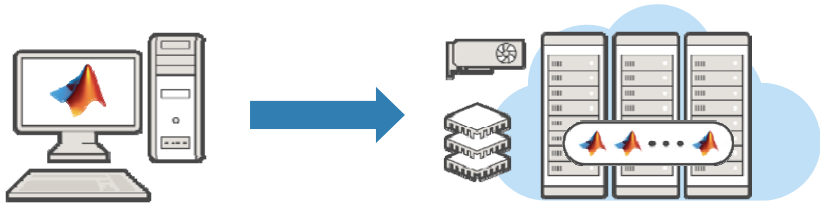
| Option | Parallel Computing Toolbox | MATLAB Parallel Cloud | Cloud Center | Custom Cloud | MATLAB Distributed Computing Server |
|---|---|---|---|---|---|
| Description | Explicit desktop scaling | Single-user, basic scaling to cloud | Scale to EC2 with some customization | Scale to custom cloud | Scale to clusters |
| Hardware | Desktop | MathWorks Compute Cloud | Amazon EC2 | Amazon EC2, Microsoft Azure,Others | Any |

# Summary and Takeaways

- Speed up your MATLAB and Simulink applications without being an expert
  - Apps and functions enabled for parallel computing
  - `parfor`, `parsim`, `gpuArray`
- Overcome memory limitations by
  - Distributing data to available hardware
  - Using `datastore` and `tall` or `distributed` arrays
- Develop applications on the desktop and scale to clusters

MathWorks

# What's new in Parallel Computing Toolbox?



## R2016b

- `tall` array support for big data

## R2017a

- Simplified parallel Simulink simulations using `parsim`
- Train a single deep learning network with multiple CPUs or multiple GPUs
- Access data stored in the cloud on Amazon S3™

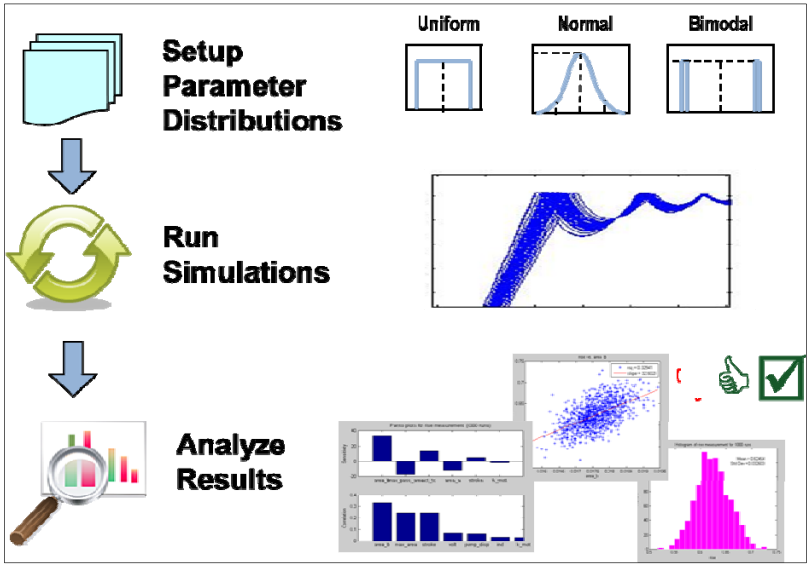
## R2017b

- Access data stored in the cloud on Microsoft® Azure® Blob Storage

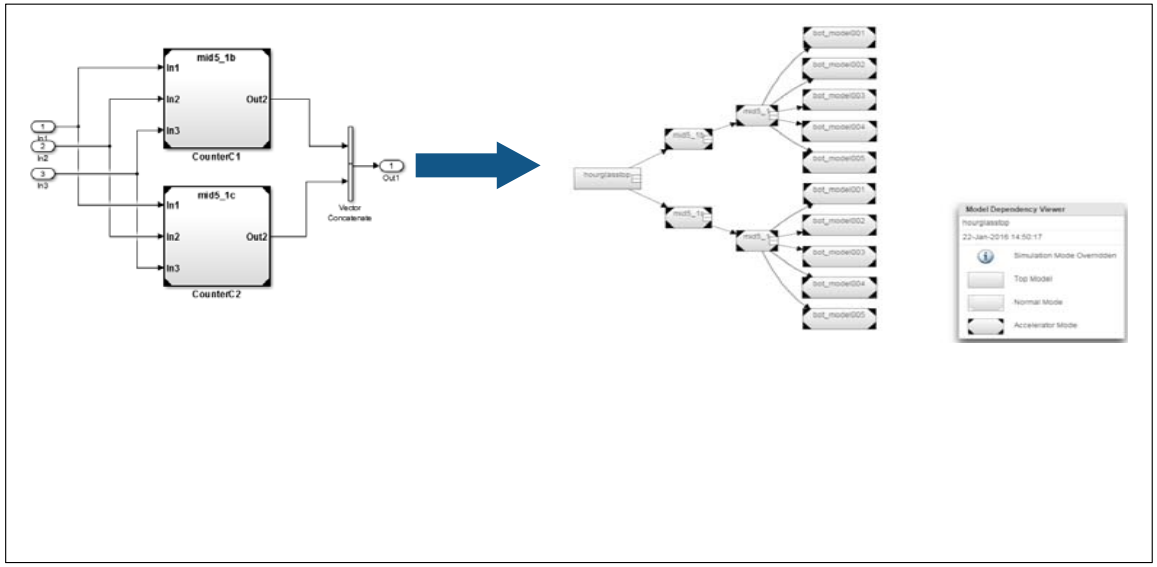
# Leverage Parallel Computing with Simulink

Reduce the total amount of time it takes to…

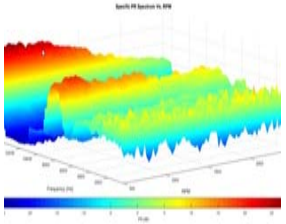
Run multiple independent simulations (E.g. Parameter sweeps, Monte Carlo Analysis)




Update models containing large model reference hierarchies



# Benefits of parallel computing



**Automotive Test Analysis and Visualization**
3-4 months of development time saved

**Design and Build Wave Energy Farm**
Sensitivity studies accelerated 12x

**Calculating Derived Market Data**
Implementation time reduced by months
Updates loaded 8X faster

**Discrete-Event Model of Fleet Performance**
Simulation time reduced from months to hours
20X faster simulation time
Linkage with Neural Network Toolbox

**Heart Transplant Studies**
4 weeks reduced to 5 days
6X speedup in process time

# Energy Production – World's First Operating Wave Farm

## Carnegie Wave Energy

**Goal:** Develop unique technology for generating electric power from ocean waves

A CETO unit ready for deployment in the wave farm.

## Challenges

- Analyze loads and estimate energy output without building scale model of entire system
- Run simulations for a range of configurations, sea conditions and faults

## Why Parallel Computing

- Sensitivity studies accelerated

*"Our sensitivity studies require numerous simulations because we typically simulate 15 to 20 sea states for each parameter value we vary. With Parallel Computing Toolbox we can run **simulations in parallel**, and with a twelve-core computer we see an almost **twelvefold increase in speed**."*

*Jonathan Fiévez*
*Carnegie Wave Energy*



Learn More

# Signal Processing – Acoustic Data Analysis

## NASA Langley Research

**Goal:** Accelerate the analysis of sound recordings from wind tunnel tests of aircraft components

## Challenges

- Legacy code took 40 mins to analyze single wind tunnel test data
- Reduce processing time to make on-the-fly decisions and identify hardware problems

## Why GPU Computing

- Computations completed 40 times faster

*"Many operations we perform, including FFTs and matrix multiplication, are **GPU-enabled MATLAB functions** .Once we developed the initial MATLAB code for CPU execution, it took 30 minutes to get our algorithm working on the GPU— **no low-level CUDA programming** was needed. The addition of GPU computing with Parallel Computing Toolbox cut it to **under a minute**, with most of that time spent on data transfer"*

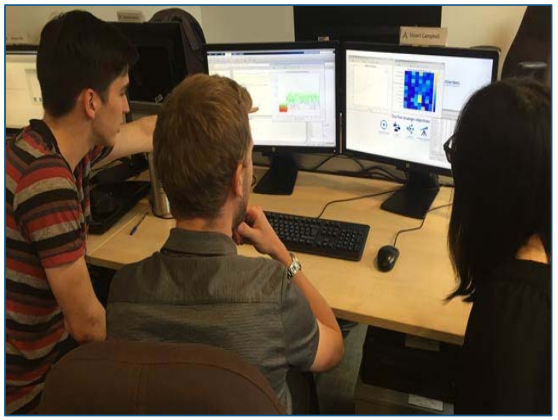
*Christopher Bahr*
*NASA*

Learn More

# Machine Learning Portfolio Allocation Models in the Cloud

## Aberdeen Asset Management

**Goal:** Improve asset allocation strategies by creating model portfolios with machine learning techniques



Interns using MATLAB at Aberdeen Asset Management.

## Challenge:

- Train and backtest the machine learning algorithms using available market data stretching back more than 15 year

## Why parallel computing:

- Processing times cut from 24 hours to 3 running on Microsoft® Azure cloud
- Multiple types of data easily accessed

*With MATLAB we can develop prototypes to test new machine learning techniques quickly. Once we've refined the techniques, MATLAB Distributed Computing Server enables us to get rapid, reliable results by running the algorithms with large financial data sets on a distributed computing cluster."*

*Emilio Llorente-Cano, Aberdeen Asset Management*

# Bosch Develops a Single Platform for Automotive Test Data Analysis and Visualization

## Challenge

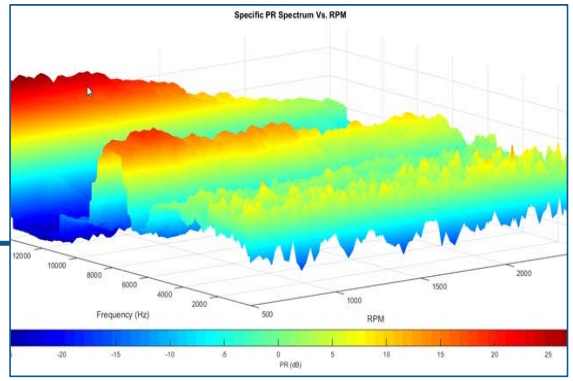
Reduce the time and steps needed to process and interpret data from automotive test benches

## Solution

Use MATLAB to develop and deploy a platform for analyzing and visualizing engineering data from various domains

## Results

- Validation times reduced by an average of 40–50%
- Three to four months of development time saved
- Analysis accuracy increased




ENValyzer plot showing prominence ratio (PR) vs. RPM spectrum results. Prominence ratio is commonly used in acoustics data analysis

**"MATLAB enabled us to speed the development of ENValyzer, a customizable, easy-to-use tool for analyzing, visualizing, and interpreting engineering data in a wide variety of formats. Now, our engineers can validate components faster and more accurately than was possible with spreadsheets and third-party tools."**

**Sharath SL**
**Bosch**

# Discrete-Event Models to Predict Fleet Performance

## Lockheed Martin

**Goal:** rapidly simulate thousands of detailed, easily configurable models that can rapidly simulate parameter combinations and scenarios to maximize F-35 fleet performance

## Challenges

- Data intensive problem, due to complexity of the F-35 aircraft and global logistics system to support it
- Complexity of solution due to thousands of Monte Carlo simulations.

## Why Parallel Computing

- **Simulation setup time reduced from months to hours**
- **Development effort lessened**
- **Simulation time cut by months**

# Lund University Develops an Artificial Neural Network for Matching Heart Transplant Donors with Recipients

## Challenge

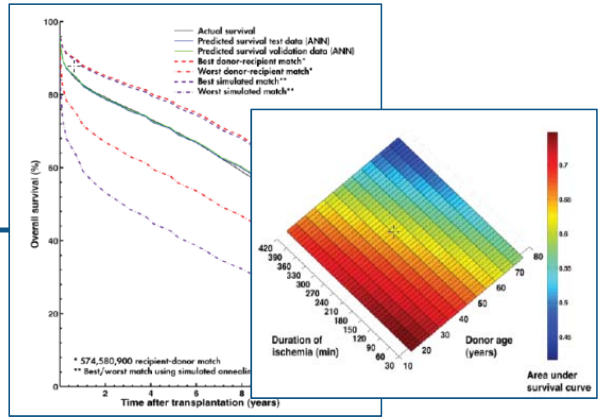
Improve long-term survival rates for heart transplant recipients by identifying optimal recipient and donor matches

## Solution

Use MathWorks tools to develop a predictive artificial neural network model and simulate thousands of risk-profile combinations on a 56-processor computing cluster

## Results

- Prospective five-year survival rate raised by up to 10%
- Network training time reduced by more than two-thirds
- Simulation time cut from weeks to days




Plots showing actual and predicted survival, best and worst donor-recipient match, best and worst simulated match (left); and survival rate by duration of ischemia and donor age (right).

**"I spend a lot of time in the clinic, and don't have the time or the technical expertise to learn, configure, and maintain software. MATLAB makes it easy for physicians like me to get work done and produce meaningful results."**

**Dr. Johan Nilsson**
**Skåne University Hospital**
**Lund University**



Link to user story

# Commerzbank Develops Production Software System for Calculating Derived Market Data

Commerzbank headquarters in Frankfurt.

## Challenge

Compute a variety of derived market data from raw market data

## Solution

Use MATLAB to read data from a data management system in a Windows and Linux architecture, perform analyses and optimizations, visualize results, and deploy mission-critical calculations

## Results

- Integration with existing system simplified
- Implementation time reduced by months
- Updates made in days, not weeks

**"Our solution required a Windows client and Linux server software. We used MATLAB to rapidly develop both by taking advantage of distributed computing, a MEX-file interface to access our financial data, and fast, built-in functions for optimization, regression, and more."**

**Julian Zenglein**
**Commerzbank**